

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



TRABAJO DE GRADO:

"PROPUESTA DE UN PROTOTIPO QUE PERMITA A PERSONAS CON DISCAPACIDAD VISUAL LA LECTURA DE TEXTOS UTILIZANDO INTELIGENCIA ARTIFICIAL PARA LA BIBLIOTECA DE LA FACULTAD DE ORIENTE"

PRESENTADO POR:

KEVIN EDUARDO CLAROS ZELAYA

MARVIN JOSÉ BENÍTEZ BENÍTEZ

MAURICIO RIGOBERTO MARTÍNEZ ROMERO

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

DOCENTE ASESOR:

ING. LUDWIN ALDUVI HERNÁNDEZ VÁSQUEZ

CIUDAD UNIVERSITARIA, OCTUBRE DE 2020

SAN MIGUEL, EL SALVADOR, CENTRO AMÉRICA

UNIVERSIDAD DE EL SALVADOR

AUTORIDADES

MASTER ROGER ARMANDO ARIAS ALVARADO

RECTOR

DOCTOR RAÚL ERNESTO AZCÚNAGA LÓPEZ

VICERRECTOR ACADÉMICO

INGENIERO JUAN ROSA QUINTANILLA

VICERRECTOR ADMINISTRATIVO

INGENIERO FRANCISCO ANTONIO ALARCÓN SANDOVAL

SECRETARIO GENERAL

LICENCIADO RAFAEL HUMBERTO PEÑA MARÍN

FISCAL GENERAL

LICENCIADO LUIS ANTONIO MEJÍA LIPE

DEFENSOR DE LOS DERECHOS UNIVERSITARIOS

FACULTAD MULTIDISCIPLINARIA ORIENTAL

AUTORIDADES

LICENCIADO CRISTÓBAL HERNÁN RÍOS BENÍTEZ

DECANO

LICENCIADO OSCAR VILLALOBOS

VICEDECANO

LICENCIADO ISRAEL LÓPEZ MIRANDA

SECRETARIO INTERINO

LICENCIADO JORGE PASTOR FUENTES CABRERA

DIRECTOR GENERAL DE PROCESOS DE GRADUACIÓN

INGENIERA MILAGRO DE MARÍA ROMERO DE GARCÍA

COORDINADORA DE PROCESOS DE GRADUACIÓN

TRIBUNAL EVALUADOR

INGENIERO LUDWIN ALDUVI HERNÁNDEZ VÁSQUEZ

DOCENTE JURADO ASESOR

INGENIERO LUIS JOVANNI AGUILAR

DOCENTE JURADO CALIFICADOR

INGENIERO MILAGRO ALICIA GONZÁLEZ DE REYES

DOCENTE JURADO CALIFICADOR

Agradecimientos

A Dios Todopoderoso que con su bendita misericordia y guianza a lo largo de mi vida, en especial durante este proceso, me ha permitido finalizar con éxito; por la sabiduría, el temple y los valores que sin su ayuda no hubiera sido capaz de construir.

Mis padres porque son el motivo de querer ser una mejor persona y siempre querer superarme en todo momento, por su amor incondicional, a mi padre René Alberto Claros Chileno por mostrarme el valor de la disciplina, la dedicación y la satisfacción proveniente del trabajo duro y el esfuerzo, a mi madre Leonorilda Zelaya de Claros por brindarme apoyo y ánimos durante toda mi vida y ser la primera en resaltar mis esfuerzos, por enseñarme la importancia del respeto, la integridad, el velar por otros y no ceder ante las adversidades.

A mis hermanos, René y Gabriela ustedes son una luz en los momentos de dificultad y su apoyo es incondicional ante cualquier situación.

Agradezco de manera especial a Norma Alicia Martínez Cruz por estar conmigo en momentos difíciles, por ayudarme a nunca rendirme y ser mi inspiración para siempre mirar al futuro con una sonrisa en el rostro.

A mis docentes, no solo durante mi periodo en la universidad, sino durante toda mi carrera educativa que siempre vieron por mí y con sus palabras y ejemplo pudieron tallar en mí los conocimientos que me acompañarán hasta el último de mis días.

A nuestro asesor de tesis, Ing. Ludwin Alduvi Hernández, por compartirme con interés y empeño su tiempo y conocimientos para el desarrollo de este proyecto.

A mis compañeros de carrera, con quienes compartí gratos momentos, y fueron parte importante de mí, por todo su apoyo y amistad, les agradezco mucho.

A mis compañeros de tesis, durante este proceso hemos podido conocernos mejor y hemos sabido trabajar de la mano en todo momento, su paciencia y colaboración fue clave para que pudiéramos finalizar este proyecto, gracias por todo y con alegría podemos decir que finalizamos esta carrera.

KEVIN EDUARDO CLAROS ZELAYA

Agradecimientos

A mí familia a la que estaré eternamente agradecido ya que fueron ellos quienes me impulsaron siempre a seguir adelante en este sueño, gracias a todos por siempre estar hay para mi en momentos de dificultad y alegría y por ayudarme a cumplir esta meta.

A mi madre Joana que pese a las dificultades siempre ha tenido fe en mí y ha servido como inspiración moral y profesional, gracias por todo el amor y cariño que me has entregado a lo largo de estos años siempre guiándome hacia adelante, siempre convenciéndome que se puede lograr todo si te lo propones.

A mis tías Mirna y Delmy que siempre me ayudaron y apoyaron en todo lo posible para ayudarme a cumplir este sueño, a mi tía Wendy que pese a la distancia siempre me ha brindado su apoyo incondicional, gracias por estar siempre.

A mí querida abuela Natividad que siempre me ha brindado su amor, su comprensión, su cariño y sobre todo por ser el motor que me impulsa para avanzar en las metas que me he propuesto alcanzar, muchísimas gracias por todo, siempre has sido el modelo de persona a seguir.

A mi hermanos Richard y Samantha, a mi primo Jonathan, gracias por sus palabras de aliento y apoyo.

A los docentes que me guiaron en mi proceso formativo, quienes estuvieron presentes no solo en el ámbito universitario sino también como amistades que brindaron su consejo y ejemplo para ser un profesional íntegro.

A nuestro asesor de tesis, Ing. Ludwin Hernández, por ayudarnos con empeño y dedicación y entregarnos su tiempo, conocimiento y consejos para desarrollar este proyecto.

A mis compañeros de carrera, ya que con ellos compartí alegría y decepciones como estudiante, gracias por esos recuerdos gratos que compartimos juntos y por todo su apoyo y amistad, aunque es verdad que el tiempo y la distancia separa las personas sepan que estoy agradecido por todo lo que vivimos juntos.

A mis compañeros de tesis, a pesar de las dificultades hemos logrado finalizar este proyecto, gracias por su entrega y dedicación en el desarrollo de las diferentes etapas que componen esta meta, gracias por todo.

A todas las personas que conocí en la universidad, se que fueron muchas, tanto como para no recordarlas a todas, pero todas ellas aportaron en mayor o menor medida en el desarrollo de la persona que escribe estas líneas, muchas gracias a todos.

MARVIN JOSÉ BENÍTEZ BENÍTEZ

Agradecimientos

Quién podría imaginar que estoy redactando este texto que representa el final de mi etapa como estudiante universitario, el llegar a este punto no fue una tarea muy fácil que se diga, ya que representó una buena cantidad de retos que fueron muy valiosos afrontarlos y otros que tal vez no fueron muy agradables. Las siguientes palabras que quedan plasmadas en este documento son un intento de expresar mis agradecimientos por todo lo que me tocó vivir en el camino hasta el punto en el que me encuentro al momento de redactarlas. Será un breve resumen de experiencias desde que inicié a ser estudiante universitario, dejar de serlo, “salir al mundo”, hasta el momento de la defensa final, una especie de historial/bitácora de palabras que quiero leer en un futuro.

Sin duda que quiero empezar agradeciendo a las personas que han estado siempre desde el inicio al final; mi familia. Gracias a mi madre Carlota, mi padre Manuel, mi hermano Manuel, mi hermana Sandra y mi abuela Santos que han sido las personas que siempre han estado en las buenas y en las malas, las personas con las que de vez en cuando compartía las frustraciones de ser estudiante —sin duda que me escucharon muchas veces quejarme— aparte que fueron y siguen siendo un apoyo emocional también lo fueron en el aspecto económico, algo vital para poder conseguir una carrera universitaria. Gracias a mi mamá que siempre que regresaba de estudiar ella me preguntaba si iba a cenar, mi papá por siempre preguntarme si quería café y llevarlo hasta la mesa donde estaba estudiando, gracias a mi hermano Manuel y mi hermana Sandra que siempre colaboraban en casa cuando yo no podía hacerlo y se encargaban de hacer las tareas que eran mi responsabilidad, gracias a mi abuela Santos que siempre me preguntaba cómo me iba en los estudios y cuándo los iba a terminar —le mentí más de una vez— gracias por todos esos momentos en los que estuvieron y siguen estando, en las buenas y en las malas, no hubiera podido conseguir nada por mi cuenta sin el apoyo que me dieron en los momentos oportunos y, eso es algo de lo que estoy y seguiré estando agradecido hasta el final de los días. Ojalá que se me permita compartir varios logros con ellos.

Es imposible no pensar en todo lo que viví como estudiante universitario y no hacer un suspiro un poco profundo, aunque hubieron malos momentos, los primeros que recuerdo son los buenos y divertidos que fueron muchos a lo largo de los años. Recordar cómo empezó todo, el ser una persona sin mucha experiencia en la vida y sin grandes habilidades aventurándose a conseguir una carrera universitaria, como todo en la vida lo más difícil siempre es empezar y me consta que en la universidad sí que lo es, el ver pizarras y folletos con cosas que se supone debería de saber pero, todo lo contrario, sin algo de conocimiento, el empezar desde cero a intentar aprender para

“sobrevivir”, desde el principio salir mal en los exámenes, prácticamente “apretando todos los botones” y esperar resultado de las evaluaciones con los dedos cruzados, el tener que despertar a las 4:30am para poder llegar a la clase de las 7:00am, el salir a las 5:00pm de clases y correr para poder llegar tiempo a todos los buses, acumular casi 3 hrs diarias en transporte, las dificultades económicas que estuvieron presentes en todo momento, el aguantar hambre por las extensas jornadas que habían entre cada clase... Una buena cantidad de experiencias que agradezco de haber podido vivirlas.

También un gran agradecimiento a mi compañeros de grupo de tesis, un honor haber podido hacerla con ellos, unas grandes personas que saben lo que hacen y que sin la colaboración de ellos no sería posible redactar estas palabras. Cuando se tenía que hacer las cosas siempre al pendiente, personas ordenadas con las que da gusto trabajar. Kevin, Marvin, gracias por todo el apoyo, son unas personas con un potencial inmenso y que saben demostrarlo cuando se tiene que demostrar, el proceso de tesis con el apoyo que brindaron hizo la experiencia menos frustrante. Estuvimos desde el principio de la universidad compartiendo, aprendí mucho de ellos, yo siempre escuchando todo lo que comentaban cuando estudiábamos en grupo para los parciales, sin nada de eso tampoco hubiera podido llegar hasta este punto ya que era una persona que no sabía nada de nada, fueron pieza fundamental en todo el proceso. Espero que todos podamos llegar lejos y haciendo lo que nos gusta, tenemos el potencial para destacar donde queramos y espero que sepamos aprovecharlo. Ojalá que se dé la oportunidad de seguir sabiendo el camino que han seguido para poder alegrarme junto con ellos de los logros que sin duda sé que serán varios.

No puedo olvidar mencionar los agradecimientos a todas las personas que conocí en la universidad, me es imposible mencionar a cada una de ellas porque fueron muchas y cada una de ellas aportó en mi desarrollo como persona. Las personas que empezaron siendo desconocidas pero que al final terminamos formando una amistad, compartimos muchas cosas de principio a fin, con algunas llegamos a ser más cercanos, me alegro de haberlas conocido, cómo no recordar todas las veces que jugábamos “UNO”, eran risas, muchas risas, las veces que íbamos a “Metrocentro” a distraernos y platicar de cualquier cosa con tal de divertirnos, también, cómo olvidar la vez que fuimos al volcán de San Miguel y que terminamos perdiéndonos, todos esos momentos que fueron espontáneos, las veces que estudiamos en grupo, las veces que en almuerzo nos poníamos a platicar en la biblioteca, las veces que esperábamos los “tacos sudados” para almorzar... que buenos tiempos. Si bien la universidad quedó mucho a deber en cuanto a aprendizaje, no creo que haya sido una pérdida de tiempo y dinero porque tuve la oportunidad de conocer

a estas personas, ese fué el verdadero valor de estudiar; conocer personas, hicieron que todo valiera la pena y hacer que la lucha de ser estudiante hasta resultara divertida, la motivación de ir a recibir clases porque sabía que podía platicar con ellas antes de que empezaran. No sé si podré seguir sabiendo de cada una de ellas pero espero que sí.

Luego de salir de la universidad tuve la oportunidad de participar en ciertas iniciativas que me permitió conocer y acercarme más a personas que ya tenía el gusto de conocer, todas ellas me ayudaron mucho a ver la vida desde otro punto de vista, me ayudaron a ser una mejor persona, el sentirme tan cómodo platicando y compartiendo con ellas es algo de lo que agradezco mucho, a pesar que en cuestión de meses no fué mucho tiempo pero al nivel de amistad que llegamos tan rápido me hace sentir contento porque pasó de una forma fluida sin forzarlo, personas que marcaron un antes y un después de mí persona. Quisiera empezar con las personas que conocí; Fabricio, Imelda, Sara, unas grandes personas que me demostraron la amistad sin intereses ni nada, las veces que almorzábamos y platicábamos, el contarnos metas, gustos, y un sin fin de cosas, en poco tiempo éramos grandes amigos, esa tranquilidad y sencillez que compartimos, creo que sin ellas no fuera posible ser la persona que soy al momento de esta redacción, las personas que me permitieron conocer otros aspectos de la vida. Ahora continuo con las personas que ya tenía el gusto de conocer desde los tiempos de la universidad pero que se me permitió poder conocerlas más, Roxana(rox), Paiz, Antony(tony), Edwin, Melvin(el chele), Fátima, tuve la oportunidad de hacernos más amigos, de acercarnos más y tal como se los dije a ellos en su momento, yo muy a gusto de compartir su compañía porque todo pasó de forma fluida, las grande pláticas en hora de almuerzo cuando íbamos a “comer moscas”, los chistes, pero, algo muy importante; nunca hablar de grandezas ni aparentar algo que no es, siempre de una forma auténtica, dando una amistad sincera sin esperar algo a cambio, todo de una forma natural, personas con una gran calidad humana. Tuve la oportunidad de poder compartir con ellas en el momento que presentaron sus trabajos de tesis, algo que aprecié mucho, el haber estado en ese momento y que fué algo que no esperaba poder compartir con ellas, el estar apoyando en ese momento, además el momento cuando recibieron sus títulos de ingenieros poder dar las felicitaciones por el logro. También espero poder seguir sabiendo de ellas durante mucho tiempo, todos los logros que cumplan, seguir conservando su amistad. Esa etapa de dejar de ser estudiante universitario no comenzó mal porque pude conocer y acercarme a las personas ya mencionadas, llegaron en el momento oportuno para generar un cambio positivo en mi forma de ser.

En la época de redacción de estas palabras a nivel mundial se está viviendo una situación difícil debido a

una pandemia, es una época un poco complicada para cada una de las personas, muchas cosas malas, muchas, pero por mi lado sucedió algo que me agrada, conocer a más personas de una forma espontánea, ya que creo que de esa forma surgen las mejores amistades. Participando en comunidades online pude conocer a unas grandes personas; Ana, Christopher(Chris), Juan, a quienes agradezco por su amistad y su apoyo, a quienes se les puede compartir pensamientos sin ser juzgado, comprendiendo la situación, aportando comentarios, un gran apoyo en general. Que placer haber podido conocer personas tan buena onda y de un gran nivel humano.

El día de mañana espero poder ser una persona que viva relativamente feliz con lo que ha conseguido, todo este proceso solo es un paso en la vida, algo que a lo mejor no es mucho en comparación de todo lo que está por delante, retos más grandes, retos de más tiempo, muchas metas que espero cumplir, pero lo más importante es que espero poder divertirme, dejar que las cosas fluyan, cada cosa poniéndose en su lugar sin forzarlas, que se me abran oportunidades, conocer a más personas, estar por aquí, estar por allá, pero siempre disfrutando el viaje. Terminando una fase de vida luego de muchos años —que pudieron haber sido menos—, y por la cual estoy inmensamente agradecido.

Y como agradecimiento final, gracias a la vida por lo que se me ha cruzado en mi camino, todo es aprendizaje para mejorar constantemente.

MAURICIO RIGOBERTO MARTÍNEZ ROMERO

**"PROPUESTA DE UN PROTOTIPO QUE PERMITA A PERSONAS CON DISCAPACIDAD VISUAL LA
LECTURA DE TEXTOS UTILIZANDO INTELIGENCIA ARTIFICIAL PARA LA BIBLIOTECA DE LA
FACULTAD DE ORIENTE"**

Kevin Eduardo Claros Zelaya

Marvin José Benítez Benítez

Mauricio Rigoberto Martínez Romero

Universidad de El Salvador

Notas del autor

Kevin Eduardo Claros Zelaya

Marvin José Benítez Benítez

Mauricio Rigoberto Martínez Romero

Tesis para optar al título de Ingeniero de Sistemas Informáticos

Universidad de El Salvador

Facultad Multidisciplinaria Oriental

Resumen

Este documento contiene información sobre el desarrollo del proyecto de investigación "PROPUESTA DE UN PROTOTIPO QUE PERMITA A PERSONAS CON DISCAPACIDAD VISUAL LA LECTURA DE TEXTOS UTILIZANDO INTELIGENCIA ARTIFICIAL PARA LA BIBLIOTECA DE LA FACULTAD DE ORIENTE." que fue realizado como trabajo de graduación para optar al grado de Ingeniero de Sistema Informático. La Investigación fue realizada debido a la problemática encontrada que indicaba que las personas con discapacidad visual cuentan con una casi inexistente ayuda para nivelar sus condiciones en materia académica, a nivel superior, en comparación con alguien que no padece discapacidad visual. Se realizaron diversas pruebas que concluyeron en la necesidad de desarrollar un prototipo de asistente académico que sirva como puente de unión entre la información y la persona con discapacidad visual. El prototipo o "asistente" cuenta con tres áreas de servicio: Lectura de textos físicos, Lectura de textos digitales y Búsquedas en línea; tomando como base las necesidades básicas que un estudiante promedio de nivel superior debe suplir para estudiar y prepararse académicamente. El prototipo está diseñado para ser utilizado de manera regular y su uso se pretende que esté al alcance de todo aquel que lo necesite.

Palabras clave: Prototipo, Discapacidad visual, Asistente automatizado, Inteligencia artificial, TSS, STT, OCR, Raspado web, Educación superior, UES.

Abstract

This document contains information on the development of the research project "PROPOSAL OF A PROTOTYPE THAT ALLOWS PEOPLE WITH VISUAL DISABILITIES TO READ TEXTS USING ARTIFICIAL INTELLIGENCE FOR THE LIBRARY OF THE EAST FACULTY." which was carried out as a graduation project to qualify for the degree of Computer System Engineer. The research was carried out due to the problems which indicated that people with visual disabilities have almost no help to level their academic conditions in higher-level studies, compared to someone who does not suffer from visual impairment. Various tests were carried out that concluded in the need to develop an academic assistant prototype that serves as a bridge between the information and the person with visual impairment. The prototype or "assistant" has three service areas: Reading of physical texts, Reading of digital texts and Online searches; based on the basic needs that an average higher-level student must have to study and prepare academically. The prototype is designed to be used on a regular basis and its use is intended to be available to everyone who needs it.

Keywords: Prototype, Visually Impaired, Automated Assistant, Artificial Intelligence, TSS, STT, OCR, Web Scraping, Higher Education, UES.

Índice general

| | |
|---|----------|
| Introducción | 1 |
| 1. Generalidades | 2 |
| 1.1. Situación problemática | 2 |
| 1.2. Justificación | 3 |
| 1.3. Objetivos | 4 |
| 1.3.1. General | 4 |
| 1.3.2. Específicos | 4 |
| 1.4. Alcances | 5 |
| 1.5. Limitaciones | 6 |
| 2. Marco Referencial | 7 |
| 2.1. Marco histórico | 7 |
| 2.1.1. Historia de la Facultad Multidisciplinaria Oriental | 7 |
| 2.1.2. Unidad Bibliotecaria | 9 |
| 2.1.3. Evolución en los últimos 20 años | 10 |
| 2.2. Marco teórico | 11 |
| 2.2.1. Discapacidad visual | 11 |
| 2.2.2. Causas de la discapacidad visual | 11 |
| 2.2.3. Derechos e inclusión en El Salvador a personas con discapacidad visual | 12 |
| 2.2.4. La educación a personas con discapacidad visual | 13 |
| 2.2.5. Tiflotecnología | 16 |
| 2.2.6. De la escritura Braille a la tiflotecnología digital | 17 |
| 2.2.7. Visión artificial | 18 |
| 2.2.8. OCR | 32 |
| 2.2.9. Reconocimiento y síntesis de voz | 58 |
| 2.2.10. Herramientas voz a texto - Speech-to-Text | 67 |

| | |
|---|------------|
| 2.2.11. Herramientas texto a voz - Text-to-Speech | 73 |
| 2.2.12. Web scraping | 80 |
| 2.2.13. Lenguaje de programación | 81 |
| 2.2.14. Raspberry PI 3 | 90 |
| 3. Metodología de la Investigación | 95 |
| 3.1. Metodología | 95 |
| 3.1.1. Tipo de estudio | 95 |
| 3.1.2. Descripción del Contenido | 95 |
| 3.1.3. Población y muestra | 96 |
| 3.1.4. Sistema de hipótesis | 97 |
| 3.1.5. Operacionalización de hipótesis | 99 |
| 3.1.6. Proceso experimental | 100 |
| 3.2. Análisis e interpretación de resultados | 105 |
| 3.2.1. Tabulación de encuestas preexperimentales | 105 |
| 3.2.2. Análisis de datos preexperimentales | 117 |
| 3.2.3. Discusión de resultados | 119 |
| 3.2.4. Conclusiones | 120 |
| 4. Desarrollo, Experimentación y Resultados | 122 |
| 4.1. Desarrollo del prototipo | 122 |
| 4.1.1. Determinación de requerimientos | 122 |
| 4.1.2. Estudio de factibilidad | 122 |
| 4.2. Casos de uso | 128 |
| 4.2.1. Paquete de actividades de los actores involucrados | 128 |
| 4.2.2. Descripción de las actividades mediante casos de uso, por actor involucrado. | 129 |
| 4.3. Recursos de Hardware para el desarrollo del proyecto | 150 |
| 4.4. Desarrollo de hardware | 150 |

| | |
|---|------------|
| 4.4.1. Diseño | 150 |
| 4.4.2. Hardware | 171 |
| 4.4.3. Plataforma | 175 |
| 4.4.4. Soporte de cámara e iluminación | 177 |
| 4.5. Desarrollo de software | 198 |
| 4.5.1. Organización de archivos | 198 |
| 4.5.2. Adquisición y procesamiento de voz | 198 |
| 4.5.3. Adquisición y procesamiento de imágenes | 210 |
| 4.5.4. Web scraping o raspado web | 225 |
| 4.5.5. Procesamiento de Texto a voz | 228 |
| 4.5.6. Script individuales | 231 |
| 4.6. Estándares de código para el proyecto | 237 |
| 4.6.1. Nomenclatura | 237 |
| 4.7. Pruebas de rendimiento | 242 |
| 4.7.1. Rendimiento con datos controlados | 242 |
| 4.7.2. Valoraciones de rendimiento | 248 |
| 4.8. Experimentación | 251 |
| 4.8.1. Tabulación de encuestas experimentales | 251 |
| 4.8.2. Análisis de datos experimentales | 254 |
| 5. Conclusiones y Recomendaciones | 260 |
| 5.1. Conclusiones | 260 |
| 5.2. Recomendaciones | 262 |
| 6. Anexos | 264 |
| 6.1. Anexo: Censo de Escuelas de Educación Especial para el año 2006 | 264 |
| 6.2. Anexo: Instalación del sistema operativo | 265 |
| 6.3. Anexo: Instalación de Librerías para el funcionamiento del prototipo | 268 |

| | |
|--|-----|
| 6.4. Anexo: Manual de requerimientos para el funcionamiento de la toma de fotografías a libros de texto | 270 |
| 6.5. Anexo: Código fuente | 276 |
| 6.6. Anexo: Encuesta dirigida a personal de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador | 348 |
| 6.7. Anexo: Cuestionario dirigido a Jefatura de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador | 350 |
| 6.8. Anexo: Cuestionario dirigido a la Jefatura de la Unidad de Sistemas Informáticos de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador | 352 |

Índice de figuras

| | | |
|-------|--|-----|
| 2.1. | Iluminación según su dirección | 21 |
| 2.2. | Cámaras modernas de semiconductores | 22 |
| 2.3. | Proceso de adquisición | 24 |
| 2.4. | Generación de imágenes digitales | 25 |
| 2.5. | Ejemplo de píxeles | 27 |
| 2.6. | Ejemplo de profundidad de bits | 27 |
| 2.7. | Ejemplo de modos de color | 28 |
| 2.8. | Ejemplo de sesgado | 36 |
| 2.9. | Ejemplo de reducción de ruido | 37 |
| 2.10. | Ejemplo de binarización | 37 |
| 2.11. | Ejemplo de eliminación de líneas | 38 |
| 2.12. | Ejemplo de zonificación | 38 |
| 2.13. | Imagen a procesar en Tesseract-ocr | 47 |
| 2.14. | Imagen de referencia OCR.Space | 49 |
| 2.15. | Imagen de referencia Google Drive OCR | 55 |
| 2.16. | Imagen de referencia Computer Vision | 57 |
| 2.17. | Espectrograma de la palabra leer. | 58 |
| 2.18. | Ejemplo simplificado de nodos de una cadena de Markov. | 60 |
| 2.19. | Ejemplo de red neuronal de tres capas | 61 |
| 2.20. | Órganos generadores de sonido | 62 |
| 2.21. | Logo raspberry PI | 90 |
| 2.22. | Raspberry PI 3 Modelo B+ | 92 |
| 4.1. | Paquete usuario. | 128 |
| 4.2. | Paquete asistente. | 128 |
| 4.3. | Caso de uso: comando de voz. | 129 |

| | | |
|-------|--|-----|
| 4.4. | Caso de uso: Seleccionar tipo de uso. | 130 |
| 4.5. | Caso de uso: Leer libro físico. | 131 |
| 4.6. | Caso de uso: Hacer búsquedas en Internet. | 132 |
| 4.7. | Caso de uso: Escuchar indicaciones de uso. | 133 |
| 4.8. | Caso de uso: Leer libro digital. | 134 |
| 4.9. | Caso de uso: Pasar páginas del libro. | 135 |
| 4.10. | Caso de uso: Repetir texto. | 136 |
| 4.11. | Caso de uso: Cargar siguiente lista de resultados. | 137 |
| 4.12. | Caso de uso: Leer lista de resultados anterior. | 138 |
| 4.13. | Caso de uso: Leer página siguiente. | 139 |
| 4.14. | Caso de uso: Leer página anterior. | 140 |
| 4.15. | Caso de uso: Conectar fuente eléctrica. | 141 |
| 4.16. | Caso de uso: Iniciar prototipo. | 142 |
| 4.17. | Caso de uso: Colocar micrófono. | 143 |
| 4.18. | Caso de uso: Acomodar libro. | 144 |
| 4.19. | Caso de uso: Acomodar cámara. | 145 |
| 4.20. | Caso de uso: Corregir iluminación. | 146 |
| 4.21. | Caso de uso: Finalizar prototipo. | 147 |
| 4.22. | Caso de uso: Desconectar fuente eléctrica. | 148 |
| 4.23. | Caso de uso: Conectar a Internet. | 149 |
| 4.24. | Pieza A. | 151 |
| 4.25. | Pieza A. | 151 |
| 4.26. | Pieza B. | 152 |
| 4.27. | Pieza B. | 152 |
| 4.28. | Pieza C. | 153 |
| 4.29. | Pieza C. | 153 |
| 4.30. | Pieza D. | 154 |

| | |
|---|-----|
| 4.31. Pieza D. | 154 |
| 4.32. Pieza E. | 155 |
| 4.33. Pieza F. | 155 |
| 4.34. Pieza G. | 156 |
| 4.35. Pieza H. | 157 |
| 4.36. Soporte de cámara | 158 |
| 4.37. Cable UTP | 158 |
| 4.38. Cinta LED | 158 |
| 4.39. Batería 9V | 159 |
| 4.40. Conector batería 9V | 159 |
| 4.41. Switch On/Off | 159 |
| 4.42. Tornillos M2. | 159 |
| 4.43. Tuercas M2. | 159 |
| 4.44. Pieza I. | 160 |
| 4.45. Pieza J. | 160 |
| 4.46. Pieza J. | 161 |
| 4.47. Pernos ¼ | 161 |
| 4.48. Tuercas mariposa ¼ | 161 |
| 4.49. arandelas ¼ | 161 |
| 4.50. Switch para lámpara | 162 |
| 4.51. LED GU10 | 162 |
| 4.52. Base LED GU10 | 162 |
| 4.53. Cable duplex número 16 | 163 |
| 4.54. Enchufe tipo americano 2 puntas | 163 |
| 4.55. Pasadores de madera | 163 |
| 4.56. Pieza K. | 164 |
| 4.57. Pieza L. | 164 |

| | |
|---|-----|
| 4.58. Pieza L. | 164 |
| 4.59. Pieza M. | 165 |
| 4.60. Pieza M. | 165 |
| 4.61. Pieza N. | 165 |
| 4.62. Pieza N. | 165 |
| 4.63. Pieza O. | 166 |
| 4.64. Pieza O. | 166 |
| 4.65. Pieza P. | 167 |
| 4.66. Pieza P. | 167 |
| 4.67. Pieza Q. | 168 |
| 4.68. Pieza Q. | 168 |
| 4.69. Pieza Q. | 169 |
| 4.70. Pieza R. | 169 |
| 4.71. Pieza R. | 170 |
| 4.72. Pieza S. | 170 |
| 4.73. Pieza S. | 171 |
| 4.74. Raspberry Pi 3 modelo B. | 171 |
| 4.75. Raspberry Pi 3 Camera V2. | 171 |
| 4.76. Pantalla de 7 Pulgadas para Raspberry Pi con armazón. | 172 |
| 4.77. Cable HDMI. | 172 |
| 4.78. Conector MicroUSB macho a macho. | 172 |
| 4.79. Adaptador USB para micrófono y Audífono. | 172 |
| 4.80. Cable para audio macho a macho de 3.5mm a 6.5mm Mono. | 173 |
| 4.81. Adaptador para audio hembra a hembra de 6.5mm a XLR. | 173 |
| 4.82. Micrófono direccional con conector XLR. | 173 |
| 4.83. Ventilador para ordenador con conector estándar. | 173 |
| 4.84. Cable de carga rápida USB. | 174 |

| | |
|--|-----|
| 4.85. Mouse y teclado inalámbrico con conector USB. | 174 |
| 4.86. Cable conector de Pi Camera V2 a Raspberry Pi 3. | 174 |
| 4.87. Conectar A con B. | 175 |
| 4.88. Conectar A con C. | 175 |
| 4.89. Conectar B con D. | 175 |
| 4.90. Conectar A con E. | 176 |
| 4.91. Conectar F con G. | 176 |
| 4.92. Conectar F con H. | 176 |
| 4.93. Conectar A con F, G y H. | 177 |
| 4.94. Pinza de agarre. | 177 |
| 4.95. Corte pinza. | 177 |
| 4.96. Corte pinza. | 178 |
| 4.97. Pegar piezas I con J. | 178 |
| 4.98. Perforaciones para tornillos. | 178 |
| 4.99. Cortar cinta LED. | 179 |
| 4.100. Pegar las tuercas. | 179 |
| 4.101. Fijar Switch en J. | 179 |
| 4.102. Sacar trenzado de cable UTP. | 180 |
| 4.103. Trenzar cables. | 180 |
| 4.104. Unir circuito cintas LED con switch. | 180 |
| 4.105. Unir circuito cintas LED con switch. | 180 |
| 4.106. Introducir cable en la funda plástica. | 181 |
| 4.107. Pasar extremo del cable en la ranura de la pieza B. | 181 |
| 4.108. Unir extremo del cable con conector de batería de 9V. | 181 |
| 4.109. Colocar batería y tratar de encender la cinta LED. | 181 |
| 4.110. Pegar las piezas I sobrantes. | 182 |
| 4.111. Introducir I en la ranura B4. | 182 |

| | |
|--|-----|
| 4.112. Sujetar pinza en las piezas I. | 182 |
| 4.113. Introducir la pinza en la hendidura J1. | 182 |
| 4.114. Unir piezas K y S. | 183 |
| 4.115. Unir piezas P y L. | 183 |
| 4.116. Introducir LP en S1. | 183 |
| 4.117. Atravesar L1 y M1 con pasadores. | 184 |
| 4.118. Atravesar N1 y M4 con pasadores. | 184 |
| 4.119. Colocar arandela en L1. | 184 |
| 4.120. Colocar arandela entre N y L. | 185 |
| 4.121. Colocar arandela entre N3 y O1. | 185 |
| 4.122. Colocar arandela en el perno entre N y O. | 185 |
| 4.123. Unir Q y R. | 186 |
| 4.124. Introducir base de foco en el agujero Q1. | 186 |
| 4.125. Conectar switch a cable duplex. | 186 |
| 4.126. Atornillar conector switch. | 187 |
| 4.127. Atornillar cable duplex al enchufe. | 187 |
| 4.128. Colocar foco en su base. | 187 |
| 4.129. Introducir pieza O en la ranura R1. | 188 |
| 4.130. Conectar y probar. | 188 |
| 4.131. Colocar conectores de pantalla. | 189 |
| 4.132. Fijar pantalla. | 189 |
| 4.133. Colocar tornillos de la Raspberry PI. | 189 |
| 4.134. Colocar protector Raspberry PI. | 190 |
| 4.135. Colocar memoria MicroSD en la Raspberry PI. | 190 |
| 4.136. Conexión cable de cámara. | 190 |
| 4.137. Colocar protector medio superior Raspberry PI. | 191 |
| 4.138. Colocar el protector superior de la Raspberry PI. | 191 |

| | |
|--|-----|
| 4.139. Fijar Raspberry PI. | 191 |
| 4.140. Conectar cable HDMI. | 192 |
| 4.141. Conectar cable microUSB. | 192 |
| 4.142. Conectar receptor USB. | 192 |
| 4.143. Conectar adaptador USB para micrófono/audífono. | 193 |
| 4.144. Conectar cable de audio 3.5mm. | 193 |
| 4.145. Conectar cable de 6.5mm. | 193 |
| 4.146. Conectar micrófono al puerto XLR. | 194 |
| 4.147. Acomodar micrófono en su base. | 194 |
| 4.148. Pegar ventilador. | 194 |
| 4.149. Conectar cable USB con salida de corriente al ventilador. | 195 |
| 4.150. Conectar ventilador a la Raspberry PI. | 195 |
| 4.151. Colocar la pantalla sobre las piezas G y H. | 195 |
| 4.152. Introducir el cable de la cámara por C4 y sacarlo por B5. | 196 |
| 4.153. Conectar cámara al cable conector. | 196 |
| 4.154. Atornillar la cámara. | 197 |
| 4.155. Conectar cable de corriente a la Raspberry PI. | 197 |
| 4.156. Ejemplo de remuestreo de audio de 44,1kHz a 16kHz. | 200 |
| 4.157. Ejemplo de supresión de ruido. | 200 |
| 4.158. Espectrograma de la palabra leer | 201 |
| 4.159. Diagrama comando leer | 203 |
| 4.160. Diagrama comando libro | 204 |
| 4.161. Diagrama comando buscar en línea | 204 |
| 4.162. Diagrama comando navegación avanzar | 205 |
| 4.163. Diagrama comando navegación regresar | 205 |
| 4.164. Diagrama comando resultado | 206 |
| 4.165. Diagrama comando repetir lista. | 206 |

| | |
|---|-----|
| 4.166. Diagrama comando repetir texto. | 207 |
| 4.167. Imagen original | 212 |
| 4.168. Conversión de imagen a gris. | 213 |
| 4.169. Binarización. | 213 |
| 4.170. Marcado de caracteres. | 214 |
| 4.171. Corrección de ángulo: parte A. | 215 |
| 4.172. Corrección de ángulo: parte B. | 215 |
| 4.173. Limpieza de bordes y sombras: parte A. | 216 |
| 4.174. Limpieza de bordes y sombras: parte B. | 216 |
| 4.175. Segunda corrección de ángulo: parte A. | 217 |
| 4.176. Segunda corrección de ángulo: parte B. | 217 |
| 4.177. Escalado de la imagen. | 218 |
| 4.178. Segmentación de la imagen. | 219 |
| 4.179. Áreas mínimas de los segmentos. | 219 |
| 4.180. Orden de los segmentos. | 220 |
| 4.181. Conversión de imagen a gris. | 221 |
| 4.182. Corrección de ángulo. | 222 |
| 4.183. Diagrama del reproductor | 230 |
| 4.184. Diagrama comando leer | 235 |
| 6.1. Selección del sistema operativo | 265 |
| 6.2. Formateo de la tarjeta SD | 266 |
| 6.3. Interfaz del programa Etcher | 266 |
| 6.4. Arranque de Raspbian Buster | 267 |
| 6.5. Imágenes de muestra de requerimientos de libros. Parte A | 270 |
| 6.6. Imágenes de muestra de requerimientos de libros. Parte B | 271 |
| 6.7. Representación del ángulo del texto en las imágenes. | 272 |
| 6.8. Ejemplo de página reflectante y no reflectante. Parte A | 273 |

6.9. Ejemplo de página reflectante y no reflectante. Parte B 274

6.10. Ejemplo del desajuste en paginas. 275

Índice de tablas

| | |
|--|-----|
| 2.1. Información de alcance para OAuth 2.0 | 51 |
| 2.2. Recursos URL para descargar Google Drive API | 54 |
| 3.1. Operacionalización de hipótesis | 99 |
| 3.2. Tabla de distribución de frecuencias - encuesta a la jefatura de la biblioteca | 108 |
| 3.3. Respuestas pregunta N° 1 | 109 |
| 3.4. Respuestas pregunta N° 2 | 109 |
| 3.5. Respuestas pregunta N° 3 | 110 |
| 3.6. Respuestas pregunta N° 4 | 110 |
| 3.7. Respuestas pregunta N° 5 | 110 |
| 3.8. Respuestas pregunta N° 6 | 111 |
| 3.9. Respuestas pregunta N° 7 | 111 |
| 3.10. Respuestas pregunta N° 8 | 111 |
| 3.11. Respuestas pregunta N° 9 | 112 |
| 3.12. Respuestas pregunta N° 10 | 112 |
| 3.13. Tabla de distribución de frecuencias - encuesta a empleados de la biblioteca | 113 |
| 3.14. Tabla de distribución de frecuencias - resumen - encuesta a empleados de la biblioteca | 113 |
| 3.15. Respuestas pregunta N° 1 | 113 |
| 3.16. Respuestas pregunta N° 2 | 114 |
| 3.17. Respuestas pregunta N° 3 | 114 |
| 3.18. Respuestas pregunta N° 4 | 114 |
| 3.19. Respuestas pregunta N° 5 | 115 |
| 3.20. Respuestas pregunta N° 6 | 115 |
| 3.21. Respuestas pregunta N° 7 | 115 |
| 3.22. Respuestas pregunta N° 8 | 116 |
| 3.23. Respuestas pregunta N° 9 | 116 |

| | |
|---|-----|
| 3.24. Respuestas pregunta N° 10 | 117 |
| 3.25. Tabla de distribución de frecuencias - encuesta a estudiantes | 117 |
| 4.1. Requerimientos funcionales | 122 |
| 4.2. Lenguaje para desarrollo | 123 |
| 4.3. Sistema operativo para Raspberry PI | 124 |
| 4.4. Perfil del programador-analista | 125 |
| 4.5. Perfil del administrador | 125 |
| 4.6. Recursos humanos | 126 |
| 4.7. Recursos de desarrollo | 126 |
| 4.8. Recursos para armado del prototipo. | 127 |
| 4.9. Costo total del proyecto. | 127 |
| 4.10. Caso de uso: Hacer comando de voz | 129 |
| 4.11. Caso de uso: Seleccionar tipo de uso | 130 |
| 4.12. Caso de uso: Leer libro físico | 131 |
| 4.13. Caso de uso: Hacer búsquedas en Internet | 132 |
| 4.14. Caso de uso: Escuchar indicaciones de uso. | 133 |
| 4.15. Caso de uso: Leer libro digital. | 134 |
| 4.16. Caso de uso: Pasar páginas del libro. | 135 |
| 4.17. Caso de uso: Repetir Texto. | 136 |
| 4.18. Caso de uso: Cargar siguiente lista de resultados. | 137 |
| 4.19. Caso de uso: Leer lista de resultados anterior. | 138 |
| 4.20. Caso de uso: Leer página siguiente. | 139 |
| 4.21. Caso de uso: Leer página anterior. | 140 |
| 4.22. Caso de uso: Conectar fuente eléctrica. | 141 |
| 4.23. Caso de uso: Conectar fuente eléctrica. | 142 |
| 4.24. Caso de uso: Colocar micrófono. | 143 |
| 4.25. Caso de uso: Acomodar libro. | 144 |

| | |
|--|-----|
| 4.26. Caso de uso: Acomodar cámara. | 145 |
| 4.27. Caso de uso: Corregir iluminación. | 146 |
| 4.28. Caso de uso: Finalizar prototipo. | 147 |
| 4.29. Caso de uso: Desconectar fuente eléctrica. | 148 |
| 4.30. Caso de uso: Conectar a Internet. | 149 |
| 4.31. Recursos de hardware para el desarrollo del proyecto | 150 |
| 4.32. Características de audio requeridas por las herramientas | 199 |
| 4.33. Proceso de mejora de imagen | 211 |
| 4.34. Ejemplo del array de resultados | 225 |
| 4.35. Parámetros necesarios para en raspado web | 226 |
| 4.36. Características a extraer | 226 |
| 4.37. Modificaciones a realizar | 227 |
| 4.38. Codificación y longitud de cadenas soportadas | 228 |
| 4.39. Ratios y ganancias. | 231 |
| 4.40. Modos de ejecución. | 236 |
| 4.41. Prueba de rendimiento TTS N° 1 | 242 |
| 4.42. Prueba de rendimiento TTS N° 2 | 242 |
| 4.43. Prueba de rendimiento TTS N° 3 | 242 |
| 4.44. Prueba de rendimiento STT N° 1 | 243 |
| 4.45. Prueba de rendimiento STT N° 2 | 243 |
| 4.46. Prueba de rendimiento STT N° 3 | 243 |
| 4.47. Prueba de rendimiento STT N° 4 | 244 |
| 4.48. Prueba de rendimiento STT N° 5 | 244 |
| 4.49. Prueba de rendimiento STT N° 6 | 244 |
| 4.50. Prueba de rendimiento STT N° 7 | 244 |
| 4.51. Prueba de rendimiento STT N° 8 | 245 |
| 4.52. Prueba de rendimiento STT N° 9 | 245 |

| | |
|--|-----|
| 4.53. Prueba de rendimiento STT N° 10 | 245 |
| 4.54. Prueba de rendimiento STT N° 11 | 245 |
| 4.55. Prueba de rendimiento STT N° 12 | 246 |
| 4.56. Prueba de rendimiento STT N° 13 | 246 |
| 4.57. Prueba de rendimiento STT N° 14 | 246 |
| 4.58. Prueba de rendimiento STT N° 15 | 246 |
| 4.59. Prueba de rendimiento STT N° 16 | 247 |
| 4.60. Prueba de rendimiento STT N° 17 | 247 |
| 4.61. Prueba de rendimiento OCR N° 1 | 247 |
| 4.62. Prueba de rendimiento OCR N° 2 | 247 |
| 4.63. Prueba de rendimiento OCR N° 3 | 248 |
| 4.64. Prueba de rendimiento OCR N° 4 | 248 |
| 4.65. Prueba de rendimiento OCR N° 5 | 248 |
| 4.66. Resumen de rendimiento de conversión texto a voz | 249 |
| 4.67. Resumen de rendimiento de conversión de voz a texto | 249 |
| 4.68. Resumen de rendimiento de conversión de imagen a texto | 250 |
| 4.69. Respuesta pregunta N° 1 | 251 |
| 4.70. Respuesta pregunta N° 2 | 251 |
| 4.71. Respuesta pregunta N° 3 | 251 |
| 4.72. Respuesta pregunta N° 4 | 252 |
| 4.73. Respuesta pregunta N° 5 | 252 |
| 4.74. Respuesta pregunta N° 1 | 252 |
| 4.75. Respuesta pregunta N° 2 | 253 |
| 4.76. Respuesta pregunta N° 3 | 253 |
| 4.77. Respuesta pregunta N° 4 | 253 |
| 4.78. Respuesta pregunta N° 5 | 253 |
| 4.79. Respuesta pregunta N° 6 | 254 |

| | |
|---|-----|
| 4.80. Resultados primer instrumento aplicado | 254 |
| 4.81. Resultados segundo instrumento aplicado | 255 |
| 4.82. Escala de respuestas | 256 |
| 4.83. Respuesta lectura de textos físicos P | 256 |
| 4.84. Respuesta lectura de textos digitales P | 256 |
| 4.85. Respuesta búsqueda en línea P | 257 |
| 4.86. Respuesta lectura de textos físicos Q | 258 |
| 4.87. Respuesta lectura de textos digitales Q | 258 |
| 4.88. Respuesta búsqueda en línea Q | 258 |
| 6.1. Anexo: Censo de escuelas de educación especial para el año 2006. | 264 |

Introducción

En la actualidad de nuestra sociedad, se puede observar en el campo laboral la exigencia constante de preparación de grado superior a los postulantes en un lugar de trabajo, la preparación especializada, la experiencia laboral y buenas oportunidades de trabajo se pueden alcanzar con mayor facilidad si se posee un título universitario.

Una buena parte de la población nacional sufre de algún tipo de discapacidad visual y muchos de ellos tienen casos severos como ceguera parcial o total, para estas personas es especialmente difícil poder valerse por sí mismos en sus actividades diarias, más aún poder realizar un trabajo que les permita mantener un estilo de vida digno y seguro.

Las universidades en el país con el tiempo han desarrollado programas de inclusión que faciliten a las personas con este tipo de discapacidad poder acceder a una educación de calidad, pero, estas iniciativas aunque brindan una gran oportunidad está limitada a clases y evaluaciones, no pudiendo brindar material de estudio adecuado para personas con discapacidad visual, por lo que estas personas deben valerse de recursos propios para estudiar, la Universidad de El Salvador sufre de este problema, aunque abre sus puertas a personas con discapacidad visual para que estudien y sean profesionales en la carrera de su elección, no puede proveerles material bibliográfico de estudio que puedan utilizar, ya que los libros que pone a disposición de los estudiantes no se encuentran en braille (sistema de lectura para invidentes); esto ralentiza el proceso de aprendizaje del estudiante y complica su preparación profesional, teniendo que limitar sus fuentes de información a medios adaptados para personas discapacitadas.

Las nuevas tecnologías en reconocimiento de imágenes, procesamiento de texto y comandos por voz son de mucha ayuda para solucionar esta situación, al desarrollarse un asistente de lectura que traduzca el texto de los libros y/o computadoras a sonido, utilizando comandos de voz para simplificar la interacción entre usuario-ordenador.

1. Generalidades

1.1. Situación problemática

Las personas con discapacidad visual en contextos de acceder a la información, siempre se les ha visto limitados en esta labor, sin embargo, debido al crecimiento tecnológico que ha tenido la humanidad, y el acceso a tecnologías como lo son los teléfonos inteligentes, ha provocado el vuelco de gran cantidad de información en versión digital, sumándose los ya existentes en versiones físicas como los libros, revistas, periódicos, etc.

En la Facultad Multidisciplinaria de Oriente en la actualidad no se brindan servicios adecuados para poder atender a personas con discapacidad visual. Las personas afectadas con este padecimiento se quejan mayormente por no ser independientes a la hora de informarse, siempre dependen de alguien más para acceder a la información, ya que no hay dispositivos específicos para que una persona no vidente pueda hacer uso a su antojo sin requerir de mucha intervención de alguien más.

La discapacidad visual es un gran inconveniente para poder acceder libremente al contenido digital, hay que tener en cuenta que este cambia día con día, ya que siempre hay contenido nuevo para leer, dificultando así mantenerse actualizado, existen alternativas como lo son los audiolibros, pero conseguir este tipo de material es bastante complicado, aún para una persona con todas sus facultades físicas funcionando correctamente.

Es preciso levantar información cualitativa respecto a la lectura de textos tanto digitales como físicos, con la ayuda de Inteligencia Artificial, ya que en la actualidad no existe una investigación bastante clara que pueda resolver la dificultad de las personas con discapacidad visual para informarse, por tal motivo es sumamente relevante analizar esta temática para obtener y analizar resultados claros y contundentes, que produzcan una muy buena base para llevar a realidad los alcances esperados del proyecto, que es facilitar las actividades que estas personas realizan a diario y así mejorar su calidad de vida.

1.2. Justificación

Hoy en día, toda la información existente, está al alcance desde cualquier parte del mundo, tan lejos como presionar un botón o abrir las páginas de un libro, y las personas durante búsqueda personal conocimiento o de superación académica, la interacción con este mar de palabras es inevitable. Ya sea por necesidad o solo porque si; el lograr esta superación, en forma de un título universitario, significa asimilar incontables textos, acceder a numerosos libros, páginas web, artículos, etc; por lo que la “lectura” es fundamental.

En El Salvador hay muchas personas que sufren discapacidad visual, es decir, tienen una afección parcial o total de ceguera, por lo que los estudios, a cualquier nivel educativo, se convierten en un reto muy grande para ellos. El gobierno del país en los últimos años ha avanzado en materia de inclusión de estas personas en el sistema educativo nacional, como desarrollo de materiales especializados (libros en braille, en formato de audio, etc.), adaptación de infraestructuras adecuadas y muchas cosas más. La mayoría, si no es que todos, los esfuerzos de inclusión se concentran en los niveles básico y medio de educación, por lo que el nivel superior o universitario está quedando un tanto relegado en este tema. Debido a estas razones es que se ha decidido enfocar la implementación del prototipo en el nivel educativo superior; para dar a los estudiantes una herramienta más en su travesía de superación académicas.

Se a tomado como lugar de implementación la Unidad Bibliotecaria de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador (FMO-UES), siendo esta Facultad una rama institucional que brinda la educación necesaria para preparar académicamente a un profesional en una o diversas áreas del conocimiento; siendo la Unidad Bibliotecaria en centro principal de acceso a la información que esta Facultad pone a disposición de sus estudiantes, la implementación del prototipo en este lugar se considera idónea.

El motivo por el cual puede utilizarse lectura de textos usando inteligencia artificial es que la gran mayoría de la información académica que la FMO-UES tiene a su disposición se encuentra en formato de texto, ya sea físico o digital, este texto es inalcanzable de manera regular para una persona con discapacidad visual, por lo que necesitan un “asistente” que les transmita el contenido; regularmente sería una persona sin problemas de vista quien desempeñaría el papel del asistente, leyendo los textos para la persona con discapacidad, pero esto reduce en gran manera el estado de independencia del asistido y duplica los esfuerzos invertidos. El uso de inteligencia artificial pretende minimizar los esfuerzos, es decir, hacer el papel del asistente y reducir la cantidad de personas involucradas a la mitad, para obtener el mismo resultado.

1.3. Objetivos

1.3.1. General

Facilitar a personas con discapacidad visual el acceso a la información mediante un prototipo asistido por computadora, que facilite la lectura de textos físicos y digitales.

1.3.2. Específicos

- Aplicar técnicas de visión artificial para reconocimiento de textos, que facilite a personas con discapacidad visual la lectura de textos físicos y digitales.
- Diseñar una interfaz de comandos de voz que permita la interacción entre el ordenador y el usuario en un ambiente controlado.
- Desarrollar una aplicación de software que funcione como controlador para lograr la comunicación entre el hardware y software, con el fin de mejorar la experiencia del usuario.
- Contribuir al mejoramiento de la calidad de vida de las personas con discapacidad visual, mediante el aprendizaje asistido por computadora.

1.4. Alcances

- Ejecutar una práctica de la metodología planteada en el estudio que consistirá en el diseño del prototipo y la elaboración del mismo para la obtención de datos de referencia para la investigación en la biblioteca de la Facultad Multidisciplinaria de Oriente.
- Incorporar tecnología de reconocimiento de voz para la ejecución de comandos en un ambiente controlado definido previamente con el personal a cargo de la biblioteca.
- Realizar el procesamiento de imágenes para la obtención de textos a partir de imágenes tomadas mediante cámara en tiempo real utilizando hardware de bajo costo.
- Procesar el texto recogido mediante el prototipo y transformarlo a audio utilizando una API para su posterior reproducción.
- En el proceso de diseño y creación del prototipo se estructuro paralelamente la documentación necesaria para construir y utilizar dicho prototipo.
- Se realizaron pruebas de funcionamiento del prototipo en personas con discapacidad visual para la obtención de información que sea la base de posibles mejoras o correcciones respecto al diseño, funcionamiento o manejo del mismo.
- Debe capacitarse al personal que gestionara dicho prototipo para su uso correcto.

1.5. Limitaciones

- El estudio se realizó dentro de los límites de la Facultad Multidisciplinaria de Oriente.
- Debido a las características del prototipo, este podrá ser utilizado únicamente de manera controlada en una zona específica que debe ser adecuada para su propósito en especial.
- La decisión de implementar, utilizar y mantener dicho prototipo recae enteramente en la universidad.
- Aun siendo funcional dicho prototipo, es probable que no sea utilizado por personas a las cuales es destinado el mismo.
- Al tratarse de un prototipo es probable que el mismo pueda tener fallas al tratarse de una implementación nueva en lo que respecta a hardware y software.
- La tecnología utilizada en el prototipo debe ser tecnología de fácil y económico acceso, así como bien documentada y actualizada.

2. Marco Referencial

2.1. Marco histórico

2.1.1. Historia de la Facultad Multidisciplinaria Oriental

El 17 de junio de 1966, en sesión No. 304, el Consejo Superior Universitario fundó el Centro Universitario de Oriente (C.U.O) en la ciudad de San Miguel, como una extensión de los estudios universitarios de la Universidad de El Salvador.

Los Objetivos principales que motivaron su creación fueron:

- Ampliar la capacidad de servicio docente de la Unidad.
- Satisfacer las necesidades educativas y culturales de la Zona Oriental.
- Contribuir con el desarrollo y progreso.
- Crear los instrumentos técnicos y culturales a sectores de la población que no tienen acceso a la Educación Universitaria.
- Descentralizar los servicios de Educación Superior.

En Abril de 1967, se adquirió un terreno de 108 manzanas de extensión en el Cantón el Jute a 6.5 kms. Al Sur Oriente de la Ciudad de San Miguel, donde se pretendía construir el Campus Universitario del CUO. Las actividades académicas se iniciaron el 17 de mayo de 1969 y el primer Director fue el Dr. José Vinnatea. Las actividades académicas se iniciaron a través de tres Departamentos que impartían el servicio de áreas comunes a todas las carreras de la Universidad. Los Departamentos de los cuales estaba constituida la Universidad eran:

- Departamento de Física y Matemática
- Departamento de Ciencias Biológicas y Química.
- Departamento de Ciencias Sociales, Filosofía y Letras

La asignación presupuestaria inicial fue de ₡96,582.00 para cubrir costos Docentes y Administrativos.

En 1988 el Consejo Superior Universitario aprobó el Reglamento de Gobierno de los Centros Regionales, el cual establece una nueva cultura académica administrativa que permitiría ampliar su capacidad de servicio; creándose los departamentos homólogos o las facultades, exceptuando odontología y permitiéndole crecer de manera espontánea las diferentes carreras que hoy se tienen.

Es importante conocer que el acuerdo de creación del Centro Universitario de Oriente contenía 20 Artículos, pero es llamativo el Art. 11 que decía: “El consejo Directivo del Centro estará compuesta por el Director, por los Decanos de las Facultades que existan en el Centro, por un Representante de cada cuerpo de Profesores de los Departamentos Locales que no forman parte de una Facultad y por Representantes Estudiantiles.”

El Centro Universitario de Oriente se denominó “Facultad Multidisciplinaria Oriental”, el 4 de junio de 1992 según acuerdo 39-91-95-IX del Consejo Superior Universitario, con todas las atribuciones y deberes del resto de facultades. La Facultad Multidisciplinaria Oriental es una unidad regional de la Universidad de El Salvador creada para extender los servicios de Educación Superior a la población de la zona oriental. Dicha Facultad está organizada académicamente en los siguientes departamentos:

- Departamento de Medicina.
- Departamento de Ciencias Agronómicas.
- Departamento de Ciencias Jurídicas.
- Departamento de Ciencias Económicas.
- Departamento de Química y Farmacia.
- Departamento de Ciencias Naturales y Matemática.
- Departamento de Ciencias y Humanidades.
- Departamento de Ingeniería y Arquitectura.

Además de los departamentos de la facultad cuenta con las siguientes Unidades Administrativas:

- Administración General.
- Administración Financiera.
- Administración Académica.
- Unidad de Proyección Social.

- Unidad de Planificación.
- Unidad Bibliotecaria.

Los órganos directrices de esta facultad son:

- **Órganos de gobierno:** Junta Directiva y Decano.
- **Funcionarios ejecutivos:** El Vicedecano.
- **Organismos asesores:** La Asamblea del Personal Académico, El Comité Técnico Asesor y los contemplados en el reglamento de cada facultad.
- **Funcionarios Auxiliares:** Secretario de la Facultad.

2.1.2. Unidad Bibliotecaria

La Biblioteca de la FMO-UES, forma parte del Sistema Bibliotecario de la Universidad de El Salvador (SIBUES), conjunto de bibliotecas de las diferentes facultades que constituyen la Universidad de El Salvador.

La Bibliotecaria de la Facultad es la responsable de apoyar el desarrollo curricular de la Facultad Multidisciplinaria Oriental mediante la selección, adquisición y catalogación, de los diferentes materiales organizados para las disciplinas albergadas por la FMO-UES.

La Biblioteca funciona actualmente en una edificación de 2 niveles de 4,090 mts², en ella se encuentran ubicadas:

Primer Nivel:

- El Departamento de Servicios al Público (Circulación y Préstamo, Colecciones Generales, Hemeroteca-Mapoteca-Tesario, Referencia, Sala de Estudio Colectiva, Sala de Estudio Individual, Catálogo en Línea de Acceso al Público, Videoteca, Sala de Internet).
- Área de Conservación y Preservación.
- Sala de Internet y Mediateca.
- Jefatura.

Segundo Nivel:

- Departamento de Procesos Técnicos y Desarrollo de las Colecciones.
- Departamento de Sistemas Informáticos.

- Área de Servidores.
- Área de Reparación y Mantenimiento de Equipos.

2.1.3. Evolución en los últimos 20 años

En términos generales, la prevalencia de la discapacidad visual ha disminuido desde comienzos de los años noventa. Esa disminución se asocia a:

- El desarrollo socioeconómico en general.
- Una actuación concertada de salud pública.
- Un aumento de los servicios de atención oftalmológica disponibles.
- El conocimiento por parte de la población general de las soluciones a los problemas relacionados con la discapacidad visual (por ejemplo, cirugía o dispositivos correctores).

Sin embargo, se calcula que el número de personas con discapacidad visual podría triplicarse debido al crecimiento de la población y a su envejecimiento. Por ejemplo, para 2050 podría haber 115 millones de personas ciegas, en comparación con los 38,5 millones de 2020¹.

¹Seth R Flaxman et al. "Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis". En: *The Lancet Global Health* 5.12 (2017), e1221-e1234.

2.2. Marco teórico

2.2.1. Discapacidad visual

La discapacidad visual se define con base en la agudeza visual de la vista de los ojos, así como el campo visual. Se habla de discapacidad visual del ojo cuando existe una disminución significativa de la agudeza visual del ojo aún con el uso de lentes, o bien, una disminución significativa del campo visual del ojo.

La discapacidad visual puede originarse por un inadecuado desarrollo de los órganos visuales o por padecimientos o accidentes que afecten los ojos, las vías visuales o el cerebro. El inadecuado desarrollo en la gestación da como resultado esta discapacidad.

La dificultad que presentan algunas personas para participar en actividades propias de la vida cotidiana, que surge como consecuencia de la interacción entre una dificultad específica relacionada con una disminución o pérdida de las funciones visuales y las barreras presentes en el contexto en que desenvuelve la persona.

2.2.2. Causas de la discapacidad visual

Según estimaciones recientes¹, las principales causas mundiales de discapacidad visual moderada a grave son:

- Errores de refracción no corregidos: 53 %
- Cataratas no operadas: 25 %
- Degeneración macular relacionada con la edad: 4 %
- Retinopatía diabética: 1 %

Las principales causas de ceguera son:

- Cataratas no operadas: 35 %
- Errores de refracción no corregidos: 21 %
- Glaucoma: 8 %

¹Rupert RA Bourne et al. "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis". En: *The Lancet Global Health* 5.9 (2017), e888-e897.

2.2.3. Derechos e inclusión en El Salvador a personas con discapacidad visual

Así como en muchos países del mundo, la integración plena de personas con discapacidad visual en la sociedad a sido un tema de interés en El Salvador, tanto para personas que sufren el problema, como organizaciones sin fines de lucro e instituciones del Estado, por lo que en abril de 2014 el Gobierno de El Salvador presentó la Política Nacional de Atención Integral a las Personas con Discapacidad¹.

En la búsqueda del cumplimiento de los derechos de las personas con discapacidad, mandatada en la Legislación y demarcando la necesidad urgente de eliminar prácticas discriminatorias que conduzcan a proceso de inclusión real y efectivo de las personas con discapacidad en El Salvador, se realizaron consultas públicas a nivel nacional para identificar las necesidades sentidas y percibidas con la finalidad de construir objetivos, estrategias y líneas de acción que permitan la reducción de las brechas de la desigualdad en aspectos de derechos humanos y de inclusión social de las personas con discapacidad².

En El Salvador la PNAIPD se basa en los estatutos definidos en la Convención sobre los Derechos de las Personas con Discapacidad y su Protocolo Facultativo realizada en la Organización de las Naciones Unidas (ONU) en el 2006, afirmando en su artículo uno, inciso primero que se debe “promover, proteger y asegurar el goce pleno y en condiciones de igualdad de todos los derechos humanos y libertades fundamentales de todas las personas con discapacidad y promover el respeto de su dignidad inherente”³.

En las consideraciones generales de discapacidad que se establecen en El Salvador se considera la discapacidad visual como deterioro o falta de la función sensorial de ver⁴.

En El Salvador, en términos generales, las personas con discapacidad se han enfrentado históricamente a situaciones y actos de exclusión y discriminación por parte de la sociedad, lo cual ha limitado el goce pleno y oportuno de sus derechos y libertades fundamentales, como la salud, educación, empleo, entornos físicos, información, comunicaciones, justicia y seguridad social, entre otros, en el camino hacia una vida autónoma e independiente.

La situación actual de las personas con discapacidad, en El Salvador, está caracterizada por limitaciones o falta de medidas concretas de equiparación de oportunidades, entendidas éstas como aquellas que facilitan el acceso

¹ PN.

² PNAIPD. *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*. República de El Salvador, 2014, pág. 1.

³ ONU. *Convención sobre los Derechos de las Personas con Discapacidad*. Artículo 1. Organización de las Naciones Unidas, 2006.

⁴ PNAIPD, *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*.

efectivo y sin discriminación de ninguna índole, a los servicios generados por el sistema, entendido éste como el Estado, el Mercado y la Sociedad en general, y de los cuales tienen derecho todas las personas: educación, empleo, transporte, espacio público, vivienda, información, comunicación, deporte, recreación, cultura, entre otros; servicios que facilitan el disfrute de la vida plena en términos de bienestar económico, social, laboral, salud, educación, seguridad, justicia y otros, que permitan la independencia y autonomía en la condición particular de la persona con discapacidad¹.

2.2.4. La educación a personas con discapacidad visual

La niñez y adolescencia con discapacidad, enfrenta mayor desventaja por su condición en cuanto al acceso a los servicios de atención temprana a la salud, habilitación y rehabilitación como también a la educación².

En el ámbito de la educación se han realizado esfuerzos importantes en la búsqueda de mejores respuestas a las necesidades que la población con discapacidad demanda en materia educativa. No obstante algunos avances, todavía resultan insuficientes, ya que las personas con discapacidad aún se enfrentan con barreras para el acceso a la educación, permanencia y egreso. Algunas dificultades se acrecientan por el hecho de que los planes de formación docente no han tomado en cuenta el enfoque de atención a la diversidad y educación inclusiva, y las acciones de actualización a docentes en servicio, se encuentran en una implementación gradual³.

Los niños con discapacidad no tienen las mismas oportunidades de acceso al nivel de educación inicial en igualdad de condiciones que el resto de la población, debido a la falta de un programa más efectivo de inclusión educativa para niños y niñas, esto ha propiciado que generalmente se incorporen a las instituciones educativas a mayor edad, con el agravante de que no son integrados en sus niveles correspondientes según su edad y se encuentran constantemente sin el apoyo de facilitadores, asimismo los procesos de habilitación o rehabilitación están separados de los educativos⁴.

La falta de adecuación a la infraestructura en los centros educativos es una de las barreras más evidentes. En algunos casos, los niños, aún habiendo ingresado a la escuela enfrentan dificultades debido a la falta de sensibilidad,

¹ PNAIPD. *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*. República de El Salvador, 2014, pág. 4.

² PNAIPD. *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*. República de El Salvador, 2014, pág. 7.

³ PNAIPD. *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*. República de El Salvador, 2014, pág. 4.

⁴ PNAIPD. *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*. República de El Salvador, 2014, pág. 4.

de adecuaciones curriculares y criterios de evaluación para el aprendizaje que garanticen el desarrollo de sus potencialidades¹.

En los esfuerzos de mejorar la realidad que se vive en El Salvador para estar personas, han aparecido diversas instituciones que están encaminadas a trabajar para la inclusión y desarrollo pleno de personas con discapacidad visual en materia de educación, tanto instituciones gubernamentales como no gubernamentales son, al día de hoy, aquellas que pueden hacer una diferencia en favor de las personas que sufren esta discapacidad, entre estas se encuentran:

- **Asociación de Ciegos de El Salvador (ASCES):** Fundada en 1992 por un grupo de pacientes del centro de rehabilitación para ciegos de El Salvador, esta desarrolla diversos programas en materia educativa como lo es el Proyecto de Apoyo Itinerante de Estudiantes Ciegos integrados en la Escuela Regular y Dotación de Material Adaptado en el que se brinda material educativo adecuado a estudiantes ciegos y apoyo pedagógico a docentes de la institución.
- **Centro de Rehabilitación de Ciegos “Eugenia de Dueñas”(CRC):** Fundado en 1943 por parte del Instituto Salvadoreño de Rehabilitación Integral² que cuenta con una serie de servicios en pro de la habilitación y rehabilitación de personas con discapacidad visual en su integración en la sociedad, cuenta con servicios como, rehabilitación funcional (enseñanza del método braille, adiestramiento manual, orientación y movilidad), terapia sensorial, educativa, musical y física (diseñadas para lograr en niños de tres a quince años el desarrollo de las sensopercepciones, apegado a la escala de maduración de un niño sin discapacidad).
- **Consejo Nacional de Atención Integral a la Persona con Discapacidad (CONAIPD):** Es una entidad descentralizada de la Presidencia de la República de El Salvador que por Ley es el ente Rector de la Política Nacional de Equiparación de Oportunidades para las Personas con discapacidad. El CONAIPD surge en diciembre del año de 1993 y tiene como labor asesorar, promover, educar y divulgar sobre los derechos de las personas con discapacidad, desde un esquema de derechos humanos, inclusión y respeto, además de velar por el cumplimiento de los marcos legales que amparan a las personas con discapacidad.
- **Escuelas de Educación Especial (EEE):** Son instituciones de Gobierno que tienen como enfoque especial

¹ PNAIPD. *Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador*. República de El Salvador, 2014, pág. 4.

² ISRI. *Memorándum UAIP ISRI, El Salvador*. República de El Salvador, 2014.

la educación de personas con diferentes tipos de discapacidad, como mentales, físicas o sensoriales, estas son instituciones que buscan la integración de las personas con discapacidad en la sociedad productiva del país, así como brindarles la oportunidad de una vida plena e independiente (Ver anexo 6.1).

En El Salvador la educación de personas con discapacidad visual y el completo desarrollo de estas, en lo académico, hasta su culminación en la obtención de un título de educación superior es bastante escaso, esto se refleja en la cantidad de estudiantes universitarios que han habido en los últimos años con discapacidad visual en comparación con la cantidad de estudiantes que no sufren una discapacidad.

En El Salvador a partir de la implementación de la Ley de Educación Superior en 1995 se a desarrollado el proceso de Información Estadística de Educación Superior, responsabilidad que está a cargo directamente de la Gerencia de Evaluación e Información Estadística de la Dirección Nacional de Educación Superior, cada año refleja el estado numérico de estas instituciones, como la cantidad de estudiantes, las ramas de estudio más solicitadas y muchas otras, sin embargo, hasta el 2016 se incluyó por primera vez información referente a personas con discapacidad, aunque es un avance muy importante, las clasificaciones que se establecieron para reflejar los tipos de discapacidad son muy generales (discapacidad física, intelectual, sensorial y mental o psíquica), un ejemplo, es que la discapacidad visual está contenida en la categoría de “discapacidad sensorial” que incluye discapacidades de vista, oído y habla; para uso práctico (ya que no se cuenta con valores más específicos), se utiliza este valor para reflejar el estado de las personas con discapacidad visual en El Salvador en la educación superior.

Según El Ministerio de Educación (MINED), a nivel nacional solamente 582 personas con discapacidad sensorial estudiaron a nivel superior para el año 2016¹, de las 76 209 personas con discapacidad visual mayores de 18 años registradas en el interior del país para ese mismo año² según el Registro Nacional de Personas Naturales (RNPN), siendo este un 0.7637 % de la población con discapacidad visual (ya el valor de discapacidad sensorial incluye más que solo discapacidades visuales, el porcentaje real que este representa de la discapacidad visual es mucho menor); por otra parte se registraron 180 373 estudiantes que no tenían algún tipo de discapacidad sensorial³, estudiando a nivel superior en el 2016, de las 3 542 907 personas sin discapacidad visual entre las edades de 18 y 60

¹MINED. *Resultados de la Información Estadística de Instituciones de Educación Superior 2016*. Ministerio de Educación, República de El Salvador, 2016.

²RNPN. *Datos Estadísticos de Personas con Discapacidad a Marzo del 2016*. Registro Nacional de las Personas Naturales, República de El Salvador, 2016.

³MINED, *Resultados de la Información Estadística de Instituciones de Educación Superior 2016*.

años registradas ese mismo año¹, se obtiene un 4.9839 % que es la población que estaba estudiando a nivel superior y no tenía una discapacidad visual. Estos datos reflejan que la cantidad de personas con discapacidad visual en el país que asiste a un centro de educación superior es mucho menor que el de las personas que asisten y no tienen una discapacidad visual.

2.2.5. Tiflotecnología

Durante siglos las personas con ceguera han enfrentado limitaciones para acceder a la información y la educación, cuando no fueron directamente excluidas. Primero se apoyaron en lectores, copistas y relatos orales; desde el siglo pasado el sistema Braille les permitió cierto grado de autonomía en la lectoescritura². En la actualidad, la revolución digital promete igualdad de oportunidades técnicas: los programas lectores de pantalla permiten escuchar los libros sin una capacitación especial y desde varios dispositivos.

Las tiflotecnologías³ (tifo=ciego) basadas en TIC se han convertido en un puente indispensable entre las personas ciegas y los libros. Sin embargo, la lectura está muy lejos para la mayoría de los 300 millones de discapacitados visuales (y la ceguera aumenta con la población creciente y envejeciendo)⁴. Sucede que las innovaciones, que en teoría podrían igualar el acceso al conocimiento, enfrentan barreras para alcanzar a los sectores menos favorecidos. Una de ellas es la falta de materiales accesibles, que ha sido denominada “hambre de libros” por la Unión Mundial de Ciegos: en 2012 se estimaba que sólo 1 % de los libros en el mundo se ofrece en copias accesibles.

Se puede ver la correlación entre el desarrollo de la ciencia moderna y los desarrollos en las incipientes tecnologías para ciegos, en la ubicación de estas innovaciones y de las personalidades que los llevaron a cabo: primero Italia, luego Francia e Inglaterra, tal como las primeras instituciones de la revolución científica.

Los primeros textos específicamente sobre ciegos fueron escritos en 1646, en Italia, en forma de carta firmada con iniciales que nunca se identificaron; luego en forma de libro, llegó a Francia, donde tuvo mayor difusión. En 1670 se publica un libro de un jesuita, Francesco Lana Terzi, sobre la educación de las personas ciegas, que propone, siguiendo las ideas de Girolano, del siglo anterior, que los ciegos lean al tacto siguiendo un código⁵.

¹DIGESTYC. *Encuesta de Hogares de Propósitos Múltiples 2016*. Dirección General de Estadística y Censos, República de El Salvador, 2016.

²Gabriel Farrell. *The story of blindness*. Harvard University Press, 1956.

³Tiflotecnología – Web de la ONCE. 2018. URL: <https://www.once.es/servicios-sociales/tecnologiayrecursosadaptados/tiflotecnologia> (visitado 20-05-2018).

⁴A Foster y S Resnikoff. “The impact of Vision 2020 on global blindness”. En: *Eye* 19.10 (2005), pág. 1133.

⁵Farrell, *The story of blindness*.

2.2.6. De la escritura Braille a la tiflotecnología digital

Los cambios en el acceso a la información y las posibilidades de integración de las personas ciegas a partir de los desarrollos de los siglos XVII y XVIII, dieron lugar a que se creará un campo incipiente de “tiflotecnología”, en el que participa una trama de actores sociales más amplia, en la que se incluyen instituciones públicas, docentes, especialistas, inventores y empresas, que se suman a los esfuerzos aislados de las propias personas con discapacidad. La tiflotecnología o tecnologías adaptativas para ciegos buscan solucionar algunos de los problemas habituales de esta población para integrarse plenamente a la sociedad, con dispositivos, técnicas y herramientas que buscan facilitar la vida cotidiana de las personas con alguna discapacidad visual. En particular, el acceso a la información es uno de los focos de estas tecnologías, ya que podemos considerar leer y escribir como las tecnologías básicas para la educación. Las innovaciones recientes de mayor impacto en este campo han sido en relación con la lectoescritura, y en menor medida con la movilidad, como es el caso del bastón blanco, adoptado a partir de los años 30, no sólo para la orientación sino también para la identificación ante quienes ven. Muchos de los desarrollos en tiflotecnología durante el siglo XX tuvieron que ver con la difusión masiva del sistema Braille, la adaptación a diversos idiomas y necesidades, y la simplificación de su escritura y lectura: máquinas de escribir, braille recargable, nuevas formas de impresión de libros, entre otros.

En las últimas décadas del siglo XX, comienzan a abrirse nuevos horizontes para la accesibilidad de las personas ciegas a la información a través de las TIC, en forma autónoma y masiva.

En la actualidad una personas ciega puede acceder a la computadora a través de programas lectores de pantalla, que por medio de una voz robotizada reproducida por los parlantes de una PC, laptop o celular transmiten la información que aparece visible en el monitor.

El acceso a la informática significó un gran salto en la autonomía, permitiendo, por ejemplo, leer el diario del día a través de Internet, algo que hasta hace pocos años era imposible; o leer un libro cualquiera a través de un scanner.

El scanner toma la imagen del papel que en él se coloque y la transfiere a la computadora. Luego un programa llamado de OCR (Reconocimiento Óptico de Caracteres) procesa la imagen y la convierte en texto reconocible por los procesadores de texto, y por ende por los programas lectores de pantalla. Es así como la lectura por medios electrónicos se ha convertido, en los últimos años, en una importante posibilidad de acceso a textos

además del Braille o las grabaciones con voces humanas.

El Royal National Institute for the Blind principal institución para ciegos del Reino Unido, ha declarado que “Internet es uno de los desarrollos más significativos desde la invención del Braille, por primera vez en la historia mucha gente ciega o con discapacidad visual puede acceder a la misma cantidad de información y en las mismas condiciones que una persona que ve.” Sin embargo, solucionadas en parte las barreras técnicas, nuevas barreras económicas, sociales y educativas aún mantienen a un alto porcentaje de las personas con discapacidad visual al margen de estas innovaciones que pueden abrirles las puertas a una vida plena.

2.2.7. Visión artificial

2.2.7.1. Introducción

El desarrollo de la industria en procesos relacionados a la visión por computadora ha sido impulsado en favor de las necesidades de automatización de actividades normalmente desempeñados por humanos tales como detección, reconocimiento de caracteres, clasificación de formas; generalmente realizados de modo más rápido, más controlado y automático, uno de estos campos es el relacionado a la transcripción de textos físicos a digitales.

2.2.7.2. Definición

La visión artificial forma parte de un conjunto aún mayor de aplicaciones que engloba la inteligencia artificial, este es un grupo de teorías, métodos y técnicas que permiten emular el proceso de visión biológica, a su vez la posibilidad de obtener, procesar y analizar de manera automática información de las imágenes obtenidas. La visión artificial permite la creación de algoritmos, métodos y aplicaciones para la interpretación del significado de una imagen.

Los procesos controlados por la visión artificial pueden definirse en tres niveles de acuerdo a sus niveles de complejidad e implementación de los mismos¹.

- **Procesos de bajo nivel:** Utiliza generalmente operaciones tales como la captura de las imágenes, el preprocesamiento de las imágenes para reducir el ruido, la mejora del contraste de las mismas, así como filtros de enfoque, dichos procesos se caracterizan porque sus entradas son imágenes al igual que sus salidas.
- **Procesos de nivel medio:** Utiliza procesos tales como la segmentación y la clasificación de objetos

¹Rafael C Gonzalez y Richard E Woods. “Digital Image Processing, Prentice Hall”. En: *New Jersey* (2002).

individuales, esto con el fin de transformarlos en una forma adecuada para su tratamiento informático, para su reconocimiento y clasificación, estos se caracterizan porque sus entradas generalmente son imágenes, pero sus resultados suelen ser características de las imágenes tales como bordes, contornos, forma o identidad de las formas presentes en la imagen.

- **Procesos de alto nivel:** Utiliza operaciones encargadas de reconocer un conjunto de objetos en la imagen además de realizar las funciones cognitivas que generalmente se asocian a la visión, también es conocido como el proceso de interpretación.

Aunque pueda parecer que estos procesos forman una secuencia esto no es estrictamente necesario y puede generarse interacciones entre los diferentes niveles tales como retroalimentaciones de los niveles superiores a los inferiores.

2.2.7.3. Elementos

2.2.7.3.1. Iluminación

La iluminación es necesaria para el desarrollo de sistemas de visión artificial para la correcta obtención de resultados óptimos, al elegir la iluminación adecuada se puede obtener una mejor y mayor exactitud en las mediciones, una mayor confiabilidad y un menor tiempo de procesamiento y operación.

Los objetivos principales en lo que respecta a los sistemas de visión artificial son: mantener la intensidad y la dirección de luz de manera constante y la optimización en el contraste para diferenciar los objetos del fondo.

Comúnmente en los sistemas dedicados a la visión artificial sea para procesos industriales o de laboratorio se suele contar con un diseño de iluminación adecuado que se encargue de resaltar las características de los objetos que se pretenden analizar. Al contar con un sistema de iluminación adecuado no es necesario la corrección posterior de iluminación mediante algoritmos, por tanto, un buen sistema de iluminación conlleva un menor tiempo de procesamiento pues permite resaltar las características deseadas e independizarlas del entorno.

2.2.7.3.2. Tipos de iluminación

Los tipos de iluminación pueden clasificarse básicamente de acuerdo a los siguientes aspectos: intensidad, dirección y su fuente de origen.

Dependiendo de la intensidad

Cuando la intensidad de la luz es variante se obtienen diferentes efectos resultantes en la imagen obtenida, ejemplo de ello al utilizar una intensidad de luz fuerte genera la aparición de grandes contrastes entre las zonas más claras y las más oscuras, al mismo tiempo, debido al limitado rango dinámico de esta se pierden detalles en la imagen tanto en las zonas claras como en las oscuras. El efecto opuesto al utilizar una intensidad de luz más suave se pueden apreciar mejor los detalles de una imagen en ambas regiones claras y oscuras, pero con ello se pierden detalles en las texturas de las mismas.

Dependiendo de la dirección

- **Frontal:** en esta se requiere iluminar el objeto de manera frontal, al hacerlo la luz incide de forma directa en el objeto, permite detallar las características externas de los objetos sea su forma, colores o superficies, este es el tipo de iluminación más utilizados, pero en ciertos casos no se logra obtener un buen contraste entre el objeto y el fondo debido a los posibles reflejos o sombras que puedan aparecer.
- **Posterior:** Se basa en colocar el objeto entre la cámara y la fuente de luz, este tipo de iluminación es adecuado para el reconocimiento y medición de objetos mediante la detección de bordes, debido al alto contraste entre el objeto y el fondo, generalmente se pierden detalles en la escena, una de sus mayores desventajas es cuando existen objetos posicionados uno detrás de otro.
- **Direccional:** se basa en proyectar la luz de manera direccionada en algún punto en el espacio para poder destacar características determinadas del objeto, la dirección de la luz está hacia el objeto, lo cual genera sombras sobre dicho objeto, lo que a su vez aumenta el contraste en las partes de los objetos si estos son tridimensionales, lo cual permite obtener más información tridimensional, este tipo de iluminación se utiliza más comúnmente en la localización y reconocimiento, la inspección de superficies, etc.
- **Estructurada:** se basa en la proyección de patrones modulados de iluminación sobre un objeto para así adquirir información sobre su superficie utilizando el reflejo de la luz, este tipo de iluminación se utiliza generalmente en reconstrucción de objetos en 3D así como en el reconocimiento de formas.

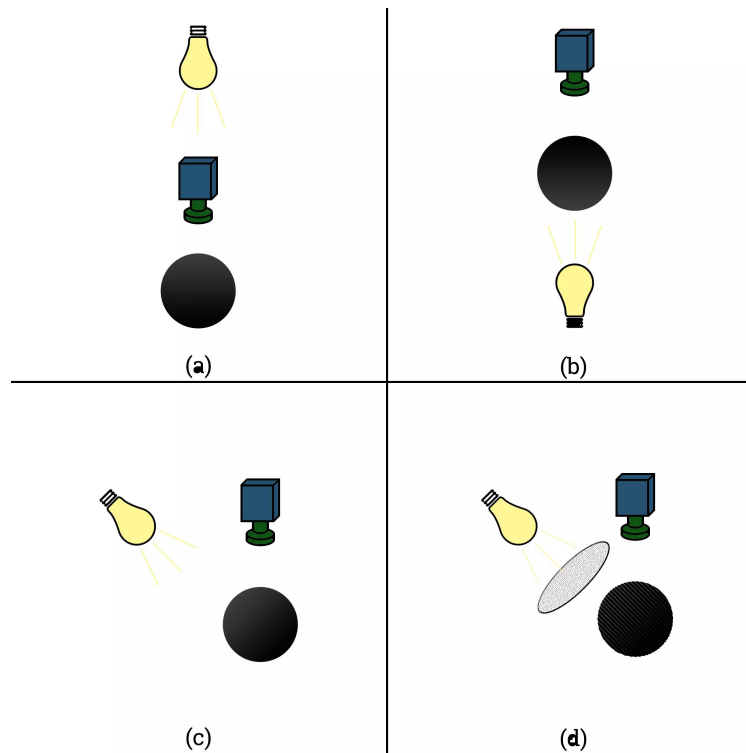


Figura 2.1: Iluminación según su dirección: (a) Frontal, (b) Posterior, (c) Direccional, (d) Estructurada - Fuente: Elaboración propia.

Dependiendo de la fuente

- Incandescente:** Fue la primera fuente de luz de origen eléctrico y es también la fuente más común de iluminación, la fuente produce luz mediante el calentamiento de un filamento metálico hasta ponerse al rojo blanco debido al efecto Joule por el paso de la corriente eléctrica. La principal ventaja de esta es que existe en una gran variedad de potencias, y entre sus desventajas están su calentamiento excesivo, su corto tiempo de vida y la disminución de la luminosidad a través del tiempo.
- Fluorescente:** Esta proporciona una fuente de luz brillante y sin sombras, pero debido a su limitada variedad de formas esta limita su aplicación en sistemas de visión artificial, es más eficaz que las lámparas incandescentes y proporciona una luz más difusa, debido a esto se utiliza más generalmente en objetos que sean muy reflectantes.
- LED (Light-Emitting Diode):** El LED (en español diodo emisor de luz), es una fuente de luz que funciona mediante el paso de corriente eléctrica a través de un semiconductor, este tipo de fuente proporciona una luz difusa de intensidad moderada la cual es muy útil, posee una amplia variedad de ventajas con respecto a

otras fuentes tradicionales, entre ellas una larga vida útil, baja emisión de calor, bajo consumo energético, pequeño tamaño, rápida respuesta, menor sensibilidad a vibraciones y una gran robustez.

- **Fibra óptica:** Genera una gran intensidad de luz la cual es fría, uniforme y en especial ausencia de sombras, se basa en dirigir la luz originada de una bombilla halógena o de xenón a través de una serie de fibras ópticas, lo cual culmina en un tipo de adaptador para suplir las necesidades de la aplicación.
- **Láser:** Generalmente utilizado en la iluminación estructurada, su mayor desventaja es el hecho de que no es eficiente en aquellas superficies que absorben la luz, este tipo de fuente se utiliza mayormente analizando su distorsión en aplicaciones tales como detección de irregularidades en superficies, medidas de profundidades, reconocimiento 3D, etc.

2.2.7.3.3. Cámaras

Se define como cámara digital al dispositivo que, encargado de capturar y almacenar imágenes de manera digital, esto gracias a un dispositivo conocido como sensor. De manera general las cámaras digitales utilizan dos tipos de sensores principalmente, uno de ellos el sensor CCD (Charge Couple Device, en español dispositivo de carga acoplada) y el otro CMOS (Complementary Metal Oxide Semiconductor, en español Semiconductor Complementario de óxido metálico), aparte de ello la calidad de la resolución de la cámara también se ve afectada por otros factores además de la distribución y número de píxeles que la cámara pueda poseer, características tales como el tipo de sensor y el lente también influyen.



Figura 2.2: Cámaras modernas de semiconductores: (a) CCD (recuperada: <https://www.imaging-resource.com/PRODS/D3000/ZD3000SENSOR-LG.JPG>), (b) CMOS (recuperada de: <https://www.dataray.com/blog-cmos-vs-ccd-overview.html>)

Cámaras web o webcam: "Webcam" significa cámara para uso en red. Es un dispositivo que se conecta al puerto USB de la computadora, y así permite captar vídeo y tomar fotos digitales con resolución baja, por lo que no ofrece una gran calidad de gráficos a diferencia de una cámara fotográfica digital, videocámara digital o un teléfono

celular moderno. El vídeo que capta, lo codifica especialmente para enviarlo por Internet (red mundial de redes) en tiempo real (lo más instantáneamente posible), hacia otra computadora donde otro usuario puede visualizarlo al momento. Son muy utilizadas para conversaciones vía Internet y hacer más personalizada la charla, así como también para actividades de vigilancia. Este tipo de cámaras carecen de sistema operativo¹.

Cámaras en red o cámaras IP: Una cámara IP es una combinación de una cámara y una computadora en una sola unidad, la cual captura y transmite imágenes en vivo a través de una red IP, habilitando a los usuarios autorizados a ver, almacenar y administrar el vídeo sobre una infraestructura de red estándar basada en el protocolo IP².

2.2.7.3.4. Resolución de cámaras

La resolución en un mundo digital o analógico es parecida, pero existen algunas diferencias importantes sobre su definición. En el vídeo analógico, una imagen consta de líneas o líneas de TV, puesto que la tecnología de vídeo deriva de la industria de la televisión. En un sistema digital, una imagen está formada por píxeles cuadrados.

2.2.7.4. Medio y software de proceso

El medio de proceso de la imagen pueden ser una computadora o un sistema integrado, es el sistema que obtiene, guarda las imágenes y las procesa a través de algoritmos específicos para extraer la información necesaria. Luego de ello tomar decisiones según sea la aplicación del sistema de visión artificial.

Los sistemas integrados son aquellos que incorporan el software y todo el hardware necesario en un mismo sistema, disponen de un procesador integrado con capacidad de tomar decisiones. El software de procesamiento es el conjunto de métodos, instrumentos y herramientas encargadas de analizar la imagen y extraer información de la misma según los algoritmos en los que estén basados. Su base es la interpretación y análisis de los píxeles.

2.2.7.5. Etapas

Las etapas de la visión artificial cambian dependiendo de la aplicación del sistema de visión artificial al que está enfocado el proceso, así también las condiciones del ambiente y la información objetivo que se desea obtener.

¹Cámara Web - Webcam , características y capacidades .:: [www.informaticamoderna.com](http://www.informaticamoderna.com/Camara_web.htm) .: URL: http://www.informaticamoderna.com/Camara_web.htm (visitado 10-08-2018).

²Cámaras IP. URL: http://www.rnds.com.ar/articulos/031/RNDS_084W.pdf (visitado 10-08-2018).

2.2.7.5.1. Adquisición

La obtención del objeto en formato digital es la primera etapa en un sistema de visión artificial, consiste básicamente en la digitalización de la imagen y para ello es necesario que existan dos elementos fundamentales. El primero es un dispositivo físico sensible al espectro de la energía electromagnética reflejada por el objeto y segundo un elemento conocido como digitalizador, este dispositivo es el encargado de convertir la salida del dispositivo de detección física a un formato digital¹.

Esta etapa da como resultado una imagen, o un grupo de imágenes, que se representan generalmente a través de una matriz bidimensional de valores enteros, para evitar procesamientos más complejos es necesario que esta etapa sea realizada correctamente.

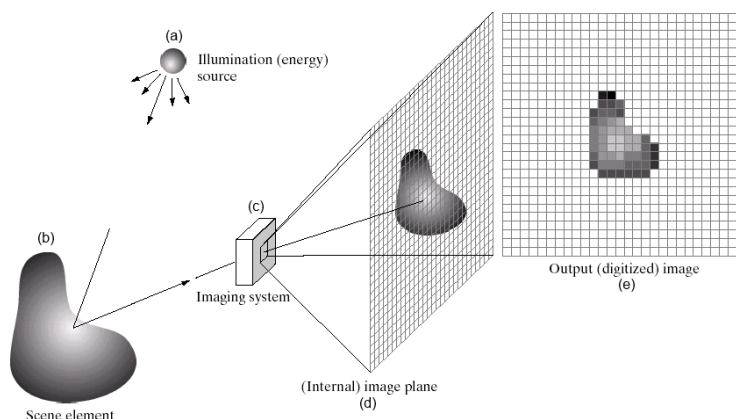


Figura 2.3: Proceso de adquisición: (a) Fuente de energía. (b) Elemento de la escena. (c) Sistema de formación de imagen. (d) Proyección de la escena sobre el plano de la imagen. (e) imagen digitalizada. Fuente: Rafael C Gonzalez y Richard E Woods. “Digital Image Processing, Prentice Hall”. En: *New Jersey* (2002)

2.2.7.5.2. Muestreo y cuantificación

Las imágenes están compuestas por una combinación entre una fuente de iluminación y el reflejo de la energía que absorben los elementos de la escena que se van a examinar. La energía ingresada se transforma en una tensión por la combinación de la energía eléctrica de entrada y el material contenido en el sensor, que es sensible al tipo energía particular que se desea detectar². Los sensores son instrumentos que alteran su señal eléctrica de acuerdo a la función de la intensidad de la luminosidad percibida. Usualmente la respuesta de dichos sensores es una onda de tensión de forma continua, de la cual su amplitud corresponde al fenómeno que se está detectando. Para poder crear una imagen digital se deben digitalizar este continuo de datos que proporcionan los sensores, al hacerlo se

¹Gonzalez y Woods, “Digital Image Processing, Prentice Hall”.

²Gonzalez y Woods, “Digital Image Processing, Prentice Hall”.

descompone la imagen real en una matriz discreta de puntos de un tamaño determinado.

Para tener una mejor comprensión del proceso de digitalización se debe conocer de antemano conceptos tales como el muestreo y la cuantificación. La digitalización de imagen continua se debe realizar en sus coordenadas (x, y) y la amplitud, el muestreo consiste en digitalizar los valores de las coordenadas y la cuantificación consiste en digitalizar los valores de la amplitud, el número de ambos, muestras y niveles de intensidad son los que determinan la calidad final de una imagen digital.

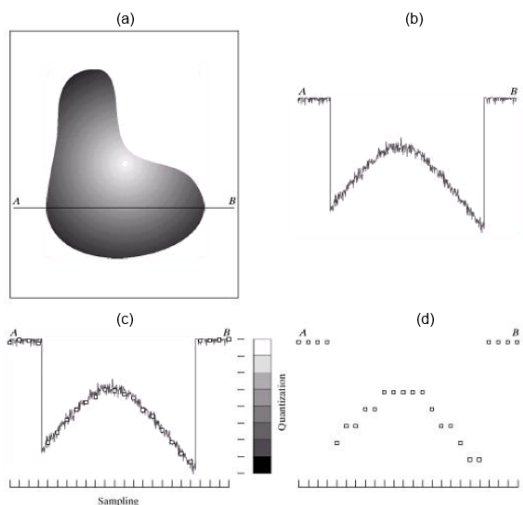


Figura 2.4: Generación de imágenes digitales. (a) Imagen continua. (b) Amplitud a lo largo de “A” a “B” de la imagen continua. (c) Muestreo. (d) Cuantificación. Fuente: Rafael C Gonzalez y Richard E Woods. “Digital Image Processing, Prentice Hall”. En: *New Jersey* (2002)

2.2.7.5.3. Representación de imágenes digitales

Sea $f(s, t)$ que representa una imagen bidimensional definida como una función de dos variables continuas, s y t . El valor o amplitud f en las coordenadas espaciales (x, y) es una magnitud escalar positiva; del cual su significado físico es la intensidad o nivel de gris de la imagen. Convertimos esta función en una imagen digital mediante el muestreo y la cuantificación, resultando la función $f(x, y)$, que contiene M filas y N columnas, donde (x, y) son las coordenadas discretas. Por razones de claridad y conveniencia de notación, utilizamos valores enteros para estas coordenadas discretas: $x = 0, 1, 2, \dots, M - 1$ e $y = 0, 1, 2, \dots, N - 1$. Se define la imagen como digital cuando las coordenadas y los valores de intensidad son cantidades discretas finitas¹.

¹Gonzalez y Woods, “Digital Image Processing, Prentice Hall”.

En forma de ecuación, escribimos la representación de una matriz numérica $M \times N$ como:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}$$

Se representa como una matriz de números reales y cada elemento de esta matriz se denomina un elemento de imagen o pixel (*picture x element*).

Debido al almacenamiento y consideraciones de hardware de cuantificación, el número de niveles de intensidad L es una potencia entera de 2 ($L = 2^k$, $k = \text{número de bits}$). Los niveles de intensidad están igualmente espaciados y son números enteros en el intervalo $[0, L - 1]$.

En las imágenes en escala de grises, el tono de la señal está generalmente representado por 8 bits (1 byte). Con esta dimensión se puede codificar $2^8 = 256 \text{ tonos}$ ($0 \text{ a } 255$). usualmente la correspondencia que se usa entre el valor digital y el tono, es negro (0) y blanco (255).

El número total de bits b para guardar una imagen digitalizada es: $b = M \times N \times k$. Si tenemos una imagen de 1280×720 píxeles con 256 niveles de gris, para guardar dicha imagen se necesitan: $1280 \times 720 \times 8 = 7,372,800 \text{ bits}$ de memoria = 921600 bytes . En las imágenes a color el método de almacenamiento es parecido, por lo que se requiere tres veces más espacio para su almacenamiento.

2.2.7.5.4. Formatos de imágenes

El píxel: Las imágenes digitales se forman a partir de retícula rectangular formada por celdas. Cada una de las celdas recibe el nombre de píxel. Dicho nombre proviene de la expresión que se obtiene al unir las palabras inglesas *picture* y *element*. Los píxeles son las unidades de color que componen la imagen. Un píxel no tiene una medida concreta, no se puede decir si mide 1 mm. o 1 cm. o 1 m., simplemente es la medida de división de una retícula en celdillas¹.

¹El píxel | GIMP, aplicaciones didácticas. URL: http://www.ite.educacion.es/formacion/materiales/86/cd/m2/el_pxel.html (visitado 10-10-2018).



Figura 2.5: Ejemplo de píxeles en una imagen. Fuente: Elaboración propia.

Toda la información que corresponde a una imagen digitalizada consta de una matriz de píxeles que vemos en nuestra pantalla y guardamos en los sistemas de almacenamiento de nuestro ordenador.

Profundidad de píxel: Una imagen digital está compuesta por una matriz de píxeles ($a \times b \times c$), en la cual a y b representan la anchura y altura, c es la profundidad de color o profundidad de bit, podríamos decir que es la tercera dimensión de la matriz, la que permite que cada píxel pueda tener un número determinado de colores distintos.

La profundidad de píxel es una unidad de medida binaria porque cada píxel está formado por bits. Cuando la profundidad de píxel es de 1 bit, la imagen solamente tiene dos colores o dos niveles de gris representados por los valores 0 y 1; por ejemplo, solamente puede ser o blanco o negro. Una profundidad de píxel de 8 bits permite que cada píxel pueda tener 256 colores distintos o 256 niveles distintos de grises que provienen de combinar las series de ocho elementos posibles obtenidas de combinar ceros y unos. Si la profundidad del píxel es de 24 bits podemos llegar a 16 millones de colores distintos en cada píxel.

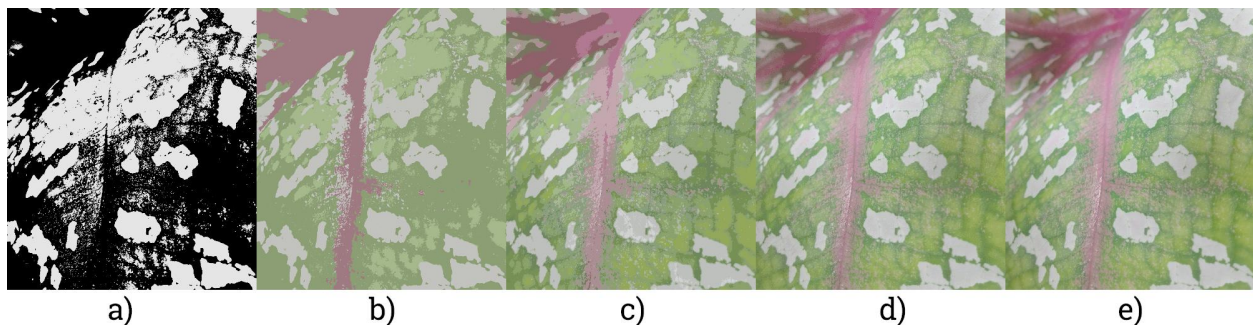


Figura 2.6: Ejemplo de profundidad de bits en imágenes digitales: a) 1 bit, b) 2 bits, c) 4 bits, d) 8 bits, e) 24 bits. Fuente: Elaboración propia.

Color real: Cada píxel únicamente puede tomar un color, cuando una imagen es de 256 colores, significa que cada píxel puede tomar uno de esos 256 colores. Si la profundidad es de 24 bits, se maneja la posibilidad de utilizar millones de colores con la combinación de los tres colores primarios: rojo (red), verde (green) y azul (blue), en el llamado modo RGB. Cada canal de color tiene 8 bits para controlar cada color, por lo tanto, los colores que se pueden obtener son $256 \times 256 \times 256 = 16.777.216$, por eso se llama "millones de colores" se denomina como color real. Con esta cantidad de colores es suficiente para que el ojo humano perciba las imágenes con calidad fotográfica.

Modos de color: Se define como modo de color al sistema de coordenadas que se utiliza para describir los colores de forma numérica. Los tipos principales son el RGB (red, green, blue o en español rojo, verde y azul), el HSL (hue, saturation, lightness, en español tono, luminosidad, saturación) y el CMYK (cyan, magenta, yellow & key, en español, cian, magenta, amarillo y negro)¹.

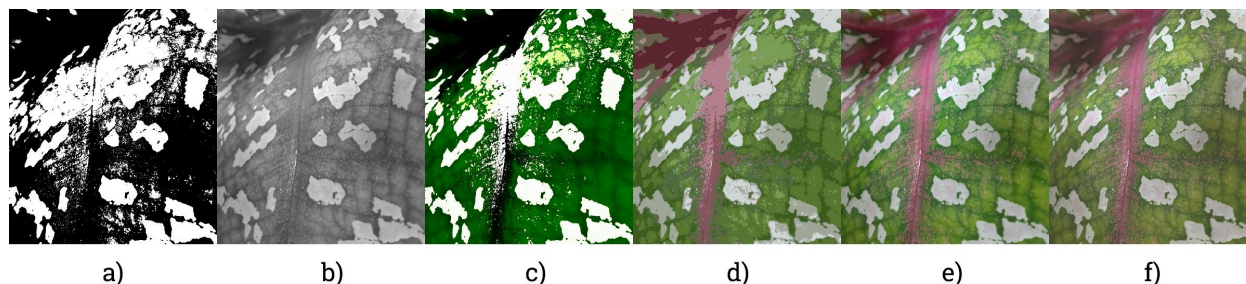


Figura 2.7: Ejemplo de modos de color en imágenes digitales: a) Mapa de bits, b) Escala de grises, c) Color indexado, solo verdes, d) color indexado, 10 colores, e) Color RGB, f) Color CMYK. Fuente: Elaboración propia.

Los modos de color influyen en el tamaño de la imagen en disco y al número de canales de color que utilizan. Si incrementamos el número de colores aumentará el tamaño del archivo que contiene la imagen. En la tabla se muestran los diferentes modos de color y sus características.

Resolución de una imagen: La definición de resolución es confusa de entender en imagen digital debido a que se utiliza refiriéndose a diferentes elementos²:

- La resolución en la impresión de imágenes digitales, está definida por el número de puntos impresos por unidad de superficie.
- En una imagen digital como tal la resolución depende del número de píxeles por unidad de superficie.
- La resolución de pantalla de un monitor se expresa con la multiplicación del número de columnas y filas de

¹Modos de color | GIMP, aplicaciones didácticas. URL: http://www.ite.educacion.es/formacion/materiales/86/cd/m2/modos_de_color.html (visitado 10-10-2018).

²Resolución de una imagen | GIMP, aplicaciones didácticas. URL: http://www.ite.educacion.es/formacion/materiales/86/cd/m2/resolucion_de_una_imagen.html (visitado 10-10-2018).

píxeles. Las más comunes son 800x600, 1024x768, 1920x1080 píxeles de resolución.

Cuando se utiliza el término “resolución de la imagen digital” se refiere a la cantidad de píxeles por unidad de superficie. Se suele utilizar como unidad mínima la pulgada cuadrada (una pulgada es igual a 2,54 cm) y así una imagen puede tener una resolución de 72x72 ppp (píxeles por pulgada), 96 ppp, 150 ppp ó 300 ppp.

La calidad de una imagen está directamente relacionada con su resolución: a mayor resolución mayor calidad de imagen. Si una imagen tiene una resolución de 200 ppp, en cada pulgada cuadrada nos encontramos la cantidad de 200 x 200 píxeles, lo que da un resultado de 40.000 píxeles que contienen la información digital de esa imagen. Si esa misma imagen tiene una resolución de 72 ppp, la información digital está contenida en 5.184 píxeles, con lo que la calidad de la imagen será menor: menor resolución, menor detalle y menor información.

Para poder entender mejor dicho aspecto es necesario marcar la diferencia en los conceptos que estamos tratando: tamaño de la imagen, resolución de la imagen y tamaño del archivo.

El tamaño de la imagen depende de sus dimensiones (anchura y altura) medidas en píxeles u otras unidades: puntos, picas, milímetros, etc.

La resolución de la imagen se expresa en píxeles por pulgada. A mayor resolución más píxeles caben en una pulgada y menor será el tamaño de los píxeles.

El tamaño del archivo de la imagen, como unidad de información digitalizada se expresa en bits. A mayor tamaño de imagen (más píxeles contiene la imagen), mayor tamaño del archivo.

2.2.7.5.5. Formatos de imagen digital

Las imágenes que pueden ser manipuladas en un ordenador básicamente se clasifican en dos tipos: imágenes bitmap (mapa de bits) e imágenes vectoriales.

Bitmap o mapa de bits: Dichas imágenes están formadas por una matriz de píxeles. Cuando se modifica una imagen bitmap lo que se hace es cambiar el aspecto de los píxeles con programas de tratamiento de imagen como GIMP. Los principales formatos de mapas de bits o bitmap son los siguientes¹:

BMP: Formato de calidad. Los archivos tienen gran peso, por lo que suelen usarse en aplicaciones en CD-ROM. Se utiliza como el formato de almacenamiento de mapa de bits estándar en el entorno de Microsoft

¹Bitmap | GIMP, aplicaciones didácticas. URL: <http://www.ite.educacion.es/formacion/materiales/86/cd/m2/bitmap.html> (visitado 10-10-2018).

Windows. Aunque se basa en estructuras de datos de mapas de bits internos de Windows, es compatible con muchas aplicaciones que no son de Windows y no de PC¹. Extensión: *.bmp

TIFF: Se utiliza para imágenes de alta calidad que van a ser impresas. El formato TIFF es quizás el formato de mapa de bits más versátil y diverso que existe. Su naturaleza extensible y el soporte para numerosos esquemas de compresión de datos permiten a los desarrolladores personalizar el formato TIFF para adaptarse a cualquier necesidad de almacenamiento de datos². Extensión: *.tif

XCF: Formato nativo de GIMP. Permite almacenar las imágenes con capas y modificarlas posteriormente. El nombre XCF honra el origen de Gimp en la instalación eXperimental Computing de la Universidad de California en Berkeley³. Extensión: *.xcf

PICT: El formato PICT (Imagen de Macintosh) está asociado con las aplicaciones en Macintosh y es uno de los formatos mejor soportados en esa plataforma. Los archivos PICT están diseñados para encapsular la funcionalidad de QuickDraw, el protocolo de dibujo de gráficos nativos en Macintosh, y consisten principalmente en llamadas de QuickDraw organizadas sin ningún orden en particular. Han existido dos versiones principales de QuickDraw, v1.0 y v2.0 (Color QuickDraw). También se han realizado numerosas revisiones secundarias de QuickDraw, cada una asociada con una versión PICT de Macintosh correspondiente⁴. Extensión: *.pic

JPG: Es el formato más utilizado en las páginas de Internet para la reproducción de fotografías. Permite comprimir las imágenes, pero produce pérdidas de calidad. El formato de intercambio de archivos JPEG (JFIF) es un desarrollo de C-Cube Microsystems con el fin de almacenar datos codificados en JPEG. Está diseñado para permitir que los archivos que contienen flujos de datos codificados en JPEG se intercambien entre sistemas y aplicaciones incompatibles⁵. Extensión: *.jpg

GIF: Este formato también se utiliza en las páginas web de Internet, pudiendo comprimir las imágenes sin pérdidas. Utiliza el modo de color indexado para las imágenes que no tienen muchas tonalidades de color. Permite gráficos animados y transparencia. Originalmente fue diseñado para facilitar la transferencia de imágenes y el

¹Microsoft Windows Bitmap: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/bmp/egff.htm> (visitado 10-10-2018).

²TIFF: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/tiff/egff.htm> (visitado 10-10-2018).

³Henning Makhholm. henning.makhholm.net/xcftools/xcf-spec-saved. URL: <http://henning.makhholm.net/xcftools/xcf-spec-saved> (visitado 10-10-2018).

⁴Macintosh PICT: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/macpict/egff.htm> (visitado 10-10-2018).

⁵JPEG File Interchange Format: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/jpeg/egff.htm> (visitado 10-10-2018).

almacenamiento en línea para que su uso por CompuServe y sus clientes, GIF es principalmente un formato de intercambio y almacenamiento, aunque se basa y es compatible con muchas aplicaciones¹. Extensión: *.gif

PNG: Tiene las ventajas de los formatos GIF y JPG. Comienza a ser muy utilizado en Internet por su gran capacidad de compresión sin pérdida y por tener la posibilidad de manejar transparencia. PNG es capaz de almacenar datos de imagen truecolor (color verdadero) de dos niveles a 48 bits sin pérdida². Extensión: *.png

PSD: Formato nativo de Photoshop que permite guardar todas las presentaciones, retoques y nuevas creaciones realizadas con este programa. Un formato flexible que se lee y escribe fácilmente pero que no tiene un esquema de compresión superior. Proporciona un soporte especialmente bueno para diferentes esquemas de almacenamiento de color³. Extensión: *.psd

Cada uno de estos formatos poseen diferentes características y permite guardar los archivos que contienen las imágenes de acuerdo a sus especificaciones.

2.2.7.5.6. Compresión

Algunos formatos de imagen digital utilizan la llamada Compresión con pérdidas. Se aplican algoritmos que permiten decidir cuál es la información menos relevante para el ojo humano y esta es desechada. Cuanto mayor cantidad se deseche, menos tamaño de almacenamiento ocupa el archivo, pero también tiene una calidad de imagen menor. Si guardamos una imagen en un formato con Compresión por pérdidas, cada vez que dicha imagen es almacenada se obtienen pérdidas de píxeles por lo que podemos llegar a perder completamente la imagen y obtener una imagen inservible. Por ejemplo el formato JPG.

Otros formatos de imagen digital utilizan Compresión sin pérdidas. Se utilizan algoritmos de compresión, pero no se pierde ningún dato con lo que la calidad de la imagen se mantiene. Por ejemplo, XCF de GIMP.

Vectorial: Dichos formatos están representados mediante trazos geométricos que están controlados por operaciones matemáticas que ejecuta el ordenador. Las líneas que componen la imagen están definidas por vectores (de ahí su nombre). La ventaja principal de este tipo de imagen es que pueden reducirse o ampliarse sin ningún tipo

¹GIF: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/gif/egff.htm> (visitado 10-10-2018).

²PNG: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/png/egff.htm> (visitado 10-10-2018).

³Adobe Photoshop: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/psd/egff.htm> (visitado 10-10-2018).

de pérdida de calidad. Los principales formatos de las imágenes vectoriales son¹:

SVG: Gráficos de Vectores escalables. Formato muy reciente pero que puede ser usado fácilmente en GIMP.

Extensión: *.svg

WMF: Formato de archivo vectorial de la plataforma Windows. Es el formato que mejor se adapta a los distintos programas de dibujo. Extensión: *.wmf

SWF: Imágenes vectoriales para animación web. Extensión: *.swf

EPS: Formato utilizado habitualmente para exportar imágenes de mapa de bits con trazados vectoriales.

Extensión: *.eps

PDF: Es compatible entre las plataformas MAC, Linux y PC. Usado fundamentalmente para la confección de documentos que puedan descargarse de Internet. Extensión: *.pdf

2.2.7.5.7. *Procesamiento*

El procesamiento digital de imágenes es el grupo de técnicas y prácticas que transforman imágenes digitales para optimizar la visibilidad de determinadas características de los objetos presentes en la misma, para su análisis posterior o para mejorar la visualización de la imagen.

El procesamiento digital de imágenes no aumenta la cantidad de información que se puede extraer solo realza las características de la imagen, para el correcto procesamiento de la imagen se debe considerar el proceso correcto de formación así como las características que se desean de dicha imagen.

2.2.8. OCR

2.2.8.1. Definición

El reconocimiento óptico de caracteres o el lector óptico de caracteres, a menudo abreviado como OCR (del inglés Optic Character Recognition), es la conversión mecánica o electrónica de imágenes de texto mecanografiado, escrito a mano o impreso en texto codificado en máquina, ya sea de un documento escaneado, una foto de un documento, una fotografía de escena. (por ejemplo, el texto en carteles y vallas publicitarias en una foto de paisaje) o de texto de subtítulos superpuestos en una imagen (por ejemplo, de una emisión de televisión).

¹Vectorial | GIMP, aplicaciones didácticas. URL: <http://www.ite.educacion.es/formacion/materiales/86/cd/m2/vectorial.html> (visitado 10-10-2018).

Es ampliamente utilizado como una forma de entrada de información de registros de datos impresos en papel, ya sean documentos como pasaportes, facturas, extractos bancarios, recibos informatizados, tarjetas de visita, correo, impresiones de datos estáticos o cualquier documentación adecuada, es un método común de digitalización de textos impresos que puedan editarse electrónicamente, realizar búsquedas, almacenarse de forma más compacta, mostrarse en línea y utilizarse en procesos mecánicos tales como computación cognitiva, traducción automática, extracción de texto a voz, datos clave y minería de texto. OCR es un campo de investigación en reconocimiento de patrones, inteligencia artificial y visión artificial.

Las primeras versiones necesitaban ser entrenadas con imágenes de cada carácter y trabajadas en una fuente o tipografía una a una. Los sistemas avanzados capaces de producir un alto grado de precisión de reconocimiento para la mayoría de las fuentes son ahora comunes y son compatibles con una variedad de entradas de formato de archivo de imagen digital. Algunos sistemas son capaces de reproducir resultados con formato que se asemejan mucho a la página original, incluyendo imágenes, columnas y otros componentes no textuales.

2.2.8.2. Historia

El reconocimiento óptico de caracteres temprano puede rastrearse a tecnologías tales como telegrafía y la creación de dispositivos que permitan lectura para ciegos¹. En 1914, Emanuel Goldberg desarrolló una máquina que leía caracteres y los convertía en un código telegráfico estándar. A su vez, Edmund Fournier d'Albe desarrolló el Optophone, un escáner de mano que cuando se movía a través de una página impresa, producía tonos que correspondían a letras o caracteres específicos².

A finales de la década de 1920 y en la década de 1930, Emanuel Goldberg desarrolló lo que llamó una "Máquina estadística" para buscar archivos de microfilm utilizando un sistema de reconocimiento de código óptico. En 1931 se le concedió el número de patente de los Estados Unidos 1.838.389 para dicha invención. La patente fue adquirida por IBM.

Con la llegada de los teléfonos inteligentes y las gafas inteligentes, el OCR se puede utilizar en aplicaciones de dispositivos móviles conectados a Internet que extraen texto capturado con la cámara del dispositivo. Estos dispositivos que no tienen la funcionalidad de OCR incorporada en el sistema operativo normalmente utilizarán una

¹Herbert F Schantz. *The history of OCR, optical character recognition*. Recognition Technologies Users Association Manchester, VT, 1982.

²EE Fournier d'Albe. "On a type-reading optophone". En: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 90.619 (1914), págs. 373-375.

API de OCR para extraer el texto del archivo de imagen capturado y proporcionado por el dispositivo. La API de OCR devuelve el texto extraído, junto con la información sobre la ubicación del texto detectado en la imagen original a la aplicación del dispositivo para su posterior procesamiento (como texto a voz) o visualización.

2.2.8.3. Usuarios ciegos y deficientes visuales

En 1974, Ray Kurzweil comenzó la compañía Kurzweil Computer Products, Inc. y continuó con el desarrollo del OCR omni-font, que podía reconocer el texto impreso en prácticamente cualquier tipo de letra (Kurzweil a menudo se le atribuye la invención del OCR omni-font, pero fue utilizado por empresas, incluida CompuScan, a finales de los años sesenta y setenta¹). Kurzweil decidió que la mejor aplicación de esta tecnología sería crear una máquina de lectura para ciegos, que permitiera a las personas ciegas que una computadora les leyera el texto en voz alta. Este dispositivo requirió la invención de dos tecnologías habilitadoras: el escáner de superficie plana CCD y el sintetizador de texto a voz.

El 13 de enero de 1976, el exitoso producto terminado se dio a conocer durante una conferencia de prensa ampliamente difundida encabezada por Kurzweil y los líderes de la Federación Nacional de Ciegos. En 1978, Kurzweil Computer Products comenzó a vender una versión comercial del programa informático de reconocimiento óptico de caracteres. LexisNexis fue uno de los primeros clientes y compró el programa para cargar documentos legales y documentos de noticias en sus emergentes bases de datos en línea. Dos años después, Kurzweil vendió su compañía a Xerox, que tenía interés en comercializar aún más la conversión de texto de papel a computadora. Xerox finalmente lo hizo girar como Scansoft, que se fusionó con Nuance Communications.

El grupo de investigación encabezado por AG Ramakrishnan en el laboratorio de inteligencia médica e ingeniería lingüística, Indian Institute of Science, ha desarrollado la herramienta PrintToBraille, una interfaz GUI de código abierto² que cualquier OCR puede utilizar para convertir imágenes escaneadas de Libros impresos a libros en braille.

En la década de 2000, OCR se puso a disposición en línea como un servicio (WebOCR), en un entorno de computación en la nube y en aplicaciones móviles como la traducción en tiempo real de carteles en idiomas extranjeros en un teléfono inteligente.

¹Schantz, *The history of OCR, optical character recognition*.

²*PrintToBraille Tool. ocr-gui-frontend. MILE Lab, Dept of EE, IISc.* URL: <https://web.archive.org/web/20141225115650/https://code.google.com/p/ocr-gui-frontend/> (visitado 10-10-2018).

Hay varios sistemas de OCR comerciales y de código abierto disponible para los sistemas de escritura más comunes, incluyendo caracteres latinos, cirílicos, árabes, hebreos, indios, bengalíes (Bangla), devanagari, tamiles, chinos, japoneses y coreanos.

2.2.8.4. Aplicaciones

Los motores de OCR se han desarrollado en muchos tipos de aplicaciones de OCR específicas de dominio, como recibo OCR, factura OCR, cheque OCR, documento legal de facturación OCR.

Se pueden utilizar para:

- Entrada de datos para documentos comerciales, por ejemplo, cheques, pasaportes, facturas, extractos bancarios y recibos
- Reconocimiento automático de matrículas
- En aeropuertos, para el reconocimiento de pasaportes y extracción de información.
- Aseguramiento automático de la extracción de información clave en un documento.
- Extracción de información de tarjetas de visita en una lista de contactos.
- Generación más rápida de versiones textuales de documentos impresos, por ejemplo, el escaneo de libros para el Proyecto Gutenberg
- Permite que las imágenes electrónicas de documentos impresos permitan la búsqueda dentro de ellos, por ejemplo, Google Books
- Conversión de la escritura a mano en tiempo real para controlar una computadora (computación con lápiz)
- Tecnología asistencial para usuarios ciegos y deficientes visuales.

2.2.8.5. Tipos

- **Reconocimiento óptico de caracteres (OCR):** se orienta al texto escrito a máquina, un glifo o un carácter a la vez.
- **Reconocimiento óptico de palabras:** se orienta al texto escrito a máquina, una palabra a la vez, para idiomas que usan un espacio como divisor de palabras, (por lo general, sólo se llama “OCR”).
- **Reconocimiento inteligente de caracteres (ICR):** Usualmente se orienta a la escritura a mano o texto

cursorio, un glifo o un carácter a la vez, generalmente con aprendizaje automático.

- **Reconocimiento de palabras inteligente (IWR):** Está dirigido a la escritura a mano o texto en cursiva, una palabra a la vez. Esto es especialmente útil para los idiomas en los que los glifos no están separados en guiones cursivos.

OCR es generalmente un proceso "fuera de línea", que analiza un documento estático. El análisis del movimiento de escritura a mano se puede utilizar como entrada para el reconocimiento de escritura a mano¹. En lugar de utilizar simplemente las formas de los glifos y las palabras, esta técnica es capaz de capturar movimientos, como el orden en que se dibujan los segmentos, la dirección y el patrón de dejar el lápiz y levantarlo. Esta información adicional puede hacer que el proceso de extremo a extremo sea más preciso. Esta tecnología también se conoce como reconocimiento de caracteres en línea", reconocimiento dinámico de caracteres", reconocimiento de caracteres en tiempo real y reconocimiento inteligente de caracteres".

2.2.8.6. Técnicas

2.2.8.6.1. Pre-procesamiento

El software de OCR a menudo "procesa" las imágenes para mejorar las posibilidades de reconocimiento exitoso. Las técnicas incluyen:

Deshacer sesgado: Si el documento no se ha alineado correctamente cuando se escanea, puede que tenga que ser rotado unos pocos grados hacia la derecha o hacia la izquierda con el fin de hacer que las líneas de texto perfectamente horizontal o vertical.

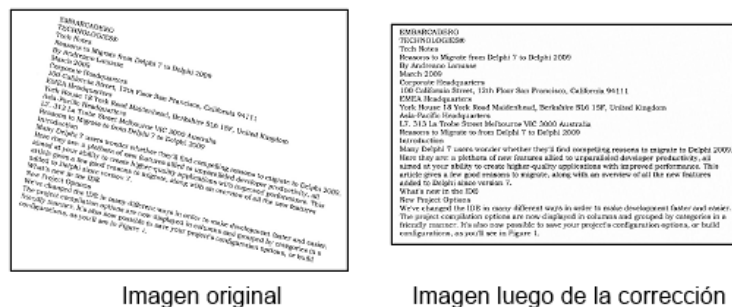


Figura 2.8: Ejemplo de sesgado. Recuperada de: <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>

¹Charles C. Tappert et al. "The state of the art in online handwriting recognition". En: *IEEE Transactions on pattern analysis and machine intelligence* 12.8 (1990), págs. 787-808.

Reducción de ruido: remueve puntos positivos y negativos, suavizando bordes.

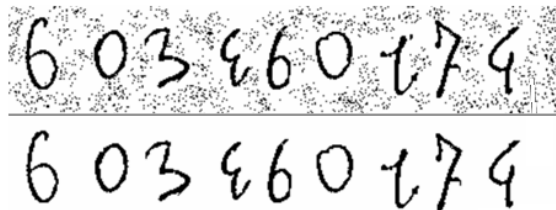


Figura 2.9: Ejemplo de reducción de ruido. Recuperada de: <https://yunmai.wordpress.com/2016/10/10/basics-that-you-need-to-know-about-ocr/>

Binarización: convierte una imagen de color o en escala de grises a blanco y negro (llamada “imagen binaria” porque hay dos colores). La tarea de binarización se realiza como una forma sencilla de separar el texto (o cualquier otro componente de imagen deseado) del fondo¹. La tarea de la binarización en sí misma es necesaria, ya que la mayoría de los algoritmos de reconocimiento comercial funcionan solo en imágenes binarias, ya que resulta más simple de realizar el proceso en ellas². Además, la tasa de eficacia de la etapa de binarización influye en gran medida en la calidad de la etapa de reconocimiento de caracteres y se toman decisiones cuidadosas en la elección de la binarización empleada para un tipo de imagen de entrada determinado; ya que la calidad del método de binarización empleado para obtener el resultado binario depende del tipo de imagen de entrada (documento escaneado, imagen del texto una fotografía, documento degradado histórico, etc.)³.

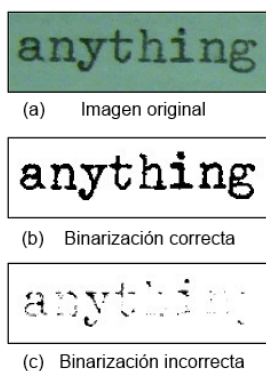


Figura 2.10: Ejemplo de binarización. Recuperada de: <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>

¹Mehmet Sezgin y Bülent Sankur. “Survey over image thresholding techniques and quantitative performance evaluation”. En: *Journal of Electronic imaging* 13.1 (2004), págs. 146-166.

²Maya R Gupta et al. “OCR binarization and image pre-processing for searching historical documents”. En: *Pattern Recognition* 40.2 (2007), págs. 389-397.

³Øivind Due Trier y Anil K Jain. “Goal-directed evaluation of binarization methods”. En: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 12 (1995), págs. 1191-1201; Sergey Milyaev et al. “Image binarization for end-to-end text understanding in natural images”. En: *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, págs. 128-132.

Eliminación de líneas: limpia cajas y líneas que no son glifos¹.

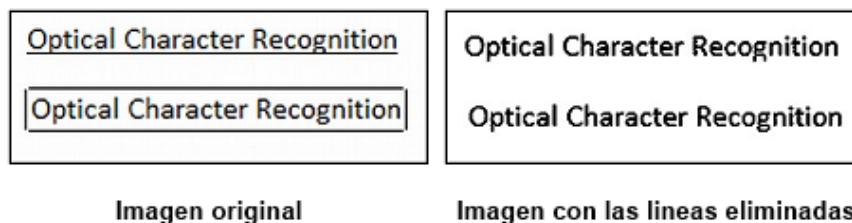


Figura 2.11: Ejemplo de eliminación de líneas. Recuperada de: <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>

Análisis de diseño o "zonificación": identifica columnas, párrafos, citas, leyendas, etc. como bloques separados y distintos. Es especialmente importante en diseños y tablas de varias columnas.



Figura 2.12: Ejemplo de zonificación. Recuperada de: <https://yunmai.wordpress.com/2016/10/10/basics-that-you-need-to-know-about-ocr/>

Detección de líneas y palabras: establece una línea de base para las formas de las palabras y los caracteres, separa las palabras si es necesario.

Reconocimiento del idioma: en documentos multilingües, la secuencia del idioma puede cambiar en el nivel de las palabras y, por lo tanto, es necesaria la identificación de la secuencia del idioma en sí, antes de que se pueda invocar el OCR correcto para manejar la secuencia de comandos específica².

¹Image Optimization with Grooper. URL: <https://grooper.com/image-optimization.html> (visitado 10-10-2018).

²Peeta Basa Pati y AG Ramakrishnan. "Word level multi-script identification". En: *Pattern Recognition Letters* 29.9 (2008), págs. 1218-1229.

Aislamiento de caracteres o "segmentación": para OCR por carácter, los múltiples caracteres que están conectados debido a artefactos de imagen deben separarse; Los caracteres individuales que se dividen en varias partes debido a artefactos deben estar conectados.

Normalizar la relación de aspecto y la escala: Se define la región de cada carácter y se escala para mejorar su procesamiento¹.

La segmentación de las fuentes de paso fijo se logra de manera relativamente simple al alinear la imagen con una cuadrícula uniforme en función de donde las líneas verticales de la cuadrícula interceptarán las áreas negras con menos frecuencia. Para las fuentes proporcionales, se necesitan técnicas más sofisticadas porque el espacio en blanco entre las letras a veces puede ser mayor que entre las palabras, y las líneas verticales pueden cruzar más de un carácter².

2.2.8.6.2. El reconocimiento de caracteres

Hay dos tipos básicos de algoritmos OCR, que pueden producir una lista clasificada de caracteres candidatos, la coincidencia matricial y la extracción de características³.

La coincidencia matricial implica comparar una imagen con un glifo almacenado píxel por píxel; también se conoce como “coincidencia de patrones”, “reconocimiento de patrones” o “correlación de imágenes”. Esto se basa en que el glifo de entrada esté aislado correctamente del resto de la imagen y que el glifo almacenado esté en una fuente similar y en la misma escala. Esta técnica funciona mejor con texto escrito a máquina y no funciona bien cuando se encuentran nuevas fuentes. Esta es la técnica que implementó el OCR basado en fotocélulas físicas, de manera bastante directa.

La extracción de características descompone los glifos en “entidades” como líneas, bucles cerrados, dirección de línea e intersecciones de línea. Las características de extracción reducen la dimensionalidad de la representación y hacen que el proceso de reconocimiento sea eficiente computacionalmente. Estas características se comparan con una representación vectorial abstracta de un carácter, que podría reducirse a uno o más prototipos de glifos. Las técnicas de detección generales de características en la visión por computadora son aplicables a este tipo

¹*Basic OCR in OpenCV | Damiles Blog*. URL: <http://blog.damiles.com/2008/11/20/basic-ocr-in-opencv.html> (visitado 10-10-2018).

²*Ray Smith (2007). An Overview of the Tesseract OCR Engine*. URL: <https://github.com/tesseract-ocr/docs/blob/master/tesseract-ocr-2007.pdf> (visitado 10-10-2018).

³*OCR Introduction*. URL: <http://www.dataid.com/aboutocr.htm> (visitado 10-10-2018).

de OCR, que se ve comúnmente en el reconocimiento de escritura a mano “inteligente” y, de hecho, en el software OCR más moderno. Clasificadores de vecinos más cercanos, como el “algoritmo de vecinos K más cercanos o KNN” se utilizan para comparar características de imagen con características de glifo almacenadas y elegir la coincidencia más cercana¹.

Software como Cuneiform y Tesseract utilizan un enfoque de dos pasos para el reconocimiento de caracteres. La segunda pasada se conoce como reconocimiento adaptativo y utiliza las formas de las letras reconocidas con gran confianza en la primera pasada para reconocer mejor las letras restantes en la segunda pasada. Esto es ventajoso para fuentes inusuales o escaneos de baja calidad donde la fuente está distorsionada (por ejemplo, borrosas o difuminadas)².

El resultado de OCR se puede almacenar en el formato estandarizado ALTO (Analyzed Layout and Text Object), un esquema XML dedicado, mantenido por la Biblioteca del Congreso de los Estados Unidos.

2.2.8.6.3. Post-procesamiento

La precisión de OCR puede aumentarse si la salida está limitada por un léxico, una lista de palabras que pueden aparecer en un documento³. Esto podría ser, por ejemplo, todas las palabras en un idioma concreto, o un léxico más técnico para un campo específico. Esta técnica puede ser problemática si el documento contiene palabras que no están en el léxico, como los nombres propios. Tesseract utiliza su diccionario para influir en el paso de la segmentación de caracteres, para mejorar la precisión⁴.

El flujo de salida puede ser un flujo de texto sin formato o un archivo de caracteres, pero los sistemas de OCR más sofisticados pueden conservar el diseño original de la página y producir, por ejemplo, un PDF anotado que incluye tanto la imagen original de la página como una representación textual con capacidad de búsqueda.

El “análisis del vecino cercano” puede hacer uso de las frecuencias de co-ocurrencia para corregir errores, señalando que ciertas palabras a menudo se ven juntas⁵. Por ejemplo, “Washington, DC” es generalmente mucho más común en inglés que “Washington DOC”.

¹*The basic pattern recognition and classification with openCV* Damiles Blog. URL: <http://blog.damiles.com/2008/11/14/the-basic-pattern-recognition-and-classification-with-opencv.html> (visitado 10-10-2018).

²Ray Smith (2007). *An Overview of the Tesseract OCR Engine*.

³*Optical Character Recognition (OCR) – How it works*. URL: <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/> (visitado 10-10-2018).

⁴Ray Smith (2007). *An Overview of the Tesseract OCR Engine*.

⁵*How does OCR document scanning work? - Explain that Stuff*. URL: <https://www.explainthatstuff.com/how-ocr-works.html> (visitado 10-10-2018).

El conocimiento de la gramática del lenguaje que se está escaneando también puede ayudar a determinar si es probable que una palabra sea un verbo o un sustantivo, por ejemplo, permitiendo una mayor precisión.

El algoritmo de distancia Levenshtein también se ha utilizado en el procesamiento posterior de OCR para optimizar aún más los resultados de una API de OCR.

2.2.8.6.4. Soluciones

Existen varias técnicas para resolver el problema del reconocimiento de caracteres por medios distintos a los algoritmos OCR mejorados.

Forzando una mejor entrada: Las fuentes especiales como OCR-A, OCR-B o MICR, con tamaños, espacios y formas de caracteres distintivos especificados con precisión, permiten una mayor tasa de precisión durante la transcripción en el procesamiento de cheques bancarios. Sin embargo, irónicamente, varios motores de OCR prominentes fueron diseñados para capturar texto en fuentes populares como Arial o Times New Roman, y son incapaces de capturar texto en estas fuentes que son especializadas y muy diferentes de las fuentes utilizadas popularmente. Como Google Tesseract puede ser entrenado para reconocer nuevas fuentes, puede reconocer las fuentes OCR-A, OCR-B y MICR¹.

Los “campos de peine” son cuadros preimpresos que animan a los humanos a escribir de forma más legible, un glifo por cuadro. Estos a menudo se imprimen en un “color de deserción” que puede ser eliminado fácilmente por el sistema OCR².

Palm OS utilizó un conjunto especial de glifos, conocido como “Graffiti”, que son similares a los caracteres impresos en inglés, pero se simplifican o modifican para facilitar el reconocimiento en el hardware de computación limitada de la plataforma. Los usuarios tendrían que aprender a escribir estos glifos especiales.

La OCR basada en la zona restringe la imagen a una parte específica de un documento. Esto a menudo se conoce como “Plantilla OCR”.

Crowdsourcing: El hecho de que los humanos realicen el crowdsourcing para realizar el reconocimiento de caracteres puede procesar rápidamente imágenes como OCR impulsado por computadora, pero con mayor precisión para reconocer las imágenes que las que se obtienen con las computadoras. Los sistemas prácticos incluyen el

¹*Train your Tesseract.* URL: <http://trainyourtesseract.com> (visitado 10-10-2018).

²*How does OCR document scanning work? - Explain that Stuff.*

Amazon Mechanical Turk y reCAPTCHA. La Biblioteca Nacional de Finlandia ha desarrollado una interfaz en línea para que los usuarios corrijan los textos de OCR en el formato estandarizado ALTO¹. Crowdsourcing también se ha utilizado no para realizar el reconocimiento de caracteres directamente, sino para invitar a los desarrolladores de software a desarrollar algoritmos de procesamiento de imágenes, por ejemplo, mediante el uso de torneos de orden jerárquico².

2.2.8.6.5. *Precisión*

Encargado por el Departamento de Energía de EE. UU. (DOE), el Instituto de Investigación de Ciencias de la Información (ISRI) tuvo la misión de fomentar la mejora de las tecnologías automatizadas para comprender documentos impresos en máquina, y realizó la prueba más autorizada de la exactitud de OCR de 1992 a 1996³.

El reconocimiento del texto escrito a máquina en escritura latina aún no es 100 % exacto, incluso cuando hay imágenes claras disponibles. Un estudio basado en el reconocimiento de las páginas de periódicos del siglo 19 y principios del siglo 20 concluyó que la precisión de OCR realizado carácter por carácter para el software comercial de OCR varió⁴ de 81 % a 99 %; la precisión total se puede lograr mediante revisión humana o autenticación del diccionario de datos. Otras áreas, incluido el reconocimiento de la impresión manual, la escritura cursiva y el texto impreso en otros idiomas (especialmente aquellos caracteres del idioma de Asia oriental que tienen muchos trazos para un solo carácter), siguen siendo objeto de una investigación activa. La base de datos MNIST se usa comúnmente para probar la capacidad de los sistemas para reconocer dígitos escritos a mano.

Las tasas de precisión se pueden medir de varias maneras, y la forma en que se miden puede afectar en gran medida la tasa de precisión informada. Por ejemplo, si el contexto de la palabra (básicamente un léxico de palabras) no se usa para corregir el software encontrando palabras inexistentes, una tasa de error de caracteres del 1 % (99 % de precisión) puede resultar en una tasa de error del 5 % (95 % de precisión). o peor, si la medición se basa en si cada palabra completa fue reconocida sin letras incorrectas⁵.

¹What is the point of an online interactive OCR text editor? | Fenno-Ugrica. URL: <https://blogs.helsinki.fi/fennougrica/2014/02/21/ocr-text-editor/> (visitado 10-10-2018).

²Christoph Riedl et al. "Detecting figures and part labels in patents: competition-based development of graphics recognition algorithms". En: *International Journal on Document Analysis and Recognition (IJ DAR)* 19.2 (2016), págs. 155-172.

³Code and Data to evaluate OCR accuracy, originally from UNLV/ISRI. URL: <https://code.google.com/archive/p/isri-ocr-evaluation-tools/> (visitado 10-10-2018).

⁴Rose Holley. "How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs". En: *D-Lib Magazine* 15.3/4 (2009).

⁵CY Suen et al. "Future challenges in handwriting and computer applications". En: *3rd International Symposium on Handwriting and Computer Applications*. 1987.

Un ejemplo de las dificultades inherentes a la digitalización del texto antiguo es la incapacidad de OCR para diferenciar entre los caracteres S largos y F¹.

2.2.8.7. OpenCV

OpenCV (Open Source Computer Vision) es una librería multiplataforma desarrollada originalmente como un proyecto por Intel para dar apoyo a los primeros compiladores Intel C++ y Microsoft Visual C++ en x86. Es ampliamente utilizado actualmente para el procesamiento de imágenes y la visión artificial en general, así también se utiliza en infinidad de aplicaciones como control de procesos, sistemas de seguridad que soportan detección de movimiento, reconocimiento de objetos, etc.

Según la página oficial de OpenCV², es una librería completamente gratuita y libre, esta se distribuye bajo licencia BSD (Berkeley Software Distribution), la cual permite que sea utilizada libremente en proyectos con propósitos de investigación y comerciales, siempre y cuando, cumpla con las condiciones de dicha licencia. OpenCV está optimizado para ser utilizada en procesadores Intel, pero puede ser usado bajo cualquier otro tipo de procesador y puede usar la ventaja que proporcionan los procesadores multinúcleo. Las librerías OpenCV pueden instalarse bajo Linux como en Windows³.

OpenCV posee una buena eficiencia computacional y además cuenta con un fuerte enfoque en aplicaciones en tiempo real. Está escrito en C y C++, y se puede ejecutar en GNU/Linux, Windows, iOS, Android y Mac OS X. También es importante ver que hay un activo desarrollo en las interfaces de Python, tales como Ruby, Matlab, y otros lenguajes. OpenCV se estructura en cinco componentes principales⁴:

- El componente CV contiene los algoritmos de procesamiento de imágenes y de visión por computadora de nivel básico y superior.
- El componente ML es la biblioteca de aprendizaje de máquina, este incluye muchos clasificadores estadísticos y herramientas de agrupación.
- El componente HighGUI contiene rutinas y funciones de entrada y salida para el almacenamiento y carga de vídeo e imágenes.

¹Sarantos Kapidakis et al. *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries, TPD L 2015, Poznań, Poland, September 14-18, 2015, Proceedings*. Vol. 9316. Springer, 2015.

²OpenCV library. URL: <https://opencv.org> (visitado 10-10-2018).

³Platforms - OpenCV library. URL: <https://opencv.org/platforms/> (visitado 10-10-2018).

⁴Gary Bradski y Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

- CXCore contiene las estructuras básicas y algoritmos de datos y de contenido, apoyo a XML, y funciones gráficas. Recibe información de CV, MIL y HighGUI.
- CvAux, que contiene algunos algoritmos experimentales (HMM de reconocimiento facial y segmentación de fondo/primer plano).

En el campo del procesamiento de imágenes, el objetivo de OpenCV es proporcionar una infraestructura sencilla y accesible. La biblioteca de OpenCV posee funciones utilizadas en diferentes áreas de la visión por computadora, tales como la inspección de productos, identificación de objetos o personas en movimiento, imágenes médicas, reconocimiento facial en una imagen, seguridad, reconstrucción 3D, interfaces de usuario, robótica, etc.

2.2.8.8. Tesseract-OCR

Tesseract es un motor de reconocimiento óptico de caracteres para varios sistemas operativos.¹ Es un software gratuito, publicado bajo la Licencia Apache, Versión 2.0,² y el desarrollo ha sido patrocinado por Google desde 2006.³ En 2006, Tesseract fue considerado uno de los motores de OCR de código abierto más precisos disponibles en la actualidad⁴.

2.2.8.8.1. Historia

El motor de Tesseract se desarrolló originalmente como software propietario en los laboratorios Hewlett Packard en Bristol, Inglaterra y Greeley, Colorado entre 1985 y 1994, con algunos cambios más realizados en 1996 a Windows y una migración de C a C++ en 1998. Mucho del código se escribió en C, y luego se escribió algo más en C++. Desde entonces, todo el código se ha convertido al menos en compilación con un compilador de C++.⁵ Se hizo muy poco trabajo en la siguiente década. Luego fue lanzado como fuente abierta en 2005 por Hewlett Packard y la Universidad de Nevada, Las Vegas (UNLV). El desarrollo de Tesseract ha sido patrocinado por Google desde 2006⁶.

¹*Tesseract: an Open-Source Optical Character Recognition Engine* \ *Linux Journal*. URL: <https://www.linuxjournal.com/article/9676> (visitado 10-10-2018).

²*GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository)*. URL: <https://github.com/tesseract-ocr/tesseract/> (visitado 10-10-2018); *Google Code - Updates: Announcing Tesseract OCR*. URL: <https://web.archive.org/web/20061026075310/http://google-code-updates.blogspot.com/2006/08/announcing-tesseract-ocr.html> (visitado 10-10-2018); *OCR - Community Help Wiki*. URL: <https://help.ubuntu.com/community/OCR> (visitado 10-10-2018).

³*Announcing Tesseract OCR - The official Google Code blog*. URL: <http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html> (visitado 10-10-2018).

⁴*OCR - Community Help Wiki; Google's Tesseract OCR engine is a quantum leap forward* \ *Linux.com* \ *The source for Linux information*. URL: <https://www.linux.com/news/googles-tesseract-ocr-engine-quantum-leap-forward> (visitado 10-10-2018).

⁵*Google Code - Updates: Announcing Tesseract OCR*.

⁶*Announcing Tesseract OCR - The official Google Code blog*.

2.2.8.8.2. Características

Tesseract estaba en los tres principales motores de OCR en términos de precisión de caracteres en 1995¹. Está disponible para Linux, Windows y Mac OS X.

Tesseract hasta la versión 2 sólo podía aceptar imágenes TIFF de texto simple de una columna como entradas. Estas primeras versiones no incluían el análisis del diseño, por lo que la introducción de texto, imágenes o ecuaciones de varias columnas generaba un resultado confuso. Desde la versión 3, Tesseract ha admitido el formateo de texto de salida, la información posicional de hOCR² y el análisis de diseño de página. El soporte para una serie de nuevos formatos de imagen se agregó usando la biblioteca Leptónica. Tesseract puede detectar si el texto es monoespaciado o espaciado proporcionalmente³.

Las versiones iniciales de Tesseract solo podían reconocer texto en inglés. Tesseract v2 agregó seis idiomas occidentales adicionales (francés, italiano, alemán, español, portugués brasileño, holandés). El soporte de idioma extendido de la versión 3 incluye significativamente idiomas ideográficos (chino y japonés) y de derecha a izquierda (por ejemplo, árabe, hebreo), así como muchos más scripts. Los nuevos idiomas incluyen árabe, búlgaro, catalán, chino (simplificado y tradicional), croata, checo, danés, alemán (escritura fraktur), griego, finlandés, hebreo, hindi, húngaro, indonesio, japonés, coreano, letón, lituano, noruego, polaco, portugués, rumano, ruso, serbio (eslovaco (estándar y Fraktur)), esloveno, sueco, tagalo, tamil, tailandesa, turca, ucraniana y vietnamita. La versión 3.04, lanzado en julio de 2015, agregó 39 combinaciones adicionales de idioma / escritura, lo que elevó el número total de idiomas de soporte a más de 100. Los nuevos códigos de idioma incluyen: amh (amárico), asm (asamés), aze_cyrl (azerbaijana (en script cirílico)), bod (tibetano), bos (bosnio), ceb (cebuano), cym (galés), dzo (dzongkha), fas (persa), gle (irlandés), guj (gujarati), sombrero (haitiano y criollo haitiano), iku (Inuktitut), jav (javanés), kat (georgiano), kat_old (georgiano antiguo), kaz (kazajo), khm (Khmer central), kir (kirguiz), kur (kurdo), lao (Lao), lat (latín), mar (marathi)⁴. Tesseract puede ser entrenado para trabajar en otros idiomas también⁵.

La salida de Tesseract tendrá una calidad muy baja si las imágenes de entrada no están preprocesadas para

¹*The Fourth Annual Test of OCR Accuracy*. URL: http://www.expervision.com/wp-content/uploads/2012/12/1995_The_Fourth_Annual_Test_of_OCR_Accuracy.pdf (visitado 10-10-2018).

²*Issue 263 - tesseract-ocr - patch to enable hOCR output - An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google*. - Google Project Hosting. URL: <https://web.archive.org/web/20121113065732/http://code.google.com/p/tesseract-ocr/issues/detail?id=263> (visitado 10-10-2018).

³*OCR - Community Help Wiki*.

⁴*GitHub - tesseract-ocr/langdata: Source training data for Tesseract for lots of languages*. URL: <https://github.com/tesseract-ocr/langdata> (visitado 10-10-2018).

⁵*OCR - Community Help Wiki*.

ajustarse a ellas: las imágenes (especialmente las capturas de pantalla) deben ampliarse de manera tal que la altura x del texto sea de al menos 20 píxeles, cualquier rotación o sesgo debe corregirse o no se reconocerá ningún texto, los cambios de baja frecuencia en el brillo deben filtrarse en el paso alto , o la etapa de binarización de Tesseract destruirá gran parte de la página, y los bordes oscuros deberán eliminarse manualmente, o se interpretarán erróneamente como caracteres.

2.2.8.8.3. Ejemplo

Al procesar una imagen utilizando una configuración adecuada mediante línea de comandos, tal que:

```
tesseract imagen.tif out -l eng --oem 0 --psm 3
```

De la línea anterior se puede detallar lo siguiente:

- **-l eng** El idioma utilizado para el procesamiento del texto de la imagen, en este caso inglés, tesseract-ocr soporta un total de 116 idiomas
- **-oem 0** El motor utilizado para realizar el proceso, en este caso el motor "legacy", los motores disponibles son:
 - 0 - Legacy engine only.
 - 1 - Neural nets LSTM engine only.
 - 2 - Legacy + LSTM engines.
 - 3 - Default, basado en lo que esté disponible.
- **-psm 3** Modo de segmentación de página, los modos disponibles son:
 - 0 - Orientación y detección de scripts (OSD) solamente.
 - 1 - Segmentación de página automática con OSD.
 - 2 - Segmentación de página automática, pero no hay OSD u OCR.
 - 3 - Segmentación de página completamente automática, pero sin OSD. (Por defecto)
 - 4 - Asume una sola columna de texto de tamaños variables.
 - 5 - Asume un solo bloque uniforme de texto alineado verticalmente.
 - 6 - Asume un solo bloque de texto uniforme.
 - 7 - Trata la imagen como una sola línea de texto.

- 8 - Trata la imagen como una sola palabra.
- 9 - Trata la imagen como una sola palabra en un círculo.
- 10 - Trata la imagen como un solo carácter.
- 11 - Texto disperso. Encuentra la mayor cantidad de texto posible sin un orden en particular.
- 12 - texto escaso con OSD.
- 13 - Línea en puro. Trate la imagen como una sola línea de texto.

This is a lot of 12 point text to test the ocr code and see if it works on all types of file format.
 The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox.

Figura 2.13: Imagen a procesar en Tesseract-ocr

Al procesar la imagen anterior con la línea de comandos detallada da como resultado lo siguiente:

```
This is a lot of 12 point text to test the
cor code and see if it works on all types
of file format.
```

```
The quick brown dog jumped over the
lazy fox. The quick brown dog jumped
over the lazy fox. The quick brown dog
jumped over the lazy fox. The quick
brown dog jumped over the lazy fox.
```

2.2.8.9. OCR.Space API

OCR.Space tiene soporte para los siguientes idiomas: árabe, búlgaro, chino (simplificado), chino (tradicional), croata, checo, danés, holandés, inglés, finlandés, francés, alemán, griego, húngaro, coreano, italiano, japonés, polaco, portugués, ruso, esloveno, español, sueco, turco¹.

2.2.8.9.1. Terminología

Clave API: la clave necesaria para poder utilizar la API, necesita de registro de usuario para obtener una.

EndPoint: Dirección URL con la que se interactúa mediante consultas HTTP GET o HTTP POST, para enviar y recibir la información de la API.

¹Free OCR API. URL: <https://ocr.space/ocrapi> (visitado 10-10-2018).

2.2.8.9.2. Características

La API posee diferentes niveles detallados a continuación:

- Free
 - 25,000 consultas/mes
 - Tamaño máximo de archivo: 1Mb
 - Límite de páginas PDF: 3
 - Ratio: 500 consultas/día
- PRO (US\$24.95/mes)
 - 250,00 consultas/mes
 - Tamaño límite de archivo: 5Mb
 - Límite de páginas PDF: No soportado
 - Ratio: 600 consultas/minuto
- PRO PDF (US\$49.95/mes)
 - 250,00 consultas/mes
 - Tamaño límite de archivo: 5Mb
 - Límite de páginas PDF: No soportado
 - Ratio: 600 consultas/minuto

Utilizando el EndPoint mediante consultas POST enviamos los parámetros y la imagen a procesar para obtener su texto, los parámetros requeridos para procesar una consulta son los siguientes:

- **apikey:** La clave para utilizar la API, ya sea gratuita o de pago
- **url o archivo o imagen en base64:** la imagen a procesar
- **language:** El idioma que utiliza la imagen o el PDF a procesar, por defecto el api utiliza eng el cual es el idioma inglés, para utilizar español utilizaríamos spa
- **isOverlayRequired:** Si es verdadero (True), devuelve las coordenadas de los cuadros delimitadores para cada palabra. Si es falso (False), el texto de OCR se devuelve sólo como un bloque de texto (esto hace que el JSON sea más pequeño). Los datos de superposición se pueden utilizar, por ejemplo, para mostrar texto sobre la imagen, por defecto está definido en False.

- **detectOrientation:** Si se establece en verdadero (True), el api autor rotará la imagen correctamente y establece el parámetro TextOrientation en la respuesta JSON. Si la imagen no se gira, entonces TextOrientation será igual a 0, de lo contrario es el grado de rotación, por ejemplo. "270".
- **Scale:** Si se establece en verdadero (True), el api realiza algunas mejoras internas. Esto puede mejorar significativamente el resultado de OCR, especialmente para escaneos de PDF de baja resolución, por defecto está definido en false (False).

2.2.8.9.3. Ejemplo

Utilizando la herramienta cURL al ejecutar algo como lo siguiente:

```
curl "https://api.ocr.space/Parse/Image" -H "apikey:helloworld" --form "file=@lazyfox.jpg;type=image/jpeg" --form "language=eng" --form "isOverlayRequired=true"
```

Utilizando el comando anterior, definiendo que el idioma del texto que contiene la imagen es inglés (eng),

con la siguiente imagen nos retorna el texto de la misma:

This is a lot of 12 point text to test the ocr code and see if it works on all types of file format.
The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox.

Figura 2.14: Imagen de referencia para probar la API OCR.Space

Texto retornado:

```
This is a lot of 12 point text to test the
ocr code and see if it works on all types
of file format.

The quick brown dog jumped over the
lazy fox. The quick brown dog jumped
over the lazy fox. The quick brown dog
jumped over the lazy fox. The quick
brown dog jumped over the lazy fox.
```


2.2.8.10. Google Drive OCR

2.2.8.10.1. Introducción

La plataforma Drive brinda un grupo de APIs junto con bibliotecas cliente, ejemplos específicos de idioma y documentación para ayudar a desarrollar aplicaciones que se integren con Drive. La funcionalidad principal de las aplicaciones de Drive es descargar y cargar archivos en Google Drive. Sin embargo, la plataforma Drive proporciona mucho más que solo almacenamiento.¹

2.2.8.10.2. Habilitar la API de Google Drive

Para comenzar a integrarse con la interfaz de usuario de Google Drive, se debe habilitar la API de Drive dentro del proyecto de la plataforma en la nube de la aplicación y proporcionar detalles de configuración. Esto da acceso a la API, así como acceso a las características de integración de UI². Para habilitar la API de Drive, se deben completar los siguientes pasos:

- Ir a la consola de Google API.
- Seleccionar un proyecto.
- En la barra lateral a la izquierda, expandir las API y autenticación y seleccione las API.
- En la lista que se muestra de las API disponibles, hacer clic en el enlace de Drive API y hacer clic en Habilitar API.

2.2.8.10.3. Autenticación

Cada solicitud que se envíe de la aplicación a la API de Drive debe incluir un token de autorización. El token también identifica la aplicación a Google³.

2.2.8.10.4. Autorizando peticiones con OAuth 2.0

Todas las solicitudes a la API de Drive deben ser autorizadas por un usuario autenticado. Los detalles del proceso de autorización, o "flujo", para OAuth 2.0 varían sutilmente dependiendo del tipo de aplicación que se esté

¹Introduction to Google Drive API | Drive REST API | Google Developers. URL: <https://developers.google.com/drive/api/v3/about-sdk> (visitado 10-10-2018).

²Enable the Google Drive API | Drive REST API | Google Developers. URL: <https://developers.google.com/drive/api/v3/enable-drive-api> (visitado 10-10-2018).

³Authenticate your users | Drive REST API | Google Developers. URL: <https://developers.google.com/drive/api/v3/about-auth> (visitado 10-10-2018).

escribiendo. El siguiente proceso general se aplica a todos los tipos de aplicaciones:

- Cuando se crea una aplicación, se registra usando la Consola API de Google. Luego, Google proporciona la información que necesitará más adelante, como una ID de cliente y un código secreto de cliente.
- Activar la API de Drive en la consola de la API de Google.
- Cuando su aplicación necesita acceso a los datos del usuario, solicita a Google un ámbito de acceso particular.
- Google muestra una pantalla de consentimiento para el usuario, pidiéndole que autorice su aplicación para solicitar algunos de sus datos.
- Si el usuario lo aprueba, Google le otorga a su aplicación un token de acceso de corta duración.
- Su aplicación solicita datos de usuario, adjuntando el token de acceso a la solicitud.
- Si Google determina que su solicitud y el token son válidos, devuelve los datos solicitados.
- Algunos flujos incluyen pasos adicionales, como el uso de tokens de actualización.

Esta es la información del alcance de OAuth 2.0 para la API de Drive:

| Alcance | Sentido |
|--|--|
| https://www.googleapis.com/auth/drive | Ámbito completo y permisivo para acceder a todos los archivos de un usuario, excluyendo la carpeta de datos de la aplicación |
| https://www.googleapis.com/auth/drive.readonly | Permite el acceso de solo lectura a los metadatos del archivo y al contenido del archivo. |
| https://www.googleapis.com/auth/drive.appfolder https://www.googleapis.com/auth/drive.file | Permite el acceso a la carpeta de datos de la aplicación. Acceso por archivo a los archivos creados o abiertos por la aplicación. La autorización del archivo se otorga por usuario y se revoca cuando el usuario desautoriza la aplicación. |
| https://www.googleapis.com/auth/drive.install | Ámbito especial utilizado para permitir a los usuarios aprobar la instalación de una aplicación. |
| https://www.googleapis.com/auth/drive.metadata | Permite solamente el acceso de lectura y escritura a los metadatos del archivo (excluyendo downloadUrl y contentHints.thumbnail). |
| https://www.googleapis.com/auth/drive.metadata.readonly | Permite el acceso de solo lectura a los metadatos del archivo (excluyendo downloadUrl y contentHints.thumbnail), pero no permite ningún acceso para leer o descargar el contenido del archivo. |
| https://www.googleapis.com/auth/drive.scripts https://www.googleapis.com/auth/drive.apps.readonly | Permite el acceso a los archivos de Script de aplicaciones. Permite el acceso de solo lectura a las aplicaciones instaladas. |

Tabla 2.1: Información de alcance para OAuth 2.0

2.2.8.10.5. *Gestionar archivos y carpetas de vista general*

Los archivos almacenados en Google Drive se representan como un recurso “File” o “Archivo” en la API de Drive.

2.2.8.10.6. *Atributos de archivo*

A continuación, se muestra una lista de elementos comunes a todos los archivos almacenados en Google Drive:

- ID de archivo
- Metadatos
- Permiso
- Contenido
- Revisión histórica
- Miniatura

2.2.8.10.7. *Tipos de archivo*

A continuación, se muestra una lista de los tipos de archivos almacenados en Google Drive:

Blob - Un archivo que contiene texto o contenido binario, como imágenes, vídeos y PDF.

Carpeta - Un contenedor utilizado para organizar otros tipos de archivos en Drive. Las carpetas son archivos con el tipo MIME `application/vnd.google-apps.folder`.

Atajo o shortcut - Un archivo que enlaza con el contenido almacenado en un sistema de almacenamiento de terceros. Tipo MIME `application/vnd.google-apps.drive-sdk`.

Documento de G Suite - Un archivo creado por una aplicación de G Suite como Google Docs, Hojas de cálculo y Diapositivas. Están representados con tipos MIME en el formulario `application/vnd.google-apps.*app*` donde aplicación es el nombre de la aplicación.

2.2.8.10.8. *Subir archivos*

La API de Drive permite cargar datos de archivo al crear o actualizar un recurso de archivo. Cuando subes contenido multimedia, se utiliza una URI especial. Los métodos que admiten las cargas de medios tienen dos puntos

finales o “EndPoint” de URI.

El recurso estándar URI /upload, para los archivos. El formato del EndPoint /upload es el URI de recurso estándar con un prefijo /upload. Se utiliza esta URI cuando transfiera los datos de los recursos multimedia. Ejemplo: POST /upload/drive/v3/files.

El recurso estándar URI, para los metadatos. Si el recurso contiene campos de datos, esos campos se usan para almacenar metadatos que describen el archivo cargado. Puede utilizar este URI al crear o actualizar valores de metadatos. Ejemplo: POST /drive/v3/files.

2.2.8.10.9. Tipos de carga

Hay tres tipos de cargas que puedes realizar:

- **Carga simple: uploadType=media.** Para la transferencia rápida de un archivo pequeño (5 MB o menos).
- **Carga de varias partes: uploadType=multipart.** Para la transferencia rápida de un archivo pequeño (5 MB o menos) y los metadatos que describen el archivo, todo en una sola solicitud.
- **Carga reanudable: uploadType=resumable.** Para una transferencia más confiable, especialmente importante con archivos grandes. Las cargas reanudables son una buena opción para la mayoría de las aplicaciones, ya que también funcionan para archivos pequeños al costo de una solicitud HTTP adicional por carga.

La mayoría de las bibliotecas de cliente API de Google implementan al menos uno de los métodos.

2.2.8.10.10. Descargar archivos

La Drive API permite descargar archivos que están almacenados en Google Drive. Además, puede descargar versiones exportadas de Documentos de Google (Documentos, Hojas de cálculo, Presentaciones, etc.) en formatos que las aplicación puedan manejar. Google Drive también admite que los usuarios tengan acceso directo a un archivo a través de URL a través de la propiedad webViewLink.

Dependiendo del tipo de descarga que desee realizar (un archivo, un documento de Google o un enlace de contenido), se utilizará una de las siguientes URL:

| Objetivo | URL |
|---|---|
| Descarga un archivo | files.get con alt=media con el recurso de archivo |
| Descargar y exportar un documento de Google | files.export |
| Vincular un usuario a un archivo | webContentLink desde el recurso de archivo |

Tabla 2.2: Recursos URL para descargar Google Drive API

2.2.8.10.11. Realizar OCR Utilizando Google Drive API

Para realizar OCR en una imagen utilizando Google Drive API deben seguirse los siguientes pasos:

- Crear un nuevo documento en blanco con el MIME correcto
- Actualizar el documento con una imagen a la que se desee realizar el proceso de OCR
- Descargar el documento utilizando como formato de salida “plain/text”

Ejemplo utilizando Python con la librería PyDrive Es necesario un archivo de configuración de extensión

.yaml con el siguiente formato: Utilizando la herramienta cURL al ejecutar algo como lo siguiente:

```
client_config_backend: settings
client_config:
client_id: [ID_CLIENTE].apps.googleusercontent.com
client_secret: [SECRETO_CLIETE]
save_credentials: True
save_credentials_backend: file
save_credentials_file: credentials.json
get_refresh_token: True
oauth_scope:
- https://www.googleapis.com/auth/drive.file
- https://www.googleapis.com/auth/drive.install
```

A continuación, utilizando Python se realiza lo siguiente:

```
google_auth = GoogleAuth(settings_file='settings.yaml')
google_auth.LocalWebserverAuth()
google_drive = GoogleDrive(google_auth)
archivo = google_drive.CreateFile({'title': 'imagen a OCR'})
archivo['mimeType'] = 'application/vnd.google-apps.document'
archivo.Upload({'ocrLanguage': 'en'})
archivo.SetContentFile(' lazyfox.jpg')
archivo.Upload()
texto_obtenido = archivo.GetContentString('text/plain').encode('utf-8',
↳ 'replace').decode('utf-8-sig', replace')
```

Utilizando el código anterior, definiendo que el idioma del texto que contiene la imagen es inglés (ENG), con la siguiente imagen nos retorna el texto de la misma:

This is a lot of 12 point text to test the ocr code and see if it works on all types of file format.

The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox.

Figura 2.15: Imagen de referencia Google Drive OCR

Texto retornado:

```
This is a lot of 12 point text to test the
cor code and see if it works on all types
of file format.
```

```
The quick brown dog jumped over the
lazy fox. The quick brown dog jumped
over the lazy fox. The quick brown dog
jumped over the lazy fox. The quick
brown dog jumped over the lazy fox.
```

2.2.8.11. Computer Vision API

La API de Computer Vision, forma parte de los Microsoft Azure Cognitive Services, en su sección OCR, detecta texto en una imagen utilizando el reconocimiento óptico de caracteres y extrae las palabras reconocidas en un flujo de caracteres legible por máquina. Analiza imágenes para detectar texto incrustado, generar flujos de caracteres y habilitar la búsqueda. Ahorra tiempo y esfuerzo al poder utilizar fotos como fuente de texto en lugar de copiarlo¹.

2.2.8.11.1. Terminología

Clave API: la clave necesaria para poder utilizar la API, necesita de registro de usuario para obtener una.

EndPoint: Dirección URL con la que se interactúa mediante consultas HTTP GET o HTTP POST, para enviar y recibir la información de la API.

2.2.8.11.2. Características

Puede usar Computer Vision para extraer el texto de una imagen en un flujo de caracteres legible por máquina utilizando el reconocimiento óptico de caracteres (OCR). Si es necesario, OCR corrige la rotación del texto reconocido y proporciona las coordenadas del marco de cada palabra. OCR admite 25 idiomas (árabe, chino

¹Image Processing with the Computer Vision API | Microsoft Azure. URL: <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/> (visitado 10-10-2018).

simplificado, chino tradicional, checo, danés, holandés, inglés, finlandés, francés, alemán, griego, húngaro, italiano, japonés, coreano, noruego, polaco, portugués, rumano, ruso, serbio (cirílico y latín), eslovaco, español, sueco y turco), y detecta automáticamente el idioma del texto reconocido¹.

También puede usar la API de lectura para extraer texto impreso y manuscrito de imágenes y documentos con gran cantidad de texto. La API de lectura utiliza modelos actualizados y funciona para una variedad de objetos con diferentes superficies y fondos, como recibos, carteles, tarjetas de visita, cartas y pizarras². Computer Vision puede analizar imágenes que cumplan con los siguientes requisitos:

- La imagen debe presentarse en formato JPEG, PNG, GIF o BMP
- El tamaño del archivo de la imagen debe ser inferior a 4Mb
- Las dimensiones de la imagen deben ser superiores a 50 x 50 píxeles.
- Para OCR, las dimensiones de la imagen deben estar entre 50 x 50 y 4200 x 4200 píxeles.

La API posee diferentes niveles detallados a continuación:

- Free

20 consultas por minuto

5,000 consultas por mes

- S1

10 consultas por segundo

0 a 1 Millón de consultas - USD\$1.50 por cada 1,000 consultas

Utilizando el EndPoint mediante los encabezados enviamos los parámetros y la imagen a procesar para obtener su texto, los parámetros requeridos para procesar una consulta son los siguientes:

- **Ocp-Apim-Subscription-Key:** La clave API
- **Content-Type:** Opcional. Define cómo se envía la imagen al EndPoint, los valores válidos son: “application/json”, “application/octet-stream”, “multipart/form-data”.

Además de lo anterior los siguientes parámetros son enviados mediante POST:

- **imageFile:** El contenido binario de la imagen a procesar.

¹Recognize printed, handwritten text - Computer Vision - Azure Cognitive Services | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-recognizing-text> (visitado 10-10-2018).

²What is the Computer Vision API? - Computer Vision | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home> (visitado 10-10-2018).

- language: Opcional. El idioma del contenido de la imagen, por defecto utiliza “unk” con lo cual el servicio auto detecta el idioma del texto reconocido.
- detectOrientation: Opcional. Si se define como verdadero (True) el servicio intentará rotar la imagen en caso de que esta esté rotada.

2.2.8.11.3. Ejemplo

Utilizando la herramienta cURL al ejecutar algo como lo siguiente:

```
curl.exe -X POST "https://westcentralus.api.cognitive.microsoft.com/vision/v2.0/ocr?language=en&detectOrientation=true" -H "Ocp-Apim-Subscription-Key: API KEY" -H "Content-Type: multipart/form-data" -F "imageFile=@lazyfox.jpg;type=image/jpeg"
```

Utilizando el comando anterior, definiendo que el idioma del texto que contiene la imagen es inglés (en), con la siguiente imagen nos retornara el texto de la misma:

This is a lot of 12 point text to test the ocr code and see if it works on all types of file format.
The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox.

Figura 2.16: Imagen de referencia para probar la API Computer Vision OCR

Texto retornado:

```
This is a lot of 12 point text to test the ocr code and see if it works on all types of file format.  
  
The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox. The quick brown dog jumped over the lazy fox.
```


2.2.9. Reconocimiento y síntesis de voz

2.2.9.1. Antecedentes generales

La audición del ser humano se caracteriza por detectar sonido en un determinado rango de frecuencias que van desde 16Hz a 16KHz¹, y por comprender y diferenciar de forma sencilla distintas fuentes sonoras. Contrario a ello, para que las máquinas obtengan los mismos resultados en esta última se han trabajado durante muchos años, estos esfuerzos han concluido en resultados ya presentes en mercados mundiales y que se integran en la vida diaria poco a poco.

El proceso de reconocimiento de voz puede explicarse en dos pasos diferenciados:

Extracción de fonemas: Las unidades lingüísticas llamados fonemas son las que al agruparlos conforman sonidos que componen palabras. Los fonemas son la unidad más pequeña en la que se puede dividir un conjunto fónico; por ejemplo la palabra /paso/, "paso" está formada por una serie de 5 fonemas, ya que es el máximo de mínimas unidades en el que puede dividirse: /p/+/a/+/s/+/o/².

Para poder extraer los fonemas de una voz de entrada la señal se debe analizar espectralmente, a través de transformadas de Fourier. El espectrograma de la palabra "LEER" se ve en la figura:

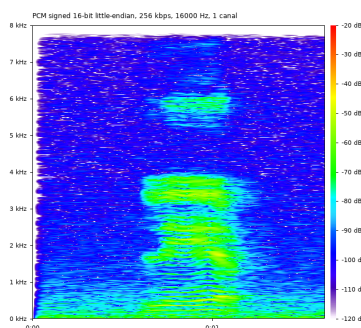


Figura 2.17: Espectrograma de la palabra leer. Fuente: Elaboración propia.

Conversión de los fonemas en palabras identificables: Este proceso se puede realizar con ayuda de métodos topológicos, probabilísticos y de redes neuronales.

¹Antonio Quilis y Joseph A Fernández. "Curso de fonética y fonología españolas: para estudiantes angloamericanos". En: (1969).

²Antonio Quilis y Joseph A.. Fernández. *Antonio Quilis y Joseph A. Fernández. Curso de fonética y fonología españolas. Para estudiantes anglo-americanos.* Consejo superior de investigaciones científicas, 1966.

2.2.9.2. Ramas del Reconocimiento de voz

Las técnicas de reconocimiento de voz se dividen en tres ramas principales¹:

Reconocimiento de voz o Reconocimiento del habla: Proceso que consiste en convertir un mensaje hablado en texto. Es la rama que más ha crecido en los últimos años.

Conversión texto-a-voz: Proceso que consiste en la generación de audio que simule la voz humana (síntesis de voz, TTS del inglés Text-To-Speech) a partir de información en formato de texto digital.

Reconocimiento de Locutores: Se basa en la identificación o verificación de la persona que habla a un sistema; generalmente su uso se define como parte de medidas de seguridad. La codificación de voz, también forma parte como una rama del reconocimiento de voz; aunque también puede considerarse un tema complementario al estar más relacionado con los canales de comunicación y el aprovechamiento del ancho de banda. Otra área importante donde se aplica esta tecnología es el control Biométrico.

2.2.9.3. Métodos de Reconocimiento de voz

Los métodos que han marcado la historia del reconocimiento de voz, pueden definirse como tres:

“Alineamiento temporal dinámico”, “Modelos ocultos de Markov”, y “Redes neuronales”.

2.2.9.3.1. *Alineamiento temporal dinámico*

El “Alineamiento temporal dinámico” (conocido como DTW, del inglés Dynamic Time Warping) se utiliza para obtener la distorsión o diferencia entre dos palabras. Muchas veces una palabra puede no pronunciarse siempre a la misma velocidad o por el mismo locutor o bajo las mismas condiciones del ambiente, es entonces necesario, ajustarla a un patrón para poder interpretar la información de manera correcta. DTW está basado en la comparativa de todas las plantillas de referencia - resultado de anteriores de entrenamientos - contra plantillas hechas a partir de vectores de parámetros, calculados a partir de los distintos segmentos en que fue dividida la señal de entrada.

Para realizar la comparación se calcula la distancia mínima entre la entrada y la referencia, y se escoge finalmente una plantilla que entregue la menor distancia. Los reconocedores de habla que se basan en DTW son sencillos de implementar y son muy efectivos con vocabularios reducidos².

¹Luis Hernández et al. “Estado del arte en tecnología del habla”. En: *Comunicaciones de Telefónica I+ D 5* (1994), págs. 3-27.

²Xuedong Huang et al. *Spoken language processing: A guide to theory, algorithm, and system development*. Vol. 1. Prentice hall PTR Upper Saddle River, 2001, págs. 4-5, 27, 793, 796, 803-805, 807, 809.

2.2.9.3.2. Modelos ocultos de Markov

Los sistemas diseñados en base a cadenas de Markov modelan procesos aleatorios, requieren menos memoria que los basados en DTW, un “Modelo Oculto de Markov” (HMM, en inglés Hidden Markov Models) pueden verse como una máquina de estados finitos en la cual el estado siguiente depende exclusivamente del estado actual, y asociado a cada transición entre estados produce un vector de parámetros.

En el reconocimiento de voz, las cadenas de Markov ajustan los distintos fonemas capturados a fonemas de palabras establecidas previamente, adquiridas por entrenamiento. Se entiende que la cantidad de posibles estados para cada fonema es finita, por tanto, el número de estados en la cadena lo es también. Un tipo de HMM apropiado especialmente para el reconocimiento de voz son los modelos de “izquierda a derecha”, dichos modelos una vez que abandonan un estado ya no pueden regresar a él. Su plantilla está conformada por vectores que se obtienen en cada nodo recorrido; cada nodo visitado genera un vector.

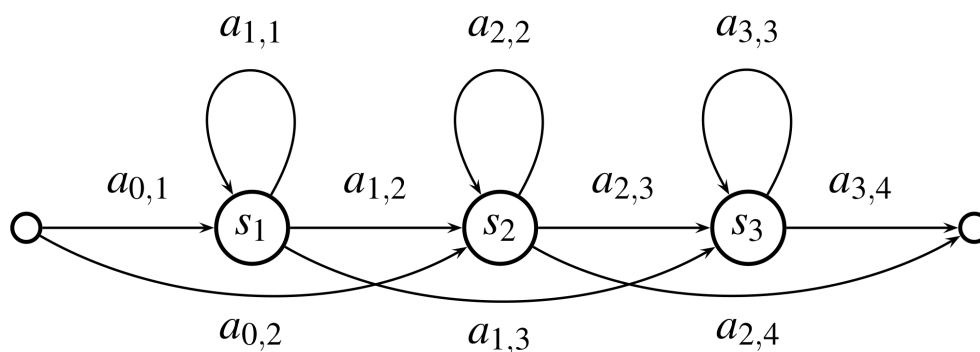


Figura 2.18: Ejemplo simplificado de nodos de una cadena de Markov. Fuente: Elaboración propia.

Las cadenas de Markov no solo sirven para extraer fonemas de señales de voz, también se aplican para unir los fonemas y transformarlos en palabras, luego de ello estas palabras se convierten en frases.

2.2.9.3.3. Redes neuronales

Las redes neuronales en el reconocimiento de voz se justifica a que estas intentan emular los complejos procesamientos cerebrales, uno de ellos precisamente es el reconocimiento del habla, además su enorme capacidad para resolver problemas que otros métodos requieren mucha carga de computación tales como: el reconocimiento de patrones, evaluaciones de hipótesis y predicciones, etc. Las redes neuronales organizan sus neuronas en forma de capas, existe una capa de entrada y una capa de salida, la capa de entrada procesa los vectores o plantillas

directamente, si el resultado de dicha operación de cada neurona individual supera un determinado umbral previamente definido, la neurona realiza la sinapsis con sus neuronas postsinápticas. De esta manera el resultado de dicha aplicación de una función de transformación no lineal, a la combinación lineal de todos los puntos contenidos en la plantilla de entrada son traspasados a las neuronas siguientes.

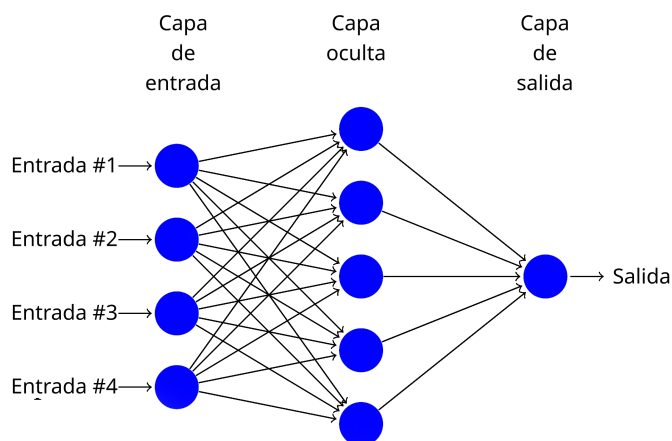


Figura 2.19: Ejemplo de red neuronal de tres capas. Fuente: elaboración propia

De esta manera las redes neuronales pueden determinar sobre cuál es la palabra que ingresa en la señal de audio proporcionada en la entrada, pese a ello su eficiencia dependerá grandemente del entrenamiento al que se haya sometido. Un entrenamiento excesivo de la red provocará que pierda efectividad, al “exigir” demasiado a las plantillas de entrada para que estas se ajusten a sus modelos internos.

La señal de voz necesita que los métodos con capacidad de procesamientos sean en dos dimensiones: espacio y tiempo. Las redes neuronales por sí mismas solo tienen capacidad de procesamiento espacial, eso obliga a combinarlas con otras técnicas de programación dinámicas tales como HMM; con esto se consigue modelar también la variable tiempo, muy acertadas clasificaciones en las entradas en la red y segmentación de la señal que entra.¹

2.2.9.4. Métodos de Síntesis de voz

La síntesis de voz puede clasificarse dentro de tres tipos, según el modelo utilizado para producir la voz. Los modelos son: síntesis por formantes, síntesis articulatoria y síntesis por concatenación.

¹José Colás Pasamontes. “Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión de habla continua en español.” En: *Estudios de Lingüística del Español* 12 (2001), pág. 1.

2.2.9.4.1. Sintetizadores por formantes

Las formantes son las resonancias características de cada articulador del tracto vocal¹. Estos determinan aspectos tales como el timbre particular de cada vocal así como definir las características individuales de cada voz. Cada palabra emitida se puede definir en términos de las frecuencias formantes de cada individuo.

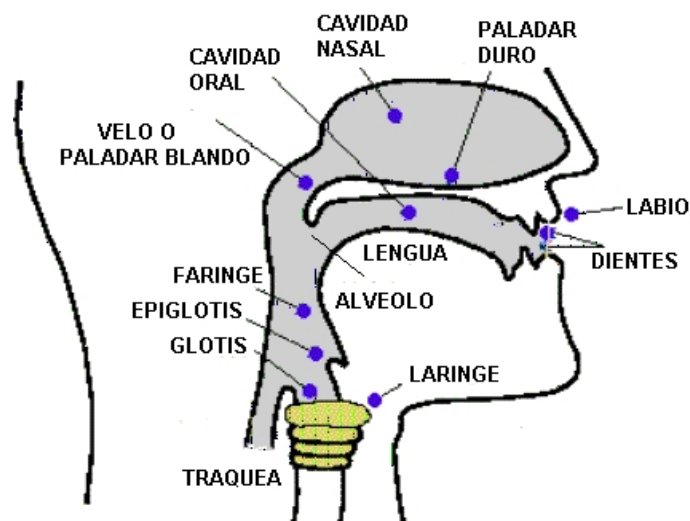


Figura 2.20: Órganos generadores de sonido. Recuperada de: <https://islandiamezzo.files.wordpress.com/2015/09/d8b98-organosdearticulacion3b2n2.jpg>

Basándose en esto, los sintetizadores por formantes, modelan dicha resonancia del tracto vocal aplicando filtros para la generación de cada formante. Dichos filtros son ajustables y tienen parámetros definibles a través de reglas. Estas reglas indican cómo se deben modificar los parámetros entre un sonido y otro sin perder la continuidad en los sistemas de generación de voz físicos².

Los sintetizadores por formantes envuelven un procedimiento flexible y manipulable, tienen la capacidad de generar diferentes voces al modificar los parámetros de sus filtros, pese a ello en la síntesis automática, se requiere un enorme número de reglas, por lo que se requieren compiladores cada vez más sofisticados, que sean capaces de integrar todo el conocimiento obtenido a partir de los experimentos en el sistema.³

¹Huang et al., *Spoken language processing: A guide to theory, algorithm, and system development*.

²Huang et al., *Spoken language processing: A guide to theory, algorithm, and system development*.

³Hernández et al., "Estado del arte en tecnología del habla".

2.2.9.4.2. *Sintetizadores articulatorios*

Los sintetizadores articulatorios utilizan un modelo físico de la generación de la voz, emulando la propagación de las ondas acústicas¹. Usan parámetros - establecidos por reglas, al igual que los sintetizadores por formantes - que modelan los movimientos mecánicos del aparato fonador, y del orden resultante de volumen y presión de aire en pulmones, laringe, tracto vocal y nasal².

Los parámetros pueden adquirirse desde la voz real a través de rayos X y resonancias magnéticas, aunque ubicar dichos sensores en el tracto vocal modifica la forma en que se genera el habla e impide sonidos totalmente naturales.

Esta técnica de síntesis no puede generar voz con una calidad similar a la síntesis por formantes y concatenación³ e incluye altos costos en investigación sobre el sistema humano de generación de voz.

2.2.9.4.3. *Sintetizadores por concatenación*

En la síntesis por concatenación, un fragmento de voz se sintetiza sencillamente reproduciendo la onda sonora con el fonema respectivo. Un discurso se sintetiza entonces concatenando varios fragmentos de voz⁴.

Al contrario de los sintetizadores anteriores, no necesita reglas ni ajuste manuales. Cada segmento es completamente natural, con lo que se puede esperar una salida del mismo tipo⁵.

El problema principal con los sintetizadores por concatenación es que, si se juntan dos segmentos de voz no adyacentes entre sí, se pueden generar discontinuidades espectrales o prosódicas. Las discontinuidades espectrales suceden cuando las formantes no coinciden en el punto de concatenación; las discontinuidades prosódicas, ocurren cuando los tonos no coinciden en el punto de concatenación. La discontinuidad puede llevar a clasificar un sistema de síntesis como inadecuado, aunque esté basado en segmentos naturales totalmente. Para solventar el problema de la discontinuidad se definieron “unidades”, que son representaciones abstractas de segmentos de voz. Las unidades suelen ir desde fonemas hasta sentencias completas. A mayor tamaño tenga la unidad, mejor será la calidad de la síntesis, pero la cantidad de unidades a almacenar podría extenderse ilimitadamente.

Lo que se plantea actualmente es utilizar híbridos que mezclan unidades grandes (como palabras) con otras

¹Hernández et al., “Estado del arte en tecnología del habla”.

²Huang et al., *Spoken language processing: A guide to theory, algorithm, and system development*.

³Huang et al., *Spoken language processing: A guide to theory, algorithm, and system development*.

⁴Huang et al., *Spoken language processing: A guide to theory, algorithm, and system development*.

⁵Huang et al., *Spoken language processing: A guide to theory, algorithm, and system development*.

más pequeñas (fonemas), manteniendo el realismo y dando flexibilidad a la pronunciación.

2.2.9.5. Formatos de audio

2.2.9.5.1. Características y terminología

Tasa de muestreo (sample rate): La tasa de muestreo (o frecuencia de muestreo) es el número de muestras de audio que se toman por segundo. La frecuencia de muestreo se mide en hercios (Hz) o kilohercios (kHz). Por ejemplo, una tasa de 16,000 muestras por segundo es igual a 16,000 Hz (o 16 kHz)¹.

Velocidad de bits: La velocidad de bits es el número de bits de datos que se envían por segundo. La velocidad de bits para un flujo de audio se mide en kilobits por segundo (kbps). La tasa de bits se calcula a partir de la tasa de muestreo y la cantidad de bits almacenados por muestra. Por ejemplo, un audio que utiliza una velocidad de muestreo de banda ancha de 16 kHz y 16 bits por muestra tiene una velocidad de bits de 256 kbps: $(16,000 * 16) / 1000$.

Compresión: La compresión es utilizada por muchos formatos de audio para reducir el tamaño de los datos de audio. La compresión reduce el número de bits almacenados por muestra y, por lo tanto, la velocidad de bits. Algunos formatos no utilizan compresión, pero la mayoría utiliza uno de los tipos básicos de compresión:

- La compresión sin pérdida reduce el tamaño del audio sin pérdida de calidad, pero la relación de compresión es generalmente pequeña.
- La compresión con pérdida reduce el tamaño del audio hasta 10 veces, pero algunos datos y cierta calidad se pierden irremediablemente en la compresión.

Los canales: Los canales indican el número de flujos del audio grabado, un canal de audio o una pista de audio es un canal de comunicación de señal de audio en un dispositivo de almacenamiento, que se utiliza en operaciones como la grabación de varias pistas y el refuerzo de sonido:

- El audio monoaural (o mono) tiene un solo canal.
- El audio estereofónico (o estéreo) normalmente tiene dos canales.

Endianness: Endianness o “extremidad” indica cómo se organizan los bytes de datos según la arquitectura de la computadora subyacente, es el orden secuencial en el que los bytes se organizan en valores numéricos más

¹Más información acerca de los formatos de audio. URL: https://online-audio-converter.com/es/help/audio_formats (visitado 10-08-2018).

grandes cuando se almacenan en la memoria o cuando se transmiten a través de enlaces digitales. El endianness es de interés en la informática, ya que dos formatos conflictivos e incompatibles son de uso común:

- Big-endian: ordena los datos por el bit más significativo.
- Little-endian: ordena los datos por el bit menos significativo.

2.2.9.5.2. *Formatos de audio más comunes*

formato audio/wav: El formato WAV (Waveform Audio File Format, en español archivo de audio de forma de onda) es un formato contenedor que a menudo se usa para transmisiones de audio sin comprimir, pero también puede contener audio comprimido¹.

Es un estándar de formato de archivo de audio de Microsoft e IBM para almacenar un flujo de bits de audio en PC. Es una aplicación del Formato de Archivo de Intercambio de Recursos (RIFF) el cual utiliza flujos de bits para almacenar los datos en "trozos", y por lo tanto también está cerca del 8SVX y el formato AIFF utilizado en las computadoras Amiga y Macintosh, respectivamente. Es el formato principal utilizado en los sistemas de Microsoft Windows para audio en bruto y normalmente sin comprimir. La codificación de flujo de bits habitual es el formato de modulación de código de pulso lineal (LPCM).

Formato audio/ogg: El formato OGG como con la mayoría de los formatos contenedores, encapsula los datos comprimidos sin procesar y permite el intercalado de datos de audio y vídeo dentro de un formato único y conveniente. Además de la encapsulación y el intercalado de múltiples flujos de datos, Ogg proporciona el encuadre de paquetes, la detección de errores y las marcas de tiempo periódicas para la búsqueda².

Ogg es un contenedor orientado a la transmisión, lo que significa que se puede escribir y leer en una sola pasada, lo que lo convierte en un ajuste natural para la transmisión por Internet y el uso en tuberías de procesamiento. Esta orientación de flujo es la principal diferencia de diseño sobre otros formatos de contenedor basados en archivos.

El formato es mantenido por la Xiph.org Foundation. Puede utilizar secuencias de audio comprimidas con los siguientes códecs con pérdida:

- Opus(audio/ogg;codecs=opus). Opus es un codec de audio totalmente abierto, libre de regalías y altamente versátil. Opus no tiene igual para la transmisión interactiva de voz y música a través de Internet, pero

¹WAVE Audio File Format. URL: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000001.shtml> (visitado 10-08-2018).

²Xiph.org: Ogg. URL: <https://www.xiph.org/ogg/> (visitado 10-08-2018).

también está diseñado para aplicaciones de almacenamiento y transmisión. Está estandarizado por el Grupo de trabajo de ingeniería de Internet (IETF) como RFC 6716 que incorporó la tecnología del códec SILK de Skype y el códec CELT de Xiph.Org¹.

- Vorbis (audio/ogg;codecs=vorbis). Ogg Vorbis es un formato de audio comprimido de uso general, no patentado, sin patentes y sin derechos de autor, para audio y música de calidad media a alta (8kHz-48.0kHz, 16+ bit, polifónico) a velocidades fijas y variables, de 16 a 128 kbps/canal. Esto coloca a Vorbis en la misma clase competitiva que las representaciones de audio como MPEG-4 (AAC), y similares, pero con mayor rendimiento que la capa de audio 3 MPEG-1/2, audio MPEG-4 (TwinVQ), WMA y PAC².

formato audio/flac: El formato FLAC (Free Lossless Audio Codec, en español codec de audio libre sin pérdida) es un formato de codificación de audio para la compresión sin pérdida de audio digital, y también es el nombre del proyecto de software libre que produce las herramientas FLAC, el paquete de software de referencia incluye una implementación de códec. El audio digital comprimido por el algoritmo de FLAC se puede reducir típicamente a entre 50 y 70 por ciento de su tamaño original y descomprimirlo a una copia idéntica de los datos de audio originales³.

FLAC es un formato abierto con licencias libres de derechos y una implementación de referencia que es software libre. FLAC admite el etiquetado de metadatos, la portada del álbum y la búsqueda rápida.

Formato audio/mp3 o audio/mpeg: El formato MP3 o MPEG es un formato de audio con pérdida. El Grupo de Expertos en Imágenes en Movimiento (MPEG) diseñó MP3 como parte de sus estándares MPEG-1 y, posteriormente, MPEG-2. El primer subgrupo de audio lo formaron varios equipos de ingenieros en CCETT, Matsushita, Philips, Sony, AT & T-Bell Labs, Thomson-Brandt y otros⁴. MPEG-1 Audio (MPEG-1 Parte 3), que incluía MPEG-1 Audio capa I, II y III, fue aprobado como borrador de comité para una norma ISO/IEC en 1991⁵, finalizado en 1992⁶, y publicado en 1993 como ISO/IEC 11172-3: 1993⁷.

¹Opus Codec. URL: <https://www.opus-codec.org> (visitado 10-08-2018).

²Xiph.org. URL: <https://xiph.org/vorbis/> (visitado 10-08-2018).

³Josh Coalson. *FLAC - format*. URL: <https://xiph.org/flac/format.html#stream> (visitado 10-08-2018).

⁴Leonardo Chiariglione. *RIDING THE MEDIA BITS*. URL: https://web.archive.org/web/20111101091827/http://ride.chiariglione.org/MPEG/s_1st_steps.php (visitado 10-08-2018).

⁵MPEG Press Release. URL: https://web.archive.org/web/20110503174827/http://mpeg.chiariglione.org/meetings/kurihama91/kurihama_press.htm (visitado 10-08-2018).

⁶MPEG Press Release. URL: https://web.archive.org/web/20100812034709/http://mpeg.chiariglione.org/meetings/london/london_press.htm (visitado 10-08-2018).

⁷ISO/IEC 11172-3:1993 - Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio. URL: <https://www.iso.org/standard/22412.html> (visitado 10-08-2018).

2.2.9.5.3. *Conversión de audio*

Se pueden utilizar varias herramientas para convertir audio a un formato diferente. Las herramientas pueden ser útiles cuando el audio está en un formato que no es compatible con el servicio a utilizar o en un formato sin compresión o sin pérdida. En este último caso, puede convertir el audio a un formato con pérdida para reducir su tamaño¹.

Las siguientes herramientas gratuitas están disponibles para convertir tu audio de un formato a otro:

- Sound eXchange (SoX)
- FFmpeg
- Audacity

Estas herramientas ofrecen soporte multiplataforma para múltiples formatos de audio. Además, puedes usar muchas de las herramientas para reproducir tu audio.

2.2.10. **Herramientas voz a texto - Speech-to-Text**

2.2.10.1. **CMU Sphinx**

Es un sistema de reconocimiento desarrollado por Kai-Fu Lee de habla continua independiente del hablante que utiliza modelos acústicos ocultos de Markov², Los decodificadores de voz vienen con modelos acústicos y aplicaciones de muestra. Los recursos disponibles incluyen, además, software para el entrenamiento de modelos acústicos, compilación de modelos de idiomas y un diccionario de pronunciación de dominio público, cmudict.

Debido a la amplia cantidad de diccionarios que Sphinx tiene agregados en su listado, hace posible darle soporte a una gran cantidad de idiomas, además de la libertad de poder generar tu propio diccionario con tu propio lenguaje, permitiendo entrenar el modelo acústico según sean las necesidades³.

2.2.10.2. **Cognitive Services Speech to Text API**

Abreviado CSST, convierte audio con voz en texto. Llamando a la API para reconocer el audio que proceda del micrófono, de otros orígenes de audio de streaming en tiempo real o de un archivo de audio grabado. Conforme se

¹Más información acerca de los formatos de audio.

²An overview of the SPHINX speech recognition system - Acoustics, Speech and Signal Processing. URL: https://www.ric.cmu.edu/pub_files/pub2/lee_k_f_1990_1/lee_k_f_1990_1.pdf (visitado 10-10-2018).

³About CMUSphinx – CMUSphinx Open Source Speech Recognition. URL: <https://cmusphinx.github.io/wiki/about/> (visitado 10-10-2018).

envía audio al servidor, se pueden devolver resultados parciales del reconocimiento si se solicita¹.

Pertenece a los servicios “Azure Cognitive Services” CSST API comparte la misma tecnología de reconocimiento de voz subyacente utilizada por otros productos de Microsoft como Cortana. Se puede usar la API para crear aplicaciones inteligentes activadas por voz.

2.2.10.2.1. Terminología

Clave API: la clave necesaria para poder utilizar la API, necesita de registro de usuario para obtener una.

EndPoint: Dirección URL con la que se interactúa mediante consultas HTTP GET o HTTP POST, para enviar y recibir la información de la API.

2.2.10.2.2. Características

La API solo admite expresiones cortas. Las solicitudes pueden contener hasta 10 segundos de audio con una duración total de 14 segundos. La API solo devuelve los resultados finales, no los resultados parciales o intermedios².

La API posee diferentes niveles detallados a continuación³:

- Free
 - 5 horas de conversión por mes - 1 a la vez
- Estándar
 - US\$1.00/hora - 20 consultas simultaneas

2.2.10.2.3. Autenticación

Utilizando el EndPoint para autenticación a través de token mediante consultas POST enviamos los parámetros requeridos para procesar una consulta son los siguientes⁴:

- **Ocp-Apim-Subscription-Key: API KEY:** la clave API para crear el token de acceso.

Ejemplo utilizando la herramienta cURL:

¹Speech to Text API | Microsoft Azure. URL: <https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/> (visitado 10-10-2018).

²API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs. URL: <https://docs.microsoft.com/es-es/azure/cognitive-services/speech-service/rest-api> (visitado 10-10-2018).

³API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs.

⁴API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs.

```
curl -v -X POST
"https://westus.api.cognitive.microsoft.com/sts/v1.0/issueToken" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Content-Length: 0" \
-H "Ocp-Apim-Subscription-Key: API KEY"
```

Lo anterior nos retorna un JSON Web Token (JWT) de 10 minutos de duración para utilizar la API.

2.2.10.2.4. *Funcionamiento*

Utilizando el EndPoint mediante consultas POST enviamos los parámetros y el archivo de audio a procesar para obtener su texto, los parámetros POST requeridos para procesar una consulta son los siguientes:

- **language:** El idioma a utilizar para identificar el texto de la voz en el caso del español sería es-ES.
- **format:** Opcional. El formato de respuesta de la consulta, puede ser simple, o detailed (detallado), detallado muestra más opciones tales como la confianza del texto reconocido así como diferentes alternativas válidas, por defecto es simple.
- **profanity:** Opcional. Especifica cómo se tratará el lenguaje soez, puede ser masket (enmascarado), removed (removido), o raw (en crudo), por defecto la opción está como masket.

Además de ello existen otros parámetros los cuales se envían mediante los encabezados (headers) de la consulta son los siguientes:

- **Ocp-Apim-Subscription-Key:** Si no se desea hacer uso de tokens aquí se definiría la clave API.
- **Authorization:** Si se utilizan tokens el contenido del token se incluiría aquí precedido por la palabra "Bearer".
- **Content-type:** Describe el formato del archivo a enviar.
- **Accept:** Si se establece debe definirse como application/json, es el formato de respuesta de la API.

2.2.10.2.5. *Ejemplo*

Ejemplo utilizando HTTP POST:

```
POST speech/recognition/conversation/cognitiveservices/v1?language=es-ES&
format=detailed HTTP/1.1
Accept: application/json
Content-Type: audio/wav; codecs=audio/pcm; samplerate=16000
Ocp-Apim-Subscription-Key: API KEY
Host: westus.stt.speech.microsoft.com
```

2.2.10.3. IBM Watson Speech-to-Text API

El servicio de la API voz a texto permite agregar capacidades de transcripción de voz a las aplicaciones.

Combina información sobre la estructura del lenguaje con la composición de la señal de audio¹.

Transcribe automáticamente audio de 7 idiomas en tiempo real. Identifica y transcribe rápidamente lo que se está discutiendo, incluso a partir de audio de menor calidad, a través de una variedad de formatos de audio e interfaces de programación (REST HTTP, WebSocket, HTTP asíncrono).

2.2.10.3.1. Terminología

Clave API: la clave necesaria para poder utilizar la API, necesita de registro de usuario para obtener una.

EndPoint: Dirección URL con la que se interactúa mediante consultas HTTP GET o HTTP POST, para enviar y recibir la información de la API.

2.2.10.3.2. Características

La API posee diferentes niveles detallados a continuación²:

- Free

100 minutos de conversión por mes gratis.

- Standard

1 - 250,000 Minutos - US\$0.02/Minuto

250,001 - 500,000 Minutos - US\$0.015/Minuto

500,001 - 1,000,000 Minutos - US\$0.0125/Minuto

1,000,000+ Minutos - \$0.01/Minuto

- Premium

Según acuerdo

¹ *Watson Speech to Text - Overview | IBM Cloud.* URL: <https://www.ibm.com/cloud/watson-speech-to-text> (visitado 10-10-2018).

² *Pricing - Pricing | IBM.* URL: <https://www.ibm.com/cloud/watson-speech-to-text/pricing> (visitado 10-10-2018).

2.2.10.3.3. *Funcionamiento*

Utilizando el EndPoint mediante consultas HTTP POST enviamos los parámetros y el archivo de audio a procesar para obtener su texto, los parámetros POST disponibles para realizar el proceso son numerosos, en esta lista sólo listamos los requeridos para procesar una consulta¹:

- **data-binary:** El contenido del audio a procesar como.
- **model:** El idioma a utilizar para identificar el texto de la voz en el caso del español sería es-ES_BroadbandModel.

Además de ello existen otros parámetros los cuales se envían mediante los encabezados (headers) de la consulta son los siguientes:

- **apikey:** La clave API para utilizar el servicio.
- **content_type:** El formato del archivo de audio a procesar, entre los disponibles estan “audio/wav”, “audio/ogg”, “audio/webm”, etc.

2.2.10.3.4. *Ejemplo*

Ejemplo utilizando cURL:

```
curl -X POST -u "apikey:API KEY"
--header "Content-Type: audio/wav"
--data-binary @archivo.wav
"https://stream.watsonplatform.net/speech-to-text/api
/v1/recognize?model=es-ES_BroadbandModel"
```

2.2.10.4. **Google Speech Recognition API**

Google Speech Recognition API en español transcripción de voz de Google Cloud es una API fácil de usar que permite a los desarrolladores convertir audio en texto fácilmente gracias a la aplicación de potentes modelos de redes neuronales. Esta API reconoce 120 idiomas y variantes para ayudarte a gestionar tu base de usuarios internacional y, entre otras cosas, permite habilitar el control por comandos de voz o transcribir las conversaciones de los centros de llamadas. Además, utiliza la tecnología del aprendizaje automático de Google para procesar audios grabados previamente o en tiempo real².

¹*Getting started tutorial.* URL: <https://cloud.ibm.com/docs/services/speech-to-text?topic=speech-to-text-gettingStarted#gettingStarted> (visitado 10-10-2018).

²*Reconocimiento de voz con Transcripción de voz de Cloud | API de Cloud Speech-to-Text | Google Cloud.* URL: <https://cloud.google.com/speech-to-text/> (visitado 10-08-2018).

La API Transcripción de voz de Cloud reconoce 120 idiomas y variantes para ayudarte a gestionar tu base de datos internacional. Además, te permite filtrar el contenido inapropiado en los resultados de texto de todos los idiomas.

Transcripción de voz de Cloud puede transcribir de forma instantánea los audios a medida que se reproducen o conforme habla el usuario. Además, esta API puede transcribir el contenido de audios guardados en archivos y puede analizar audios de formato tanto corto como largo.

Transcripción de voz de Cloud proporciona varios modelos de reconocimiento de voz prediseñados, así que puedes elegir el que se ajuste mejor a cada caso práctico (por ejemplo, los comandos de voz). Nuestro modelo de transcripción de vídeo predefinido, por ejemplo, es ideal para indexar o subtitar vídeos o contenido en los que intervienen varios interlocutores, y utiliza una tecnología de aprendizaje automático similar a la empleada para la generación de subtítulos en YouTube.

2.2.10.4.1. Funciones

- **Conversión de voz en texto con la tecnología del aprendizaje automático**
- **Reconocimiento automático de voz:** El reconocimiento automático de voz (ASR) basado en redes neuronales de aprendizaje profundo dota a tus aplicaciones de búsqueda por voz o transcripción de voz.
- **Vocabulario internacional:** Reconoce 120 idiomas y variantes con un amplio vocabulario.
- **Sugerencias lingüísticas:** Es posible personalizar el reconocimiento de voz según un contexto específico; para ello, basta con suministrar un conjunto de palabras y frases que tienen muchas probabilidades de aparecer. Esta característica es especialmente útil a la hora de añadir palabras y nombres personalizados a la lista de vocabulario, así como en casos prácticos de control por voz.
- **Transcripción de audios grabados previamente o en tiempo real:** El sonido puede proceder del micrófono de una aplicación o se puede enviar desde un archivo de audio grabado previamente (insertado o a través de Google Cloud Storage). Se admiten diversas codificaciones de audio, como FLAC, AMR, PCMU y LINEAR16.
- **Detección automática de idioma:** En las situaciones multilingües, ahora es posible especificar entre dos y cuatro códigos de idioma. Transcripción de voz de Cloud identificará correctamente el idioma que se habla y

proporcionará una transcripción.

- **Tratamiento del ruido:** Es capaz de procesar audio de una multitud de entornos ruidosos sin necesidad de recurrir a otros procesos de reducción de ruido.
- **Filtrado de contenido inapropiado:** Puedes filtrar el contenido inapropiado en los resultados de texto en algunos idiomas.

2.2.10.4.2. Precios

- Reconocimiento de voz (todos los modelos excepto vídeo)

Hasta 60 minutos - Gratis

Más de 60 minutos a 1 millón de minutos - 0.006\$USD cada 15 segundos.

- Reconocimiento de voz en vídeo

Hasta 60 minutos - 0,006\$USD

Más de 60 minutos a 1 millón de minutos - 0.012\$USD cada 15 segundos.

2.2.10.4.3. Ejemplo

Ejemplo utilizando cURL

```
curl --data-binary @voz_16kHz.flac -H 'Content-Type: audio/x-flac; rate=16000' -X POST 'http://www.google.com/speech-api/v2/recognize?key=AIzaSyBOTi4mM-6x9WDnZIjIeyEU21OpBXqWBgw&lang=es-ES&client=chromium'
```

2.2.11. Herramientas texto a voz - Text-to-Speech

2.2.11.1. Festival

Festival ofrece un marco general para la construcción de sistemas de síntesis de voz. En conjunto, ofrece texto completo a voz a través de numerosas API's: desde el nivel de shell, a través de un intérprete de comandos Scheme, como una biblioteca de C++, desde Java y una interfaz de Emacs. Festival es multilingüe (actualmente inglés (británico y estadounidense) y español), aunque el inglés es el más avanzado. Otros grupos lanzan nuevos lenguajes para el sistema. Y las herramientas completas y la documentación para construir nuevas voces están disponibles a través del proyecto FestVox de Carnegie Mellon's¹.

¹Festival. URL: <http://www.cstr.ed.ac.uk/projects/festival/> (visitado 10-10-2018).

El sistema está escrito en C++ y utiliza la Biblioteca de herramientas de voz de Edimburgo para la arquitectura de bajo nivel y tiene un intérprete de comandos basado en Esquema (SIOD) para el control. La documentación se proporciona en el formato de información FSF que puede generar, un manual impreso, archivos de información y HTML.

Festival es software libre. Festival y las herramientas de habla se distribuyen bajo una licencia de tipo X11 que permite un uso comercial y no comercial sin restricciones por igual.

2.2.11.1.1. características

- Soporte de Inglés (inglés y británico), español y galés a voz
- Módulos independientes de lenguaje configurables externamente:
 - Sets de tonos
 - léxicos
 - reglas de letras a sonido
 - tokenización
 - parte del etiquetado de voz
 - entonación y duración
- Sintetizadores de forma de onda:
 - Motor de selección de unidades multisync
 - Motor de síntesis paramétrica HTS
 - Motor de síntesis paramétrica Clustergen
 - Motor de selección de unidades clunits.
 - basado en diphone: LPC excitado residual (y PSOLA no para distribución) Soporte de base de datos MBROLA.
- Distribuido bajo una licencia libre de tipo X11
- Distribución portátil (Unix)
- Documentación en línea
- Marcado SABLE, Emacs, cliente/servidor, interfaces de scripting

2.2.11.1.2. *hisvavoces*

Las voces sintéticas en castellano que están disponibles para Linux no son muy utilizables por su escasa claridad y su monotonía, por lo que la Consejería de Innovación Ciencia y Empresa de la Junta de Andalucía ha creado estas dos voces -masculina y femenina- para su utilización en Guadalinux y en cualquier otra distribución de software libre que las precise, para los fines de esta tesis la herramienta festival.

2.2.11.2. **gTTS (Google Text to Speech)**

Una biblioteca de Python y una herramienta de CLI para interactuar con la API de texto a voz de Google Translate¹. Escribe datos mp3 hablados en un archivo, algunas de sus características son:

- Lectura de oraciones específicas para el discurso personalizable que permite la lectura de longitudes de texto ilimitadas, todo manteniendo la entonación, las abreviaturas, los decimales y demás.
- Pre-procesadores de texto personalizables que pueden, por ejemplo, proporcionar correcciones de pronunciación.
- Recuperación automática de idiomas admitidos.

2.2.11.2.1. *Ejemplo*

Haciendo uso de la terminal:

```
gtts-cli 'Prueba de gTTS' --lang es --output prueba.mp3
```

2.2.11.3. **Cognitive Services Text to Speech API**

Abreviado CSTS, con la API de texto a voz desarrollado por Microsoft, las aplicaciones puede enviar solicitudes HTTP a un servidor en la nube, donde el texto se sintetiza instantáneamente en voz humana y se devuelve como un archivo de audio. Esta API se puede utilizar en muchos contextos diferentes para proporcionar conversión de texto a voz en tiempo real en una variedad de voces e idiomas diferentes².

¹*gTTS — gTTS documentation*. URL: <https://gtts.readthedocs.io/en/latest/> (visitado 10-10-2018).

²*Text to Speech API | Microsoft Azure*. URL: <https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/> (visitado 10-10-2018).

2.2.11.3.1. Terminología

Clave API: la clave necesaria para poder utilizar la API, necesita de registro de usuario para obtener una.

EndPoint: Dirección URL con la que se interactúa mediante consultas HTTP GET o HTTP POST, para enviar y recibir la información de la API.

2.2.11.3.2. Características

El cuerpo de cada solicitud POST se envía como lenguaje de marcado de síntesis de voz (SSML). SSML permite elegir la voz y el idioma de la voz sintetizada que será devuelta por el servicio de texto a voz¹.

La API posee diferentes niveles detallados a continuación²:

- Free
 - 5 horas de conversión por mes
 - 1 consulta por vez
- Estándar
 - US\$1.00/hora
 - 20 consultas simultaneas

2.2.11.3.3. Autenticación

Utilizando el EndPoint para autenticación a través de token mediante consultas POST enviamos los parámetros requeridos para procesar una consulta son los siguientes³:

- **Ocp-Apim-Subscription-Key: API KEY:** la clave API para crear el token de acceso.

Ejemplo utilizando la herramienta cURL:

```
curl -v -X POST
"https://westus.api.cognitive.microsoft.com/sts/v1.0/issueToken" \
-H "Content-type: application/x-www-form-urlencoded" \
-H "Content-Length: 0" -H "Ocp-Apim-Subscription-Key: API KEY"
```

Lo anterior nos retorna un JSON Web Token (JWT) de 10 minutos de duración para utilizar la API, una vez caducado es necesario crear un nuevo token.

¹API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs.

²API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs.

³API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs.

2.2.11.3.4. *Funcionamiento*

Utilizando el EndPoint mediante consultas HTTP POST en el formato SSML enviamos los parámetros y el texto a procesar para obtener su audio, al ser de esta manera, el contenido de la consulta HTTP POST no puede superar los 1,000 caracteres de longitud, los parámetros requeridos para procesar una consulta son los siguientes¹:

- **lang:** El idioma a utilizar para identificar el texto de la voz en el caso del español sería es-ES.
- **gender:** El género de la voz a utilizar, puede ser Male (hombre) o Female (mujer).
- **name:** Especifica el nombre de la voz a utilizar, va estructurado de la forma “Microsoft Server Speech Text to Speech Voice(idioma, nombre)”, en caso del idioma español por ejemplo, una de las voces femeninas quedaría tal que “Microsoft Server Speech Text to Speech Voice (en-US, ZiraRUS)”.

Además de ello existen otros parámetros los cuales se envían mediante los encabezados (headers) de la consulta son los siguientes:

- **Authorization:** El contenido del token se incluiría aquí precedido por la palabra “Bearer”.
- **Content-type:** Describe el formato del texto a procesar, el único valor aceptado es application/ssml+xml.
- **X-Microsoft-OutputFormat:** Establece el formato de salida de audio.
- **User-Agent:** El nombre de la aplicación que utiliza el EndPoint, máximo 255 caracteres.

Lista de formatos de audio permitidos para la salida de audio:

- raw-16khz-16bit-mono-pcm
- raw-8khz-8bit-mono-mulaw
- riff-8khz-8bit-mono-alaw
- riff-8khz-8bit-mono-mulaw
- riff-16khz-16bit-mono-pcm
- audio-16khz-128kbitrate-mono-mp3
- audio-16khz-64kbitrate-mono-mp3
- audio-16khz-32kbitrate-mono-mp3
- raw-24khz-16bit-mono-pcm
- riff-24khz-16bit-mono-pcm

¹API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs.

- audio-24khz-160kbitrate-mono-mp3
- audio-24khz-96kbitrate-mono-mp3
- audio-24khz-48kbitrate-mono-mp3

2.2.11.3.5. Ejemplo

Ejemplo utilizando HTTP POST:

```
POST /cognitiveservices/v1 HTTP/1.1
X-Microsoft-OutputFormat: audio-16khz-32kbitrate-mono-mp3
Content-Type: application/ssml+xml
Host: westus.tts.speech.microsoft.com
Content-Length: 225
Authorization: Bearer [Base64 Contenido token]

<speak version='1.0' xml:lang='en-US'><voice xml:lang='es-ES' xml:gender='
  Female'
  name='Microsoft Server Speech Text to Speech Voice (es-ES, HelenaRUS)'>
Prueba de la API Cognitive Service Text to Speech
</voice></speak>
```

2.2.11.4. VoiceRSS API

La API Voice RSS Text-to-Speech permite la conversión de contenido textual a voz. Simplemente conectando a la API de texto a voz con unas pocas líneas de código y obtener una representación verbal de un contenido textual. Para convertir texto a voz, no se necesita hardware especial para preocuparse por el uso intensivo de la CPU y la memoria durante las operaciones de conversión.

Voz RSS proporciona voces muy humanas y soporta 26 idiomas. La API de texto a voz (TTS, por sus siglas en inglés) es compatible con el uso multiplataforma es un servicio gratuito de texto a voz en línea. Voice RSS permite que las aplicaciones entreguen información auditiva a través de la API de texto a voz sin necesidad de instalar ningún software.¹

2.2.11.4.1. Terminología

Clave API: la clave necesaria para poder utilizar la API, necesita de registro de usuario para obtener una.

EndPoint: Dirección URL con la que se interactúa mediante consultas HTTP GET o HTTP POST, para enviar y recibir la información de la API.

¹Voice RSS - Text-to-speech (TTS) API Overview. URL: <http://www.voicerss.org/api/default.aspx> (visitado 10-10-2018).

2.2.11.4.2. *Características*

La API de voz a texto Voice RSS crea una transmisión de audio de alta calidad. La API no admite callback y devuelve datos de audio o un mensaje de error de forma síncrona en la misma solicitud.

La API Text-to-Speech proporciona un uso en línea muy fácil y simple. Simplemente coloque la clave API y el idioma en la URL y recibirá datos de audio que contienen el texto especificado.¹.

La API posee diferentes niveles detallados a continuación²:

- Free
350 consultas/día
- Advanced - US\$5.00/Mes
1,000 consultas/día
- Premium - US\$15.00/Mes
5,500 consultas/día
- Premium Plus - US\$40.00/Mes
20,000 consultas/día
- Business - US\$120.00/Mes
100,000 consultas/día
- Enterprise - US\$300.00/Mes
consultas ilimitadas
- Personalizado
Según acuerdo

2.2.11.4.3. *Funcionamiento*

Utilizando el EndPoint mediante consultas HTTP POST o HTTP GET enviamos los parámetros y el texto a procesar para obtener su audio, al ser de esta manera, el contenido de la consulta HTTP POST no puede superar los 100Kb, los parámetros requeridos para procesar una consulta son los siguientes³:

¹Voice RSS - Text-to-speech (TTS) API documentation. URL: <http://www.voicerss.org/api/documentation.aspx> (visitado 10-10-2018).

²Voice RSS - Free Text-to-speech (TTS) online service. URL: <http://www.voicerss.org/default.aspx> (visitado 10-10-2018).

³Voice RSS - Text-to-speech (TTS) API documentation.

- **key:** La clave API del servicio.
- **src:** El texto a convertir a voz.
- **hl:** El lenguaje a utilizar para hacer la síntesis de voz, para el español se utiliza es-es (Español de España) o es-mx (Español de Mexico).
- **r:** Opcional. El ratio de habla, puede ser desde -10, lo más lento a 10, lo más rápido por defecto es 0.
- **hl:** El codec que se utilizará para almacenar el audio, puede ser “MP3”, “WAV”, “AAC”, “OGG” y “CAF”, por defectos es “WAV”.
- **f:** El formato de audio a utilizar, por defecto es 8khz_8bits_mod0, se utilizará en nuestro caso 16khz_16bit_mono.

2.2.11.4.4. Ejemplo

Ejemplo utilizando HTTP GET:

```
https://api.voicerss.org/?key=<API key>&hl=es-mx&r=2&c=WAV&f=16khz_16bit_mono&src=Prueba de conversion de texto a voz
```

2.2.12. Web scraping

El web scraping o raspado web, son métodos para la extracción de datos de sitios web¹. El software de rastreo web puede acceder a la World Wide Web directamente mediante el Protocolo de transferencia de hipertexto (HTTP), o mediante un navegador web. Si bien un usuario puede realizar manualmente el web scraping, el término generalmente se refiere a procesos automatizados implementados usando un bot o un rastreador web. Es una forma de copia, en la que se recopilan y copian datos específicos de la web, generalmente en una base de datos local central u hoja de cálculo, para su posterior recuperación o análisis.

El raspado web de una página web implica buscarla y extraerla². Recuperar es la descarga de una página (lo que hace un navegador cuando se ve una página). Por lo tanto, el rastreo web es un componente principal del raspado web, para recuperar páginas para su procesamiento posterior. Una vez recuperado, entonces la extracción puede tener lugar. El contenido de una página puede ser analizado, buscado, reformateado, sus datos copiados en una hoja de

¹ Geoff Boeing y Paul Waddell. “New insights into rental housing markets across the united states: web scraping and analyzing craigslist rental listings”. En: *Journal of Planning Education and Research* 37.4 (2017), págs. 457-476.

²Eloisa Vargiu y Mirko Urru. “Exploiting web scraping in a collaborative filtering-based approach to web advertising.” En: *Artif. Intell. Research* 2.1 (2013), págs. 44-54.

cálculo, etc. Los raspadores web suelen sacar algo de una página, para utilizarlo para otro propósito en otro lugar. Un ejemplo sería encontrar y copiar nombres y números de teléfono, o compañías y sus URL, en una lista (contact scraping).

El raspado web se utiliza para el raspado de contactos y como un componente de las aplicaciones utilizadas para la indexación web, la minería web y la minería de datos, el monitoreo de cambios de precios en línea y la comparación de precios, el raspado de revisión de productos (para vigilar la competencia), la recopilación de listados de bienes raíces, el monitoreo de los datos meteorológicos, detección de cambios en un sitio web, investigación, seguimiento de presencia y reputación en línea, etc.

Las páginas web se crean utilizando lenguajes de marcado basados en texto (HTML y XHTML), y con frecuencia contienen una gran cantidad de datos útiles en forma de texto. Sin embargo, la mayoría de las páginas web están diseñadas para usuarios finales humanos y no para facilitar el uso automatizado. Debido a esto, se crearon kits de herramientas que raspan el contenido web. Un raspador web es una Interfaz de programación de aplicaciones (API) para extraer datos de un sitio web. Compañías como Amazon AWS y Google proporcionan herramientas de rastreo web, servicios y datos públicos disponibles de forma gratuita para los usuarios finales.

Las formas más nuevas de raspado web implican escuchar los orígenes de datos de los servidores web. Por ejemplo, JSON se usa comúnmente como un mecanismo de almacenamiento de transporte entre el cliente y el servidor web.

Existen métodos que algunos sitios web utilizan para evitar el raspado web, como detectar y deshabilitar a los robots para que no puedan rastrear sus páginas. Como respuesta, hay sistemas de rastreo web que se basan en el uso de técnicas de análisis de DOM, visión computacional y procesamiento de lenguaje natural para simular la navegación humana para permitir la recopilación de contenido de páginas web para análisis sin conexión.

2.2.13. Lenguaje de programación

2.2.13.1. Python

2.2.13.1.1. Definición

Python es un lenguaje de programación interpretado, de alto nivel, de propósito general. Creado por Guido van Rossum, lanzado por primera vez en 1991, Python tiene una filosofía de diseño que enfatiza la legibilidad del

código, especialmente en el uso de espacios en blanco significativos. Proporciona construcciones que permiten una programación clara tanto en escalas pequeñas como grandes.¹ Van Rossum lideró la comunidad del lenguaje hasta que dejó de ser líder en julio de 2018.²

Python cuenta con un sistema de tipado dinámico y gestión automática de memoria. Es compatible con múltiples paradigmas de programación, incluyendo programación orientada a objetos, imperativo, funcional y de procedimiento, y tiene una biblioteca estándar grande y completa.³

Los intérpretes de Python están disponibles para muchos sistemas operativos. CPython, la implementación definida por referencia de Python, es un software de código abierto⁴ y tiene un modelo de desarrollo basado en la comunidad, al igual que casi todas las otras implementaciones de Python. Python y CPython son administrados por la organización sin fines de lucro Python Software Foundation.

2.2.13.1.2. *Historia*

Python fue concebido a fines de la década de 1980⁵ por Guido van Rossum en Centrum Wiskunde & Informatica (CWI) en los Países Bajos como un sucesor del lenguaje ABC (inspirado en SETL)⁶, que era capaz de manejar e interactuar con las excepciones en el sistema operativo Amoeba.⁷ Su implementación comenzó en diciembre de 1989⁸. La larga influencia de Van Rossum en Python se refleja en el título que le dio la comunidad de Python: Benevolent Dictator For Life (BDFL) - un puesto desde el cual se dio vacaciones permanentes en 12 de julio de 2018⁹.

Python 2.0 se lanzó el 16 de octubre de 2000 con muchas características nuevas importantes, que incluyen un recolector de basura cíclico y soporte para Unicode¹⁰.

¹A *Python Book: Beginning Python, Advanced Python, and Python Exercises*. URL: https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html (visitado 10-10-2018).

²*Guido van Rossum Stepping Down from Role as Python's Benevolent Dictator For Life* | *Linux Journal*. URL: <https://www.linuxjournal.com/content/guido-van-rossum-stepping-down-role-pythons-benevolent-dictator-life> (visitado 10-10-2018); *Python boss Guido van Rossum is stepping down from leading the project*. URL: <https://www.theinquirer.net/inquire/news/3035842/python-boss-guido-van-rossum-steps-down-after-30-years> (visitado 10-10-2018).

³*About Python™* | *Python.org*. URL: <https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place> (visitado 10-10-2018).

⁴*History and License — Python 3.7.2 documentation*. URL: <https://docs.python.org/3/license.html> (visitado 10-10-2018).

⁵*The Making of Python*. URL: <https://www.artima.com/intv/pythonP.html> (visitado 10-10-2018).

⁶[*Python-Dev*] *SETL (was: Lukewarm about range literals)*. URL: <https://mail.python.org/pipermail/python-dev/2000-August/008881.html> (visitado 10-10-2018).

⁷*General Python FAQ — Python 3.7.2 documentation*. URL: <https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place> (visitado 10-10-2018).

⁸*The History of Python: A Brief Timeline of Python*. URL: <https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html> (visitado 10-10-2018).

⁹*Guido van Rossum Stepping Down from Role as Python's Benevolent Dictator For Life* | *Linux Journal*.

¹⁰*What's New in Python 2.0 — Python 3.7.2 documentation*. URL: <https://docs.python.org/3/whatsnew/2.0.html> (visitado 10-10-2018).

Python 3.0 se lanzó el 3 de diciembre de 2008. Fue una revisión importante del lenguaje que no es completamente compatible con versiones anteriores¹. Muchas de sus características principales fueron reimportadas a la serie de versiones Python 2.6.x² y 2.7.x. Las versiones de Python 3 incluyen la utilería 2to3, que automatiza (al menos parcialmente) la traducción del código de Python 2 a Python 3³.

La fecha de finalización de la vida útil de Python 2.7 se fijó inicialmente en 2015 y luego se pospuso hasta 2020 debido a la preocupación de que una gran parte del código existente no podría trasladarse fácilmente a Python 3.⁴ En enero de 2017, Google anunció el trabajo en un transcompilador Python 2.7 to Go para mejorar el rendimiento bajo cargas de trabajo concurrentes⁵.

2.2.13.1.3. Características y filosofía

Python es un lenguaje de programación multi-paradigma. Es compatible totalmente con programación orientada a objetos y la programación estructurada, y muchas de sus características son compatibles con la programación funcional y la programación orientada a aspectos (incluida la metaprogramación⁶ y los metaobjetos (métodos mágicos))⁷. Muchos otros paradigmas son compatibles a través de extensiones, incluido el diseño por contrato⁸ y la programación lógica⁹.

Python utiliza la tipificación dinámica y una combinación de recuento de referencias y un recolector cíclico de basura para la administración de la memoria. También proporciona resolución de nombres dinámica (enlace tardío), que enlaza los nombres de variables y métodos durante la ejecución del programa. El diseño de Python ofrece algo de soporte para la programación funcional en el lenguaje Lisp tradicional. Tiene las funciones filter(), map() y reduce(); comprensión de listas, diccionarios y conjuntos; y generador de expresiones¹⁰. La biblioteca estándar tiene

¹Python 3.0 Release | Python.org. URL: <https://www.python.org/download/releases/3.0/> (visitado 10-10-2018).

²PEP 3000 – Python 3000 | Python.org. URL: <https://www.python.org/dev/peps/pep-3000/> (visitado 10-10-2018).

³2to3 - Automated Python 2 to 3 code translation — Python 3.7.2 documentation. URL: <https://docs.python.org/3/library/2to3.html> (visitado 10-10-2018).

⁴PEP 373 – Python 2.7 Release Schedule. URL: <https://legacy.python.org/dev/peps/pep-0373/> (visitado 10-10-2018); PEP 466 – Network Security Enhancements for Python 2.7.x | Python.org. URL: <https://www.python.org/dev/peps/pep-0466/> (visitado 10-10-2018).

⁵Grumpy: Go running Python! | Google Open Source Blog. URL: <https://opensource.googleblog.com/2017/01/grumpy-go-running-python.html> (visitado 10-10-2018).

⁶Python Metaclasses: Who? Why? When? URL: <https://www.webcitation.org/5lubkaJRC?url=http://www.python.org/community/pycon/dc2004/papers/24/metaclasses-pycon.pdf> (visitado 10-10-2018).

⁷Data model — Python v3.0.1 documentation. URL: <https://docs.python.org/3.0/reference/datamodel.html#special-method-names> (visitado 10-10-2018).

⁸Contracts for Python. URL: <http://www.nongnu.org/pydbc/> (visitado 10-10-2018).

⁹pyDatalog. URL: <https://sites.google.com/site/pydatalog/> (visitado 10-10-2018).

¹⁰PEP 289 – Generator Expressions | Python.org. URL: <https://www.python.org/dev/peps/pep-0289/> (visitado 10-10-2018).

dos módulos (itertools y functools) que implementan herramientas funcionales tomadas de Haskell y Standard ML¹.

La filosofía central del lenguaje se resume en el documento El zen de Python (PEP 20), que incluye aforismos como²:

- Hermoso es mejor que feo
- Explícito es mejor que implícito
- Simple es mejor que complejo
- Complejo es mejor que complicado
- La legibilidad cuenta

En lugar de tener toda su funcionalidad integrada en su núcleo, Python fue diseñado para ser altamente extensible. Esta compacta modularidad lo ha hecho particularmente popular como medio para agregar interfaces programables a las aplicaciones existentes. La visión de Van Rossum de un lenguaje básico pequeño con una gran biblioteca estándar y un intérprete fácilmente extensible surgió de sus frustraciones con ABC, que propugnaban el enfoque opuesto³

2.2.13.2. API

Application Programming Interface (cuya traducción es Interfaz de Programación de Aplicaciones). El concepto hace referencia a los procesos, las funciones y los métodos que brinda una determinada biblioteca de programación a modo de capa de abstracción para que sea empleada por otro programa informático.

Podemos entender la API como un código que indica a las aplicaciones cómo pueden mantener una comunicación entre sí. Estas reglas permiten que los distintos programas mantengan interacciones.

Otra manera de comprender qué es una API se vincula al suministro de funciones que tienen un uso extendido. De este modo, un programador puede recurrir a la funcionalidad de una API y así evitar iniciar la tarea de programación desde cero. Gracias a su flexibilidad, las APIs ayudan a evitar un gran volumen de trabajo que suele ser común a muchos proyectos, ofreciendo en cambio un código que ha sido probado por millones de usuarios y que, en muchos casos, continúa recibiendo mantenimiento y actualizaciones para asegurar su estabilidad.

¹*itertools — Functions creating iterators for efficient looping — Python 3.7.2 documentation.* URL: <https://docs.python.org/3/library/itertools.html> (visitado 10-10-2018).

²*PEP 20 – The Zen of Python | Python.org.* URL: <https://www.python.org/dev/peps/pep-0020/> (visitado 10-10-2018).

³*The Making of Python.*

Una API sirve para establecer una comunicación con una base de datos, un sistema operativo o un protocolo de comunicaciones, por citar algunas posibilidades. Incluso las redes sociales emplean distintas API: una herramienta para gestionar tuits, por ejemplo, puede utilizar una API para establecer una comunicación con los servidores que utiliza Twitter.

Una de las claves del funcionamiento de las API es la facilidad de integración. Estas herramientas tienen que resultar simples de integrar a otro software para que las comunicaciones puedan desarrollarse con éxito. De igual manera, sus actualizaciones no deberían generar conflictos para que su labor siga siendo óptima.

Para evitar los problemas relacionados con sus actualizaciones, es necesario que se cumpla uno de los siguientes dos puntos, o bien ambos: que el formato de los datos que envía como respuesta al programa en el cual se integra no cambie; que la documentación explique con mucha claridad cómo adaptar el programa para que pueda continuar utilizando dichos datos.

Además de las redes sociales, una de las plataformas cuya API es más usada a nivel global es PayPal. Se trata de un sistema de pagos en línea ideal para simplificar las transacciones internacionales, tanto particulares como profesionales. Cuando un desarrollador desea incluir una pasarela de pago de esta compañía en su sitio web, debe utilizar la API distribuida gratuitamente y seguir las instrucciones para dar con la configuración que más se adapte a sus necesidades.

En pocos pasos, es posible incrustar el formulario de PayPal, simplemente aprovechando las funciones que ofrece su API, las cuales son el resultado de cientos de horas de trabajo y han sido probadas por millones de personas en todo el mundo. Si quisiéramos hacer esto desde cero, independientemente de que en este caso particular no podríamos acceder directamente a los servidores de PayPal, tendríamos que dedicar muchísimo tiempo de trabajo, para escribir y testear cada línea de código, y eso atrasaría innecesariamente el desarrollo de nuestro proyecto.

El trabajo de los programadores es muy particular en cuanto a que se caracteriza por un constante aprendizaje, dado el ritmo al que evolucionan los lenguajes y las tecnologías en las que se utilizan. Entre sus actividades cotidianas se encuentra la consulta de la documentación de una o más APIs, para encontrar las funciones que necesita a cada paso.

Es importante recordar que tanto la API como los lenguajes de programación son “arbitrarios”, en cuanto a que podrían elaborarse de infinitas maneras: una función llamada CargarMensajes, por ejemplo, bien podría llamarse

ObtenerComentarios, AccederAConversacion o incluso CasaAzul; los nombres de las variables y funciones son escogidos por los desarrolladores, y por eso es necesario consultar la documentación de la API para conocerlos.

2.2.13.3. PIP

PIP es un sistema de administración de paquetes que se utiliza para instalar y administrar paquetes de software escritos en Python. Muchos paquetes se pueden encontrar en la fuente predeterminada de los paquetes y sus dependencias¹.

Python 2.7.9 y versiones posteriores (en la serie python2), y Python 3.4 y versiones posteriores incluyen PIP (PIP3 para Python 3) de forma predeterminada².

PIP es un acrónimo recursivo de "Paquetes de instalaciones de PIP"³.

Una ventaja importante de pip es la facilidad de su interfaz de línea de comandos, el cual permite instalar paquetes de software de Python fácilmente desde solo una orden, ejemplo:

```
pip install nombre-paquete
```

2.2.13.4. Virtualenv

Es un programa que permite crear entornos virtuales de Python. Un entorno virtual consta de un intérprete (podemos elegir la versión concreta) acompañado de todos los módulos que necesitemos instalar. Se pueden tener varios entornos distintos, instalando en cada uno los módulos que necesitemos, sin que unos entornos afecten a los otros⁴.

Un entorno virtual proporciona un método más sencillo para dar seguimiento a los paquetes instalados en el proyecto. Una vez que se ha activado el entorno virtual, este proporciona la facilidad de congelar el estado actual de los paquetes del entorno.

¹*pip - The Python Package Installer — pip 19.0.3 documentation.* URL: <https://pip.pypa.io/en/stable/> (visitado 10-10-2018).

²*Installation — pip 19.1.dev0 documentation.* URL: <https://pip.pypa.io/en/latest/installing/> (visitado 10-10-2018).

³*pip · PyPI.* URL: <https://pypi.org/project/pip/0.2/> (visitado 10-10-2018).

⁴*Entornos virtuales en Python con virtualenv | Oficina de Software Libre.* URL: <https://osl.ugr.es/2016/10/17/entornos-virtuales-en-python-con-virtualenv/> (visitado 10-10-2018).

2.2.13.5. Numpy

Uno de los módulos más importantes de Python es Numpy¹. Es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

Incorpora operaciones tan básicas como la suma o la multiplicación u otras mucho más complejas como la transformada de Fourier o el álgebra lineal. Además incorpora herramientas que nos permiten incorporar código fuente de otros lenguajes de programación como C/C++ o Fortran lo que incrementa notablemente su compatibilidad e implementación.

2.2.13.6. Pillow

Python Imaging Library (abreviada como PIL), en las versiones más nuevas conocidas como Pillow es una biblioteca gratuita para el lenguaje de programación Python que agrega soporte para abrir, manipular y guardar muchos formatos de archivo de imagen diferentes. Está disponible para Windows, Mac OS X y Linux. La última versión de PIL es 1.1.7, se lanzó en septiembre de 2009 y es compatible con Python 1.5.2–2.7, con el soporte de Python 3 que se lanzará "más adelante"².

El desarrollo parece estar descontinuado con el último compromiso para el repositorio de PIL en 2011³. En consecuencia, un proyecto sucesor llamado Pillow ha bifurcado el repositorio de PIL y ha agregado el soporte de Python 3.x⁴. Esta bifurcación ha sido adoptada como un reemplazo para el PIL original en distribuciones de Linux incluyendo Debian⁵ y Ubuntu (desde 13.04).

Pillow ofrece varios procedimientos estándar para la manipulación de imágenes. Éstas incluyen:

- Manipulaciones por pixel,
- Enmascaramiento y manejo de la transparencia
- Filtrado de imágenes, como borrosidad, contornos, suavizado o búsqueda de bordes.

¹4.1. Numpy — *Computación científica con Python para módulos de evaluación continua en asignaturas de ciencias aplicadas*. URL: http://webs.ucm.es/info/aocg/python/modulos_cientificos/numpy/index.html (visitado 10-10-2018).

²Python Imaging Library (PIL). URL: <http://www.pythonware.com/products/pil/> (visitado 10-10-2018).

³effbot / pil-2009-raclette — Bitbucket. URL: <https://web.archive.org/web/20150315041249/http://hg.effbot.org/pil-2009-raclette> (visitado 10-10-2018).

⁴Pillow — *Pillow (PIL Fork) 6.0.0.dev0 documentation*. URL: <https://pillow.readthedocs.io/en/latest/> (visitado 10-10-2018).

⁵Debian — *Details of package python-imaging in sid*. URL: <https://packages.debian.org/sid/python-imaging> (visitado 10-10-2018).

- Mejora de la imagen, como nitidez, ajuste de brillo, contraste o color,
- Agregar texto a las imágenes y mucho más.

2.2.13.6.1. Archivos de imágenes

La biblioteca de imágenes de Python es ideal para archivar imágenes y aplicaciones de procesamiento por lotes. Puede usar la biblioteca para crear miniaturas, convertir formatos de archivo, imprimir imágenes, etc.

La versión actual identifica y lee una gran cantidad de formatos. El soporte de escritura está restringido intencionalmente a los formatos de intercambio y presentación más utilizados.

2.2.13.6.2. Procesamiento de imágenes

La biblioteca contiene funciones básicas de procesamiento de imágenes, incluidas operaciones puntuales, filtrado con un conjunto de núcleos de circunvolución integrados y conversiones de espacios de color. La biblioteca también admite cambio de tamaño de imagen, rotación y transformaciones afines arbitrarias.

Hay un método de histograma que le permite extraer algunas estadísticas de una imagen. Esto se puede usar para la mejora automática del contraste y para el análisis estadístico global.

2.2.13.7. Request

Es un módulo de Python que se puede usar para enviar todo tipo de peticiones HTTP. Es una librería muy fácil de usar con muchas características variando desde pasar parámetros en URLs a enviar encabezados personalizados y Verificación SSL¹.

2.2.13.8. Pyaudio

PyAudio proporciona un enlace (wrappers) entre Python y PortAudio, la cual es una biblioteca de E/S de audio multiplataforma².

¹*Requests: HTTP para Humanos — documentación de Requests - 1.1.0.* URL: <http://docs.python-requests.org/es/latest/> (visitado 10-10-2018).

²*PyAudio Documentation — PyAudio 0.2.11 documentation.* URL: <https://people.csail.mit.edu/hubert/pyaudio/docs/> (visitado 10-10-2018).

2.2.13.9. SpeechRecognition

Es la librería que sirve como intermediaria para la utilización de API's con la funcionalidad del reconocimiento de voz, ya sean estas online u offline, debido a que la librería SpeechRecognition puede utilizar una amplia cantidad de motores disponibles para el reconocimiento de voz, facilita la tarea de la compatibilidad que suele ser uno de los mayores problemas a la hora de realizar reconocimiento de voz, soluciona dependencias que estas puedan necesitar al ser esta un intermediario genérico que se adapta correctamente a la mayor parte de las API's disponibles¹.

Speech recognition soporta los siguientes motores/API's:

- CMU Sphinx (funciona offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Voice Recognition Service
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection (funciona offline)

2.2.13.10. PortAudio

PortAudio es una biblioteca de E/S de audio gratuita, multiplataforma y de código abierto. Permite escribir programas de audio simples en 'C' o C++ que se compilarán y ejecutarán en muchas plataformas, incluidas Windows, Macintosh OS X y Unix (OSS/ALSA). Su objetivo es promover el intercambio de software de audio entre desarrolladores en diferentes plataformas. Muchas aplicaciones usan PortAudio para la E/S de audio².

PortAudio proporciona una API muy simple para grabar y/o reproducir sonido utilizando una función de devolución de llamada simple o una interfaz de lectura/escritura de bloqueo.

¹*SpeechRecognition · PyPI*. URL: <https://pypi.org/project/SpeechRecognition/> (visitado 10-10-2018).

²*PortAudio - an Open-Source Cross-Platform Audio API*. URL: <http://www.portaudio.com> (visitado 10-10-2018).

2.2.14. Raspberry PI 3

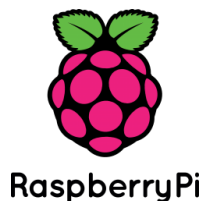


Figura 2.21: Logo raspberry PI. Recuperada de: <https://www.raspberrypi.org>

En la actualidad se cuentan con múltiples opciones para elegir la plataforma de hardware sobre la que se construirá un proyecto de índole tecnológica, tales como placas de desarrollo de microcontroladores tales como Arduino son una opción muy popular debido a su facilidad de trabajo y uso.

Otra opción es el uso de sistema de chips tales como Raspberry PI, el cual es más parecido a un computador tradicional que un Arduino, de acuerdo a las necesidades del proyecto se utilizará una u otra.

2.2.14.1. Historia y características

Raspberry Pi es la tercera marca de computadoras más vendida en el mundo. La Raspberry Pi es una computadora del tamaño de una tarjeta de crédito que se conecta a su televisor o pantalla, y un teclado y un mouse. Puede usarlo para aprender a programar y crear proyectos electrónicos, y para muchas de las cosas que hace su PC de escritorio, como hojas de cálculo, procesamiento de textos, navegar por Internet y jugar juegos. También reproduce vídeo de alta definición. La Raspberry Pi está siendo utilizada por adultos y niños de todo el mundo para aprender programación y creación digital¹.

Los modelos de Raspberry Pi que están disponibles actualmente: Pi 3 Model B, Pi 2 Model B, Pi Zero, Pi Zero W y Pi 1 Model B + y A + El Modelo A+ es la variante de bajo costo de la Raspberry Pi. Tiene 512 MB de RAM (a partir de agosto de 2016: los modelos anteriores tienen 256 MB), un puerto USB, 40 pines GPIO y ningún puerto Ethernet. El Modelo B + es la revisión final de la Raspberry Pi original. Tiene 512 MB de RAM, cuatro puertos USB, 40 pines GPIO y un puerto Ethernet.

En febrero de 2015, fue reemplazado por la placa Pi 2 modelo B, la segunda generación de Raspberry Pi. El Pi 2 comparte muchas especificaciones con el Pi 1 B+, y originalmente usaba una CPU Arm Cortex-A7 de cuatro núcleos a 900 MHz y tiene 1 GB de RAM. Algunas versiones recientes de Pi 2 (v1.2) ahora usan una CPU Arm

¹FAQs - Raspberry Pi Documentation. URL: <https://www.raspberrypi.org/documentation/faqs/> (visitado 10-10-2018).

Cortex-A53 de 900MHz.

La placa Pi 3 Modelo B se lanzó en febrero de 2016. Utiliza una CPU Arm Cortex-A53 de 1,2 GHz de 64 bits y cuatro núcleos, tiene 1GB de RAM, 802.11n LAN inalámbrica integrada y Bluetooth 4.1.

El Pi 3 Model B+ se lanzó en marzo de 2018. Utiliza una CPU Arm Cortex-A53 de 1,4 GHz de 64 bits y cuatro núcleos, tiene 1GB de RAM, Gigabit Ethernet, 802.11ac / n LAN inalámbrica integrada y Bluetooth 4.2. y es el modelo que recomendamos para su uso en las escuelas, debido a su flexibilidad para el alumno.

Pi Zero y Pi Zero W / WH son la mitad del tamaño de un Modelo A+, con una CPU de 1 GHz de un solo núcleo y 512 MB de RAM, y puertos mini-HDMI y USB On-The-Go y un conector para cámara. El Pi Zero W también tiene una LAN inalámbrica 802.11n y Bluetooth 4.1 integradas. El Pi Zero WH es idéntico al Zero W, pero viene con un cabezal previamente soldado.

El Modelo A/A+ tiene un puerto USB, el Modelo B tiene dos puertos, y el Modelo B+, Pi 2 Modelo B y Pi 3 Modelo B tienen cuatro puertos. Se pueden usar para conectar la mayoría de los dispositivos USB 2.0. Se pueden conectar dispositivos USB adicionales como ratones, teclados, adaptadores de red y almacenamiento externo a través de un concentrador USB. El Pi Zero y el Pi Zero W tienen un solo puerto micro USB, esto requiere un cable USB OTG para conectar dispositivos como teclados o concentradores.

El modelo final (no listado anteriormente) es el Módulo de cálculo, que está diseñado para aplicaciones industriales. Es un dispositivo de factor de forma pequeño que se conecta a una placa portadora, por ejemplo, una placa de circuito dentro de un producto industrial, y brinda a los fabricantes una manera fácil de usar el ecosistema de la Raspberry Pi en sus propios dispositivos.

Tras su aparición, el 28 de febrero de 2012, la Raspberry Pi vendió un millón de unidades el primer año. Las ventas alcanzaron los 2,3 millones de tarjetas a comienzos de 2014 y los 4 millones a primeros de 2015. La salida de la Raspberry Pi 2 (el 2 de febrero de 2015) aceleró el movimiento con 500.000 unidades en los dos meses siguientes a su salida. En total, son más de 5 millones de Raspberry Pi vendidas hasta hoy.

La Raspberry Pi es capaz de mostrar vídeos en calidad Bluray full HD, servir como puesto de escritorio/Internet, de consola de juegos o de herramienta de iniciación a la informática. Gracias a sus puertos de hardware también puede recibir información de dispositivos de almacenamiento externos y controlar las acciones para manejar robots, vehículos, etc.

2.2.14.2. Características técnicas Raspberry PI 3 Modelo B+



Figura 2.22: Raspberry PI 3 Modelo B+. Recuperada de: <https://www.raspberrypi.org>

Las características técnicas de la placa Raspberry PI 3 Modelo B+ están definidas a continuación¹:

- Procesador:

Chipset Broadcom BCM2387.

1,2 GHz de cuatro núcleos ARM Cortex-A53

- GPU

Dual Core VideoCore IV ® Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación.

Capaz de 1 Gpixel / s, 1.5Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA

- RAM: 1GB LPDDR2.

- Conectividad

Ethernet socket Ethernet 10/100 BaseT

802.11 b/g/n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)

- Salida de vídeo

HDMI rev 1.3 y 1.4

RCA compuesto (PAL y NTSC)

- Salida de audio

jack de 3,5 mm de salida de audio, HDMI

¹Raspberry Pi 3 Model B+ - Raspberry Pi. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (visitado 10-10-2018).

- USB 4 x Conector USB 2.0
- Conector GPIO
 - 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira
 - Proporcionar 27 pines GPIO, así como 3,3 V, +5 V y GND líneas de suministro
- Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
- Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
- Ranura de tarjeta de memoria Empuje/tire Micro SD IO

2.2.14.3. Debian Buster

Arquitecturas soportadas

Las siguientes son las arquitecturas oficialmente soportadas en Debian 10:

- PC de 32 bits (i386) y PC de 64 bits (amd64)
- ARM de 64 bits (arm64)
- ARM EABI (armel)
- ARMv7 (EABI hard-float ABI, armhf)
- MIPS (mips («big-endian») y mipsel («little-endian»))
- MIPS «little-endian» de 64 bits (mips64el)
- PowerPC «little-endian» de 64 bits (ppc64el)
- IBM System z (s390x)

Esta nueva versión de Debian trae de nuevo muchos más programas que su predecesora stretch; la distribución incluye más de 13370 paquetes nuevos, para un total de más de 57703 paquetes. La mayor parte de los programas que se distribuyen se han actualizado: más de 35532 paquetes de programas (corresponde a un 62 % de los paquetes en stretch). También se han eliminado por varios motivos un número significativo de paquetes (más de 7278, 13 % de los paquetes en stretch).

Debian trae de nuevo varias aplicaciones de escritorio y entornos. Entre otros ahora incluye los entornos de escritorio GNOME 3.30, KDE Plasma 5.14, LXDE 10, LXQt 0.14, MATE 1.20 y Xfce 4.12.

También se han actualizado las aplicaciones de productividad, incluyendo las suites de oficina:

- LibreOffice se ha actualizado a la versión 6.1
- Calligra se ha actualizado a la versión 3.1
- GNUcash se ha actualizado a la versión 33.4

Con buster, Debian introduce por primera vez un sistema de control de acceso obligatorio por omisión. Las nuevas instalaciones de Debian buster tendrán AppArmor instalado por omisión.

Además, buster es la primera publicación de Debian que se distribuye con programas basados en Rust como Firefox, ripgrep, fd, exa, etc. y un número significativo de librerías basadas en Rust (más de 450). Buster se publica con Rustc 1.34.

Las actualizaciones de otras aplicaciones de escritorio incluyen la actualización a Evolution 3.30.

El arranque seguro («Secure Boot», N. del T.) es una funcionalidad activa en la mayoría de los PCs que previene la carga de código sin firmar, protegiéndolos frente a unos tipos de virus de arranque y «rookits».

Debian puede instalarse ahora en la mayoría de los PCs que tienen el arranque seguro habilitado.

AppArmor activo por omisión

En Debian buster AppArmor está activado por omisión. AppArmor es un sistema de control de acceso obligatorio que permite restringir las capacidades de un programa (como puedan ser los permisos de lectura, escritura y ejecución sobre archivos o los permisos sobre montaje, ptrace y señales) a través de una serie de perfiles definidos por programa.

El paquete apparmor se distribuye con perfiles de AppArmor para distintos programas. Algunos otros paquetes, como por ejemplo evince, incluyen perfiles para los programas que incluyen. Puede encontrar más perfiles en el paquete apparmor-profiles-extra.

AppArmor se instala a través de las Recomendaciones del paquete del núcleo Linux de buster. Puede instalar el paquete apparmor manualmente para activar AppArmor en los sistemas que están configurados para no instalar paquetes recomendados por omisión.

3. Metodología de la Investigación

3.1. Metodología

3.1.1. Tipo de estudio

Se define la investigación como exploratoria, este tipo de investigación “Se efectúan, normalmente, cuando el objetivo es examinar un tema o problema de investigación poco estudiado o que no a sido abordado antes”², por lo que se exploró las circunstancias que rodean la problemática en específico, utilizando referencias con una vaga relación a la problemática a investigar, diferenciándose por contextos ya sea sociales, económicos, políticos o cualquier otro que pueda ser relevante en la investigación.

La temática abordada posee un escaso campo de documentación y referencias existentes dentro del país así como en el extranjero, por ende, se sentó una base para una investigación que sirva de referente a futuros investigadores que tengan un campo de estudio similar o relacionado al tema en cuestión.

3.1.2. Descripción del Contenido

El propósito del trabajo de investigación es estructurar una base de estudio en el aporte que hacen las tecnologías de bajos recursos para la asistencia de personas con problemas visuales, enfocado en la asistencia de lectura para textos físicos y digitales así como el impacto que pueden generar en el mejoramiento de la calidad de vida y el aporte que realizan como asistente de estudio en la Facultad Multidisciplinaria de Oriente de la Universidad de El Salvador.

La presente investigación pretende describir de inicio a fin el proceso de creación de un asistente para personas con discapacidad visual en el estudio de textos físicos y digitales, desde la etapa de investigación documental hasta las pruebas e implementación de un prototipo funcional.

²Roberto Hernández Sampieri. *Metodología de la investigación*; ed. McGraw-Hill: México, 1991, págs. 207-237.

3.1.3. Población y muestra

3.1.3.1. Población

Para la investigación se define una población de estudio con las siguientes características:

- Tiene una discapacidad visual que le impida la interpretación de textos físicos o digitales.
- Está en edad universitaria.
- No posee alguna discapacidad del tipo auditivo o verbal que le impida la realización de comandos por voz y la captación de las respuestas del ordenador.
- No sufre de alguna discapacidad mental que le impida el libre uso y entendimiento de la información que se le presente.
- Tiene la capacidad física y económica para desplazarse hasta las instalaciones de la universidad (FMO-UES).

Según la Dirección Nacional de Educación Superior (DNES)¹, la población estudiantil universitaria con discapacidad se divide en cuatro categorías que son:

- discapacidad física
- intelectual
- sensorial
- mental o psíquica

De las cuales la discapacidad visual se encuentra dentro de la categoría de discapacidad sensorial².

La DNES, en la Universidad de El Salvador a nivel nacional, registra a 39 estudiantes, al final del ciclo 02 del año 2016, dentro de la categoría de discapacidad sensorial; por lo que siendo esta la última información estadística liberada por la DNES se toma como referencia numérica de la población general para determinar la muestra poblacional en esta investigación.

3.1.3.2. Tipo de muestreo

Muestreo no probabilístico: Según³ “Una muestra no probabilística es adecuada pues se trata de un estudio con un diseño de investigación exploratorio, es decir, no es concluyente, sino su objetivo es documentar ciertas

¹MINED, *Resultados de la Información Estadística de Instituciones de Educación Superior 2016*.

²Sampieri, *Metodología de la investigación*; ed, pág. 89.

³Sampieri, *Metodología de la investigación*; ed, págs. 207-237.

experiencias”

3.1.3.3. Muestra

Las muestras no probabilísticas, llamadas también “Muestras dirigidas”¹ suponen un procedimiento de selección informal y un poco arbitrario. Aun así estas se utilizan en muchas investigaciones y a partir de estas se hacen inferencias sobre la población.

Para esta investigación se utilizó una muestra de sujetos-tipos, donde el objetivo es la riqueza, profundidad y calidad de la información y no la cantidad ni estandarización².

Debido a la población tan reducida con la que se cuenta, se utilizará como referente numérico para los instrumento la cantidad de la población en su totalidad.

3.1.4. Sistema de hipótesis

3.1.4.1. Hipótesis general

- **Hi:** Un asistente por computadora para lectura de textos físicos y digitales facilita el acceso a información a personas con discapacidad visual en el ámbito educativo.
- **Ho:** Un asistente por computadora para lectura de textos físicos y digitales no facilita el acceso a información por personas con discapacidad visual en el ámbito educativo.

3.1.4.2. Hipótesis específicas

- **H1:** Las técnicas de visión por computadora son un medio que facilita la lectura de textos físicos y digitales a personas con discapacidad visual.
- **Ho:** Las técnica de visión por computadora no son un medio que facilita la lectura de textos físicos y digitales a personas con discapacidad visual.
- **H2:** El diseño de una interfaz de comandos por voz es necesaria para mantener un ambiente controlado en la interacción entre el ordenador y una persona con discapacidad visual.
- **Ho:** El diseño de una interfaz de comandos por voz no es necesaria para mantener un ambiente controlado

¹Sampieri, *Metodología de la investigación; ed.* pág. 231.

²Sampieri, *Metodología de la investigación; ed.* pág. 232.

en la interacción entre el ordenador y una persona con discapacidad visual.

- **H3:** El aprendizaje asistido por computadora genera una contribución directa al mejoramiento de la calidad de vida de una persona con discapacidad visual.
- **H0:** El aprendizaje asistido por computadora no plantea una contribución al mejoramiento de la calidad de vida de una persona con discapacidad visual.

3.1.5. Operacionalización de hipótesis

| Objetivo general | Hipótesis general | Hipótesis específica | Variables | Indicadores |
|--|---|---|---|---|
| Facilitar a personas con discapacidad visual el acceso a la información mediante un prototipo asistido por computadora, que facilite la lectura de textos físicos y digitales. | <p>H1: Un asistente por computadora para lectura de textos físicos y digitales facilita el acceso a información por personas con discapacidad visual en el ámbito educativo.</p> | <p>H11: Las técnicas de visión por computadora son un medio que facilita la lectura de textos físicos y digitales a personas con discapacidad visual.</p> | <p>Vi: Técnicas de visión por computadora.</p> <p>Vd: Lectura de textos físicos y digitales a personas con discapacidad visual.</p> | <p>Vi:</p> <ul style="list-style-type: none"> ▪ Reconocimiento de caracteres por computadora. <p>Vd:</p> <ul style="list-style-type: none"> ▪ Libros, folletos, revistas de papel. ▪ Documentos digitales (.pdf). ▪ Búsquedas en línea. |
| | | <p>H2: El diseño de una interfaz de comandos por voz es necesaria para mantener un ambiente controlado en la interacción entre el ordenador y una persona con discapacidad visual.</p> | <p>Vi: Interfaz de comandos por voz.</p> <p>Vd: Control de la interacción entre el ordenador y una persona con discapacidad visual.</p> | <p>Vi:</p> <ul style="list-style-type: none"> ▪ Conversión de voz a texto (STT). ▪ Conversión de Texto a voz (TTS). <p>Vd:</p> <ul style="list-style-type: none"> ▪ Comandos de uso predeterminados. |
| | | <p>H3: El aprendizaje asistido por computadora genera una contribución directa al mejoramiento de la calidad de vida de una persona con discapacidad visual.</p> | <p>Vi: Aprendizaje asistido por computadora.</p> <p>Vd: Mejoramiento en la calidad de vida de personas con discapacidad visual</p> | <p>Vi:</p> <ul style="list-style-type: none"> ▪ Uso de recursos académicos en la FMO-UES. <p>Vd:</p> <ul style="list-style-type: none"> ▪ Adquisición de un título universitario. ▪ Posibilidad de tener un empleo bien remunerado. |

Tabla 3.1: Operacionalización de hipótesis

3.1.6. Proceso experimental

Se define la experimentación como¹: Estudios explicativos, ya que analizan las relaciones entre una o más variables independientes y una o varias dependientes y los efectos causales de las primeras sobre las segundas, asimismo, abarcan correlaciones.

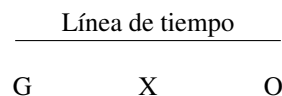
Para esta investigación se realizó un proceso experimental de dos etapas, donde se utilizarán dos diseños experimentales, diferentes para cada una. Según Sampieri², se les denomina preexperimento y experimento “verdadero”, para los cuales se utilizó la simbología sugerida por Campbell y Stanley³:

- **R** : Asignación al azar o aleatorización. Cuando aparece significa que los sujetos han sido asignados a un grupo de manera aleatoria .
- **G** : Grupo de sujetos, (G1, grupo 1; G2 grupo 2; etc.).
- **X** : Tratamiento, estímulo o condición experimental (presencia de algún nivel de variable independiente).
- **O** : Una medición a los sujetos de un grupo (una prueba, cuestionario, observación, tarea, etc.)
- **-** : Ausencia de estímulo. Indica que se trata de un grupo de control.

Cabe mencionar que la secuencia horizontal indica tiempos distintos y que cuando, en dos grupos, dos símbolos aparecen alineados verticalmente, esto señala que tienen lugar en el mismo momento del experimento.

3.1.6.1. Preexperimento

Se llaman así porque su grado de control es mínimo⁴, consiste en administrar un estímulo o tratamiento (una película, un método educativo, un comercial televisivo, etc.) a un grupo, y después aplicar una medición en una o más variables para observar cual es el nivel del grupo en estas variables. Este diseño se puede diagramar así:



Sampieri dice⁵: “Recomendamos que su uso debe restringirse a que sirvan como ensayo de otros experimentos con mayor control (experimento verdadero), y si se tiene dudas sobre el estímulo o la manera de administrar las mediciones”.

¹Sampieri, *Metodología de la investigación*; ed, pág. 172.

²Sampieri, *Metodología de la investigación*; ed, pág. 137.

³Sampieri, *Metodología de la investigación*; ed, pág. 137.

⁴Sampieri, *Metodología de la investigación*; ed, pág. 139.

⁵Sampieri, *Metodología de la investigación*; ed, pág. 140.

Asimismo, los diseños preexperimentales pueden servir como estudios exploratorios. Son útiles como un primer acercamiento con el problema de investigación en la realidad, pero no como el único y definitivo acercamiento con dicho problema.

Para esta investigación, por su carácter exploratorio, se realizó un preexperimento, donde se estableció un primer contacto con la población general, no solo la población objetivo¹, sino, toda la población que interactuará con el prototipo, por lo que se definen tres grupos para la preexperimentación (G1, G2, G3), G1 es la Jefatura de la Unidad Bibliotecaria de la FMO-UES, G2 es el personal de atención a usuarios en el Departamento de Servicio al Público de la Unidad Bibliotecaria de la FMO-UES y G3 es la población estudiantil en las inmediaciones de la Unidad Bibliotecaria de la FMO-UES.

Se pretende establecer las condiciones básicas de aceptación y apoyo que cada grupo presenta ante el prototipo y su impacto en estudiantes con discapacidad visual, cada uno, enfocado a su área de relevancia y su nivel de interacción con respecto al prototipo. Se presentó a cada grupo un documento que indique el propósito del prototipo, su forma de uso y su diseño (X), luego se pasó el instrumento de medición, adecuado a su grupo correspondiente (O1, O2, O3).

Por lo que el diseño preexperimental queda de la siguiente manera:

| | | |
|-----|---|----|
| G1 | X | O1 |
| G2 | X | O2 |
| RG3 | X | O3 |

3.1.6.2. Experimento verdadero

Este diseño incluye dos grupos, uno recibe el tratamiento experimental y el otro no (grupo de control). Es decir, la manipulación de la variable independiente alcanza 2 niveles: presencia y ausencia. Los sujetos son asignados a los grupos de manera aleatoria, después que concluye el periodo experimental, a ambos grupos se les administro una medición sobre la variable dependiente de estudio. El diseño puede diagramarse de la siguiente manera²:

| Línea de tiempo | | |
|-----------------|---|----|
| RG1 | X | O1 |
| RG2 | - | O |

En este diseño³, la única diferencia entre los grupos debe ser la presencia o ausencia de la variable

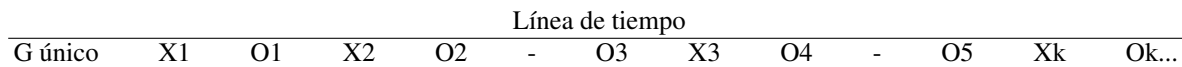
¹Véase Sección 3.1.3 Población y muestra, pág. 96

²Sampieri, *Metodología de la investigación*; ed, pág. 141.

³Sampieri, *Metodología de la investigación*; ed, pág. 142.

independiente. Estos son inicialmente equivalentes y para asegurar que durante el experimento continúen siendo equivalentes, el experimentador debe asegurar que no ocurra algo que afecte sólo a un grupo. a

En ocasiones donde se cuenta con un grupo reducido de sujetos para la experimentación se puede llevar a cabo un tratamiento múltiple en un sólo grupo. No hay asignación al azar puesto que se tiene un único grupo. La equivalencia se obtiene puesto que no hay nada más parecido a un grupo que este mismo, el mismo grupo hace las veces de “grupo experimental” y “grupo de control”. Este se podría diagramar así:



Para esta investigación las variables principales a investigar fueron:

- El uso de un prototipo de asistente para lectura de textos físicos y digitales (Vi).
- El acceso a información por personas con discapacidad visual en el ámbito educativo (Vd).

Por lo que el experimento se centrará en la variación producida sobre Vd por la ausencia o presencia de Vi; debido a las dimensiones de la población, así como el tipo de muestreo utilizado¹, el modelo experimental será uno de tratamiento múltiple sobre un solo grupo, es decir, se harán 3 pruebas iguales (O1, O2, O3) a un solo grupo (G), cada prueba tendrá un estímulo o escenario diferente (X1, X2, X3) y se hará una comparación en las respuestas de las pruebas.

Los escenarios (X1, X2, X3) que se presentarán al grupo G será la realización de 3 actividades que son las áreas de uso del prototipo, búsqueda información en internet, en un documento digital y un documento físico sobre un tema en particular, cada escenario tendrá ciertas variaciones que sirvan para diferenciarlos, así como darles el enfoque deseado. X1 será el escenario para la prueba de control (X1 = -), es decir, para determinar el nivel de dificultad que las actividades en los escenarios pueden representar para una persona que no sufre con discapacidad visual y cuenta con los medios para realizarlas.

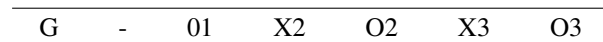
X2 y X3 serán los escenarios cuando se pretende realizar las actividades sin tener la capacidad de ver, será la ausencia o presencia de Vi, según sea el caso, lo que marque la diferencia entre ambos escenarios y permita determinar el tipo y nivel de correlación que Vd y Vi tienen entre sí. La secuencia experimental fue la siguiente:

1. El grupo realizó X1.
2. Se realizó una prueba (O1) para medir el grado de dificultad que X1 representó para ellos.

¹Véase Sección 3.1.3 Población y muestra, pág. 96

3. El grupo realizó X2, que es la realización de las actividades sin el uso del prototipo.
4. Se realizó una prueba (O2) para medir el grado de dificultad que X1 representó para ellos.
5. El grupo realizó X3, que es la realización de las actividades con el uso del prototipo.
6. Se realizó una prueba (O3) para medir el grado de dificultad que X1 representó para ellos.

Su diagrama sería:



3.1.6.3. Recopilación bibliográfica

Se hizo una investigación documental que fundamente una base teórica para el proceso de investigación que se desarrollará¹, se buscó extraer experiencias documentales de proyectos o investigaciones que puedan tener algún tipo de relación con el tema, por el tipo de investigación que se plantea, es de esperar que la información recolectada tenga un bajo margen de similitud con el tema de investigación por variantes demográficas, sociales, culturales, etc.

3.1.6.4. Recolección de datos

Una vez seleccionado el diseño experimental y la población, se definió el Método de Recolección de Datos pertinentes relacionados con las variables de la investigación².

Los instrumentos utilizados para la recolección de datos fue:

Encuesta: Este método de recolección consta de una serie de ítems relacionados con las variables a investigar, a partir de los indicadores de estas; según Sampieri³: “Cualquier investigación seria, reporta la confiabilidad y validez de su instrumento de medición [...]”; una encuesta es fácilmente valorable en su fiabilidad por lo que resulta idónea, se utilizará un total de diez ítems (preguntas) por encuesta, para que cumpla con el criterio de validación de mitades partidas⁴ que indica que al dividir en dos grupos las respuestas dadas en la encuesta, los valores medios se mantendrán iguales para cada mitad.

Se aplicaron cuatro encuestas en total, tres preexperimentales y una experimental, todas son preguntas cerradas, cada encuesta preexperimental se enfoca en presentar las variables investigadas según sea el grupo al que se dirijan, la encuesta experimental tiene una variación cuantitativa, es decir, para cada pregunta se establece una escala

¹ Sampieri, *Metodología de la investigación*; ed, pág. 42.

² Sampieri, *Metodología de la investigación*; ed, pág. 242.

³ Sampieri, *Metodología de la investigación*; ed, pág. 252.

⁴ Sampieri, *Metodología de la investigación*; ed, pág. 250.

de valores ascendente¹, del uno al cinco, para hacer valoraciones dependiendo del criterio presentado.

3.1.6.5. Técnicas para análisis de datos

3.1.6.5.1. Procedimiento para autorización de instrumentos de recolección de datos

Para poder utilizar los instrumentos definidos previamente se necesita la autorización de estos, para lo cual se establecen los pasos siguientes:

- Elaborar las encuestas para la Unidad Bibliotecaria de la Facultad Multidisciplinaria Oriental, para la jefatura, el personal de atención al público y los estudiantes.
- Presentar una petición escrita a la Jefatura de la Unidad Bibliotecaria de la Facultad Multidisciplinaria Oriental, adjuntando una copia de las encuestas para su respectiva autorización.

3.1.6.5.2. Procedimiento de recolección de datos

Las encuestas fueron escritas y presentadas de manera presencial², en la Unidad Bibliotecaria de la Facultad Multidisciplinaria Oriental, en sus lugares de trabajo respectivos, a los estudiantes de la Facultad que se encontraron en las instalaciones de esta, a los trabajadores de la Unidad y a la jefatura.

Se siguieron los siguientes pasos³:

- Se explico a los encuestados el propósito de la encuesta y se solicito su tiempo para completarla.
- Se dieron las indicaciones para completar las encuestas de forma clara y simple.
- Se solvento cualquier duda que surja antes, durante y después de la encuesta relacionada al prototipo y su posible labor con este.

3.1.6.5.3. Procedimiento de procesamiento e interpretación de datos

El procesamiento de datos se dio cuando se tomaron las encuestas pasadas en la Unidad Bibliotecaria de la Facultad Multidisciplinaria Oriental y se recopilo la información recolectada, facilitando así su análisis e interpretación⁴

¹Sampieri, *Metodología de la investigación*; ed, pág. 255.

²Sampieri, *Metodología de la investigación*; ed, pág. 299.

³Sampieri, *Metodología de la investigación*; ed, pág. 294.

⁴Sampieri, *Metodología de la investigación*; ed, pág. 349.

Según Sampieri¹:“El nivel de significancia es un valor que fija el investigador *a priori*. De certeza respecto a no equivocarse [...]”, un nivel de significancia del 0.05 indica que el investigador tiene un 95 % de seguridad para generalizar sus resultados sin equivocarse.

Para las encuestas preexperimentales se siguieron los siguientes pasos:

- Se hizo la tabulación de las encuestas realizadas a los diferentes grupos respectivamente.
- Se presentaron cada una de las preguntas de las encuestas realizadas.
- Se formulo un objetivo para cada pregunta realizada en las encuestas.
- Se hizo una tabla de distribución de frecuencias² donde se reflejara el conteo de las diferentes respuestas.
- Se hizo un cuadro general de la encuesta donde se valorara el error porcentual en las afirmaciones hipotéticas contra el nivel de significancia permitido para la investigación.
- Se realizaron conclusiones generales para el preexperimento.

Para la encuesta experimental se siguieron los siguientes pasos:

- Se hizo la tabulación de los resultados obtenidos en cada escenario presentado.
- Se presento cada una de las preguntas de las encuestas realizadas.
- Se hizo una tabla comparativa de los tres escenarios, así como una gráfica de variación de los resultados en cada escenario.
- Se hizo una valoración de la hipótesis general contra los resultados del experimento para la investigación, presentando conclusiones finales de la investigación.

3.2. Análisis e interpretación de resultados

3.2.1. Tabulación de encuestas preexperimentales

3.2.1.1. Encuesta a la jefatura de la Unidad Bibliotecaria de la FMO-UES

Personas encuestadas = 2

1. **Desde una perspectiva Administrativa, ¿Considera sostenible la implementación del dispositivo dentro de las instalaciones de la Unidad Bibliotecaria?**

Objetivo: Determinar si la Unidad Bibliotecaria de la FMO-UES es capaz de darle mantenimiento al

¹Sampieri, *Metodología de la investigación*; ed, pág. 350.

²Sampieri, *Metodología de la investigación*; ed, pág. 352.

prototipo así como si cuenta con el espacio físico necesario.

- Si, se cuenta con espacio físico y condiciones apropiadas para poner a disposición de los usuarios con necesidades de este tipo.

Conclusión: La Unidad Bibliotecaria de la FMO-UES cuenta con el Recurso Humano¹ y el espacio necesario para implementar el prototipo y darle el mantenimiento y cuidado necesario.

2. **¿La Unidad Bibliotecaria tomaría a bien gestionar la asignación de los recursos necesarios para la implementación del dispositivo?**

Objetivo: Conocer la disposición que tiene la Unidad Bibliotecaria de la FMO-UES para realizar las gestiones administrativas necesarias para implementar el prototipo.

- Si, La Unidad puede hacerse cargo de las gestiones con la Facultad.

Conclusión: La Unidad Bibliotecaria de la FMO-UES está dispuesta a llevar a cabo todas las gestiones administrativas necesarias para implementar el prototipo dentro de sus instalaciones.

3. **¿La Unidad Bibliotecaria cuenta con la capacidad de dedicar y adecuar un espacio físico para la implementación del dispositivo?**

Objetivo: Conocer si la Unidad Bibliotecaria de la FMO-UES cuenta con un espacio físico que puedan adecuar bajo los criterios establecidos en el proyecto para el uso del prototipo.

- Si, En el área de Colección General del departamento de Servicios al Público.

Conclusión: La Unidad Bibliotecaria de la FMO-UES cuenta con un espacio físico donde se puede adecuar bajo los criterios establecidos en la investigación para la implementación del prototipo.

4. **¿La Unidad Bibliotecaria cuenta con personal dedicado al área de restauración y mantenimiento de libros?**

Objetivo: Establecer si la Unidad Bibliotecaria de la FMO-UES cuenta con personal capacitado en el área de mantenimiento de libros.

- Si, la Unidad cuenta con personal capacitado.

Conclusión: El personal de la Unidad Bibliotecaria se encuentra capacitado para el mantenimiento físico de libros y restauración de estos.

¹Véase Sección 3.2.3 Discusión de Resultados, Cualitativos., pág. 119

5. **Para un mejor uso del dispositivo es necesario que los libros que se utilicen tengan un formato determinado, por ejemplo, hojas individuales tamaño carta, texto en color negro justificado, una sola columna, borde no menor a un centímetro en ancho y alto, etc... ¿Es posible adecuar libros a un formato similar al presentado?**

Objetivo: Conocer los alcances en la preparación y adecuación de libros que la Unidad Bibliotecaria posee.

- Si, la Unidad es capaz de adecuar libros a formatos similares a los presentados.

Conclusión: La Unidad cuenta con los recursos¹ para adecuar los libros al formato que se necesita para un uso óptimo del prototipo.

6. **¿Se cuenta con personal al que se le pueda asignar la asistencia de estudiantes en el uso del dispositivo?**

Objetivo: Determinar si la Unidad Bibliotecaria de la FMO-UES puede asignar personal a su cargo para que asista a los usuarios del prototipo.

- Si, el personal de la Unidad está disponible y dispuesto a brindar la ayuda necesaria a los estudiantes con esta discapacidad.

Conclusión: Existe personal disponible para asistir a los usuarios durante el tiempo que utilicen el prototipo y brindar la ayuda necesaria.

7. **¿Es posible apartar tiempo del personal en la Unidad Bibliotecaria para su capacitación en el uso del dispositivo?**

Objetivo: Descubrir si el personal de la Unidad Bibliotecaria de la FMO-UES puede ser capacitado para el uso, mantenimiento y asistencia del prototipo.

- Si.

Conclusión: La Unidad Bibliotecaria de la FMO-UES puede apartar tiempo del personal para que pueda ser capacitado en el uso y asistencia del prototipo así como su mantenimiento.

8. **¿Considera de utilidad, para la comunidad universitaria, la implementación del dispositivo en las instalaciones de la Unidad Bibliotecaria?**

Objetivo: Conocer la postura de la Unidad Bibliotecaria de la FMO-UES respecto al impacto que podría

¹Véase Sección 3.2.3 Discusión de Resultados, Cualitativos., pág. 119

tener en la comunidad universitaria la implementación del prototipo en sus instalaciones.

- Si, sería de mucha ayuda para la comunidad universitaria.

Conclusión: El prototipo es percibido como útil para los estudiantes de la universidad y su instalación en la Unidad Bibliotecaria se considera buena idea por parte de la administración.

9. **¿Según su opinión, cree que la Universidad Nacional debería implementar soluciones para facilitar el estudio de estudiantes con discapacidad?**

Objetivo: Conocer la postura de la jefatura de la Unidad Bibliotecaria de la FMO-UES respecto a la importancia de implementar programas que fomenten el ingreso de estudiantes con discapacidad.

- Sí, es de mucha importancia.

Conclusión: La Unidad Bibliotecaria de la FMO-UES considera importante la implementación de programas inclusivos a personas con discapacidad que buscan superarse profesionalmente y sienten que son alejados de esas posibilidades por la escasez de métodos o recursos disponibles para ellos.

10. **¿La Unidad Bibliotecaria cuenta en la actualidad con un plan de asistencia o apoyo que facilite el uso del material de estudio a personas con este tipo de discapacidad?**

Objetivo: Determinar si la Unidad Bibliotecaria de la FMO-UES cuenta con programas de inclusión a personas con discapacidad visual.

- No.

Conclusión: La Unidad Bibliotecaria no cuenta con planes de asistencia a personas con discapacidad visual, por lo que la implementación del prototipo puede ser de mucha ayuda para los estudiantes.

A continuación se presenta una tabla que muestra la frecuencia con la que se seleccionó cada respuesta posible a las preguntas de la encuesta a la jefatura de la Unidad Bibliotecaria de la FMO-UES.

| Tabla de distribución de frecuencias | | | | | | | | | | |
|--------------------------------------|---|---|---|---|---|---|---|---|---|----|
| Pregunta | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Si | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 |
| No | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Abstenciones | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 |

Tabla 3.2: Tabla de distribución de frecuencias - encuesta a la jefatura de la biblioteca

3.2.1.2. Encuesta a empleados de la Unidad Bibliotecaria de la FMO-UES

Personas encuestadas = 5

1. **¿En alguna ocasión ha atendido a estudiantes que sufran discapacidad visual, en el contexto de su trabajo?**

Objetivo: Conocer si algún estudiante con discapacidad ha hecho uso del material bibliotecario de la Unidad Bibliotecaria de la FMO-UES.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 4 | 1 | 0 | 5 |

Tabla 3.3: Respuestas pregunta N° 1

Conclusión: La mayoría de los trabajadores que atienden al público en la Unidad Bibliotecaria de la FMO-UES han tenido que ayudar a estudiantes con discapacidad a encontrar el material bibliográfico que necesitan, por lo que se puede presumir que el prototipo puede ser de utilidad para estos estudiantes.

2. **¿Cómo considera la atención de la Facultad a los estudiantes con discapacidad respecto a las facilidades de acceso a material de estudio que les brinda?**

Objetivo: Determinar la postura de los empleados de atención al público de la Unidad Bibliotecaria de la FMO-UES respecto a la atención que la FMO-UES les dan a los estudiantes discapacitados para el acceso a material de estudio.

| Insuficiente | Adecuada | Óptima | Excelente | Abstención | Total |
|--------------|----------|--------|-----------|------------|-------|
| 4 | 0 | 0 | 1 | 0 | 5 |

Tabla 3.4: Respuestas pregunta N° 2

Conclusión: Los empleados del área de atención al público en la Unidad Bibliotecaria de la FMO-UES en su mayoría consideran que la atención a las necesidades de acceso a material de estudio que la FMO-UES les da a los estudiantes con discapacidad necesita mejorar.

3. **¿Considera el uso de la tecnología como una herramienta útil en el proceso de aprendizaje?**

Objetivo: Establecer la opinión de los empleados de atención al público en la Unidad Bibliotecaria de la FMO-UES respecto al aprendizaje por computadora.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 5 | 0 | 0 | 5 |

Tabla 3.5: Respuestas pregunta N° 3

Conclusión: Los empleados de atención al público en la Unidad Bibliotecaria de la FMO-UES consideran útil el uso de tecnología para el aprendizaje, por lo que el concepto del prototipo no es visto como una mala idea.

4. **¿Ha tenido experiencia con el uso de ordenadores u otra herramienta tecnológica en el contexto de su trabajo?**

Objetivo: Determinar la experiencia que el personal de atención al público en la Unidad Bibliotecaria de la FMO-UES tiene en el uso de ordenadores a nivel laboral.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 4 | 1 | 0 | 5 |

Tabla 3.6: Respuestas pregunta N° 4

Conclusión: La mayoría de los empleados del área de atención al público en la Unidad Bibliotecaria de la FMO-UES tiene conocimientos en el uso de computadores por lo que no será un problema el aprendizaje para asistir a los estudiantes en el uso del prototipo.

5. **¿Considera que el dispositivo podría ser de utilidad para que estudiantes que sufren discapacidad visual utilicen las instalaciones de la Unidad Bibliotecaria para estudiar?**

Objetivo: Conocer la opinión de los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES respecto a si la implementación del prototipo fomentaría el uso más recurrente del material bibliotecario disponible por parte de los estudiantes con discapacidad visual.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 5 | 0 | 0 | 5 |

Tabla 3.7: Respuestas pregunta N° 5

Conclusión: Los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES concuerdan con la premisa que la implementación del prototipo fomentaría el uso de las instalaciones y material bibliotecario de la Unidad por parte de estudiantes con discapacidad visual.

6. **¿Considera que existe un espacio físico dentro de la Unidad Bibliotecaria que pueda ser dedicado**

para el uso del dispositivo? ¿Dónde sería?

Objetivo: Indagar la postura del personal en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES respecto a la capacidad en el espacio físico que la Unidad tiene para implementar el prototipo con sus requerimientos.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 3 | 2 | 0 | 5 |

Tabla 3.8: Respuestas pregunta N° 6

■ Lugares sugeridos:

Sala de estudio individual

Conclusión: Los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES en su mayoría consideran que la Sala de Estudios Individual cuenta con el espacio necesario para implementar el prototipo sin problemas.

7. **¿Estaría a favor con la implementación del dispositivo dentro de las instalaciones de la Unidad**

Bibliotecaria?

Objetivo: Saber la postura de los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES respecto a la implementación del prototipo en las instalaciones de la Unidad.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 5 | 0 | 0 | 5 |

Tabla 3.9: Respuestas pregunta N° 7

Conclusión: Todos los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están a favor de la implementación del prototipo en su lugar de trabajo.

8. **¿Participaría en los procesos de capacitación en el uso del dispositivo?**

Objetivo: Determinar si los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están dispuestos a realizar un curso de capacitación que los prepare para la asistencia de estudiantes con discapacidad visual en el uso del prototipo.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 5 | 0 | 0 | 5 |

Tabla 3.10: Respuestas pregunta N° 8

Conclusión: Todos los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están de acuerdo en realizar un curso de capacitación para la asistencia de estudiantes con discapacidad visual en el uso adecuado del prototipo.

9. **¿Brindaría asistencia en el uso del dispositivo a estudiantes con discapacidad visual si llega a ser requerido?**

Objetivo: Conocer si los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están dispuestos a ayudar a estudiantes con discapacidad visual a usar correctamente el prototipo.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 5 | 0 | 0 | 5 |

Tabla 3.11: Respuestas pregunta N° 9

Conclusión: Todos los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están dispuestos a ayudar a los estudiantes con discapacidad visual en el uso del prototipo, por lo que se puede determinar que apoyan la implementación del prototipo en su totalidad.

10. **¿Estaría de acuerdo en la adecuación de material bibliográfico a un formato que facilite su uso en el dispositivo?**

Objetivo: Saber si los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están dispuestos a adecuar material bibliográfico a los estándares definidos para el uso adecuado del prototipo.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 5 | 0 | 0 | 5 |

Tabla 3.12: Respuestas pregunta N° 10

Conclusión: Los empleados en el área de atención al público de la Unidad Bibliotecaria de la FMO-UES están dispuestos a adecuar los libros que se utilizarán en el prototipo para los estudiantes con discapacidad visual.

A continuación se presenta una tabla que muestra la frecuencia con la que se seleccionó cada respuesta posible a las preguntas de la encuesta a los empleados de la Unidad Bibliotecaria de la FMO-UES, la pregunta dos cuenta con un rango de respuestas más amplio por lo que su totalización será aparte.

| Tabla de distribución de frecuencias | | | | | | | | | | |
|--------------------------------------|---|---|---|---|---|---|---|---|---|----|
| Pregunta | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Si | 4 | - | 5 | 4 | 5 | 3 | 5 | 5 | 5 | 5 |
| No | 1 | - | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
| Abstenciones | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 5 | - | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

Tabla 3.13: Tabla de distribución de frecuencias - encuesta a empleados de la biblioteca

| Tabla de distribución de frecuencias | |
|--------------------------------------|---|
| Pregunta | 2 |
| Insuficiente | 4 |
| Adecuada | 0 |
| Óptima | 0 |
| Excelente | 1 |
| Abstenciones | 0 |
| Total | 5 |

Tabla 3.14: Tabla de distribución de frecuencias - resumen - encuesta a empleados de la biblioteca

3.2.1.3. Encuesta a estudiantes de la FMO-UES

Personas encuestadas = 39

1. **¿Considera útil el uso de tecnologías de reconocimiento de caracteres por computadora para la interpretación de textos físicos, con el fin de facilitar la lectura a estudiantes de la FMO-UES?**

Objetivo: Saber si los estudiantes de la FMO-UES consideran que el reconocimiento de caracteres por computadora para textos físicos facilita el estudio a los estudiantes con discapacidad visual de la FMO-UES.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 39 | 0 | 0 | 39 |

Tabla 3.15: Respuestas pregunta N° 1

Conclusión: Todos los estudiantes encuestados consideran que el uso de reconocimiento de caracteres por computadora es de utilidad para que personas con discapacidad visual puedan utilizar material bibliográfico para sus estudios, por lo que se puede suponer que el concepto general del prototipo es aceptado por la población estudiantil.

2. **¿Cree que el material bibliográfico de la FMO-UES como libros y tesis, así como recursos en línea deberían estar al alcance de estudiantes que sufren discapacidad visual?**

Objetivo: Conocer la opinión de los estudiantes de la FMO-UES respecto a si estudiantes con discapacidad

visual deberían tener a su alcance el material bibliográfico que la Facultad pone a disposición de la comunidad universitaria en general.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 39 | 0 | 0 | 39 |

Tabla 3.16: Respuestas pregunta N° 2

Conclusión: Los estudiantes de la FMO-UES opinan que toda la comunidad universitaria, incluyendo a los estudiantes con discapacidad visual deben tener igual acceso al material bibliográfico que la Facultad brinda.

3. **Según su opinión, ¿Estudiantes con discapacidad visual deberían tener acceso a las facilidades para estudio e investigación en línea que brinda la Sala de Internet de la Unidad Bibliotecaria a la población universitaria en general?**

Objetivo: Determinar la postura de los estudiantes de la FMO-UES respecto a si estudiantes con discapacidad visual deberían tener acceso a las facilidades brindadas al resto de la comunidad universitaria por parte de la Sala de Internet de la Unidad Bibliotecaria en materia de acceso en línea de contenido para estudio e investigación.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 39 | 0 | 0 | 39 |

Tabla 3.17: Respuestas pregunta N° 3

Conclusión: Todos los estudiantes encuestados están de acuerdo en que la comunidad universitaria en general, incluyendo a personas con discapacidad visual, deben tener acceso a las mismas facilidades de estudio brindadas por la Sala de Internet.

4. **¿Cree que una interfaz por comandos de voz sería de utilidad para la interacción entre un usuario con discapacidad visual y el prototipo propuesto?**

Objetivo: Sondar la postura de los estudiante de la FMO-UES frente al uso de comandos de voz para una óptima interacción con el prototipo.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 39 | 0 | 0 | 39 |

Tabla 3.18: Respuestas pregunta N° 4

Conclusión: El uso de comandos por voz es visto como de utilidad para interactuar con el prototipo si se es

una persona con discapacidad visual, ya que facilita la comunicación entre el usuario y la máquina.

5. ¿Considera que el uso de comandos predeterminados o palabras clave sea de utilidad para simplificar la interacción con el prototipo?

Objetivo: Conocer si los estudiantes de la FMO-UES consideran los comandos predeterminados o palabras clave como útiles para facilitar la interacción con el prototipo.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 36 | 3 | 0 | 39 |

Tabla 3.19: Respuestas pregunta N° 5

Conclusión: Para casi toda la muestra estudiantil utilizada para la encuesta el uso de comandos por voz utilizando palabras clave, ayuda a controlar de manera eficiente la interacción entre el prototipo y el usuario.

6. Según su opinión ¿La presentación en formato de audio de la información procesada por el prototipo, es la más útil para que un estudiante con discapacidad visual pueda comprenderla?

Objetivo: Saber si para los estudiantes de la FMO-UES el presentar los resultados de las tareas dadas al prototipo en formato de audio es la manera más práctica para comunicarlos a un estudiante con discapacidad visual.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 38 | 1 | 0 | 39 |

Tabla 3.20: Respuestas pregunta N° 6

Conclusión: La mayoría de los estudiantes encuestados consideran que el formato de audio es el formato adecuado para transmitir la información procesada, a los estudiantes con discapacidad visual.

7. ¿Considera que el uso del prototipo pueda facilitar la adquisición de un título universitario a una persona con discapacidad visual que estudie en la FMO-UES?

Objetivo: Establecer la postura de los estudiantes de la FMO-UES respecto a si el uso del prototipo por parte de una persona con discapacidad visual le facilita sus estudios universitarios permitiéndole obtener un título profesional.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 36 | 3 | 0 | 39 |

Tabla 3.21: Respuestas pregunta N° 7

Conclusión: La mayoría de los estudiantes encuestados consideran que el prototipo puede ser de utilidad a personas con discapacidad visual que pretenden finalizar una carrera universitaria.

8. **Según su punto de vista ¿Se puede relacionar el uso del prototipo y las oportunidades que representa a una persona con discapacidad visual con la capacidad de optar por un trabajo bien remunerado?**

Objetivo: Indagar en el punto de vista de los estudiantes de la FMO-UES en cuanto a la relación que se puede hacer entre el uso del prototipo, el cómo mejora la calidad de estudio de una persona con discapacidad visual y su capacidad de acceder a un nivel profesional más competitivo y mejor remunerado.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 36 | 3 | 0 | 39 |

Tabla 3.22: Respuestas pregunta N° 8

Conclusión: Los estudiantes de la FMO-UES observan una relación entre el uso del prototipo por personas con discapacidad visual y su capacidad de optar por una mejor gama de trabajos.

9. **¿Recomendaría la implementación del prototipo dentro de las instalaciones de la Unidad Bibliotecaria de la FMO-UES?**

Objetivo: Saber si los estudiantes de la FMO-UES recomendarían la implementación del prototipo en las instalaciones de la Unidad Bibliotecaria de la Facultad.

| Sí | No | Abstención | Total |
|----|----|------------|-------|
| 39 | 0 | 0 | 39 |

Tabla 3.23: Respuestas pregunta N° 9

Conclusión: Todos los estudiantes encuestados apoyan la implementación del prototipo en las instalaciones de la Unidad Bibliotecaria de la FMO-UES, por lo que se puede considerar que el prototipo es visto como de utilidad para la comunidad universitaria.

10. **¿Animaría a una persona con discapacidad visual que inicia una carrera universitaria el uso del prototipo para su asistencia en el proceso de estudio?**

Objetivo: Conocer si los estudiantes de la FMO-UES incentivarían a personas con discapacidad visual a que hagan uso del prototipo para sus estudios en la Facultad.

| Instrumento aplicado al personal de la Unidad Bibliotecaria de la FMO-UES (población=5) | | | | | | | | | | |
|---|----------------|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Pregunta | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Respuesta esperada | Si(5) | Insuficiente(5) | Si(5) | Si(5) | Si(5) | Si(5) | Si(5) | No(5) | Si(5) | Si(5) |
| Respuesta obtenidas | Si(2) No(0) | Insuficiente(4) Adecuado(0) Óptima(0) Excelente(1) | Si(5) No(0) | Si(4) No(1) | Si(5) No(0) | Si(3) No(2) | Si(5) No(0) | Si(5) No(0) | Si(5) No(0) | Si(5) No(0) |
| % de acierto | 100 | 80 | 100 | 80 | 100 | 60 | 100 | 100 | 100 | 100 |

$$Va_2 = \frac{\sum(\%deacierto)}{10(100)} = \frac{100 + 80 + 100 + 80 + 10 + 60 + 100 + 100 + 100 + 100}{1000} = \frac{920}{1000} = 0,92$$

| Instrumento aplicado a estudiantes de la FMO-UES (población=39) | | | | | | | | | | |
|---|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Pregunta | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Respuesta esperada | Si(39) | Si(39) | Si(39) | Si(39) | Si(39) | Si(39) | Si(39) | Si(39) | Si(39) | Si(39) |
| Respuesta obtenidas | Si(39) No(0) | Si(39) No(0) | Si(39) No(0) | Si(39) No(0) | Si(36) No(3) | Si(38) No(1) | Si(36) No(3) | Si(36) No(3) | Si(39) No(0) | Si(38) No(1) |
| % de acierto | 100 | 100 | 100 | 100 | 92 | 97 | 92 | 92 | 100 | 97 |

$$Va_3 = \frac{\sum(\%deacierto)}{10(100)} = \frac{100 + 100 + 100 + 100 + 92 + 97 + 92 + 92 + 100 + 97}{1000} = \frac{970}{1000} = 0,970$$

Al obtener el valor de acierto de todos los instrumentos se saca el valor de acierto medio, para así determinar el valor de acierto del preexperimento.

$$\bar{V}_a = \frac{Va_1 + Va_2 + Va_3}{3} = \frac{1 + 0,92 + 0,97}{3} = \frac{2,89}{3} = 0,9633$$

Para obtener el nivel de significancia de un experimento, se hace uso de la siguiente fórmula:

$$Nivel\ de\ significancia = 1 - Va = 1 - 0,9633 = 0,0367$$

Según Sampieri¹ el experimento de una investigación social es representativo si su nivel de significancia es menor o igual a 0.05, es decir, tiene un 95 % de veracidad, y es aceptado como herramienta para realizar

¹Sampieri, *Metodología de la investigación*; ed. pág. 350.

generalizaciones sobre las variables estudiadas y su comportamiento previamente supuesto en las hipótesis.

El nivel de significancia del preexperimento desarrollado es de 0.0367, es decir, cuenta con un 96.33 % de veracidad en sus resultados y puede ser utilizado para realizar conclusiones respecto a las variables de esta investigación.

3.2.3. Discusión de resultados

Cuantitativos: Los resultados cuantitativos son el reflejo de las respuestas cuantificables obtenidas en los tres instrumentos utilizados en el preexperimento.

- EL 100 % de los estudiantes encuestados en la FMO-UES consideran que todo el material de estudio e investigación disponible debe estar al alcance de aquellos que sufren una discapacidad visual.
- El 100 % de los empleados de la Unidad Bibliotecaria de la FMO-UES consideran que utilizar tecnología para estudiar es muy útil en el proceso de aprendizaje.
- Si bien un empleado de la Unidad Bibliotecaria de la FMO-UES considera que la Facultad brinda un excelente acceso al material bibliotecario a personas con discapacidad visual, este representa solo el 20 % de los resultados, por lo que 80 % de los empleados considera insuficiente este acceso a la información, pudiendo generalizar así la opinión de los empleados respecto al tema.
- El 60 % de los empleados de la Unidad Bibliotecaria de la FMO-UES consideran que la Sala de Estudio Individual es el lugar idóneo para implementar el prototipo.

Cualitativos: Los resultados cualitativos reflejan la ampliación y comprensión no cuantificable de las respuestas obtenidas en los instrumentos del preexperimento.

- Los empleados de la Unidad Bibliotecaria de la FMO-UES con los que se cuenta para implementar el prototipo son los cinco empleados en el Área de Atención al Público.
- El mantenimiento de los libros adecuados para usarse en el prototipo será de responsabilidad del Área de Tesarío de la Unidad Bibliotecaria de la FMO-UES.
- El método utilizado para el mantenimiento y la adecuación del material bibliográfico del prototipo se hará de la siguiente manera:

1. Se sacan las hojas individuales del libro.

2. Se limpian y recortan imperfecciones.
 3. Se Sujetan las páginas en una prensa de madera especial.
 4. Se hacen 4 cortes en diagonal de un milímetro de grosor, en el lomo del libro.
 5. Se cosen con hilo de nailon en pares las hendiduras.
 6. Con cola blanca se pega un trozo de tela de manta sobre el lomo.
 7. Se deja secar por 12 horas.
 8. Se recortan restos de tela que sobresalgan del lomo del libro.
 9. Se agrega una hoja de guarda para tapar las costuras a ambos por el frente y el revés del libro.
 10. Se pega la pasta y se prensa todo el libro.
 11. Se deja reposar 12 horas.
 12. Hojas dañadas se utiliza Metil Celulosa y papel japonés para repararlas.
- Es posible adaptar 3 libros de 200 páginas cada 2 días, por lo que en las 47 semanas laborales del año se podrían adecuar 352 libros aproximadamente.

3.2.4. Conclusiones

Según la información obtenida por los instrumentos utilizados en el proceso preexperimental de la investigación, se puede realizar generalizaciones sobre las variables dependiente e independiente de esta, cabe destacar que estas afirmaciones no son concluyentes, es decir, no pueden ser utilizadas para confirmar o refutar los planteamientos presentados en el sistema de hipótesis de la investigación.

La finalidad del preexperimento es establecer un parámetro en la tendencia hacia la cual se inclina el comportamiento de las variables estudiadas.

Los resultados revelados en el proceso preexperimental realizado, permiten concluir las siguientes afirmaciones:

- La tecnología de reconocimiento de caracteres por computadora para la asistencia de personas con discapacidad visual es una opción viable para ser implementada en las instalaciones de la FMO-UES en el área de la Unidad Bibliotecaria.

- La dinámica de interacción entre el usuario y el prototipo propuesto en esta investigación debe ser orientada al uso de comandos de voz por medio de palabras clave, para que el usuario indique las tareas que necesita que el prototipo realice y como canal para transmitir las respuestas del prototipo al usuario, el mejor medio es usar el formato de audio.
- Las actividades de lectura de textos digitales y búsqueda en línea de información es necesario que sean incluidas en las capacidades de asistencia que el prototipo pueda realizar.
- El personal de la Unidad Bibliotecaria de la FMO-UES puede ser capacitado en el mantenimiento del prototipo, así como en su uso y asistencia para la ayuda de estudiante con discapacidad visual que pueda requerirlo, también, se les puede encomendar el cuidado, adecuación y control del material bibliográfico especial que se utilizará.
- La comunidad universitaria de la FMO-UES, consideran que la implementación del prototipo en las instalaciones de la Unidad Bibliotecaria de la Facultad es de utilidad para el desarrollo profesional de estudiantes con discapacidad visual.
- El uso e implementación del prototipo en las instalaciones de la Unidad Bibliotecaria de la FMO-UES está directamente relacionado con el mejoramiento en la calidad de vida que una persona con discapacidad visual pueda tener en el área de desarrollo académico profesional.

4. Desarrollo, Experimentación y Resultados

4.1. Desarrollo del prototipo

4.1.1. Determinación de requerimientos

4.1.1.1. Requerimientos funcionales

“Cuando hablamos de una característica requerida de la cual se sabe que va a ser satisfecha por medio de la adición de un subsistema o bloque de código en el software, entonces se dice que estamos ante un requisito funcional, por cuanto es un requisito que denota una funcionalidad del sistema. Requisito Funcional: característica requerida del sistema que expresa una capacidad de acción del mismo – una funcionalidad; generalmente expresada en una declaración en forma verbal.”²

| REF. | Descripción | Prioridad |
|----------|--|-----------|
| REQ-0001 | Ejecutarse sobre python | Alta |
| REQ-0002 | Ingreso de comandos de voz | Alta |
| REQ-0003 | Procesamiento de comandos | Alta |
| REQ-0004 | Búsqueda en Internet de Información | Alta |
| REQ-0005 | Salida de audio de las respuesta de búsqueda | Alta |
| REQ-0006 | Lectura de imágenes | Baja |
| REQ-0007 | Procesamiento de texto en imágenes | Baja |
| REQ-0008 | Salida de audio del texto de la imagen | Baja |
| REQ-0009 | Lectura de archivos PDF | Baja |

Tabla 4.1: Requerimientos funcionales

4.1.1.2. Requerimientos no funcionales

Requisito no funcional: característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

Se llama requisito no funcional a toda exigencia de cualidades que se imponen al proyecto como exigencias de usar un cierto lenguaje de programación o plataforma tecnológica.³

4.1.2. Estudio de factibilidad

En el estudio de la factibilidad de un proyecto informático es normal cuestionarse sobre la viabilidad del mismo. La factibilidad del software tiene las siguientes dimensiones sólidas:

²Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix. 2008. URL: <https://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/> (visitado 01-07-2018).

³Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix.

- Tecnología: ¿Es factible un proyecto técnicamente? ¿Está dentro del estado actual de la técnica?
- Financiación: ¿Es factible financieramente? ¿Puede realizarse a un coste asumible por la empresa de software y por el cliente?
- Recursos: ¿La organización cuenta con los recursos suficientes para tener éxito?

A continuación se presenta el análisis obtenido del estudio de factibilidad para este proyecto de desarrollo de software.

4.1.2.1. Factibilidad técnica

Lenguaje para desarrollo

Para la elección de la herramienta que se utilizó en el desarrollo web del proyecto, el cual sirvió para manejar los posibles tipos de eventos, se tomaron en cuenta las siguientes herramientas de desarrollo: Java, y Python.

A continuación en el cuadro de “lenguaje de desarrollo”, se listan los criterios utilizados por el grupo para evaluar cuál lenguaje de programación se utilizó para el control de eventos en proyecto.

| Criterios | Ideal | Java | Python |
|---|-------|------|--------|
| Multiplataforma | 5 | 5 | 5 |
| Software Libre | 5 | 3 | 5 |
| Sintaxis cómoda | 5 | 4 | 5 |
| Soporte | 5 | 4 | 5 |
| Objetos y herencia | 5 | 5 | 5 |
| Compatibilidad | 5 | 3 | 4 |
| Conocimiento por parte del equipo del lenguaje de programación. | 5 | 4 | 4 |
| Total | 35 | 28 | 33 |

Tabla 4.2: Lenguaje para desarrollo

Dada la necesidad de una sintaxis clara y concisa en el desarrollo del código necesario para el funcionamiento del software se decidió el uso del estándar detallado en la sección 4.6 estándares de código.

Síntesis de calificaciones

Por la cantidad de criterios evaluados para la elección de la herramienta de desarrollo web, se determinó que para ser considerado una alternativa para este proyecto debe de cumplir arriba de 25 puntos en total. Dado que los resultados favorecen más al lenguaje de programación Python, se recomienda hacer implementación de dicho lenguaje.

Sistema Operativo para Raspberry PI

Para que el proyecto se pueda implementar necesitamos un sistema operativo como base para la Raspberry PI, por lo cual se debe de escoger el más idóneo, los sistemas operativos que se han tomado en cuenta son: Ubuntu Mate, Raspbian Stretch.

| Criterios | Ideal | Ubuntu Mate | Raspbian Buster |
|----------------------------|-------|-------------|-----------------|
| Asequibilidad | 5 | 5 | 5 |
| Compatibilidad | 5 | 5 | 5 |
| Estabilidad | 5 | 3 | 4 |
| Disponibilidad de Software | 5 | 4 | 4 |
| Facilidad de Uso | 5 | 4 | 3 |
| Seguridad | 5 | 5 | 5 |
| Total | 30 | 26 | 26 |

Tabla 4.3: Sistema operativo para Raspberry PI

Síntesis de calificaciones

Como resultado de la evaluación se seleccionará aquel sistema que sume como mínimo 21 puntos. Ya que las dos opciones suman más de lo establecido, la mejor opción que según como grupo consideramos conveniente es Raspbian Stretch debido a la gran cantidad de soporte que tiene este sistema operativo, en comparación de Ubuntu Mate.

4.1.2.2. Factibilidad operativa

En la factibilidad Operativa se hace énfasis en la capacidad de operación que posee el lugar dónde se llevará a cabo el proyecto, en este caso, la Facultad Multidisciplinaria Oriental.

Anteriormente en la factibilidad técnica se mencionó las herramientas que se deben de utilizar para el desarrollo del proyecto, las cuales requieren de conocimientos técnicos para poder darles un uso adecuado y sacar el mayor beneficio posible, por lo tanto se necesita de personal capacitado para dicha labor. Para poder implementar el proyecto se necesita de dos personas, una que sería el programador-analista que daría mantenimiento al sistema, y otra persona que debe de ser la encargada de administrarlo para las personas con problemas visuales, personal que la Unidad Bibliotecaria de la Facultad debe tener a disposición. En el siguiente cuadro se muestran las características que deben de cumplir éstas dos personas:

| Perfil del programador-analista | |
|---------------------------------|---|
| Funciones Básicas | <ul style="list-style-type: none"> ■ Mantener, configurar, monitorear, documentar y asegurar el correcto funcionamiento del asistente en su fase lógica. ■ Mantenimiento certero y correcto funcionamiento del sistema electrónico en el asistente. ■ Asesoramiento del cuerpo administrativo sobre el mantenimiento, uso y cuidado del asistente. |
| Formación Académica | <ul style="list-style-type: none"> ■ Graduado en Ingeniería de Sistemas Informáticos, Técnico en Ingeniería de Sistemas Informáticos, o afines. |
| Conocimientos Requeridos | <ul style="list-style-type: none"> ■ Dominio Intermedio de Python ■ Dominio Básico Intermedio de Electrónica. ■ Conocimiento Intermedios de HTML5, CSS3 y JavaScript |
| Competencias | <ul style="list-style-type: none"> ■ Organización y planificación. ■ Comunicación fluida. ■ Capacidad de atención y concentración. ■ Pensamiento lógico. ■ Capacidad de análisis y resolución de problemas. |

Tabla 4.4: Perfil del programador-analista

| Perfil del administrador | |
|--------------------------|---|
| Funciones Básicas | <ul style="list-style-type: none"> ■ Administrar el sitio de ayuda, con entrega de reportes periódicos sobre el uso del asistente. ■ Explicar pasos sobre el uso del asistente a los usuarios. ■ Cuido del asistente y prevención de accidentes o mal uso por parte de los usuarios. |
| Formación Académica | <ul style="list-style-type: none"> ■ Bachillerato en cualquier especialidad, preferente en informática o afines. |
| Conocimientos Requeridos | <ul style="list-style-type: none"> ■ Dominio Intermedio de ofimática. |
| Competencias | <ul style="list-style-type: none"> ■ Buenas relaciones interpersonales. ■ Organización y planificación. ■ Comunicación fluida. ■ Capacidad de atención y concentración. ■ Buena didáctica para explicación de pasos. |

Tabla 4.5: Perfil del administrador

4.1.2.3. Factibilidad económica

En el estudio de la Factibilidad Económica, se determina el presupuesto del proyecto, es decir, los costos de los recursos técnicos, humanos y materiales tanto para el desarrollo como para la implantación.

Recursos humanos

Aquí se incluyen los gastos generados por el recurso humano necesario para el desarrollo del prototipo únicamente.

| Recursos humanos | | | |
|----------------------|---------------|-----------------|-----------------------|
| Cargo | Costo por mes | Número de meses | Total |
| Analista de sistemas | USD\$ 300.00 | 8 | USD\$ 2,400.00 |
| Desarrollador | USD\$ 200.00 | 6 | USD\$ 1,200.00 |
| Total | | | USD\$ 3,900.00 |

Tabla 4.6: Recursos humanos

Recursos de desarrollo

Se presenta el costo requerido del equipo requerido para el proceso de desarrollo del Software del prototipo, el tiempo utilizado, así como la depreciación de este.

| Recursos de desarrollo | | | |
|-----------------------------|---------------------|----------|-----------------------|
| Descripción | Hardware | | Total |
| | Costo | Meses | |
| Computadora para desarrollo | USD\$ 340.00 | 8 | USD\$ 2,720.00 |
| Raspberry Pi 3 | USD\$ 80.00 | 8 | USD\$ 640.00 |
| Raspberry camera v2 | USD\$ 50.00 | 8 | USD\$ 400.00 |
| Pantalla HDMI | USD\$ 50.00 | 8 | USD\$ 400.00 |
| Subtotal | USD\$ 520.00 | 8 | USD\$ 4,160.00 |
| Depreciación (20 %) | USD\$ 104.00 | 8 | USD\$ 832.00 |
| Total | USD\$ 624.00 | 8 | USD\$ 4,992.00 |

Tabla 4.7: Recursos de desarrollo

Recursos para armado del prototipo

Aquí se presenta el costo de todas las piezas necesarias para el armado del prototipo.

| Recursos para el armado del prototipo | | | |
|--|---------------|-----------------|--------------|
| Cargo | Costo por mes | Número de meses | Total |
| Set de piezas diseñadas para plataforma e iluminación* | USD\$ 30.00 | 1 | USD\$ 30.00 |
| Raspberry Pi 3 * | USD\$ 80.00 | 1 | USD\$ 80.00 |
| Pantalla para Raspberry (con armazón) * | USD\$ 60.00 | 1 | USD\$ 60.00 |
| Raspberry Pi 3 Camera (con extensor) * | USD\$ 50.00 | 1 | USD\$ 50.00 |
| Cable HDMI * | USD\$ 10.00 | 1 | USD\$ 10.00 |
| Cable MicroUSB macho a macho * | USD\$ 5.00 | 1 | USD\$ 5.00 |
| Cable para audio (3.5 mm a 6.5 mm) * | USD\$ 2.00 | 1 | USD\$ 2.00 |
| Micrófono direccional con plataforma * | USD\$ 35.00 | 1 | USD\$ 35.00 |
| Adaptador de audio (6.5mm a XRL) | USD\$ 3.00 | 1 | USD\$ 3.00 |
| Ventilador para PC (estándar) | USD\$ 5.00 | 1 | USD\$ 5.00 |
| Cable USB de carga rápida * | USD\$ 12.00 | 1 | USD\$ 12.00 |
| Set de mouse y teclado inalámbricos * | USD\$ 29.00 | 1 | USD\$ 29.00 |
| Adaptador USB para micrófono y audífono | USD\$ 10.00 | 1 | USD\$ 10.00 |
| Enchufe (tipo americano) | USD\$ 1.50 | 1 | USD\$ 1.50 |
| Cable de corriente (duplex #16)* | USD\$ 2.00 | 1 | USD\$ 2.00 |
| Base para foco (GU10) * | USD\$ 1.00 | 1 | USD\$ 1.00 |
| Foco LED GU10 * | USD\$ 3.50 | 1 | USD\$ 3.50 |
| Switch para lámpara (110V) * | USD\$ 1.00 | 1 | USD\$ 1.00 |
| Pernos de ¼ de pulgada | USD\$ 0.35 | 2 | USD\$ 0.70 |
| Arandelas de ¼ de pulgada | USD\$ 0.35 | 4 | USD\$ 1.40 |
| Tuercas de mariposa de ¼ de pulgada | USD\$ 0.35 | 2 | USD\$ 0.70 |
| Set de tornillo y tuerca M2X4 | USD\$ 1.50 | 2 | USD\$ 3.00 |
| Switch On/Off para circuito | USD\$ 0.50 | 1 | USD\$ 0.50 |
| Conector para batería de 9V | USD\$ 0.40 | 1 | USD\$ 0.40 |
| Batería de 9V recargable | USD\$ 7.00 | 1 | USD\$ 7.00 |
| Cinta LED * | USD\$ 5.00 | 1 | USD\$ 5.00 |
| Cable trenzado UTP * | USD\$ 5.00 | 1 | USD\$ 5.00 |
| Soporte para celular * | USD\$ 3.00 | 1 | USD\$ 3.00 |
| Total Hardware | | | USD\$ 321.70 |

Tabla 4.8: Recursos para armado del prototipo. * La completa descripción del elemento se encuentra en el “Manual de Armado e Instalación”.

Costo total del proyecto

| Costo total del proyecto | |
|------------------------------------|-----------------|
| Rubro | Costo |
| Recursos humanos | USD\$ 3,600.00 |
| Recursos de desarrollo | USD\$ 4,992.00 |
| Recursos para armado del prototipo | USD\$ 321.70 |
| Subtotal | USD\$ 8,913.7 |
| Imprevistos (20 %) | USD\$ 1,782.74 |
| Total | USD\$ 10,696.44 |

Tabla 4.9: Costo total del proyecto.

4.2. Casos de uso

Se describen las actividades de cada actor que interactúa con el prototipo mediante casos de uso.

4.2.1. Paquete de actividades de los actores involucrados

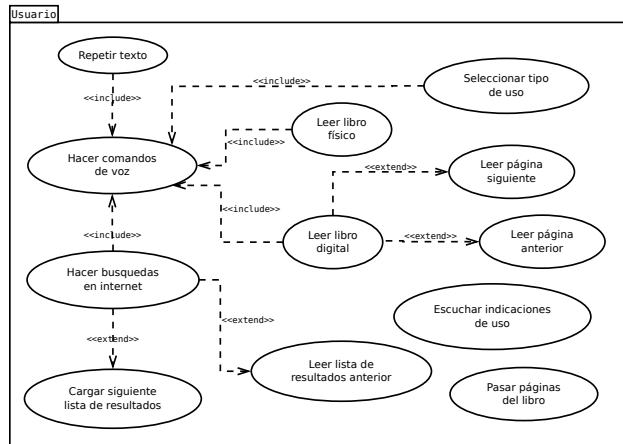


Figura 4.1: Paquete usuario. Fuente: Elaboración propia.

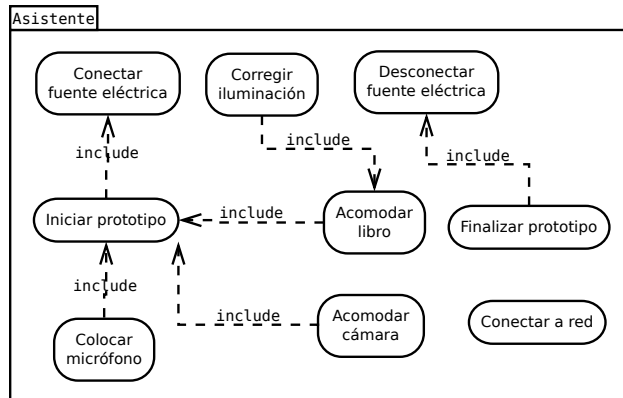


Figura 4.2: Paquete asistente. Fuente: Elaboración propia.

4.2.2. Descripción de las actividades mediante casos de uso, por actor involucrado.

Usuario

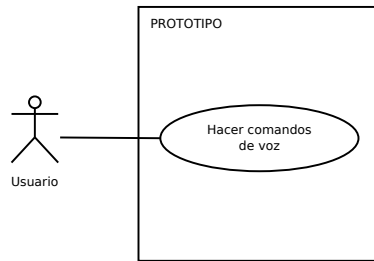


Figura 4.3: Caso de uso: comando de voz. Fuente: Elaboración propia.

| | | |
|---------------------|---|---|
| Caso de uso | Hacer comandos de voz | |
| Prioridad | Alta | |
| REF | REQ-0001 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe estar iniciado y en modo de escucha. ■ El micrófono debe estar conectado y funcional. | |
| Descripción | En éste caso es donde el administrador se posiciona en la máquina que ejecutará el cliente de la plataforma, para que luego la persona con problemas visuales haga uso de ella. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario se posiciona frente al prototipo con su cabeza en dirección y altura del micrófono. 2. El usuario dice el comando determinado por el prototipo según sea su interés. | <p>Prototipo</p> <ol style="list-style-type: none"> 3. El prototipo captura el comando por medio del micrófono. 4. El prototipo procesa el comando por voz. 5. El prototipo interpreta y realiza la acción determinada. |
| | Curso del evento (Caso alterno) | |
| | | <p>Prototipo</p> <ol style="list-style-type: none"> 5. El prototipo no comprende el comando. 6. El prototipo solicita reenviar el comando. |
| Postcondición | <ul style="list-style-type: none"> ■ La realización de una actividad del prototipo según el interés del usuario. | |

Tabla 4.10: Caso de uso: Hacer comando de voz

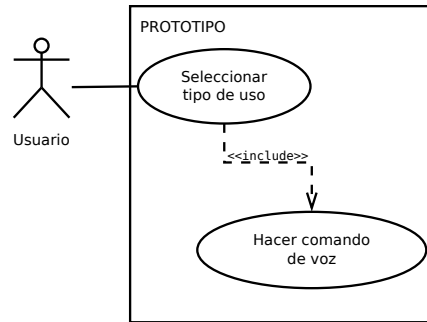


Figura 4.4: Caso de uso: Seleccionar tipo de uso. Fuente: Elaboración propia.

| | | |
|---------------------|---|---|
| Caso de uso | Seleccionar tipo de uso | |
| Prioridad | Alta | |
| REF | REQ-0002 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe estar en modo de escucha. ■ El micrófono debe estar funcionando adecuadamente. ■ El usuario debe usar audífonos para escuchar las respuestas del prototipo. | |
| Descripción | El usuario orienta el uso del prototipo al área que desee, según las capacidades de este, y hacer que las indicaciones y comandos sean acorde a la situación. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Usuario 1. El usuario dicta el comando de voz para escoger la opción de uso que desea. | Prototipo 2. El prototipo captura y procesa el comando. 3. El prototipo orienta su uso y funcionalidad al indicado. 4. El prototipo realiza la función que se le fue requerida. |
| | Curso del evento (Caso alterno) | |
| | | Prototipo 3. El prototipo no cuenta con la funcionalidad requerida. 4. El prototipo solicita reenviar el comando. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo orienta su trabajo hacia la funcionalidad que se le ha requerido. | |

Tabla 4.11: Caso de uso: Seleccionar tipo de uso

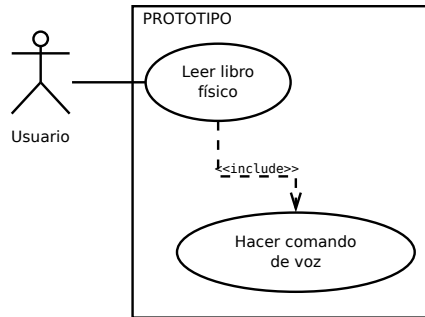


Figura 4.5: Caso de uso: Leer libro físico. Fuente: Elaboración propia.

| | | |
|---------------------|---|---|
| Caso de uso | Leer libro físico | |
| Prioridad | Alta | |
| REF | REQ-0003 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El libro a leer debe estar abierto en la página que se desea leer. ■ La cámara del prototipo debe estar posicionado adecuadamente. ■ La luz que ilumine el libro debe ser la adecuada. ■ El prototipo debe estar en modo de escucha. | |
| Descripción | El usuario indica al prototipo que desea leer el libro que se encuentra colocado en la plataforma. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario dicta el comando de voz para que el prototipo capture la imagen del libro. | <p>Prototipo</p> <ol style="list-style-type: none"> 2. El prototipo captura y procesa la imagen extrayendo el texto. 3. El prototipo pasa el texto a un procesador de voz. 4. El prototipo dicta el contenido del libro al usuario. 5. El prototipo dicta las opciones de acción que el usuario puede tomar. |
| | Curso del evento (Caso alterno) | |
| | | <p>Prototipo</p> <ol style="list-style-type: none"> 3. El prototipo captura la imagen y no puede procesar el texto. 4. El prototipo indica que no es capaz de procesar la imagen. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo ha leído el libro físico que el usuario a escogido. | |

Tabla 4.12: Caso de uso: Leer libro físico

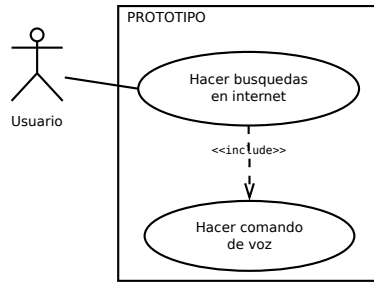


Figura 4.6: Caso de uso: Hacer búsquedas en Internet. Fuente: Elaboración propia.

| | | |
|---------------------|---|--|
| Caso de uso | Hacer búsquedas en Internet | |
| Prioridad | Alta | |
| REF | REQ-0004 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe tener una conexión estable a internet. ■ El prototipo debe estar en modo de escucha. | |
| Descripción | El usuario inicia la búsqueda de información de un tema, de diferentes fuentes, a través de Internet. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario dicta el comando de voz para iniciar la búsqueda en Internet, seguido por el tema que desea investigar. 5. El Usuario escoge la fuente que desea escuchar. | <p>Prototipo</p> <ol style="list-style-type: none"> 2. El prototipo captura y procesa el comando. 3. El prototipo inicia una búsqueda con el tema por varios sitios en internet. 4. El prototipo dicta las fuentes del tema encontradas y solicita escoger una. 6. El prototipo dicta la información recolectada del tema en la fuente que se indicó. |
| | | <p>Curso del evento (Caso alterno)</p> <p>Prototipo</p> <ol style="list-style-type: none"> 4. El prototipo no encuentra información en base al tema indicado.. 5. El prototipo solicita reenviar el comando. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo orienta su trabajo hacia la funcionalidad que se le a requerido. | |

Tabla 4.13: Caso de uso: Hacer búsquedas en Internet

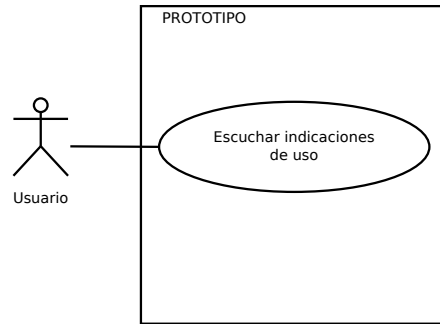


Figura 4.7: Caso de uso: Escuchar indicaciones de uso. Fuente: Elaboración propia.

| | | |
|---------------------|--|--|
| Caso de uso | Escuchar indicaciones de uso | |
| Prioridad | Alta | |
| REF | REQ-0005 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe haber realizado una acción en específico. ■ El usuario debe usar los audífonos conectados al prototipo. | |
| Descripción | El usuario escucha una lista de opciones para posibles acciones a realizar. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Usuario | Prototipo |
| | | <ol style="list-style-type: none"> 1. El prototipo dicta opciones de acciones a realizar al usuario. 2. El prototipo entra en un período de escucha para recibir las indicaciones. |
| | Curso del evento (Caso alterno) | |
| | Prototipo | |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo espera a la decisión del usuario para realizar un comando. | |

Tabla 4.14: Caso de uso: Escuchar indicaciones de uso.

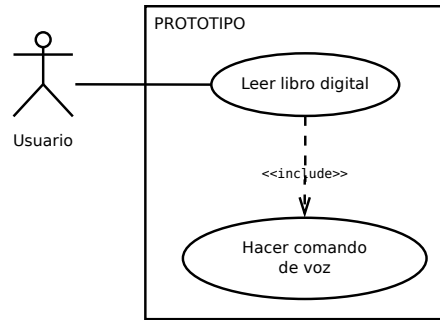


Figura 4.8: Caso de uso: Leer libro digital. Fuente: Elaboración propia.

| | | |
|---------------------|--|--|
| Caso de uso | Leer libro digital | |
| Prioridad | Alta | |
| REF | REQ-0006 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> El asistente debe haber cargado un documento digital (.pdf) al prototipo. | |
| Descripción | El usuario indica que desea leer un libro digital. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | <p>Usuario</p> <ol style="list-style-type: none"> El usuario dicta el comando de voz para leer el libro digital, seguido del número de página que desea. | <p>Prototipo</p> <ol style="list-style-type: none"> El prototipo ingresa al libro digital previamente cargado. El prototipo captura el texto de la página indicada. El prototipo pasa el texto a un sintetizador de voz. El prototipo dicta el contenido del libro al usuario. El prototipo dicta las opciones de acción que el usuario puede tomar. |
| | Curso del evento (Caso alterno) | |
| | | <p>Prototipo</p> <ol style="list-style-type: none"> El prototipo no puede capturar texto en la página indicada. El prototipo dicta las opciones de acciones que el usuario puede tomar. |
| Postcondición | <ul style="list-style-type: none"> El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.15: Caso de uso: Leer libro digital.

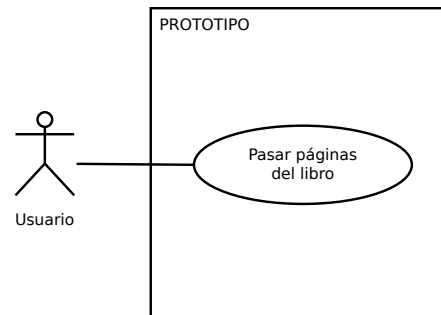


Figura 4.9: Caso de uso: Pasar páginas del libro. Fuente: Elaboración propia.

| | | | |
|--|--|--|-------------------------|
| Caso de uso | Pasar páginas del libro | | |
| Prioridad | Media | | |
| REF | REQ-0007 | | |
| Actores | Usuario | | |
| Precondición | <ul style="list-style-type: none"> ■ Haber terminado de leer una página del libro físico. | | |
| Descripción | El usuario mueve la página del libro físico que está leyendo y lo acomoda para escanear la siguiente. | | |
| Secuencia de acción | Curso del evento (Caso más probable) | | |
| | <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario toma el libro que lee por la página superior. 2. El usuario mueve la página y expone la siguiente página frontal a leer. 3. El usuario acomoda el libro para que la nueva página se posicione debajo de la cámara. </td> <td style="width: 50%; vertical-align: top;"> <p>Prototipo</p> </td> </tr> </table> | <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario toma el libro que lee por la página superior. 2. El usuario mueve la página y expone la siguiente página frontal a leer. 3. El usuario acomoda el libro para que la nueva página se posicione debajo de la cámara. | <p>Prototipo</p> |
| <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario toma el libro que lee por la página superior. 2. El usuario mueve la página y expone la siguiente página frontal a leer. 3. El usuario acomoda el libro para que la nueva página se posicione debajo de la cámara. | <p>Prototipo</p> | | |
| | Curso del evento (Caso alterno) | | |
| | Prototipo | | |
| Postcondición | <ul style="list-style-type: none"> ■ El libro físico está listo para ser leído por el prototipo. | | |

Tabla 4.16: Caso de uso: Pasar páginas del libro.

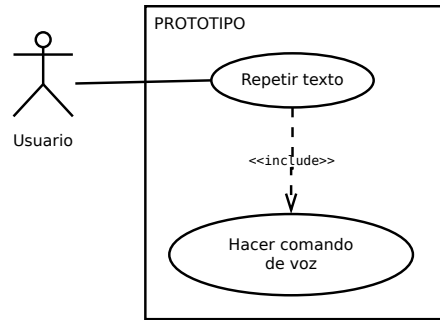


Figura 4.10: Caso de uso: Repetir texto. Fuente: Elaboración propia.

| | | |
|---------------------|---|---|
| Caso de uso | Repetir texto/lista | |
| Prioridad | Media | |
| REF | REQ-0008 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo tiene un texto procesado para dictar, de cualquier fuente posible. ■ El prototipo ya leyó por lo menos una vez el texto almacenado. | |
| Descripción | El usuario indica que desea que se lea de nuevo el texto que acaba de escuchar. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | <p>Usuario</p> <ol style="list-style-type: none"> 1. El usuario dicta el comando de voz para leer de nuevo el texto que se acaba de leer. | <p>Prototipo</p> <ol style="list-style-type: none"> 2. El prototipo captura y procesa el comando. 3. El prototipo pasa el texto almacenado a un sintetizador de voz. 4. El prototipo dicta el texto almacenado. |
| | Curso del evento (Caso alterno) | |
| | | <p>Prototipo</p> <ol style="list-style-type: none"> 3. El prototipo no tiene texto almacenado de alguna actividad previa. 4. El prototipo dicta las opciones de acciones que el usuario puede tomar. |
| Postcondición | <ul style="list-style-type: none"> ■ El usuario a escuchado nuevamente la información que desea estudiar. ■ El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.17: Caso de uso: Repetir Texto.

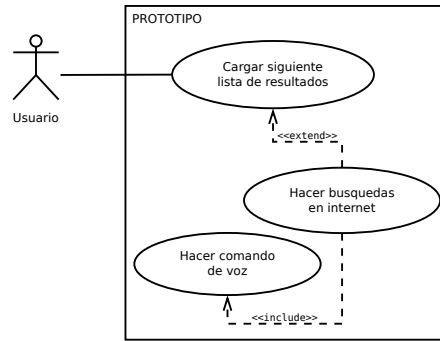


Figura 4.11: Caso de uso: Cargar siguiente lista de resultados. Fuente: Elaboración propia.

| | | |
|---------------------|---|---|
| Caso de uso | Cargar siguiente lista de resultados | |
| Prioridad | Baja | |
| REF | REQ-0009 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> El prototipo ha realizado una búsqueda en Internet sobre un tema específico y presentó una lista de fuentes para el tema deseado. El contenido deseado sobre el tema no se encuentra en la lista de fuentes presentada con anterioridad. | |
| Descripción | Presentación de una nueva lista de opciones de fuentes para el tema que se está buscando en línea. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Usuario | Prototipo |
| | <ol style="list-style-type: none"> El usuario dicta el comando de voz para listar nuevas opciones de fuentes. | <ol style="list-style-type: none"> El prototipo captura y procesa el comando. El prototipo busca nuevas fuentes del tema deseado. El prototipo dicta las nuevas opciones de fuentes encontradas. |
| | Curso del evento (Caso alterno) | |
| | Prototipo | |
| | <ol style="list-style-type: none"> El prototipo no comprende el comando. El prototipo dicta las opciones de acciones que el usuario puede tomar. | |
| Postcondición | <ul style="list-style-type: none"> El usuario ha creado una lista nueva de fuentes para escuchar referentes al tema deseado. El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.18: Caso de uso: Cargar siguiente lista de resultados.

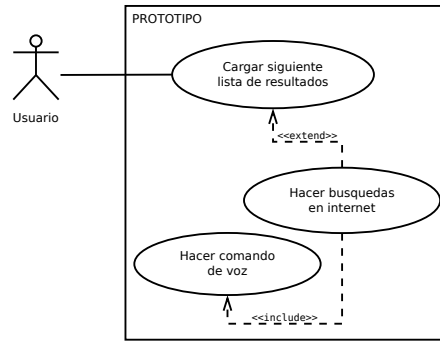


Figura 4.12: Caso de uso: Leer lista de resultados anterior. Fuente: Elaboración propia.

| | | |
|---------------------|--|--|
| Caso de uso | Leer lista de resultados anterior | |
| Prioridad | Baja | |
| REF | REQ-0010 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> El prototipo ha realizado más de una búsqueda en internet sobre un tema específico y presentado una lista de fuentes para el tema deseado. | |
| Descripción | Lectura de la anterior lista de opciones de fuentes para el tema que se está buscando en línea. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | <p>Usuario</p> <ol style="list-style-type: none"> El usuario dicta el comando de voz para listar las fuentes anteriores en la búsqueda deseada. | <p>Prototipo</p> <ol style="list-style-type: none"> El prototipo captura y procesa el comando. El prototipo lee la anterior lista de fuentes. |
| | Curso del evento (Caso alternativo) | |
| | | <p>Prototipo</p> <ol style="list-style-type: none"> El prototipo no comprende el comando. El prototipo dicta las opciones de acciones que el usuario puede tomar. |
| Postcondición | <ul style="list-style-type: none"> Se ha presentado nuevamente la lista de fuentes anterior. El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.19: Caso de uso: Leer lista de resultados anterior.

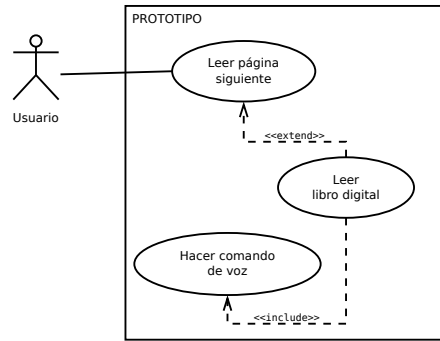


Figura 4.13: Caso de uso: Leer página siguiente. Fuente: Elaboración propia.

| | | |
|---------------------|---|--|
| Caso de uso | Leer página siguiente | |
| Prioridad | Media | |
| REF | REQ-0011 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo está en medio de la lectura de un libro digital. ■ El prototipo ha terminado de leer la página actual del libro digital. | |
| Descripción | Iniciar la lectura de la página siguiente a la actual en el libro digital. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Usuario | Prototipo |
| | <ol style="list-style-type: none"> 1. El usuario dicta el comando de voz para pasar a la página siguiente. | <ol style="list-style-type: none"> 2. El prototipo captura y procesa el comando. 3. El prototipo dicta la página siguiente a la que leyó previamente. |
| | Curso del evento (Caso alterno) | |
| | | Prototipo |
| | | <ol style="list-style-type: none"> 3. El prototipo determina que es la última página del libro. 4. El prototipo dicta las opciones de acciones que el usuario puede tomar. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo ha leído una nueva página del libro digital y la establece como la página actual. ■ El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.20: Caso de uso: Leer página siguiente.

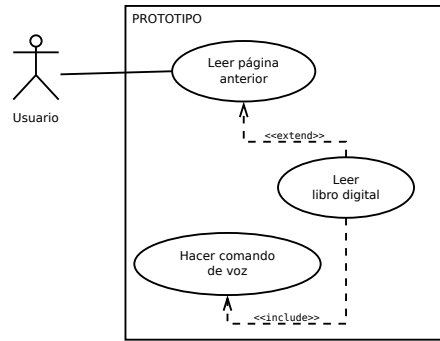


Figura 4.14: Caso de uso: Leer página anterior. Fuente: Elaboración propia.

| | | |
|---------------------|---|---|
| Caso de uso | Leer página anterior | |
| Prioridad | Media | |
| REF | REQ-0012 | |
| Actores | Usuario | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo está en medio de la lectura de un libro digital. ■ El prototipo ha terminado de leer la página actual del libro digital. | |
| Descripción | Iniciar la lectura de la página anterior a la actual en el libro digital. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Usuario | Prototipo |
| | <ol style="list-style-type: none"> 1. El usuario dicta el comando de voz para pasar a la página anterior. | <ol style="list-style-type: none"> 2. El prototipo captura y procesa el comando. 3. El prototipo dicta la página anterior a la que leyó previamente. |
| | Curso del evento (Caso alterno) | |
| | | Prototipo |
| | | <ol style="list-style-type: none"> 3. El prototipo determina que es la primera página del libro. 4. El prototipo dicta las opciones de acciones que el usuario puede tomar. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo ha leído la página anterior del libro digital y la establece como la página actual. ■ El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.21: Caso de uso: Leer página anterior.

Asistente humano

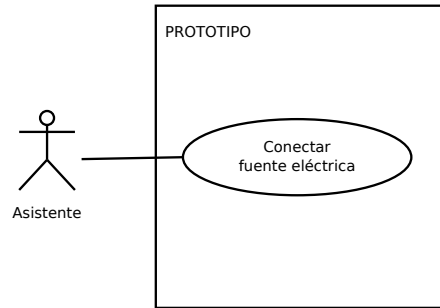


Figura 4.15: Caso de uso: Conectar fuente eléctrica. Fuente: Elaboración propia.

| | | |
|---------------------|---|--|
| Caso de uso | Conectar fuente eléctrica | |
| Prioridad | Alta | |
| REF | REQ-0013 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> ■ Debe existir una fuente eléctrica cercana a la que se pueda acceder. | |
| Descripción | Conectar el prototipo a una fuente eléctrica, estable y adecuada para su uso. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 1. El asistente toma el conector eléctrico del prototipo y lo conecta a una fuente de energía ya definida. | <ol style="list-style-type: none"> 2. El prototipo enciende el ordenador y se levanta el sistema operativo. |
| | Curso del evento (Caso alternativo) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 3. El asistente desconecta el prototipo de la fuente de energía. 4. El asistente revisa y corrige el cableado del prototipo. 5. El asistente reconecta el prototipo a la fuente de energía. | <ol style="list-style-type: none"> 2. El prototipo no logra encender el ordenador. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo ha leído la página anterior del libro digital y la establece como la página actual. ■ El prototipo queda en espera para nuevas instrucciones. | |

Tabla 4.22: Caso de uso: Conectar fuente eléctrica.

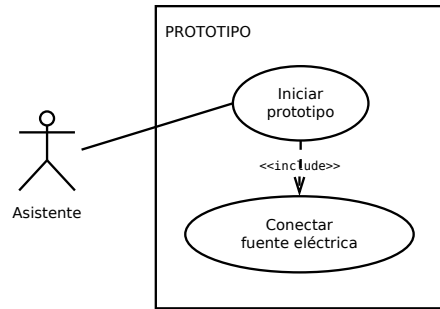


Figura 4.16: Caso de uso: Iniciar prototipo. Fuente: Elaboración propia.

| | | |
|---------------------|---|--|
| Caso de uso | Iniciar prototipo | |
| Prioridad | Alta | |
| REF | REQ-0014 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe de haber iniciado el sistema operativo sin ningún problema. ■ Todos los periféricos del prototipo deben estar conectados y funcionando. | |
| Descripción | Es el inicio del software para el uso del prototipo, así como la definición de los parámetros necesarios para el caso específico. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 1. El asistente localiza en el ordenador el software utilitario del prototipo. 2. El asistente ejecuta el software utilitario del prototipo con los parámetros correspondientes para su uso. | <ol style="list-style-type: none"> 3. El prototipo inicia el software utilitario. |
| | Curso del evento (Caso alterno) | |
| | Asistente | Prototipo |
| | - | <ol style="list-style-type: none"> 3. El prototipo no procesa los parámetros indicados y no inicia. 4. El prototipo indica un error en los parámetros definidos. |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo inicia el modo de escucha para su uso. | |

Tabla 4.23: Caso de uso: Conectar fuente eléctrica.

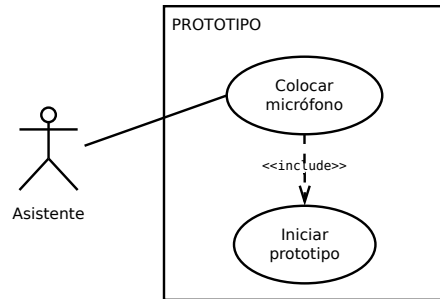


Figura 4.17: Caso de uso: Colocar micrófono. Fuente: Elaboración propia.

| | | |
|---------------------|--|------------------|
| Caso de uso | Colocar micrófono | |
| Prioridad | Alta | |
| REF | REQ-0015 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe estar colocado en una superficie plana. ■ El prototipo debe estar frente al usuario. ■ El prototipo debe estar a la altura de la parte media del pecho del usuario. | |
| Descripción | Ubicar el micrófono en una posición idónea para que el usuario pueda dictar los comandos de uso al prototipo y este pueda captarlos completa y claramente. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 1. El asistente ayuda al usuario a sentarse frente al prototipo en el área designada. 2. El asistente coloca la base del micrófono frente al usuario. 3. El asistente asegura el micrófono en su base. 4. El asistente acomoda el ángulo del micrófono acorde al usuario. | - |
| | Curso del evento (Caso alterno) | |
| | Asistente | Prototipo |
| | - | - |
| Postcondición | <ul style="list-style-type: none"> ■ El usuario es capaz de interactuar por medio de comandos con el prototipo. | |

Tabla 4.24: Caso de uso: Colocar micrófono.

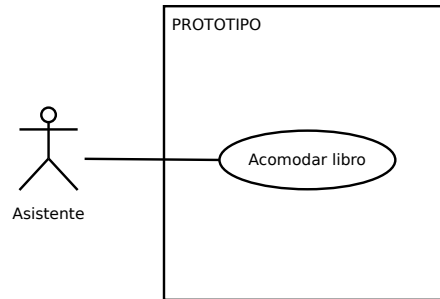


Figura 4.18: Caso de uso: Acomodar libro. Fuente: Elaboración propia.

| | | |
|---------------------|---|------------------|
| Caso de uso | Acomodar libro | |
| Prioridad | Alta | |
| REF | REQ-0016 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> El libro debe cumplir los parámetros definidos para un adecuado uso del prototipo. | |
| Descripción | El asistente acomoda el libro para que su información pueda ser procesada por el prototipo. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> El asistente coloca el libro en la plataforma designada. El asistente abre el libro con la página que el usuario desea leer apuntando a la cámara del prototipo. | - |
| | Curso del evento (Caso alternativo) | |
| | Asistente | Prototipo |
| | - | - |
| Postcondición | <ul style="list-style-type: none"> El prototipo es capaz de leer el libro para el usuario. | |

Tabla 4.25: Caso de uso: Acomodar libro.

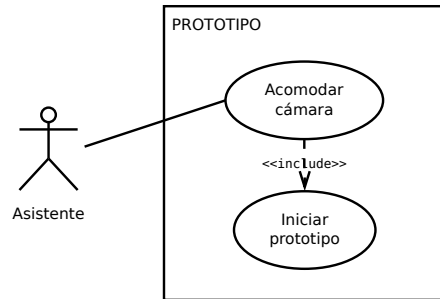


Figura 4.19: Caso de uso: Acomodar cámara. Fuente: Elaboración propia.

| | | |
|---------------------|--|--|
| Caso de uso | Acomodar cámara | |
| Prioridad | Alta | |
| REF | REQ-0017 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe de haber sido iniciado. ■ El libro a leer debe de estar posicionado sobre la plataforma del prototipo. ■ La cámara del prototipo no encuadra bien al libro en la plataforma. | |
| Descripción | El asistente coloca ajusta la posición de la cámara del prototipo para que capture la totalidad de la página deseada del libro sobre la plataforma. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 2. El asistente afloja el eje de movimiento de la cámara del prototipo. 3. El asistente apunta la cámara en dirección del libro, procurando hacer el encuadre correcto para abarcar todas las dimensiones del libro. 4. El asistente asegura el eje de movimiento del prototipo. | <ol style="list-style-type: none"> 1. El prototipo indica que no ha sido capaz de extraer el texto del libro. |
| | Curso del evento (Caso alternativo) | |
| | Asistente | Prototipo |
| | - | - |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo puede iniciar a procesar la información del libro. | |

Tabla 4.26: Caso de uso: Acomodar cámara.

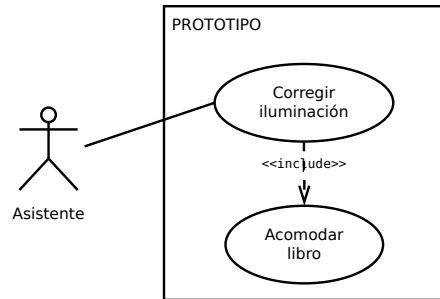


Figura 4.20: Caso de uso: Corregir iluminación. Fuente: Elaboración propia.

| | | |
|---------------------|---|------------------|
| Caso de uso | Corregir iluminación | |
| Prioridad | Alta | |
| REF | REQ-0018 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe de haber sido iniciado. ■ El libro debe de estar posicionado sobre el prototipo. ■ La cámara debe de estar colocada y apuntando al libro. | |
| Descripción | El asistente coloca el dispositivo de iluminación apuntando sobre la superficie del libro. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 1. El asistente debe de asegurarse que el dispositivo de iluminación funciona 2. El asistente coloca la iluminación en dirección del libro ya antes posicionado, procurando que toda la superficie del libro esté bien iluminada | - |
| | Curso del evento (Caso alterno) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 2. El asistente se percata que el dispositivo no funciona. 3. El asistente notifica que debe de hacerse un cambio del dispositivo. | - |
| Postcondición | <ul style="list-style-type: none"> ■ El prototipo puede iniciar a procesar la información del libro. | |

Tabla 4.27: Caso de uso: Corregir iluminación.

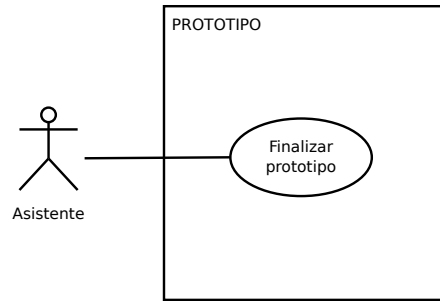


Figura 4.21: Caso de uso: Finalizar prototipo. Fuente: Elaboración propia.

| | | |
|---------------------|--|---|
| Caso de uso | Finalizar prototipo | |
| Prioridad | Alta | |
| REF | REQ-0019 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> El usuario debe haber terminado de usar el prototipo. | |
| Descripción | El asistente apaga por completo el hardware del prototipo luego de realizar el respectivo uso. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> El asistente abre el menú principal del sistema operativo del prototipo El asistente selecciona entre las opciones, la opción de "Apagar" | <ol style="list-style-type: none"> El prototipo realiza la ejecución de la instrucción La pantalla del prototipo se apaga |
| | Curso del evento (Caso alterno) | |
| | Asistente | Prototipo |
| | - | - |
| Postcondición | <ul style="list-style-type: none"> El prototipo puede iniciar a procesar la información del libro. | |

Tabla 4.28: Caso de uso: Finalizar prototipo.

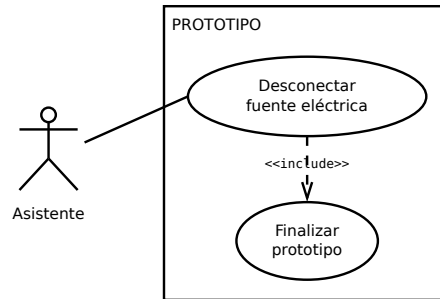


Figura 4.22: Caso de uso: Desconectar fuente eléctrica. Fuente: Elaboración propia.

| | | |
|---------------------|---|------------------|
| Caso de uso | Finalizar prototipo | |
| Prioridad | Alta | |
| REF | REQ-0020 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> El prototipo debe de haber sido apagado. | |
| Descripción | El asistente desconecta el dispositivo de la energía eléctrica. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | 1. El asistente toma con sus manos la extensión de conexión y la remueve del toma corrientes. | - |
| | Curso del evento (Caso alternativo) | |
| | Asistente | Prototipo |
| | - | - |
| Postcondición | - | |

Tabla 4.29: Caso de uso: Desconectar fuente eléctrica.

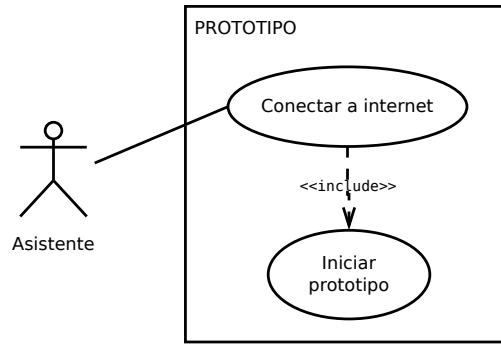


Figura 4.23: Caso de uso: Conectar a Internet. Fuente: Elaboración propia.

| | | |
|---------------------|--|--|
| Caso de uso | Conectar a Internet | |
| Prioridad | Media | |
| REF | REQ-0021 | |
| Actores | Asistente Humano | |
| Precondición | <ul style="list-style-type: none"> ■ El prototipo debe de haber sido iniciado. ■ Debe de existir conexión a internet vía Wi-Fi en la edificación. | |
| Descripción | El asistente conecta a Internet el prototipo para poder realizar búsquedas por la red. | |
| Secuencia de acción | Curso del evento (Caso más probable) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 1. El asistente ubica visualmente la barra principal (la que normalmente muestra la hora y fecha) de opciones del sistema operativo 2. El asistente selecciona la opción que posee un icono de antena para que así se puedan listar todas las conexiones disponibles 3. El asistente selecciona la red Wi-Fi a la cual quiere conectarse | <ol style="list-style-type: none"> 4. El prototipo realiza la conexión a Internet |
| | Curso del evento (Caso alterno) | |
| | Asistente | Prototipo |
| | <ol style="list-style-type: none"> 3. El asistente selecciona la red Wi-Fi a la cual quiere conectarse e introduce la contraseña de la señal Wi-Fi | - |
| Postcondición | - | |

Tabla 4.30: Caso de uso: Conectar a Internet.

4.3. Recursos de Hardware para el desarrollo del proyecto

En el cuadro “Equipo de Hardware para el desarrollo del proyecto”, se muestra el listado de equipo con el que el grupo cuenta para la elaboración del proyecto.

| Procesador | Memoria RAM | Almacenamiento | S/O |
|-----------------------|-------------|----------------|----------------------|
| Intel Core i3 3.5 GHz | 4 GB | 2TB | Windows 10/Debian 10 |
| Intel Celeron 1.6 GHz | 2 GB | 500 GB | Debian 9 |
| Intel Core i5 3.5 GHz | 8 GB | 2TB | Windows 10/Debian 9 |

Tabla 4.31: Recursos de hardware para el desarrollo del proyecto

4.4. Desarrollo de hardware

4.4.1. Diseño

4.4.1.1. Piezas

Para el adecuado funcionamiento del prototipo es necesaria una serie de piezas que formen una plataforma que sirva de soporte y facilite el uso del asistente, teniendo en cuenta su población objetivo, en consecuencia el diseño del prototipo y su proceso de armado se detalla a continuación.

Se cuenta con tres tipos de piezas:

- **Moldeadas:** éstas son diseñadas exclusivamente para el prototipo, están hechas de un material moldeable y firme como madera, madera comprimida, acrílico, tabla roca, etc; se les asigna un código particular a sus partes para identificarlas adecuadamente durante el proceso de armado.
- **Hardware:** Son todas las piezas y componentes físicos que integran el ordenador y sus periféricos para la interacción humano-máquina.
- **Otros:** Son cualquier otra pieza necesaria para el funcionamiento del prototipo, pero de índole secundaria como cables de corriente, baterías, tuercas, tornillos, luces, etc.

4.4.1.2. Plataforma

Se necesita una pieza de cada tipo, a no ser que se especifique en el caso, presentado a continuación. Es recomendable el uso de materiales con un grosor de medio centímetro (0.5cm).

A: Esta pieza está formada por una figura en forma de letra ele (L), puede entenderse como la unión de dos rectángulos de diferentes dimensiones, con las medidas mostradas a continuación (figura 4.24):

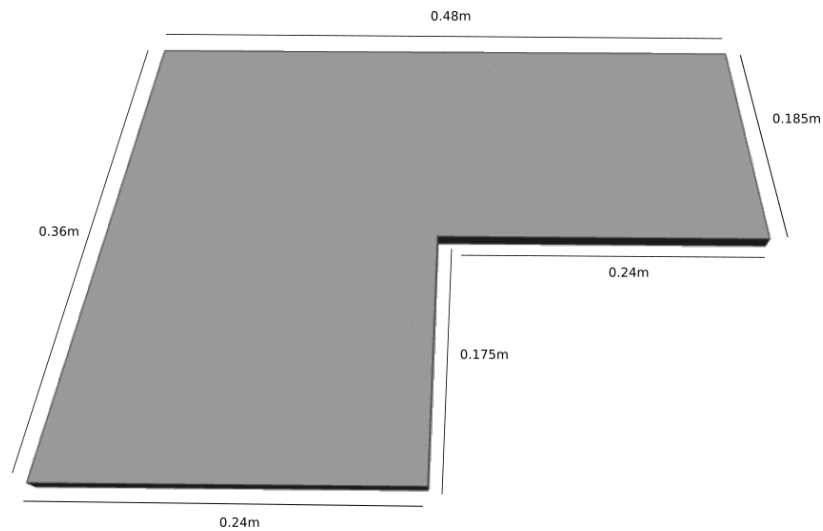


Figura 4.24: Pieza A. Fuente: Elaboración propia.

Se hacen 3 ranuras en la pieza A (A1, A2, A3) con el propósito de que sirvan de uniones en el armado de la plataforma, cada una con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), el largo y la posición se hacen según se muestra en la (Figura 4.25).

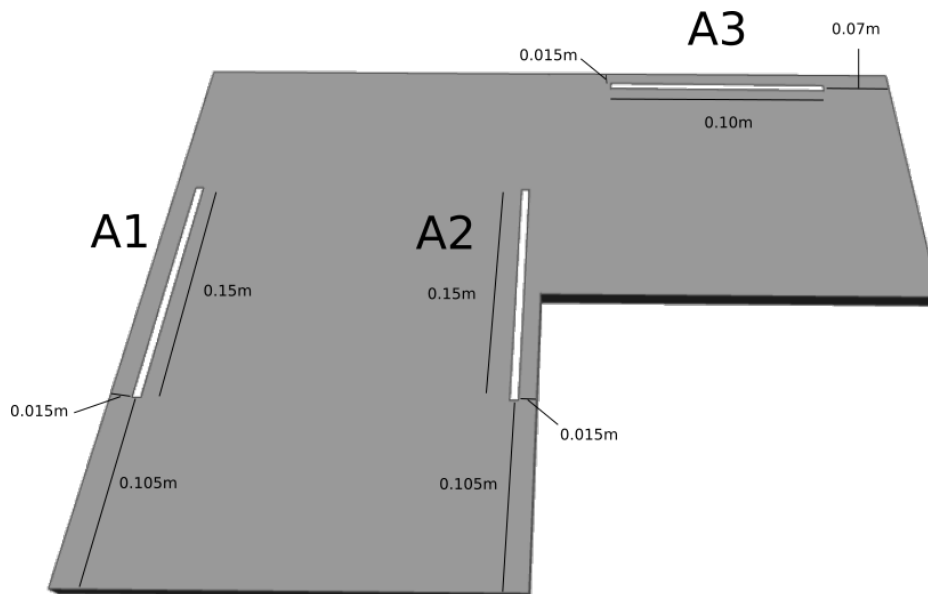


Figura 4.25: Pieza A. Fuente: Elaboración propia.

B: Esta pieza presenta 3 pestañas (B1, B2, B3) y una muesca (B4) que servirán de uniones en el proceso de armado de la plataforma, la altura de las pestañas B1, B2, B3 es igual al grosor del material con que se está trabajando (0.5cm recomendable), la profundidad de la muesca B4 es igual a la altura de las pestañas (0.5cm recomendable), las medidas generales se muestran en la figura 4.26.

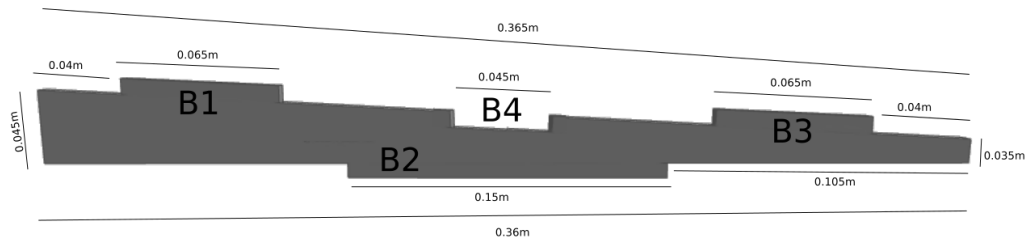


Figura 4.26: Pieza B. Fuente: Elaboración propia.

Se hace una ranura en la pieza B (B5) con el propósito de que sirva de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), el largo y la posición se hacen según se muestra en la Figura 4.27.



Figura 4.27: Pieza B. Fuente: Elaboración propia.

C: Esta pieza presenta 3 pestañas (C1, C2, C3) que servirán de uniones en el proceso de armado de la plataforma, la altura de las pestañas es igual al grosor del material con que se está trabajando (0.5cm recomendable), las medidas generales se muestran en la figura 4.28.

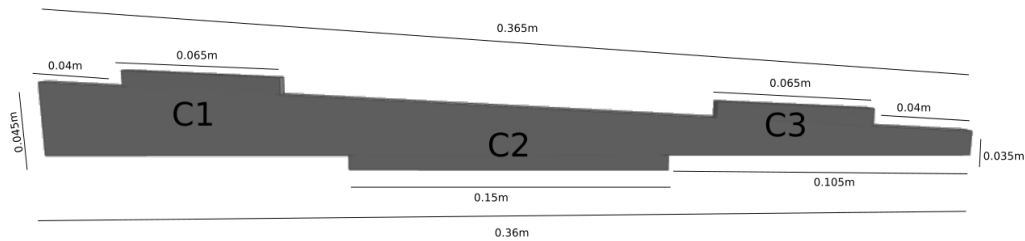


Figura 4.28: Pieza C. Fuente: Elaboración propia.

Se hace una ranura en la pieza C (C4) con el propósito de que sirva de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), el largo y la posición se hacen según se muestra en la Figura 4.29.



Figura 4.29: Pieza C. Fuente: Elaboración propia.

D: Esta pieza tiene la forma de un rectángulo, cuyas medidas se describen en la Figura 4.30 a continuación.

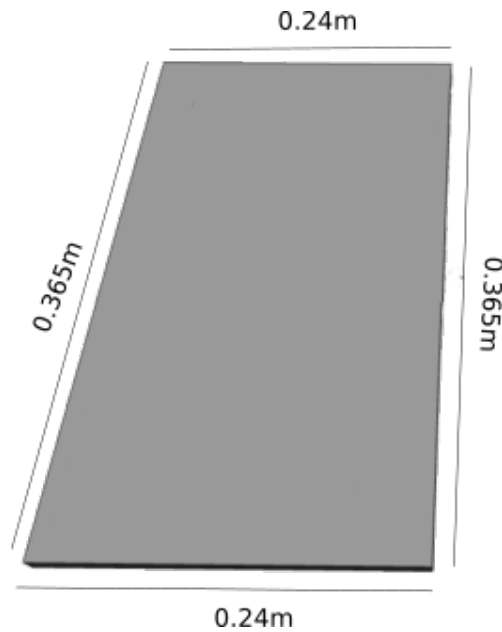


Figura 4.30: Pieza D. Fuente: Elaboración propia.

Se hacen unas ranuras en la pieza D (D1, D2, D3, D4) con el propósito de que sirvan de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), el largo y la posición se hacen según se muestra en la Figura 4.31.

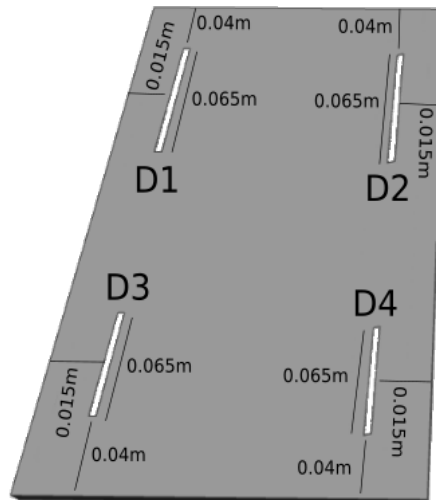


Figura 4.31: Pieza D. Fuente: Elaboración propia.

E: Esta pieza tiene la forma de un rectángulo y sirve para soporte de la ventilación del prototipo, cuenta con una serie de ranuras diseñadas para el flujo de aire y la extracción de calor producido por el prototipo, cuenta con una pestaña (E1) que servirá de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), las medidas de la pieza se muestran en la Figura 4.32.

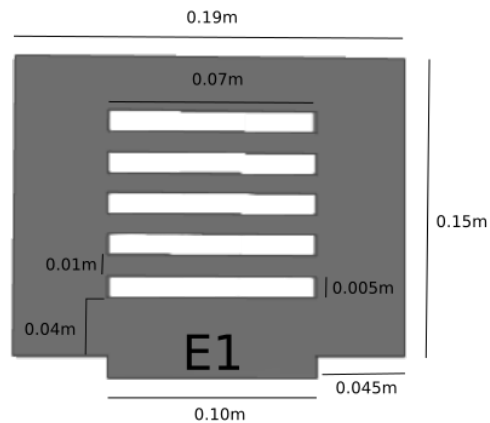


Figura 4.32: Pieza E. Fuente: Elaboración propia.

F: Esta pieza es la base del soporte de la pantalla del prototipo, cuenta con dos ranuras (F1, F2) con el propósito de que sirvan de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), el largo y la posición se hacen según se muestra en la Figura 4.33.

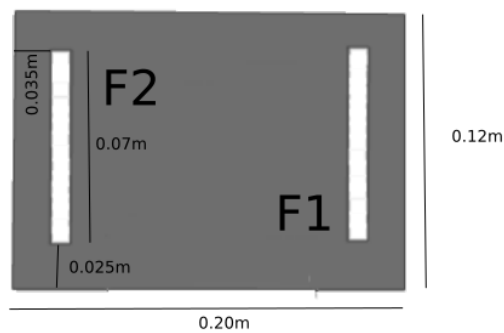


Figura 4.33: Pieza F. Fuente: Elaboración propia.

G: Esta pieza tiene como función ser el soporte de la pantalla del prototipo, cuenta con una pestaña (G1) que servirá de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), sus medidas se definen en la Figura 4.34 a continuación.

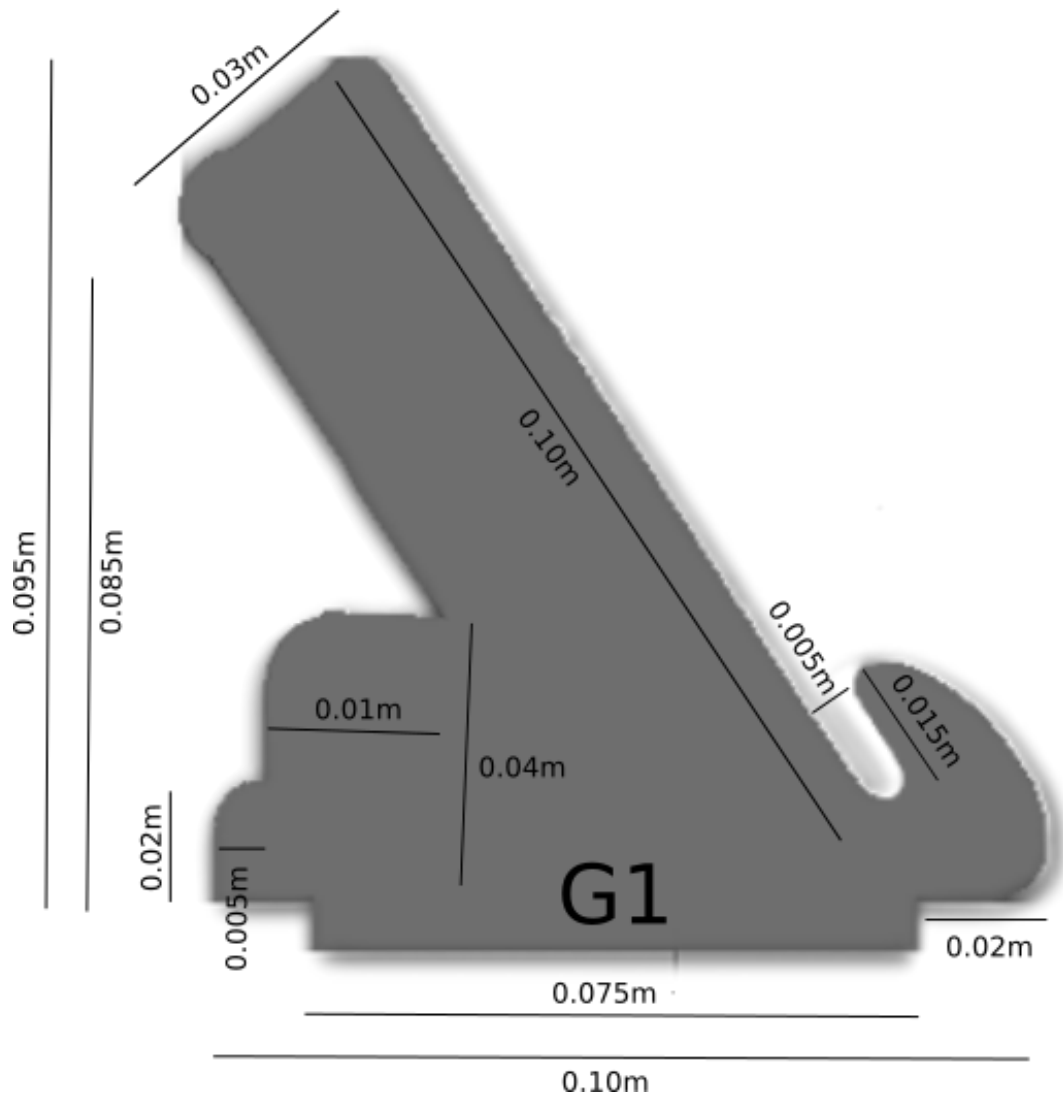


Figura 4.34: Pieza G. Fuente: Elaboración propia.

H: Esta pieza tiene como función ser el soporte de la pantalla del prototipo, cuenta con una pestaña (H1) que servirá de unión en el armado de la plataforma, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), sus medidas se definen en la Figura 4.35 a continuación.

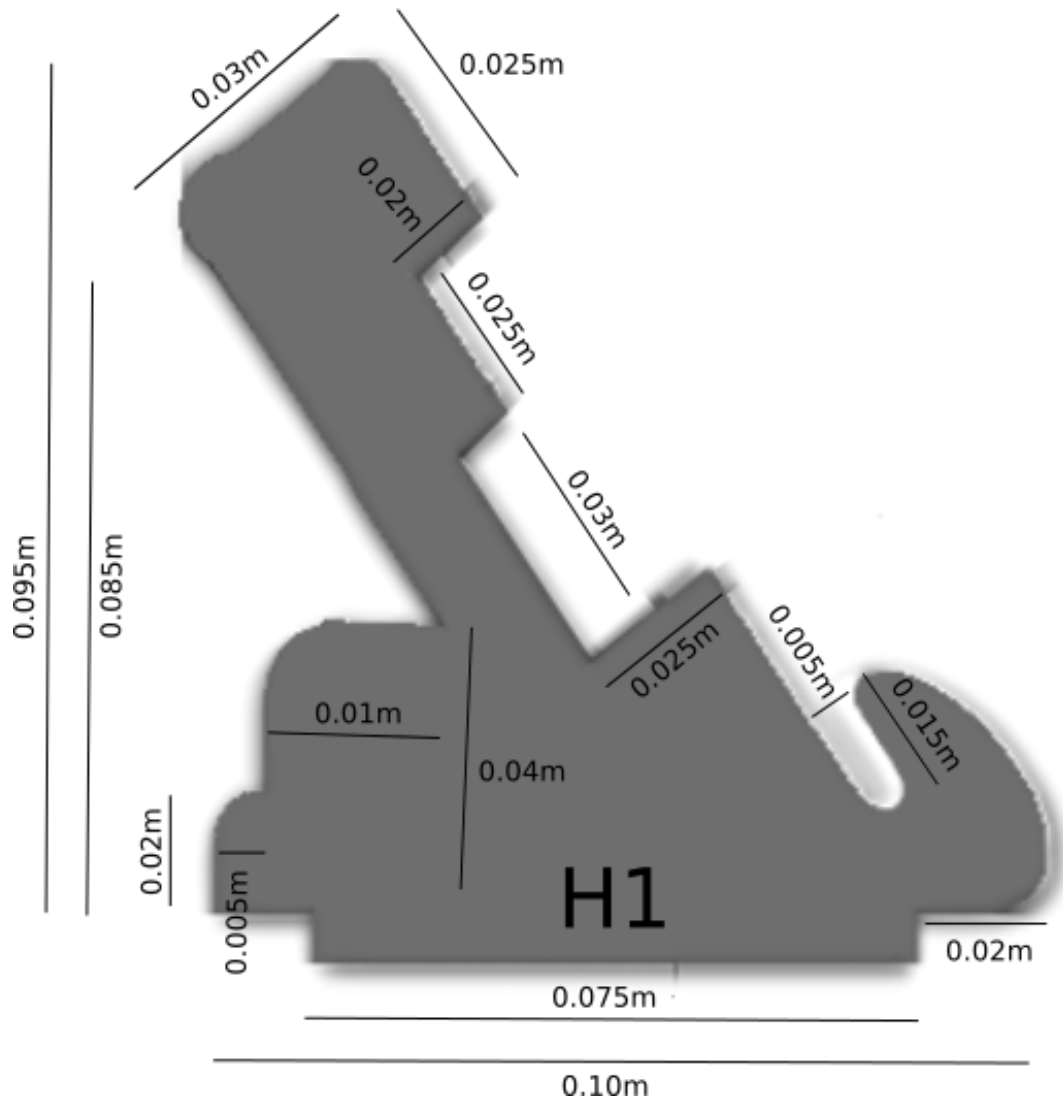


Figura 4.35: Pieza H. Fuente: Elaboración propia.

4.4.1.3. Soporte de cámara e iluminación

Estas piezas del prototipo están diseñadas para sostener y facilitar la captura de los textos que se van a procesar, así como una adecuada iluminación para que funcione en varios ambientes. Los materiales son variados, por lo que se listan a continuación.

Soporte de cámara: 1 Soporte de pinza para celular con brazo flexible de 70cm de largo.



Figura 4.36: Soporte de cámara

1 metro de cable de par trenzado UTP.

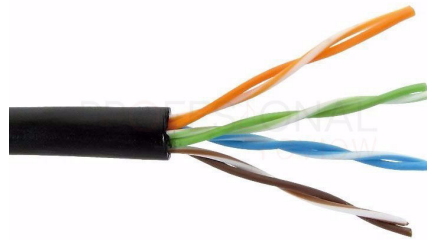


Figura 4.37: Cable UTP

10 cm de cinta LED blanca.



Figura 4.38: Cinta LED

1 batería de 9V (se recomienda recargable).



Figura 4.39: Batería 9V

1 conector para batería de 9V.



Figura 4.40: Conector batería 9V

Switch On/Off para circuito.



Figura 4.41: Switch On/Off

2 tornillos con tuerca M2X4.



Figura 4.42: Tornillos M2.



Figura 4.43: Tuercas M2.

Piezas del material con que se hicieron las partes de la plataforma.

I: Estas piezas sirven como base del soporte de la cámara. con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), se necesitan 3 piezas de estas, sus medidas se definen en la Figura 4.44 a continuación.

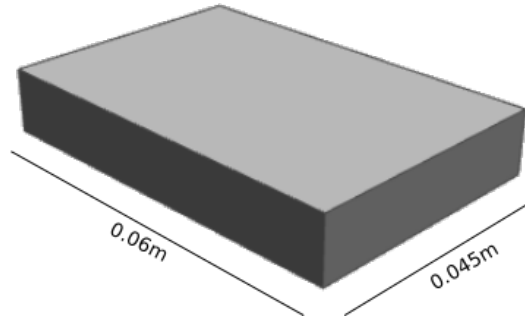


Figura 4.44: Pieza I. Fuente: Elaboración propia.

J: Estas piezas sirven como base del soporte de la cámara. con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), sus medidas se definen en la Figura 4.45 a continuación.

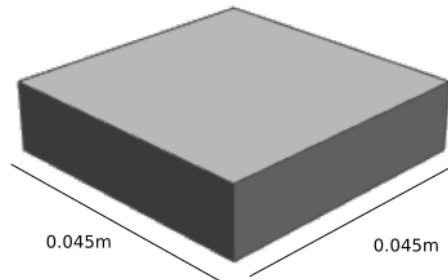


Figura 4.45: Pieza J. Fuente: Elaboración propia.

Se hace una hendidura (J1) para que sirva de unión el el soporte de cámara, sus medidas se definen en la Figura 4.46 a continuación.

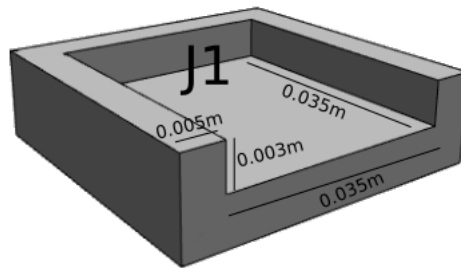


Figura 4.46: Pieza J. Fuente: Elaboración propia.

Iluminación:

2 pernos de ¼x1.



Figura 4.47: Pernos ¼

2 tuercas mariposa de ¼.



Figura 4.48: Tuercas mariposa ¼

4 arandelas de ¼.



Figura 4.49: arandelas ¼

1 switch para lámpara.



Figura 4.50: Switch para lámpara

1 foco LED GU10.



Figura 4.51: LED GU10

1 base para foco tipo GU10.

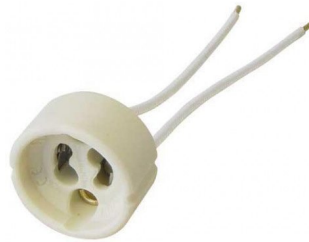


Figura 4.52: Base LED GU10

1 metro de cable duplex número 16.



Figura 4.53: Cable duplex número 16

1 Enchufe tipo americano de 2 puntas.



Figura 4.54: Enchufe tipo americano 2 puntas

2 Pasadores de madera de $\frac{1}{4}$ de pulgada de grosor y 2 pulgadas de largo.



Figura 4.55: Pasadores de madera

Piezas del material con que se hicieron las partes de la plataforma.

K: Esta pieza es un rectángulo que sirven como parte de la base del soporte de la luz, sus medidas se definen en la Figura 4.56 a continuación.

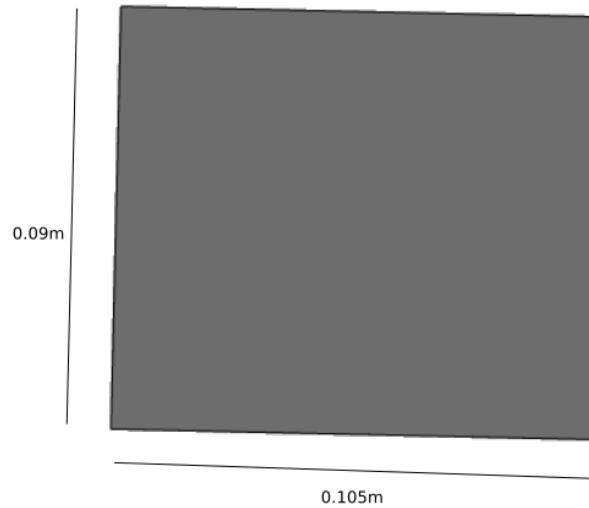


Figura 4.56: Pieza K. Fuente: Elaboración propia.

L: Esta pieza es una letra “L” que sirven como pilar de movimiento de la luz, sus medidas se definen en la Figura 4.57 a continuación.

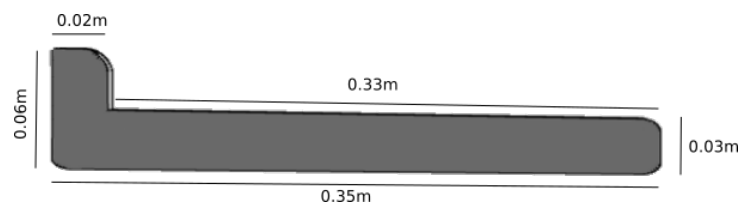


Figura 4.57: Pieza L. Fuente: Elaboración propia.

Crear una abertura (L1) de un cuarto de pulgada (0.635 cm) para el movimiento de la luz, las medidas se definen a continuación.

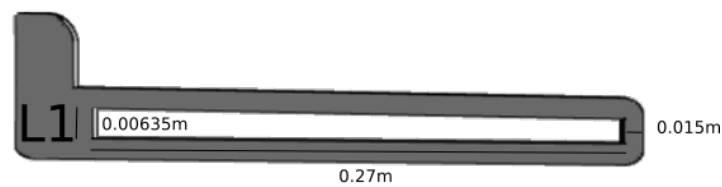


Figura 4.58: Pieza L. Fuente: Elaboración propia.

M: Esta pieza es un rectángulo con bordes redondeados que sirven como soporte para el movimiento de la luz, sus medidas se definen en la Figura 4.59 a continuación.

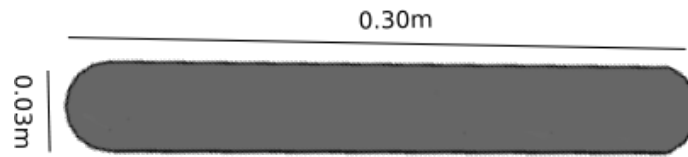


Figura 4.59: Pieza M. Fuente: Elaboración propia.

Perforar 4 agujeros (M1, M2, M3, M4) de un cuarto de pulgada (0.635cm) para los ejes de movimiento, en las posiciones mostradas a continuación.

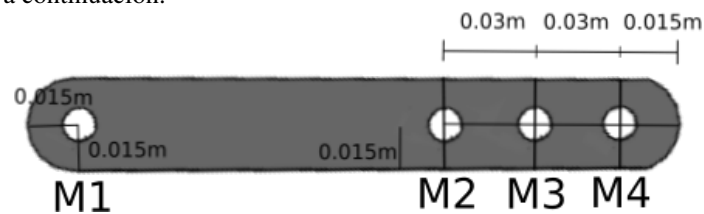


Figura 4.60: Pieza M. Fuente: Elaboración propia.

N: Esta pieza es un rectángulo que sirven como soporte para el movimiento de la luz, se necesitan 2 de estas piezas, sus medidas se definen en la Figura 4.61 a continuación.

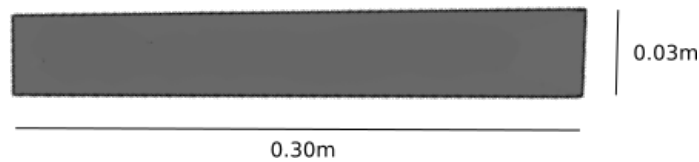


Figura 4.61: Pieza N. Fuente: Elaboración propia.

Se perforan 2 agujeros (N1, N2) en los extremos de un cuarto de pulgada (0.635cm) como ejes entre las piezas y una ranura (N3) por el medio, las medidas se muestran a continuación.

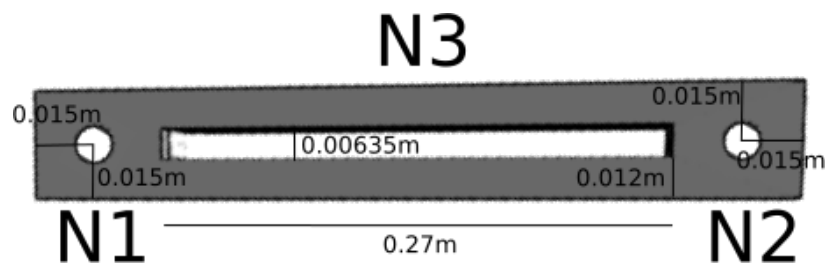


Figura 4.62: Pieza N. Fuente: Elaboración propia.

O: Esta pieza es un pequeño rectángulo redondeado de un extremo que sirve como eje de movimiento de la luz en el prototipo, sus medidas son las siguientes mostradas en la Figura 4.63.

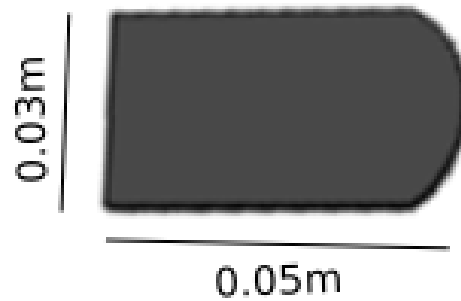


Figura 4.63: Pieza O. Fuente: Elaboración propia.

Se perfora un agujero (O1) de un cuarto de pulgada (0.635cm) para que funcione como eje de movimiento para la luz del prototipo, su posición se define en la siguiente imagen.

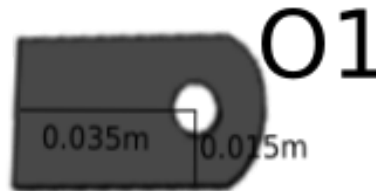


Figura 4.64: Pieza O. Fuente: Elaboración propia.

P: Esta pieza es parte del soporte para el movimiento de la luz del prototipo, sus medidas son las siguientes mostradas en la Figura 4.65.

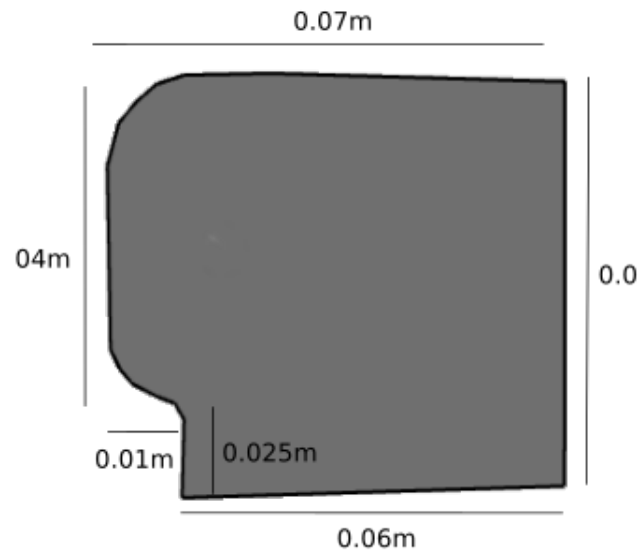


Figura 4.65: Pieza P. Fuente: Elaboración propia.

perforar un agujero de un cuarto de pulgada (0.635cm) para que funcione como eje de movimiento en la base del soporte de la luz del prototipo, su posición se define en la siguiente imagen.

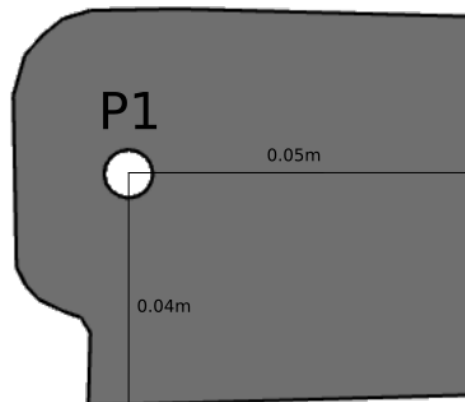


Figura 4.66: Pieza P. Fuente: Elaboración propia.

Q: Esta pieza es parte del soporte para el movimiento de la luz del prototipo, sus medidas son las siguientes mostradas en la Figura 4.67.

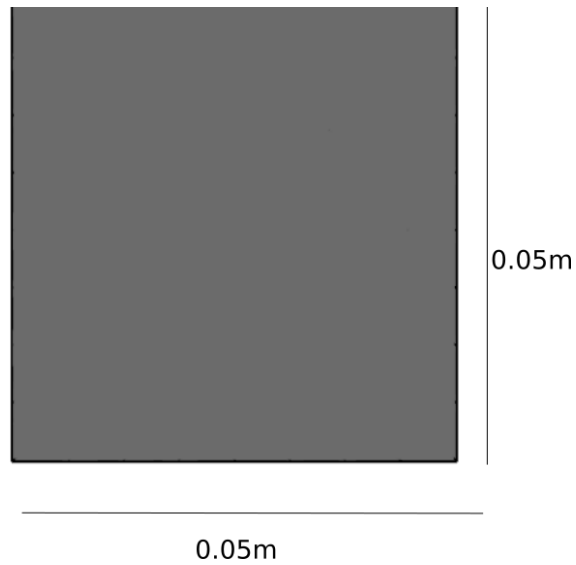


Figura 4.67: Pieza Q. Fuente: Elaboración propia.

Se perfora un agujero en el centro (Q1), para colocar la base de la luz del prototipo, con un centro y dimensiones mostradas en la siguiente figura.

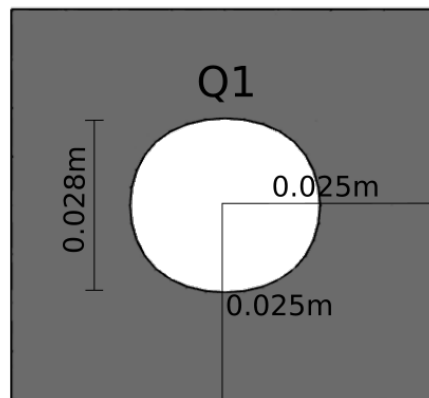


Figura 4.68: Pieza Q. Fuente: Elaboración propia.

Se hace una hendidura (Q2) en forma angular sobre la pieza para crear espacio para los cables de la base de la luz, las medidas se muestran a continuación.

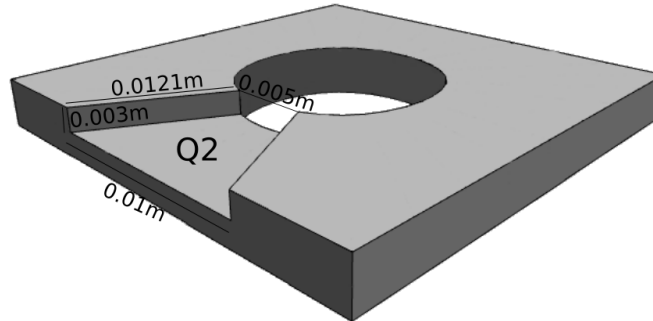


Figura 4.69: Pieza Q. Fuente: Elaboración propia.

R: Esta pieza es parte del soporte para el movimiento de la luz del prototipo, sus medidas son las siguientes mostradas en la Figura 4.70.

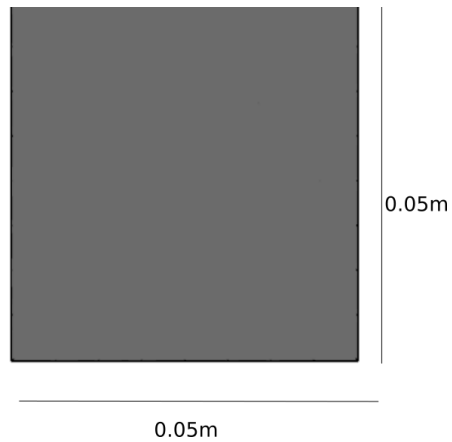


Figura 4.70: Pieza R. Fuente: Elaboración propia.

Se hace una ranura (R1), para colocar la base de la luz del prototipo, con un ancho igual al grosor del material con el que se está trabajando (0.5cm recomendable), sus medidas son las siguientes.

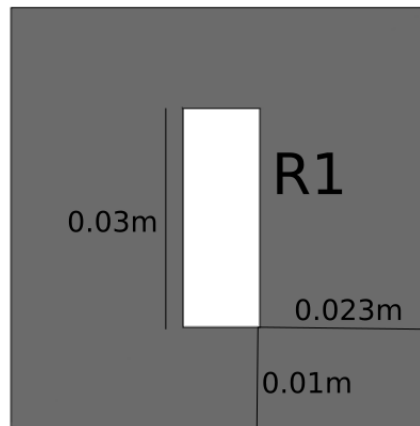


Figura 4.71: Pieza R. Fuente: Elaboración propia.

S: Esta pieza es parte del soporte para el movimiento de la luz del prototipo, sus medidas son las siguientes mostradas en la Figura 4.72.

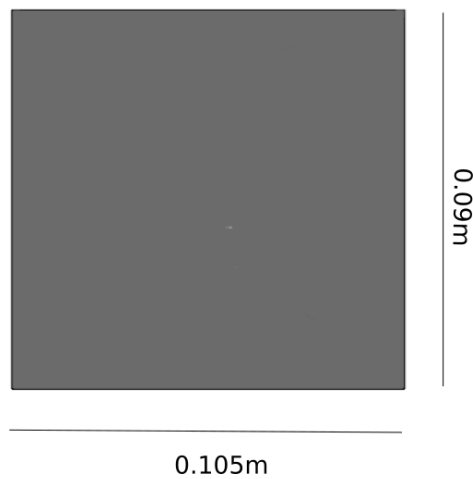


Figura 4.72: Pieza S. Fuente: Elaboración propia.

Se hace una ranura (S1) , para colocar la base del soporte de la luz del prototipo, con un ancho igual a tres veces el grosor del material con el que se está trabajando (0.5cm recomendable), sus medidas son las siguientes.

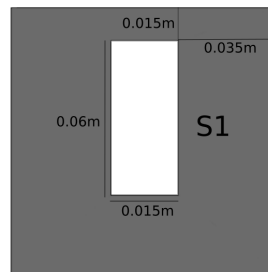


Figura 4.73: Pieza S. Fuente: Elaboración propia.

4.4.2. Hardware

Las piezas aquí detalladas son el equipo computacional requerido para el funcionamiento del prototipo, se listan a continuación.

Raspberry Pi 3 modelo B.



Figura 4.74: Raspberry Pi 3 modelo B.

Raspberry Pi 3 Camera V2.

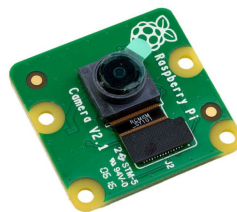


Figura 4.75: Raspberry Pi 3 Camera V2.

Pantalla de 7 Pulgadas para Raspberry Pi con armazón.

Figura 4.76: Pantalla de 7 Pulgadas para Raspberry Pi con armazón. Fuente: Elaboración propia.

Cable HDMI de 30cm.

Figura 4.77: Cable HDMI. Fuente: Elaboración propia.

Conector MicroUSB macho a macho de 30cm.

Figura 4.78: Conector MicroUSB macho a macho. Fuente: Elaboración propia.

Adaptador USB para micrófono y Audífono.

Figura 4.79: Adaptador USB para micrófono y Audífono. Fuente: Elaboración propia.

Cable para audio macho a macho de 3.5mm a 6.5mm Mono de 1m de largo.



Figura 4.80: Cable para audio macho a macho de 3.5mm a 6.5mm Mono. Fuente: Elaboración propia.

Adaptador para audio hembra a hembra de 6.5mm a XLR.



Figura 4.81: Adaptador para audio hembra a hembra de 6.5mm a XLR. Fuente: Elaboración propia.

Micrófono direccional con conector XLR.



Figura 4.82: Micrófono direccional con conector XLR. Fuente: Elaboración propia.

Ventilador para ordenador con conector estándar.



Figura 4.83: Ventilador para ordenador con conector estándar. Fuente: Elaboración propia.

Cable de carga rápida USB.



Figura 4.84: Cable de carga rápida USB. Fuente: Elaboración propia.

Mouse y teclado inalámbrico con conector USB.



Figura 4.85: Mouse y teclado inalámbrico con conector USB. Fuente: Elaboración propia.

Cable conector de Pi Camera V2 a Raspberry Pi 3 de 1m de largo.



Figura 4.86: Cable conector de Pi Camera V2 a Raspberry Pi 3. Fuente: Elaboración propia.

En esta sección se describe el armado del prototipo, se recomienda seguir las instrucciones al pie de la letra, así como las indicaciones visuales en cada paso del proceso, en diversos casos será necesario utilizar diversos adhesivos por lo que se especificará su uso de la siguiente manera:

- Si en la unión de piezas está involucrada una pieza de Hardware o cable y una pieza moldeada, se debe utilizar como adhesivo silicón caliente u otro de igual índole, según se indique.
- Si en la unión de piezas está involucrada dos piezas como cables, se debe utilizar como adhesivo cinta aislante de corriente, según se indique.
- Si en la unión de piezas están involucradas dos o más piezas moldeadas.

4.4.3. Plataforma

Se listan los pasos para el armado de la plataforma, siga en orden cada paso que se indica a continuación, cada pieza y sus partes serán llamadas por los códigos que se les asignaron previamente en este manual.

1. Introducir la pestaña B2 de la pieza B dentro de la ranura A1 de la pieza A, como se muestra a continuación.

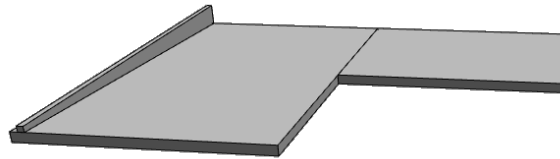


Figura 4.87: Conectar A con B. Fuente: Elaboración propia.

2. Introducir la pestaña C2 de la pieza C dentro de la ranura A2 de la pieza A, según se muestra en la imagen a continuación.

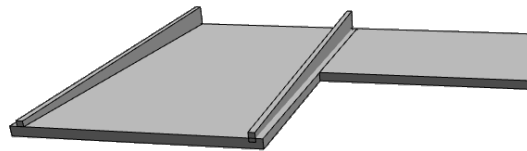


Figura 4.88: Conectar A con C. Fuente: Elaboración propia.

3. Introducir las pestañas B1 y B3 de la pieza B en las ranuras D1 y D3 de la Pieza D respectivamente, igualmente introducir las pestañas C1 y C3 de la pieza C en las ranuras D2, D4 de la pieza D, como se muestra a continuación.

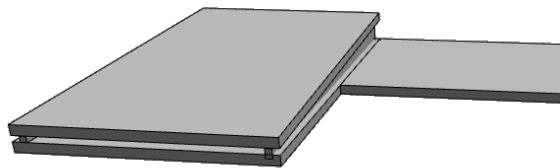


Figura 4.89: Conectar B con D. Fuente: Elaboración propia.

4. Introducir la pestaña E1 de la pieza E en la ranura A3 de la pieza A como se muestra a continuación.

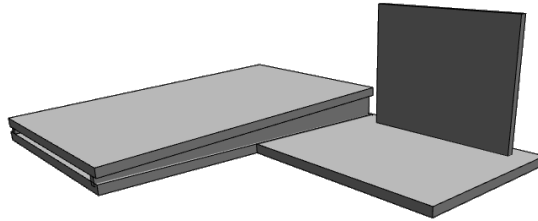


Figura 4.90: Conectar A con E. Fuente: Elaboración propia.

Introducir la pestaña G1 de la pieza G en la ranura F2 de la pieza F como se muestra a continuación.

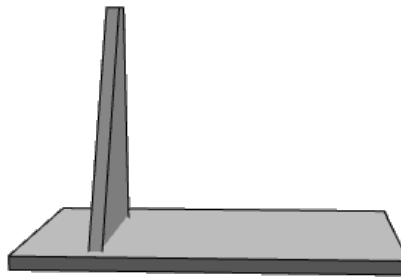


Figura 4.91: Conectar F con G. Fuente: Elaboración propia.

5. Introducir la pestaña H1 de la pieza H en la ranura F1 de la pieza F como se muestra a continuación.

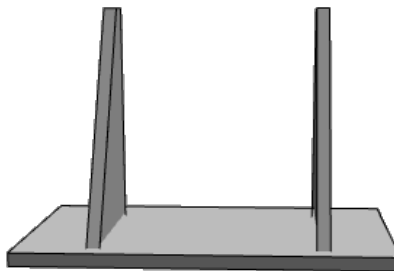


Figura 4.92: Conectar F con H. Fuente: Elaboración propia.

6. Colocar las piezas F, G y H sobre la pieza A de la siguiente manera.

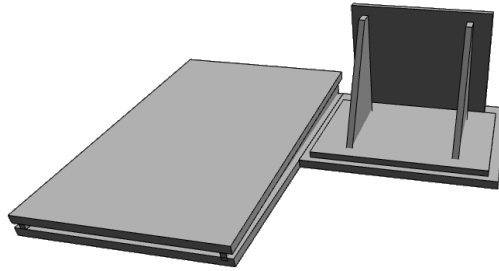


Figura 4.93: Conectar A con F, G y H. Fuente: Elaboración propia.

4.4.4. Soporte de cámara e iluminación

El ensamblado del soporte de la cámara se presenta a continuación, se recomienda seguir el proceso de armado en el orden que se especifica.

1. Tomar la pinza de agarre de la pinza para celular y separar ambas partes, solo se necesita la parte con la rosca de ajuste.



Figura 4.94: Pinza de agarre. Fuente: Elaboración propia.

2. Hacer un corte recto en la pieza por sobre la rosca.



Figura 4.95: Corte pinza. Fuente: Elaboración propia.

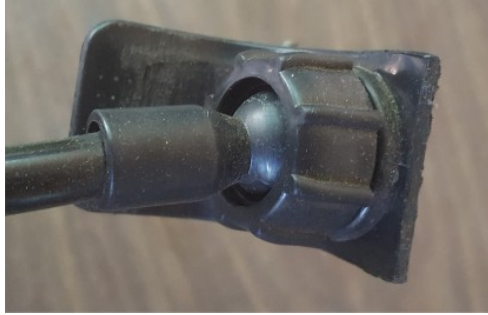


Figura 4.96: Corte pinza. Fuente: Elaboración propia.

3. Pegar una de las tres piezas I con la pieza J, de la manera que se muestra a continuación.

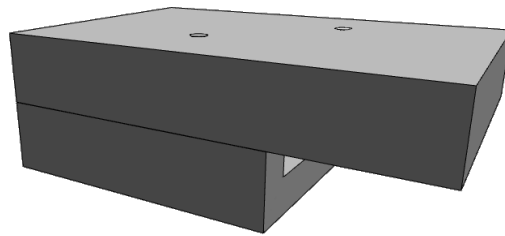


Figura 4.97: Pegar piezas I con J. Fuente: Elaboración propia.

4. Hacer dos perforaciones sobre la parte I de las piezas pegadas, el tamaño de estas perforaciones debe ser el mismo de las tuercas para los tornillos M2X4 y su posición se muestra a continuación.

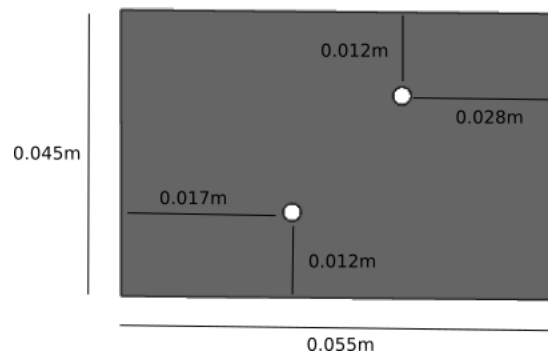


Figura 4.98: Perforaciones para tornillos. Fuente: Elaboración propia.

5. Cortar a la mitad la cinta LED y pegar a los costados de los orificios, de la siguiente manera.



Figura 4.99: Cortar cinta LED. Fuente: Elaboración propia.

6. Pegar las tuercas para tornillos M2X4 en los orificios hechos en el paso cinco a modo que se puedan atornillar con facilidad, como se muestra a continuación.



Figura 4.100: Pegar las tuercas. Fuente: Elaboración propia.

7. Fijar el switch para circuito sobre la pieza J de la siguiente manera.



Figura 4.101: Fijar Switch en J. Fuente: Elaboración propia.

- Tomar el cable UTP y sacar el trenzado de su funda plástica sin dañarla.

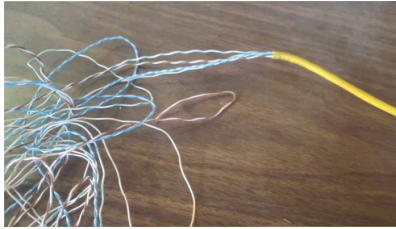


Figura 4.102: Sacar trenzado de cable UTP. Fuente: Elaboración propia.

- Trenzar dos cables de diferente color (uno para corriente positiva y otro para negativa).



Figura 4.103: Trenzar cables. Fuente: Elaboración propia.

- Unir el circuito entre las cintas LED y el switch de los pasos seis y ocho respectivamente. Respetando los colores de los cables trenzados para el tipo de corriente.

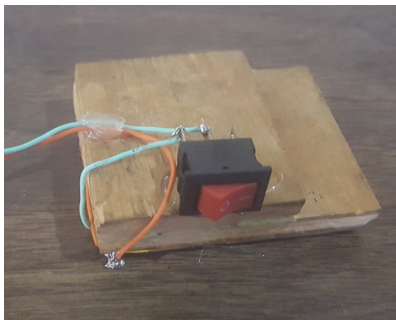


Figura 4.104: Unir circuito cintas LED con switch. Fuente: Elaboración propia.



Figura 4.105: Unir circuito cintas LED con switch. Fuente: Elaboración propia.

11. Introducir el resto del cable (sin cortarlo) en la funda plástica del paso nueve.

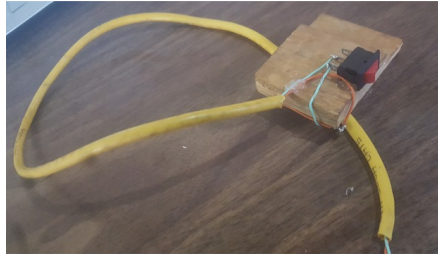


Figura 4.106: Introducir cable en la funda plástica. Fuente: Elaboración propia.

12. Pasar el otro extremo del cable trenzado en la plataforma por la ranura B5 de la pieza B.

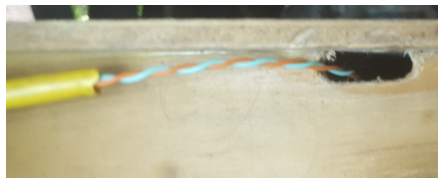


Figura 4.107: Pasar extremo del cable en la ranura de la pieza B.. Fuente: Elaboración propia.

13. Unir el extremo del cable trenzado con el conector de la batería de 9V.



Figura 4.108: Unir extremo del cable con conector de batería de 9V. Fuente: Elaboración propia.

14. Colocar la batería de 9V y tratar de encender la cinta LED, en caso de no poderse, repasar los pasos desde el 9 hasta este y hacer pruebas hasta que funcione adecuadamente.

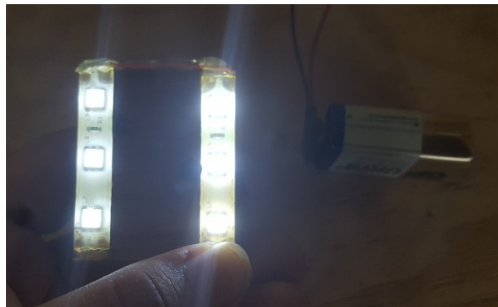


Figura 4.109: Colocar batería y tratar de encender la cinta LED. Fuente: Elaboración propia.

15. Tomar las dos piezas I sobrantes y pegarlas entre sí como se muestra a continuación.

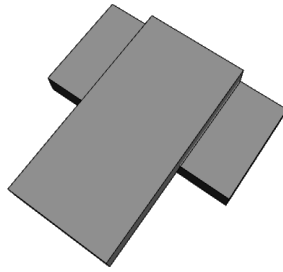


Figura 4.110: Pegar las piezas I sobrantes. Fuente: Elaboración propia.

16. Introducir las piezas I en la ranura B4 de la Pieza B.

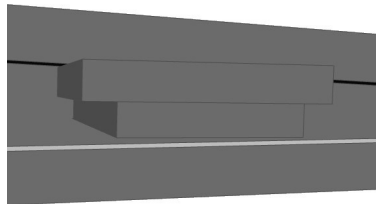


Figura 4.111: Introducir I en la ranura B4. Fuente: Elaboración propia.

17. Sujetar la pinza para celular en las piezas I, como se muestra en la figura.



Figura 4.112: Sujetar pinza en las piezas I. Fuente: Elaboración propia.

18. Introducir la pestaña de la pinza de celular en la hendidura J1 de la pieza J.



Figura 4.113: Introducir la pinza en la hendidura J1. Fuente: Elaboración propia.

A continuación se listan los pasos para ensamblar la iluminación principal del prototipo, se recomienda hacerlo en el orden presentado.

1. Unir las piezas K y S para que funcionen como soporte de todo el armazón, así como se muestra a continuación.

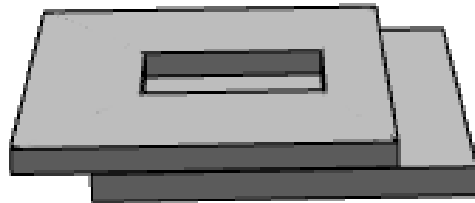


Figura 4.114: Unir piezas K y S. Fuente: Elaboración propia.

2. Unir las dos piezas P y la pieza L, dejando la pieza L en el medio.



Figura 4.115: Unir piezas P y L. Fuente: Elaboración propia.

3. Introducir las piezas L y P en la ranura S1 de la pieza S como se presenta a continuación.

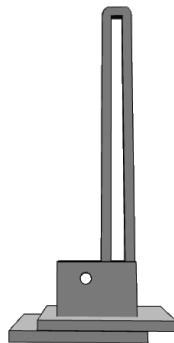


Figura 4.116: Introducir LP en S1. Fuente: Elaboración propia.

4. Atravesar un pasador de madera por los agujeros L1 de ambas piezas L y el agujero M1 de la pieza M, estando la pieza M entre las piezas L como se muestra a continuación.

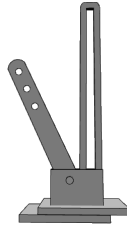


Figura 4.117: Atravesar L1 y M1 con pasadores. Fuente: Elaboración propia.

5. Introducir un pasador de madera por los agujeros N1 de las piezas N y el agujero M4, estando la pieza M entre las piezas N.

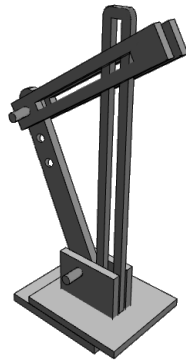


Figura 4.118: Atravesar N1 y M4 con pasadores. Fuente: Elaboración propia.

6. Colocar una arandela de $\frac{1}{4}$ en un perno de $\frac{1}{4}$ y hacerlo pasar a través de las ranuras N3 en ambas piezas N, pasando entre estas por la ranura L1 de la pieza L así como se presenta a continuación.



Figura 4.119: Colocar arandela en L1. Fuente: Elaboración propia.

7. Colocar una arandela en el perno que atraviesa las piezas N y L y ajustar una tuerca de mariposa hasta que quede firme.



Figura 4.120: Colocar arandela entre N y L. Fuente: Elaboración propia.

8. Colocar una arandela de $\frac{1}{4}$ en un perno de $\frac{1}{4}$ y hacerlo pasar a través de las ranuras N3 en ambas piezas N, pasando entre estas por la ranura O1 de la pieza O así como se presenta a continuación.



Figura 4.121: Colocar arandela entre N3 y O1. Fuente: Elaboración propia.

9. Colocar una arandela en el perno que atraviesa las piezas N y O y ajustar una tuerca de mariposa hasta que quede firme.



Figura 4.122: Colocar arandela en el perno entre N y O. Fuente: Elaboración propia.

10. Unir las piezas Q y R como se presenta a continuación.

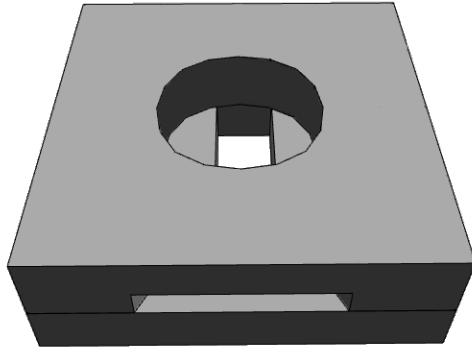


Figura 4.123: Unir Q y R. Fuente: Elaboración propia.

11. Introducir la base del foco GU10 en el agujero Q1 de la pieza Q como en la imagen siguiente.

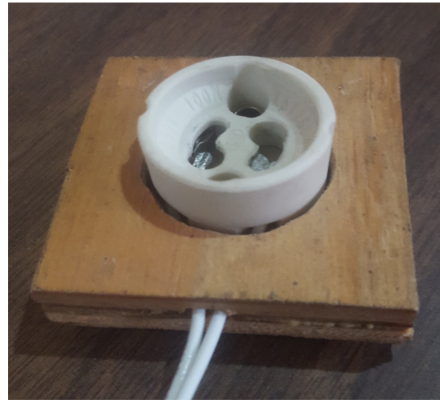


Figura 4.124: Introducir base de foco en el agujero Q1. Fuente: Elaboración propia.

12. Abrir el switch de lámpara y unir uno de los extremos de la base GU10 a una de las puntas del cable duplex.

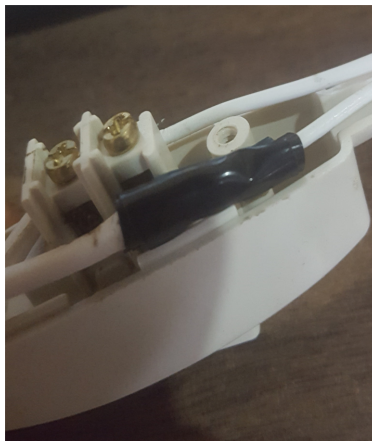


Figura 4.125: Conectar switch a cable duplex. Fuente: Elaboración propia.

13. Atornillar en un conector del switch la otra punta de la base GU10 y en el otro la punta suelta del cable duplex, uniendo el circuito.

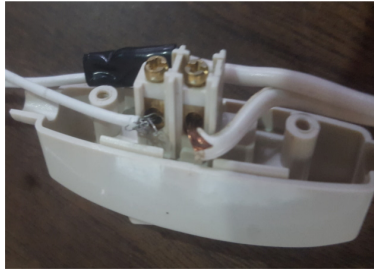


Figura 4.126: Atornillar conector switch. Fuente: Elaboración propia.

14. Atornillar las otras dos puntas del cable duplex a un enchufe de corriente.



Figura 4.127: Atornillar cable duplex al enchufe. Fuente: Elaboración propia.

15. Colocar el foco GU10 en su base y asegurarlo.



Figura 4.128: Colocar foco en su base. Fuente: Elaboración propia.

16. Introducir el extremo recto de la pieza O en la ranura R1 de la pieza R.



Figura 4.129: Introducir pieza O en la ranura R1. Fuente: Elaboración propia.

17. Conectar a la corriente y probar el circuito.



Figura 4.130: Conectar y probar. Fuente: Elaboración propia.

Se presenta el armado de las partes del prototipo que conforman el hardware, para elementos previamente armados, como la pantalla de siete pulgadas y su armazón, se recomienda seguir los manuales de armado que traen de fábrica.

1. Colocar los conectores al protector frontal de la pantalla.

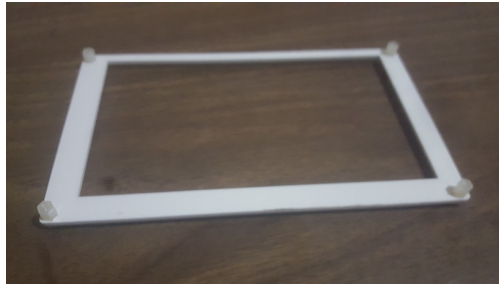


Figura 4.131: Colocar conectores de pantalla. Fuente: Elaboración propia.

2. Fijar la pantalla a su protector frontal.

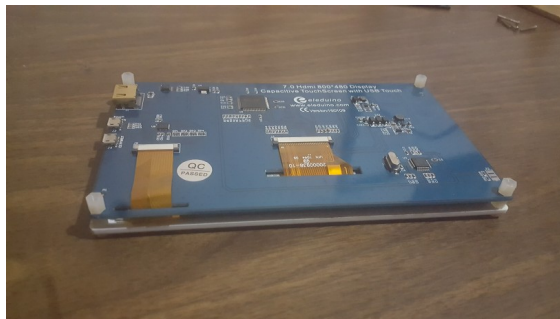


Figura 4.132: Fijar pantalla. Fuente: Elaboración propia.

3. Colocar los tornillos que sirven para sostener la Raspberry Pi en el protector trasero de la pantalla y luego asegurarlo a la pantalla.



Figura 4.133: Colocar tornillos de la Raspberry PI. Fuente: Elaboración propia.

- Colocar sobre el protector inferior de la Raspberry Pi el protector medio inferior.

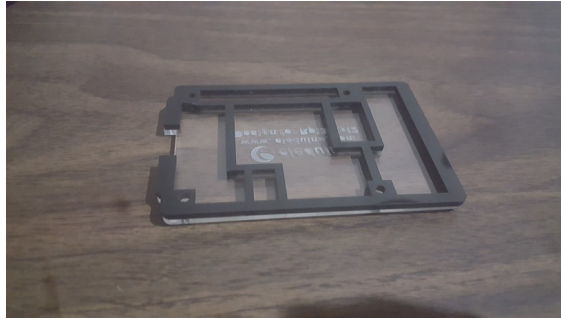


Figura 4.134: Colocar protector Raspberry PI. Fuente: Elaboración propia.

- Colocar la memoria MicroSD con el OS instalado en la Raspberry Pi.



Figura 4.135: Colocar memoria MicroSD en la Raspberry PI. Fuente: Elaboración propia.

- Conectar el cable conector de la cámara a la Raspberry Pi.



Figura 4.136: Conexión cable de cámara. Fuente: Elaboración propia.

7. Colocar el protector medio superior de la Raspberry Pi.

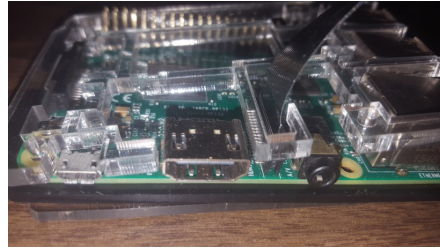


Figura 4.137: Colocar protector medio superior Raspberry PI. Fuente: Elaboración propia.

8. Colocar el protector superior de la Raspberry Pi.

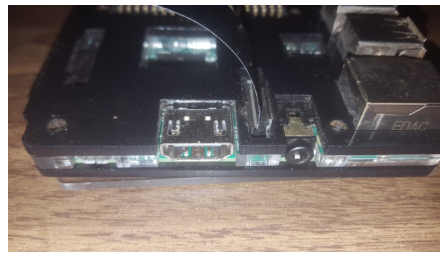


Figura 4.138: Colocar el protector superior de la Raspberry PI. Fuente: Elaboración propia.

9. Fijar la Raspberry Pi y sus protectores en los tornillos del protector trasero de la pantalla.



Figura 4.139: Fijar Raspberry PI. Fuente: Elaboración propia.

10. Conectar el cable HDMI a los puertos HDMI de la Pantalla y de la Raspberry Pi.

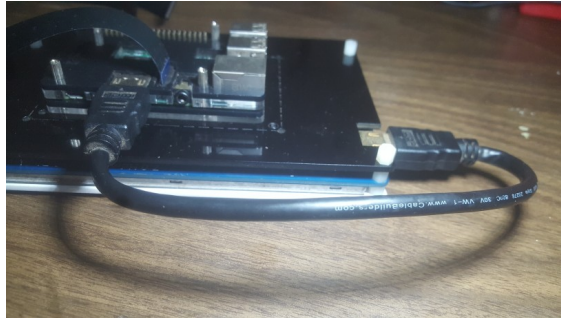


Figura 4.140: Conectar cable HDMI. Fuente: Elaboración propia.

11. Conectar el cable microUSB en uno de los puertos microUSB de la pantalla y en el de la Raspberry Pi.



Figura 4.141: Conectar cable microUSB. Fuente: Elaboración propia.

12. Conectar el receptor USB de señal del teclado y el mouse en el puerto USB de la Raspberry Pi como se especifica a continuación.

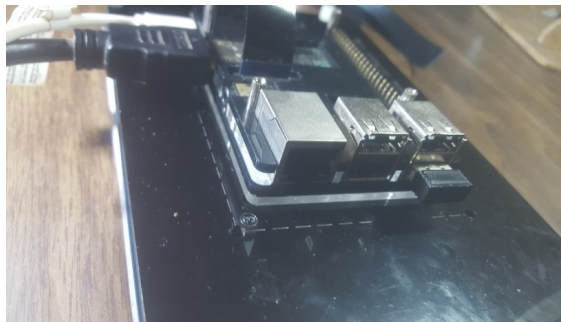


Figura 4.142: Conectar receptor USB. Fuente: Elaboración propia.

13. Conectar el adaptador USB para micrófono y audífono en el puerto USB de la Raspberry Pi como se muestra a continuación.

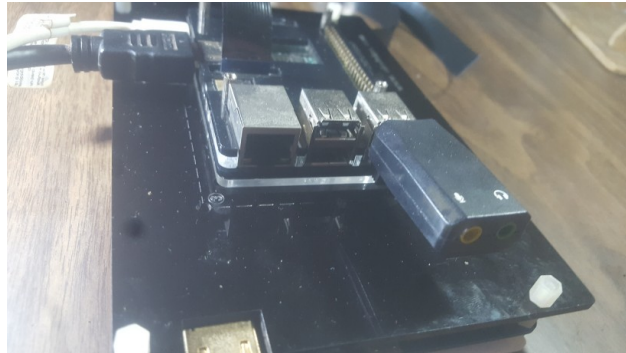


Figura 4.143: Conectar adaptador USB para micrófono/audífono. Fuente: Elaboración propia.

14. Conectar el cable de audio del extremo de 3.5mm en el puerto de micrófono del adaptador.



Figura 4.144: Conectar cable de audio 3.5mm. Fuente: Elaboración propia.

15. Conectar el cable de audio del extremo de 6.5mm en el adaptador de 6.5mm a XLR.



Figura 4.145: Conectar cable de 6.5mm. Fuente: Elaboración propia.

16. Conectar el micrófono al cable de audio desde el puerto XLR.



Figura 4.146: Conectar micrófono al puerto XLR. Fuente: Elaboración propia.

17. Acomodar el micrófono en la base del micrófono.



Figura 4.147: Acomodar micrófono en su base. Fuente: Elaboración propia.

18. Pegar el ventilador en la Pieza E de manera que cubra las ranuras marcadas en la pieza, así como se presenta a continuación.



Figura 4.148: Pegar ventilador. Fuente: Elaboración propia.

19. Conectar el cable USB con salidas de corriente al ventilador como se presenta a continuación.

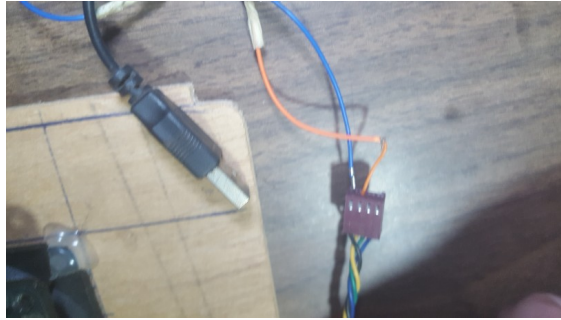


Figura 4.149: Conectar cable USB con salida de corriente al ventilador. Fuente: Elaboración propia.

20. Conectar el ventilador a la Raspberry Pi.

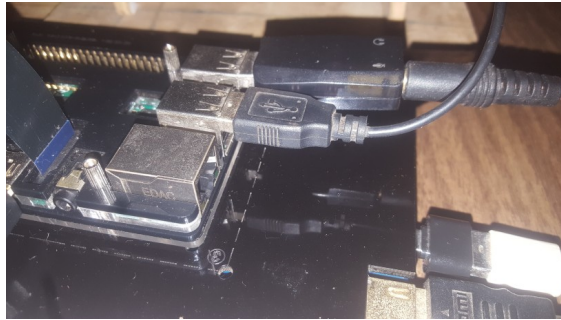


Figura 4.150: Conectar ventilador a la Raspberry PI. Fuente: Elaboración propia.

21. Colocar la pantalla sobre las piezas G y H como se muestra a continuación.



Figura 4.151: Colocar la pantalla sobre las piezas G y H. Fuente: Elaboración propia.

22. Introducir el cable conector de la cámara por ranura C4 de la pieza C y sacarlo por la ranura B5 de la pieza B.



Figura 4.152: Introducir el cable de la cámara por C4 y sacarlo por B5. Fuente: Elaboración propia.

23. Conectar la cámara al otro extremo del cable conector de la cámara.



Figura 4.153: Conectar cámara al cable conector. Fuente: Elaboración propia.

24. Atornillar la cámara en las tuercas M2X4 puestas en la pieza J.

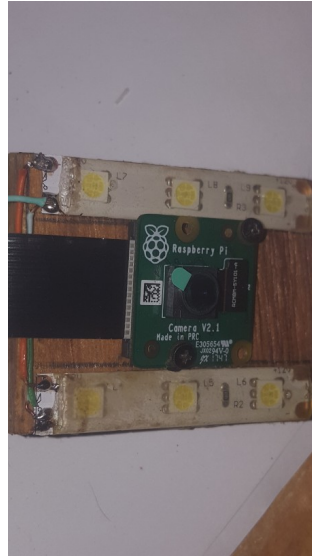


Figura 4.154: Atornillar la cámara. Fuente: Elaboración propia.

25. Conectar el cable de corriente a microUSB a la Raspberry Pi en el puerto para microUSB.



Figura 4.155: Conectar cable de corriente a la Raspberry Pi. Fuente: Elaboración propia.

4.5. Desarrollo de software

4.5.1. Organización de archivos

```

+ prototipo - base.py
            - credentials.json
            - ocr_bing.py
            - ocr_google_drive.py
            - ocr_space.py
            - player.py
            - procesador_imagen.py
            - procesador_tesseract.py
            - record.py
            - regex.py
            - run.py
            - tts_bing.py
            - tts_voicerSS.py
            - web.py
            - hunspell.py
+ audio    + bing
            + festival
            + gtts
            + voicerSS
+ images   + ocr_drive
            + ocr_bing
            + ocr_space
            + ocr_vision
+ menu     - bienvenido.mp3
            - dicte.mp3
            - error_busqueda.mp3
            - error_lectura.mp3
            - espera.mp3
            - introduccion.mp3
            - limite.mp3
            - navegacion.mp3
            - no_comando.mp3
            - no_previa.mp3

```

4.5.2. Adquisición y procesamiento de voz

4.5.2.1. Adquisición

Con el objetivo de grabar el audio para procesarlo y obtener los comandos de voz para el funcionamiento del prototipo se hizo uso de un micrófono en conjunción con el script de grabación en Python (Ver anexo 6.5). El micrófono a utilizado es un Samson Q-Tom, con las siguientes características:

- Micrófono dinámico SPL alto en una carcasa robusta.
- El patrón polar cardioide reduce la retroalimentación
- Respuesta de frecuencia plana y suave
- Conector XLR dorado-plateado
- Samplerate de 44100 Hz

4.5.2.2. Modos de adquisición

Para la entrada de comandos se utilizan fundamentalmente dos formas, mediante línea directa de escucha al micrófono y mediante grabación los cuales se limitan a una duración de 10 segundos.

Captura directa desde el micrófono: En el modo de captura directa desde el micrófono el prototipo requiere la activación por voz para iniciar la captura de comandos, dicha activación por voz puede ser iniciada por las palabras clave “prototipo” o “inicio”.

Teclado: En el modo de captura por teclado, el prototipo requiere que el usuario interactúe con el teclado para iniciar la captura de comandos, para ello debe presionarse la tecla “espacio” o “intro” del teclado numérico.

4.5.2.3. Herramientas a utilizar

Para el procesamiento de la voz se dispone de cinco herramientas, la librería CMUSphinx, la API Cognitive Services Speech to Text, la API IBM Speech to Text, la API Google Speech to Text y la API Wit.AI, el audio será obtenido en la Raspberry PI a través del micrófono conectado utilizando la librería de audio “PyAudio” de Python, debido a las limitaciones impuestas por la API Cognitive Services Speech to Text, la longitud temporal de los archivos de audio deben ser de 10 segundos o menos, el script se encargará de controlar este tiempo y almacenar el audio grabado en un archivo el cual su codificación sea transformable al utilizado por las tres herramientas, las características del audio de salida proporcionado por el micrófono son las siguientes:

Codec: WAV, Canales: 1, Ratio: 41.1KHz, Bits de profundidad: 16 Bits

Y las características necesarias para el funcionamiento de las herramientas es:

| Herramienta | Codec | Canales | Ratio | Bits de profundidad |
|---------------------------|--------------|---------|------------|---------------------|
| CMU Sphinx | WAV | 1 | 8kHz/16kHz | 16 |
| Wit.ai API | WAV | 1 | 8kHz/16kHz | 16 |
| Cognitive Services API | WAV/OGG/FLAC | 1-2 | 8kHz/16kHz | 16 |
| IBM Speech-to-Text API | WAV/OGG/FLAC | 1-9 | 8kHz/16kHz | 16 |
| Google Speech-to-Text API | WAV/OGG/FLAC | 1-9 | 8kHz/16kHz | 16 |

Tabla 4.32: Características de audio requeridas por las herramientas

4.5.2.4. Preprocesamiento

Remuestreo: Para poder utilizar entonces el audio se convertirá al formato compatible con todas las herramientas, haciendo uso de la utilidad “sox” a través de CLI que se encargará de volver a almacenar el audio en el

muestreo correcto.

Se realizará el proceso de remuestreo debido a que utilizando el Ratio de kHz correcto las herramientas funcionan mejor, en mayor medida la librería CMUSphinx, sumado a ello, el remuestreo reduce considerablemente el tamaño de los archivos de audio lo cual agiliza los procesos que involucren subida de archivos a una API.

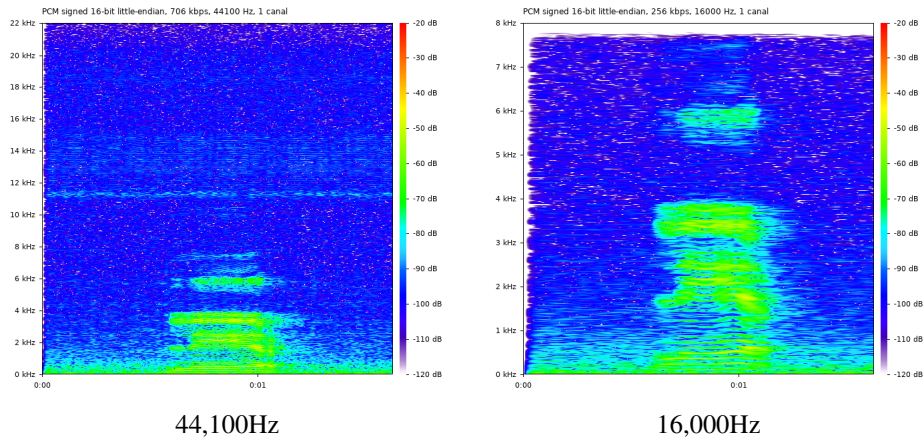


Figura 4.156: Ejemplo de remuestreo de audio de 44,1kHz a 16kHz. Fuente: elaboración propia.

Reducción de ruido: El proceso de reducción de ruido permite un mejor reconocimiento de voz en herramientas que utilizan API, pero reduce considerablemente la precisión para la herramienta CMU Sphinx, con lo cual este proceso será utilizado únicamente en todas las demás herramientas exceptuando CMU Sphinx, este proceso además reduce el tamaño de los archivos de audio.

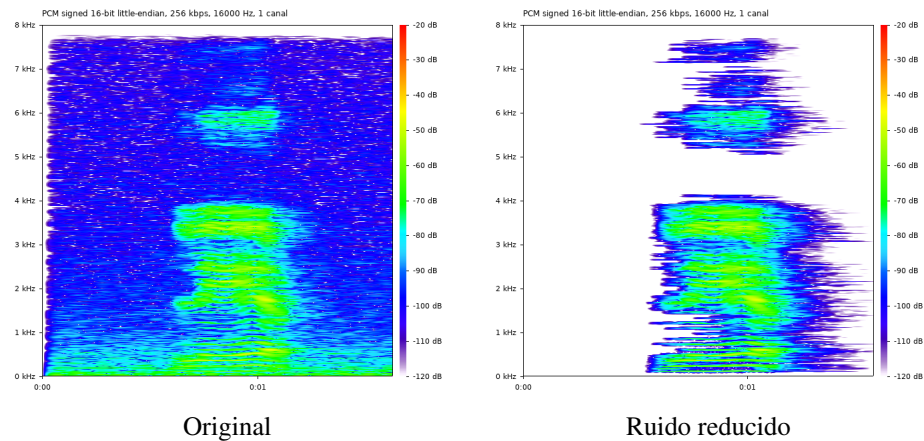


Figura 4.157: Ejemplo de supresión de ruido. Fuente: elaboración propia.

Ajuste de ruido ambiental: Utilizando la librería “SpeechRecognition” se realizará un ajuste dinámico del

umbral de energía para tener en cuenta el ruido ambiente, esto para determinar el inicio y final del dictado de voz.

4.5.2.5. Selección de los comandos

Debido a que CMUSphinx realiza sus procesos de reconocimiento de voz a través de modelos acústicos y de lenguaje basado en fonemas individuales, y estos están basados en determinados diccionarios de palabras de cada idioma, al realizar la traducción de voz a texto es posible que varias combinaciones de fonemas concuerden con una secuencia acústica o también que dichos fonemas no existan como una palabra en el diccionario de lenguaje, ya que utilizaremos la librería “SpeechRecognition”, la cual es un “wrapper” o contenedor que nos permite simplificar el acceso a diferentes librerías y API’s utilizaremos una de sus opciones de configuración al momento de realizar el reconocimiento de voz, llamada ”keyword_entries”, la cual nos permite asociar una determinada palabra a una secuencia acústica definiendo un porcentaje de confiabilidad a dicha palabra, por ejemplo:

```
salida_texto =
↳ speech_recognition.recognize_sphinx(audio_data=audio_de_entrada,
    language='es-ES',
    keyword_entries=[('palabra', 1.0)])
```

Lo anterior se encarga de verificar mediante el modelo acústico si la palabra seleccionada “existe” en el audio a procesar basado en la confiabilidad, esto mejora en gran medida el reconocimiento de palabras o frases determinadas.

Por ejemplo, el siguiente espectro de audio que contiene la palabra “leer”:

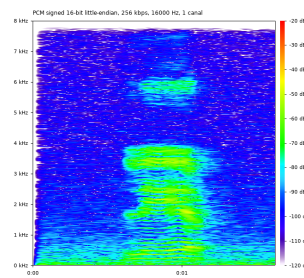


Figura 4.158: Espectrograma de la palabra "leer"

Al procesar dicho audio utilizando únicamente la función de reconocimiento de CMUSphinx:

```
salida_texto = speech_recognition.recognize_sphinx(audio_data=audio,
    language='es-ES')
```

Retorna la frase: “le error”, si se procesa de la misma manera, pero utilizando keyword_entries:


```
salida_texto = speech_recognition.recognize_sphinx(audio_data=audio,
          language='es-ES',
          keyword_entries=[('leer', 1.0)])
```

Retorna la palabra: “leer”, lo cual es correcto.

Los comandos definidos para el prototipo son los siguientes:

Lectura/Leer: Al detectar este comando el script principal ejecutará las funciones requeridas para la lectura de una página física, sea un libro o documento impreso a través de la cámara conectada al prototipo.

Libro <número de página>: Al detectar este comando el script principal ejecutará las funciones requeridas para la lectura de una página digital de un archivo PDF previamente definido, sea un libro, revista electrónica, etc.

Buscar <término o términos de búsqueda>: Al detectar este comando el script principal ejecutará las funciones para realizar una búsqueda en internet utilizando el motor de búsqueda definido al iniciar la aplicación, los posibles buscadores son: Google, Bing y DuckDuckGo y presentará los resultados de la misma, los resultados estarán numerados del número 1 al 10 y en algunas ocasiones de 1 al 9 en el caso del buscador DuckDuckGo.

Navegación avanzar: Al detectar este comando el script principal intentará cargar la siguiente lista de resultados de navegación, si no se ha efectuado ninguna búsqueda previa notificara un mensaje señalando lo anterior.

Navegación regresar: Al detectar este comando el script principal intentará cargar la anterior lista de resultados de navegación, en caso de que la primera lista de resultados de búsqueda ya haya sido cargada y se intente ejecutar este comando se notificará un mensaje señalando lo anterior.

Resultado <número de 1 a 9/10>: Al detectar este comando el script principal buscara si existen búsquedas previas, en caso de existir una búsqueda entonces cargará el recurso web al que esté asociado el número del resultado, es decir cargará la página web asociada a dicho resultado y realizará el scraping del contenido de la misma.

Repetir listado: Al detectar este comando el script principal buscara si existe una búsqueda previa, en caso de existir repetirá nuevamente la lista de resultados, caso contrario informará que no existen resultados previos.

Repetir texto: Al detectar este comando el script principal buscara si existe un recurso web previamente cargado o una lectura física previamente realizada, si existe repetirá el texto de dichos resultados, caso contrario informará que no existe texto previo.

Salir: Al detectar este comando el script principal se cerrará.

4.5.2.6. Diagramas de los comandos

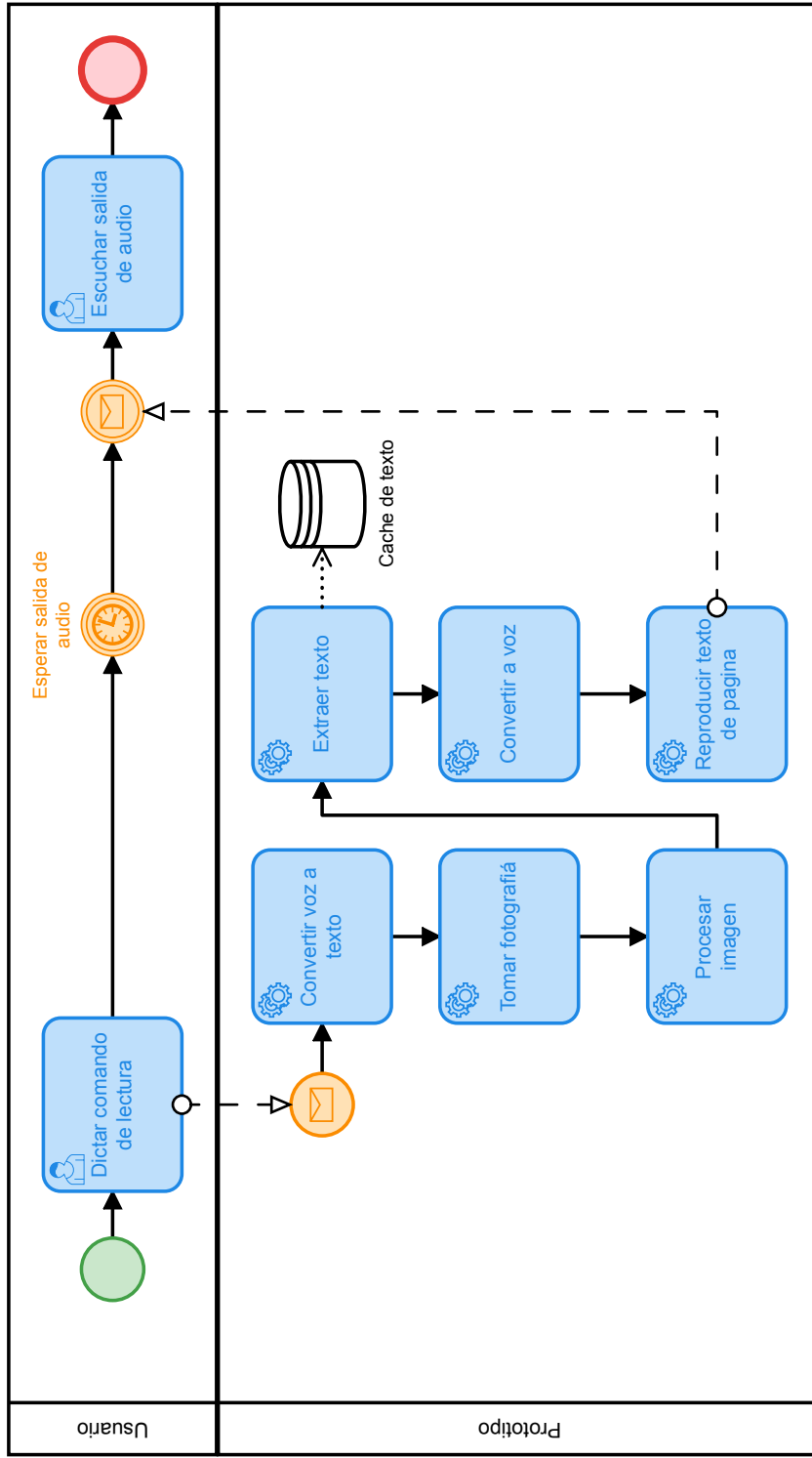


Figura 4.159: Diagrama comando leer. Fuente: elaboración propia.

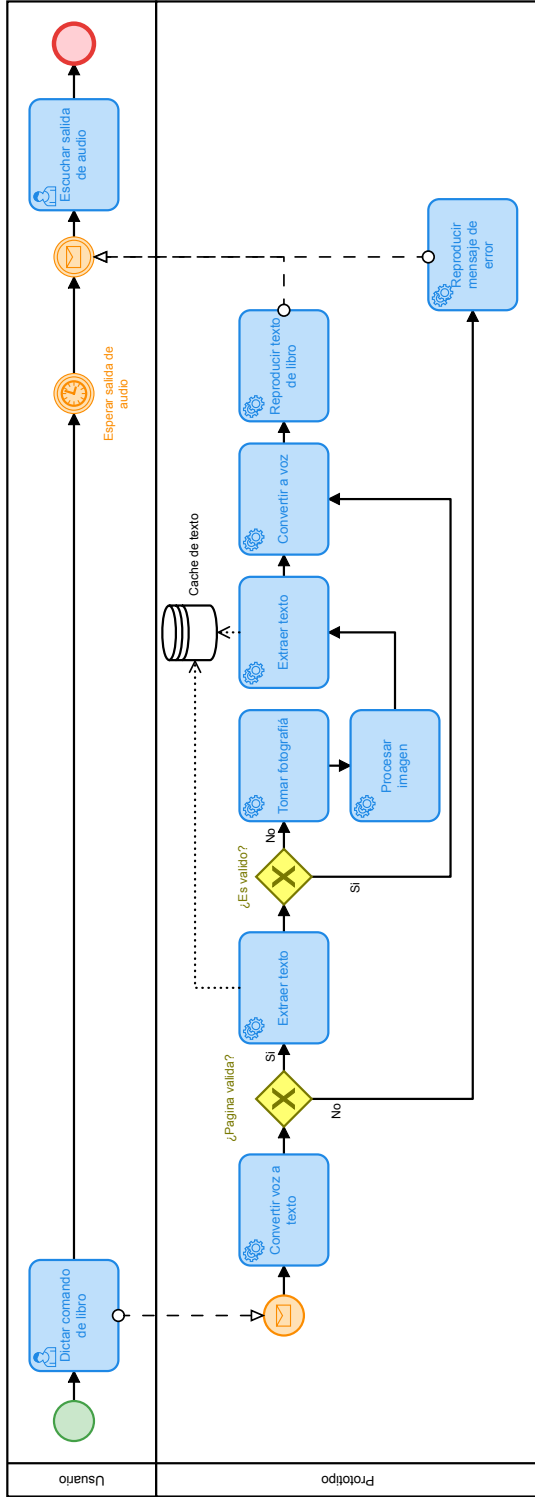


Figura 4.160: Diagrama comando libro. Fuente: elaboración propia.

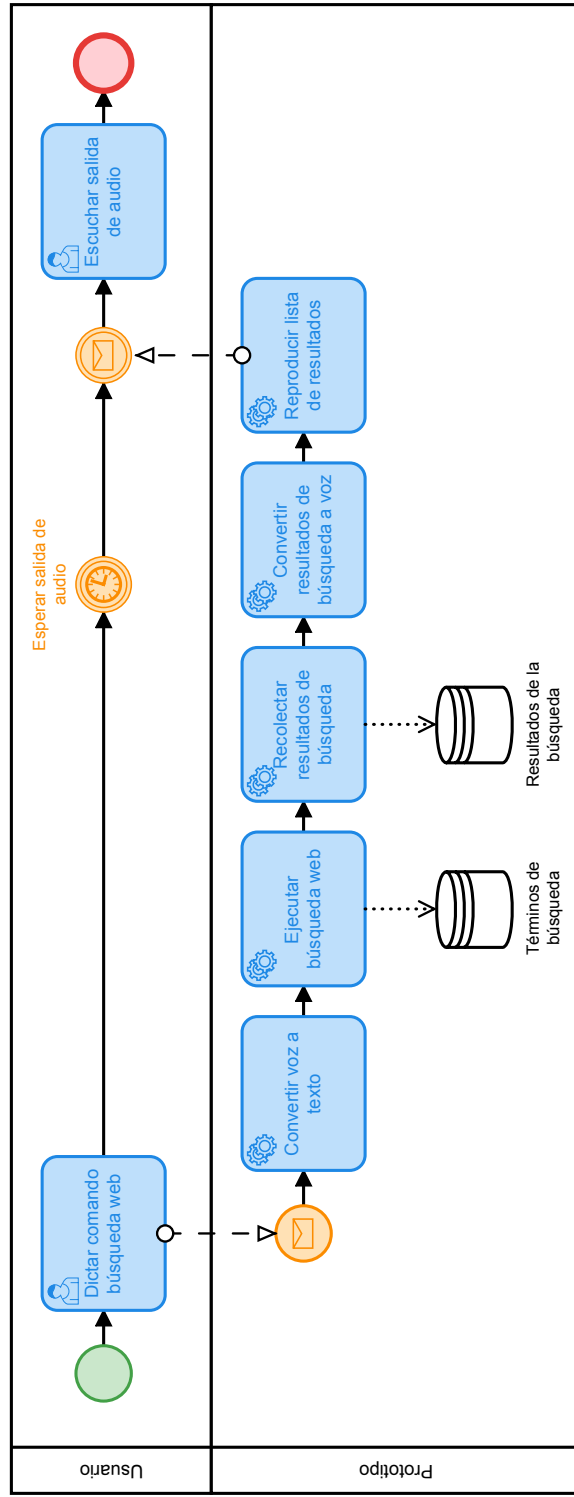


Figura 4.161: Diagrama comando buscar en línea. Fuente: elaboración propia.

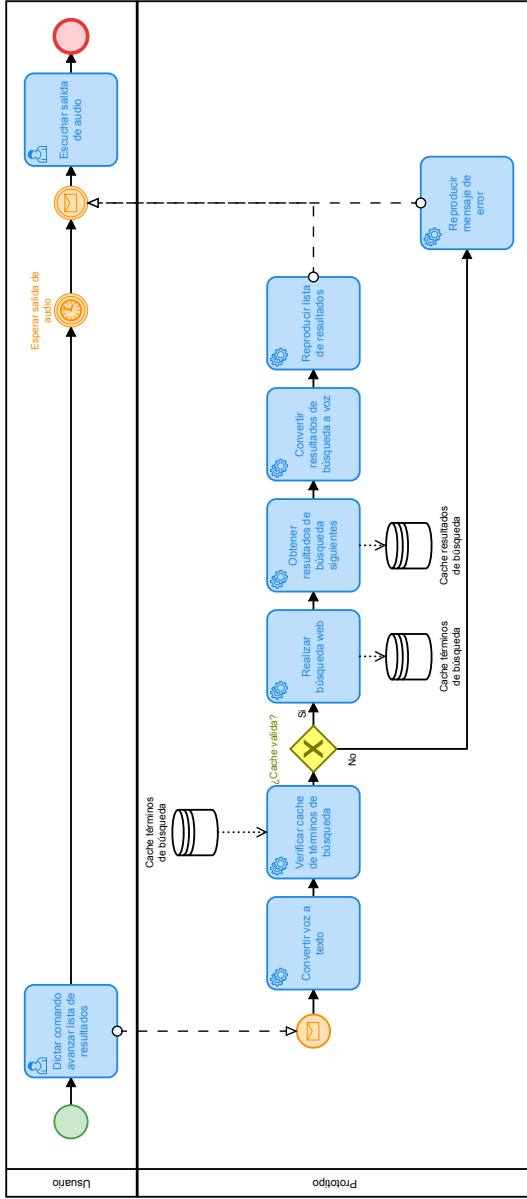


Figura 4.162: Diagrama comando navegación avanzar. Fuente: elaboración propia.

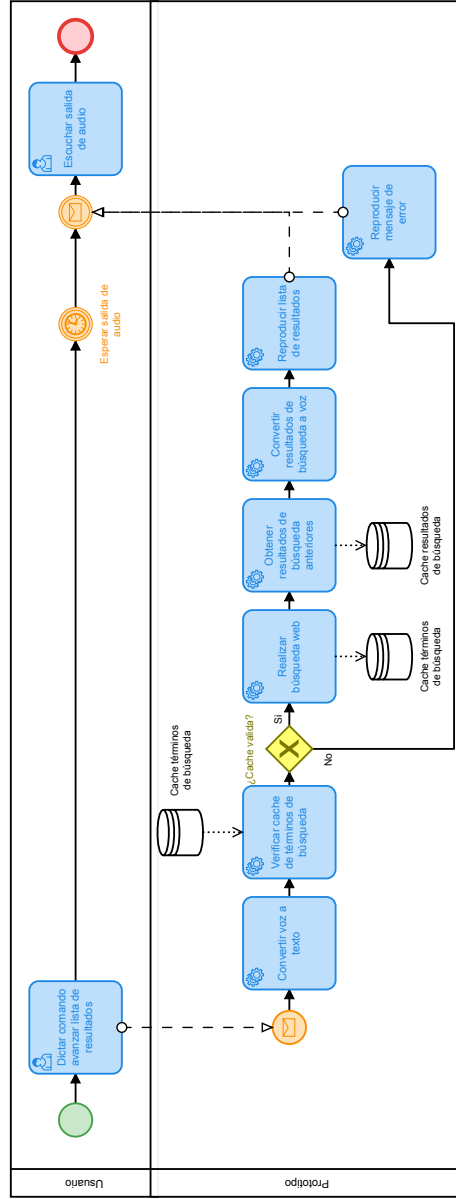


Figura 4.163: Diagrama comando navegación regresar. Fuente: elaboración propia.

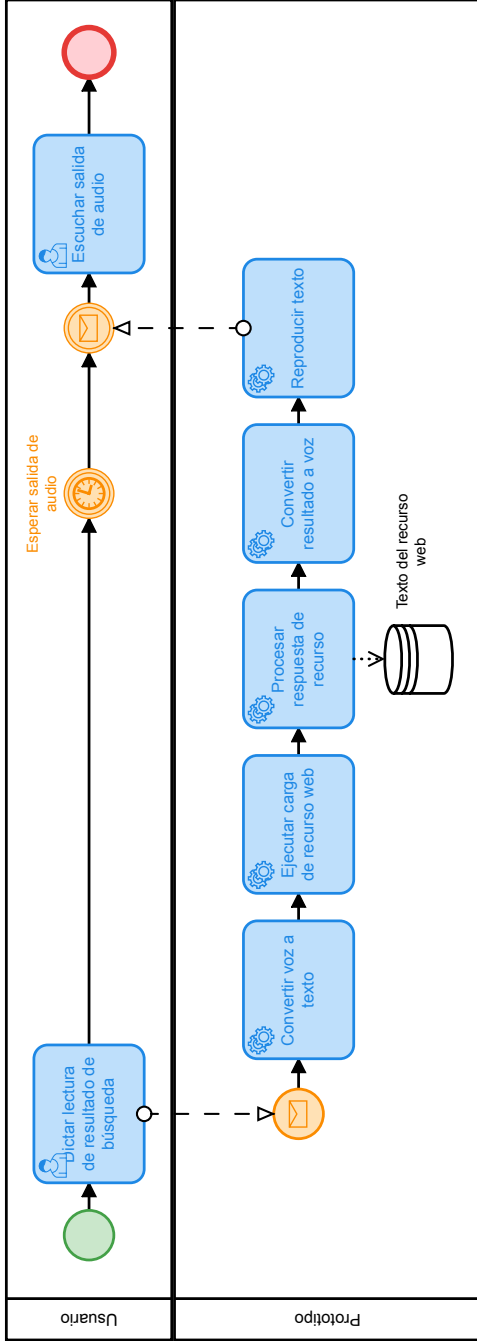


Figura 4.164: Diagrama comando resultado. Fuente: elaboración propia.

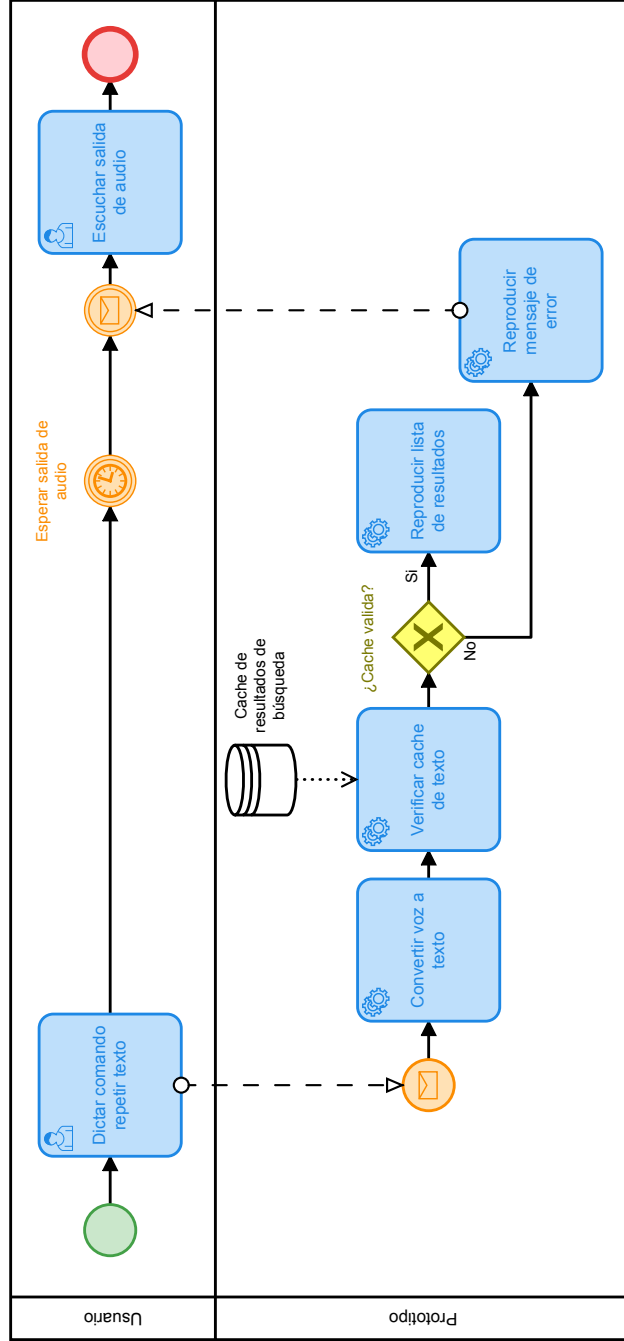


Figura 4.165: Diagrama comando repetir lista. Fuente: elaboración propia.

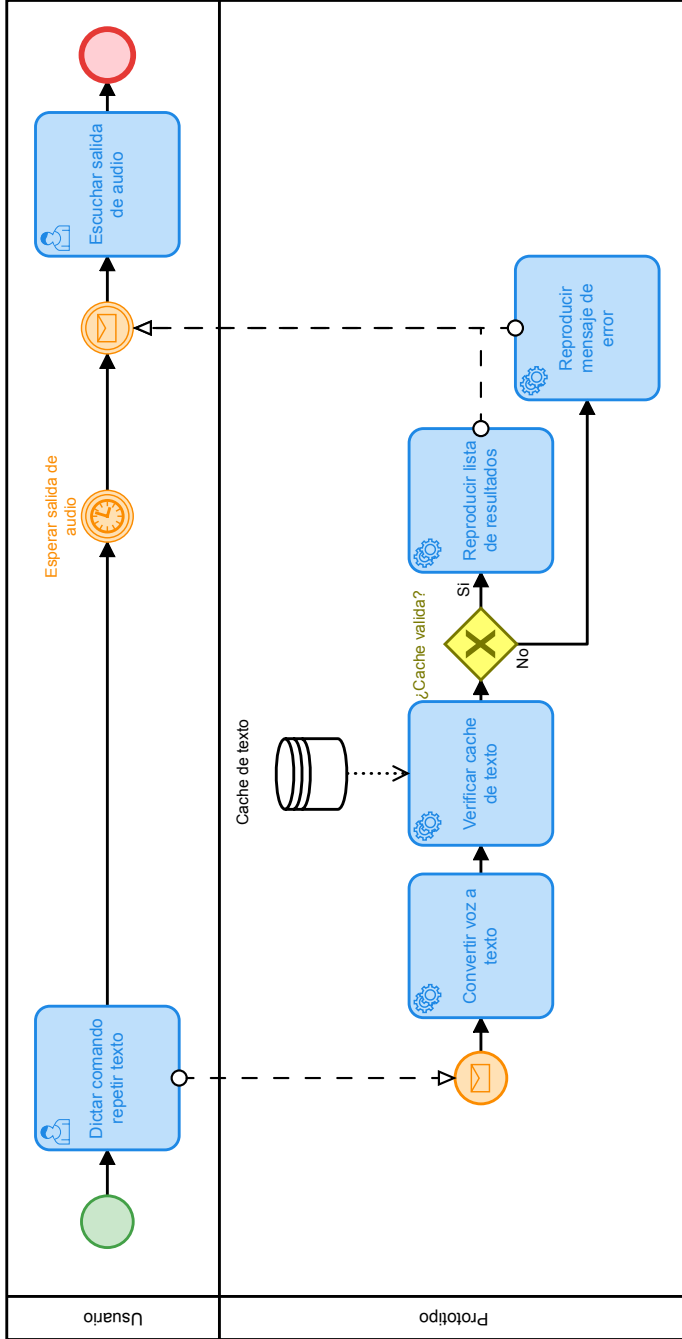


Figura 4.166: Diagrama comando repetir texto. Fuente: elaboración propia.

4.5.2.7. Procesamiento

Una vez que el audio a procesar ha sido transformado al formato requerido es necesario pasarla a través de la herramienta de reconocimiento de voz deseada, cada una tiene su manera diferente de funcionar, a continuación, se detalla la manera en la que se efectúa dicho procesamiento, dichos procesos son idénticos tanto si se utiliza audio grabado, como línea directa del micrófono:

Para el procesamiento de los comandos se plantea el uso de cinco modos de funcionamiento acorde a las herramientas disponibles: CMUSphinx, Cognitive Services, IBM, Google, Wit.ai e híbrido (juntando CMUSphinx y Wit.ai). Dado que CMUSphinx trabaja directamente en la Raspberry PI y además utiliza un diccionario modificado para reducir aún más el tiempo de respuesta, es relativamente rápido para procesar comandos individuales y concisos tales como “lectura, repetir listado, repetir texto, resultado <número>, navegación avanzar y navegación regresar”, ya que no presentan varianza pero puede presentar problemas si por ejemplo se dicta un comando de “buscar” y uno o varios términos de dicha búsqueda no existen en el diccionario ni en los modelos de CMUSphinx lo cual llevará a que la librería retorne el término más similar o cercano “acústicamente” a lo que se dictó, por otro lado las API’s Cognitive Services, IBM, Google y Wit.ai al ejecutarse mediante el uso de consultas HTTP requiere que el archivo de audio sea enviado a la API lo cual consume tiempo agregado al procesamiento necesario por parte de dicha API, pero estas al contar con diccionarios mucho más amplios, variados y actualizados que los proporcionados por CMUSphinx su respuesta será por tanto mucho más precisa y rápida.

Sphinx: Todos los comandos se procesarán a través de CMUSphinx, (esta opción es más rápida que todas las demás si el equipo en el que se ejecuta el código es más potente), pero puede producir errores al momento de realizar búsquedas.

Wit.ai: Todos los comandos se procesarán a través de Wit.ai API, más rápido que CMUSphinx y mucho más acertado, sobre todo al realizar búsquedas.

Cognitive Services: Todos los comandos se procesarán a través de Cognitive Services Speech to Text API, más rápido que CMUSphinx y mucho más acertado, sobre todo al realizar búsquedas.

IBM: Todos los comandos se procesarán a través de IBM Speech to Text API, más rápido que la de Cognitive Services, pero mucho más acertado que CMUSphinx, sobre todo al realizar búsquedas.

Google: Todos los comandos se procesarán a través de Google Speech to Text API, más rápido que todas las

demás herramientas además mucho más acertado que CMUSphinx, sobre todo al realizar búsquedas.

Híbrido mínimo: Los comandos se procesan a través de CMUSphinx, y si el reconocimiento falla o si se detecta que el comando dictado es una búsqueda esta se vuelve a procesar a través de Wit.ai API para conseguir el mejor resultado, mayor velocidad en comandos definidos y más lento al realizar las búsquedas.

CMUSphinx: A través de la librería SpeechRecognition utilizando las siguientes instrucciones:

Para reconocer comandos concretos a partir de palabras claves:

```
texto_salida = speech_recognition.recognize_sphinx(audio_data=audio,
    language='es-ES_hotwords',
    keyword_entries=[('comando', 1.0)])
```

Para reconocer texto como tal:

```
texto_salida = speech_recognition.recognize_sphinx(audio_data=audio,
    language='es-ES')
```

De las dos líneas anteriores podemos detallar lo siguiente

- **recognize_sphinx:** Le ordenamos a la librería “SpeechRecognition” que reconozca el audio utilizando la herramienta CMUSphinx.
- **audio_data=audio:** El contenido binario del archivo de audio.
- **language='es-ES_hotwords'/'es-ES':** Definimos que la librería CMUSphinx utilice el idioma español (sus modelos acústicos, de datos y diccionario) para reconocer el texto del audio.
- **keyword_entries=[('comando', 1.0)]:** Un listado con las palabras claves a buscar utilizando construcciones acústicas, cada palabra o frase ira entre paréntesis y se definirá un nivel de confianza para procesar dicha palabra o frase, siendo 1.0 como fiabilidad total, y 0.0 como fiabilidad nula, utilizando la versión modificada de diccionario para mayor velocidad.

Cognitive Services Speech to Text API: A través de la librería SpeechRecognition utilizando la instrucción:

```
texto_salida = speech_recognition.recognize_bing(audio_data=audio,
    language='es-ES',
    key='API KEY',
    show_all=True)
```

De la línea anterior podemos detallar lo siguiente:

- **Recognize_bing:** Le ordenamos a la librería “SpeechRecognition” que reconozca el audio utilizando la API

Cognitive Services.

- **audio_data=audio:** El contenido binario del archivo de audio.
- **language='es-ES':** Definimos que la Cognitive Services Speech to Text API utilice el idioma español para reconocer el texto del audio.
- **key='API KEY':** Definimos la clave API a utilizar.
- **show_all=True:** Puede ser verdadero o falso (True / False), por defecto es falso, si se define como verdadero la respuesta de texto mostrará los múltiples resultados posibles y su grado de fiabilidad.

IBM Speech to Text API: Esta API tiene su propia librería de desarrollo para Python llamada “Watson Developer Cloud Python SDK”, la cual brinda acceso a todas las capacidades de procesamiento necesarias:

```
texto_salida = ibm_speech_recognition.recognize(audio=audio,
        content_type='audio/wav',
        model='es-ES_BroadbandModel')
```

De las líneas anteriores podemos detallar lo siguiente:

- **audio=audio:** El contenido binario del archivo de audio.
- **content_type:** El tipo de audio que se envía, en este caso WAV.
- **model='es-ES_BroadbandModel':** Definimos que la IBM Speech to Text API utilice el modelo asociado al idioma español para reconocer el texto del audio.

4.5.3. Adquisición y procesamiento de imágenes

Con el objetivo de obtener la mejor calidad de imagen para el procesamiento del texto del contenido en formato físico se construyó el prototipo compuesto por: una cámara, la iluminación, la Raspberry PI y el software.

La cámara se ubicó en la parte superior del brazo mediante el uso de un brazo flexible captando las imágenes desde arriba, para obtener las fotografías, en formato JPEG, RGB de dimensiones 2464x3280.

La cámara a utilizar necesita una resolución adecuada, que cumpla con las necesidades de densidad de píxeles para que las herramientas OCR funcionen, en este caso se utilizó el módulo de cámara oficial de Raspberry PI en su revisión número dos.

Lo ideal es tener una iluminación difusa que libere a la imagen capturada de sombras. Para que la luz ambiental no afecte al procesamiento de imágenes, se construyó un módulo de iluminación.

El funcionamiento es básicamente el siguiente: Se coloca el libro, revista o documento físico a procesar en su lugar designado, en la parte superior de este la cámara tomará la imagen de dicho documento, estas imágenes serán recogidas por la Raspberry PI mediante un cable de conexión a la Raspberry PI para su procesamiento.

4.5.3.1. Herramientas a utilizar

Para el preprocesamiento debido a que las imágenes requieren determinados procesamientos para mejorar la tasa de reconocimiento de caracteres se hará uso de diferentes librerías para realizar dichos procesos, las cuales son “OpenCV” y “PIL” también llamada “Pillow”.

Para el procesamiento OCR de las imágenes utilizaremos tres herramientas, Tesseract-OCR, OCR.Space API, Azure Computer Vision API y Google Drive OCR API, las imágenes serán obtenidas a través de la cámara conectada a la Raspberry PI utilizando la librería “OpenCV”.

4.5.3.2. Preprocesamiento

Para poder ejecutar el OCR con las herramientas Tesseract-OCR, OCR.Space API, Azure Computer Vision API y Google Drive API en la imagen obtenida previamente, primero deben hacerse una serie de tratamientos de mejora para realzar las características de interés en la misma, en este caso los caracteres contenidos en la misma; los procesos de mejora dependen de la herramienta en sí, dichos procesos se detallan en la siguiente tabla:

| Tesseract OCR | OCR Space API Computer Vision API Google Drive OCR API |
|--|--|
| Transformar a gris | Transformar a gris |
| Binarizado | Corregir ángulo de rotación |
| Marcado de caracteres | Conversión a formato JPG |
| Corregir ángulo de rotación | |
| Limpieza de bordes y sombras | |
| Segunda detección de ángulo de rotación y ajuste | |
| Alargar a x2 cúbico | |
| Segmentar áreas | |
| Demarcar áreas mínimas | |
| Ordenar segmentos | |

Tabla 4.33: Proceso de mejora de imagen

A continuación, se detalla el proceso de mejora para las herramientas, se utilizará la misma imagen de referencia para ejemplificar dicho proceso:

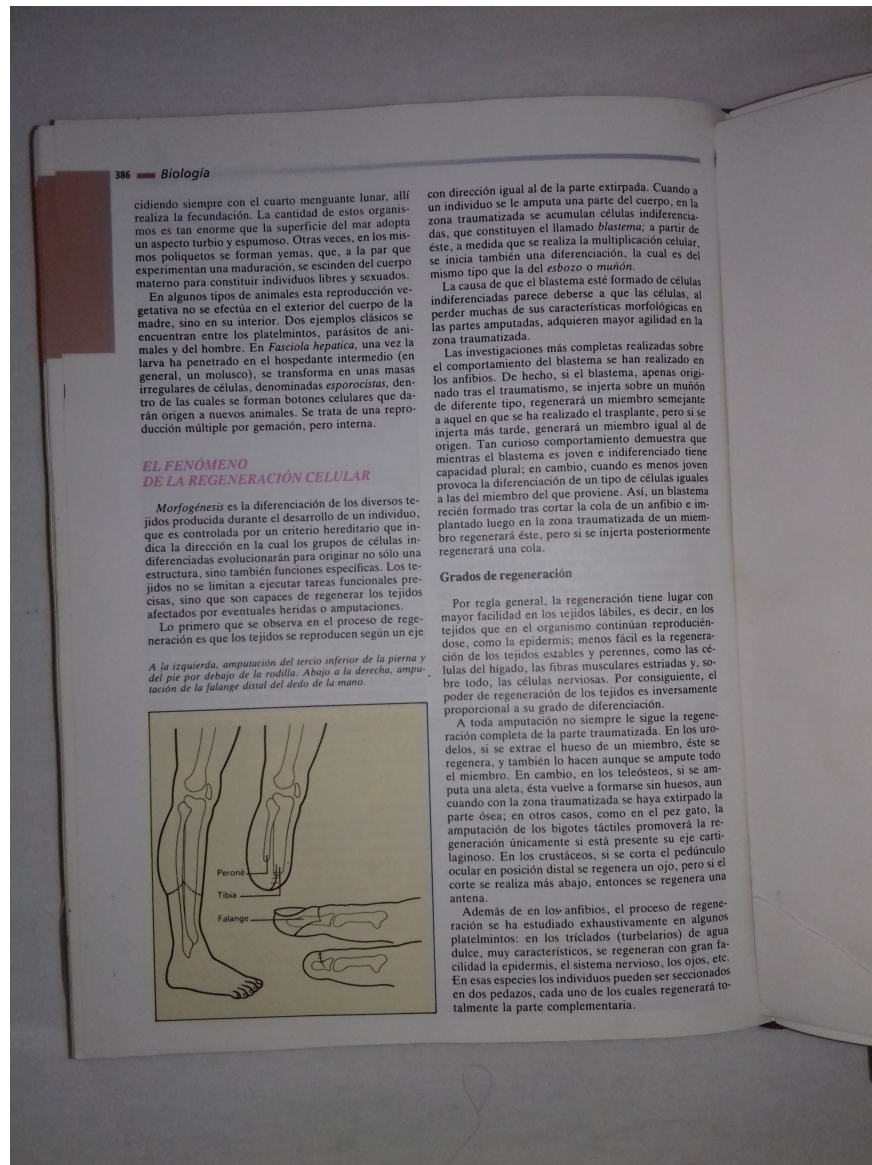


Figura 4.167: Imagen original - Resolución: 2464x3280 - Formato: JPEG - Profundidad de Bits: 8 - Tamaño en disco: 1.57 MB

4.5.3.2.1. Preprocesado para Tesseract-OCR

Transformar a gris: Esto con fin de mejorar el resultado del siguiente proceso, la reducción de ruido y sobre todo la binarización:

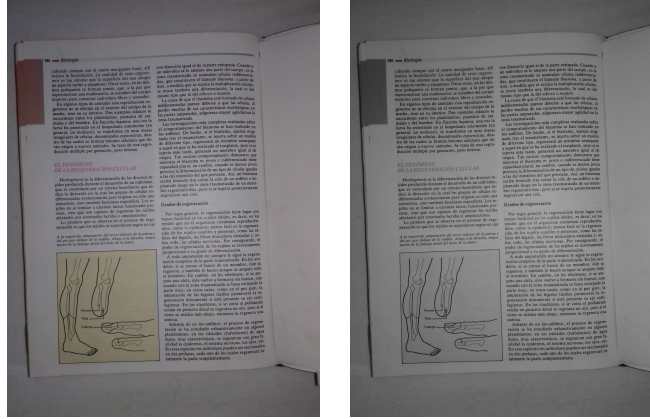


Imagen original

Conversión a gris

Figura 4.168: Conversión de imagen a gris. Fuente: elaboración propia.

Binarizado: Esto convertirá la imagen de escala de grises a blanco y negro, se utilizará el algoritmo de binarización “MEAN” o más cercano, ya que permite una salida más limpia ya que escoge automáticamente el umbral para una correcta binarización:

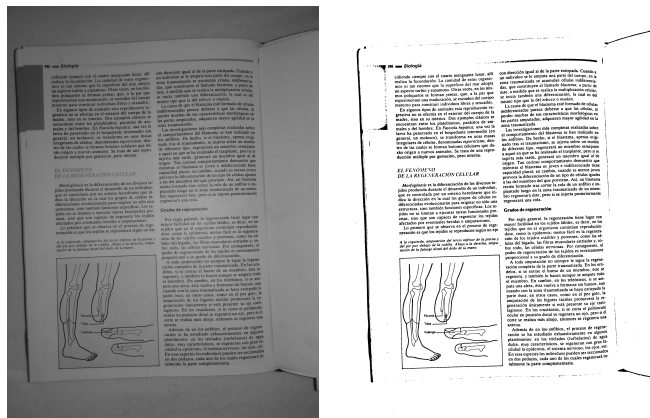
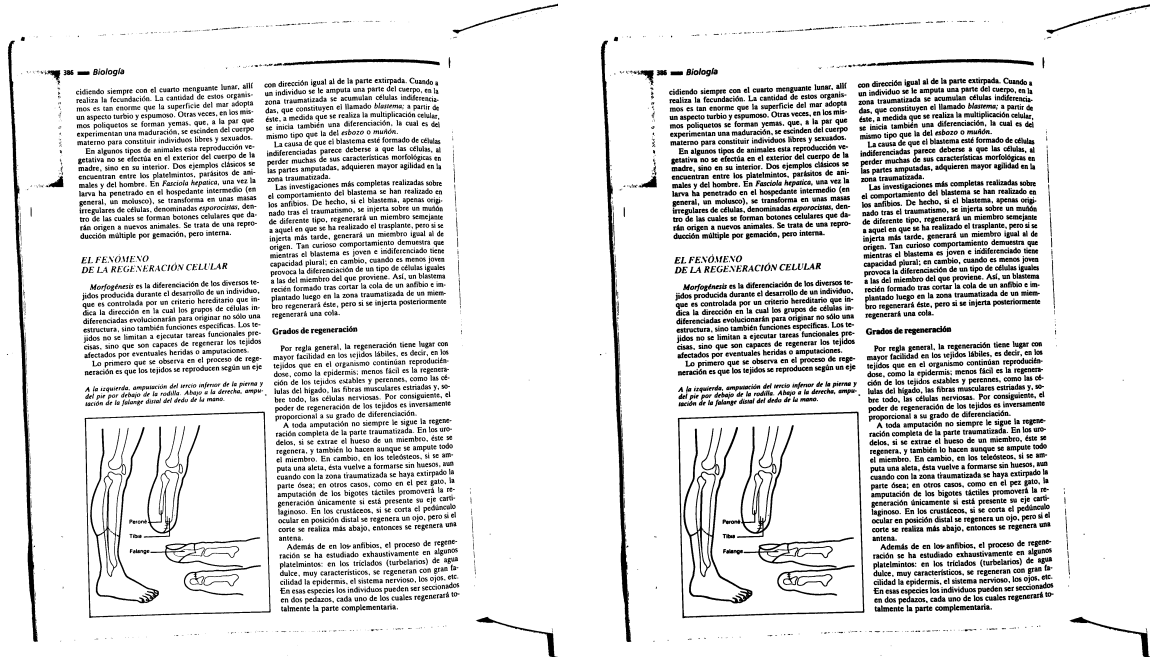


Imagen en gris

Conversión a imagen binaria

Figura 4.169: Binarización. Fuente: elaboración propia.

Marcado de caracteres: Al resultado anterior se le aplicará un marcado de caracteres, esto con el fin de conectar posibles caracteres rotos y disminuir aún más la aparición de falsos positivos o resultados erróneos:



idiendo siempre con el cuarto menguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mixtos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuados.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en una masa irregular de células, denominadas *esporocistas*, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

Imagen binaria

idiendo siempre con el cuarto menguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mixtos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuados.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en una masa irregular de células, denominadas *esporocistas*, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

Marcado de caracteres

Figura 4.170: Marcado de caracteres. Fuente: elaboración propia.

Corrección de ángulo: Si la imagen a procesar tiene una inclinación pronunciada, la herramienta utilizada

para realizar el proceso de extracción de texto presentara muchos errores o simplemente fallará, por tanto, es necesario hacer un ajuste de la orientación de la página, esto únicamente se realizará si la rotación de la imagen es grande, para rotaciones de imagen pequeñas el resultado no cambia mucho:

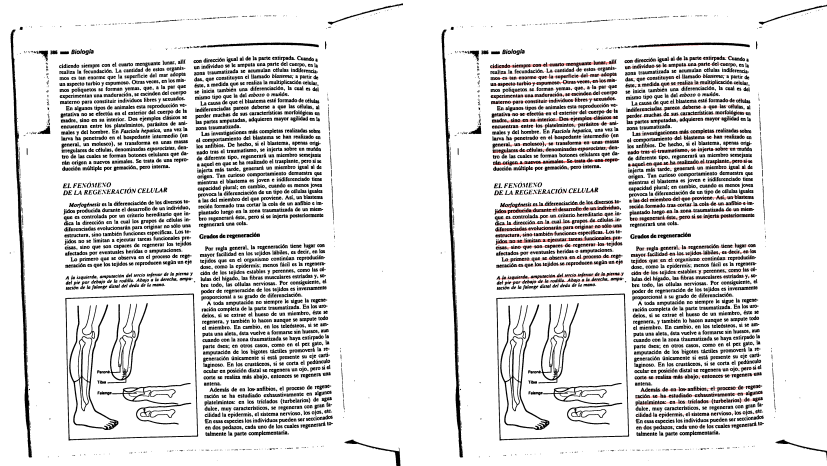
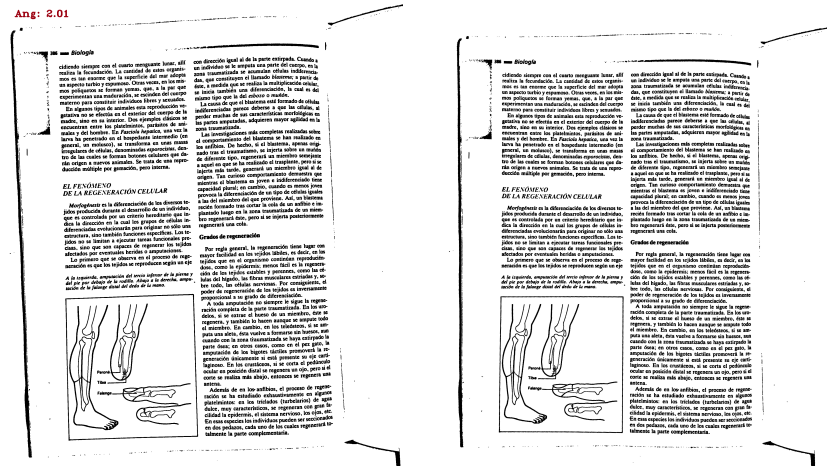


Imagen marcada

Detección de líneas

Figura 4.171: Corrección de ángulo: parte A. Fuente: elaboración propia.



Detección de ángulo

Corrección de ángulo

Figura 4.172: Corrección de ángulo: parte B. Fuente: elaboración propia.

Limpieza de bordes y sombras: Mediante la aplicación de diversas técnicas a través de la librería OpenCV

se creará una "caja" de ajuste la cual se utilizará para eliminar cosas tales como, bordes, imágenes, ruido, etc., con el fin de reducir o limitar al mínimo la aparición de falsos positivos al momento de aplicar el OCR.:

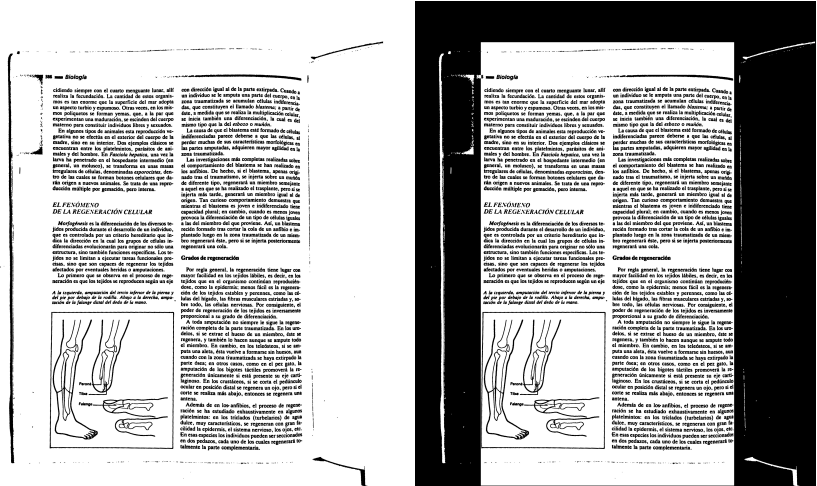


Figura 4.173: Limpieza de bordes y sombras: parte A. Fuente: elaboración propia.

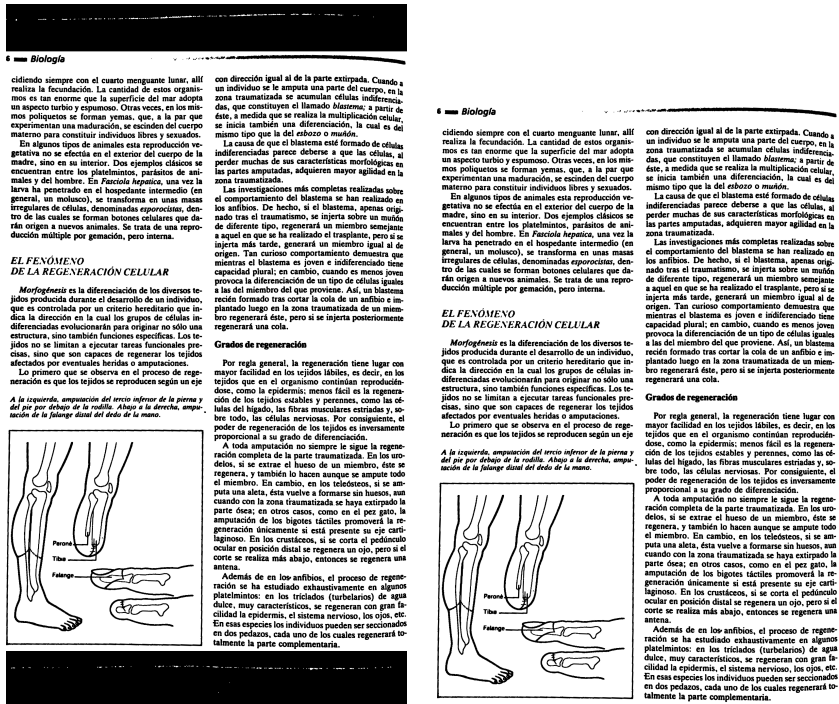


Figura 4.174: Limpieza de bordes y sombras: parte B. Fuente: elaboración propia.

Segunda corrección de ángulo: Al haber procesado la limpieza es posible mejorar el resultado del ángulo de rotación de la imagen:

Biología

ciendo siempre con el cuarto manguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuales.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en una masa irregular de células, denominadas esporocistos, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morphogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones.

El primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje. A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

Biología

ciendo siempre con el cuarto manguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuales.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en una masa irregular de células, denominadas esporocistos, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morphogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones.

El primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje. A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

Imagen corregida

Detección de líneas

Figura 4.175: Segunda corrección de ángulo: parte A. Fuente: elaboración propia.

Biología

ciendo siempre con el cuarto manguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuales.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en una masa irregular de células, denominadas esporocistos, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morphogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones.

El primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje. A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

Biología

ciendo siempre con el cuarto manguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuales.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en una masa irregular de células, denominadas esporocistos, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morphogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones.

El primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje. A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

Corrección de ángulo

Detección de líneas

Figura 4.176: Segunda corrección de ángulo: parte B. Fuente: elaboración propia.

Escalar imagen a x2: Tesseraact-OCR funciona mejor si los caracteres contenidos en la imagen tienen un

tamaño grande (aproximadamente entre 30 a 40 píxeles por caracteres es óptimo), para ello se alargará la imagen utilizando el algoritmo de interpolación bicúbica, esto con el fin de suavizar el resultado final y no perder tanta calidad en el re escalado:

6 Biología

ciendo siempre con el cuarto menguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuados. En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En Fasciola hepatica, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en unas masas irregulares de células, denominadas esporocistos, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morfogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones. Lo primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje

A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

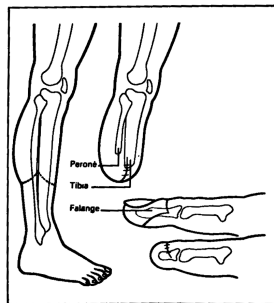


Imagen corregida

6 Biología

con dirección igual al de la parte extirpada. Cuando a un individuo se le amputa una parte del cuerpo, en la zona traumatizada se acumulan células indiferenciadas, que constituyen el llamado blastema; a partir de éste, a medida que se realiza la multiplicación celular, se inicia también una diferenciación, la cual es del mismo tipo que la del esbozo o muición. La causa de que el blastema esté formado de células indiferenciadas parece deberse a que las células, al perder muchas de sus características morfológicas en las partes amputadas, adquieren mayor agilidad en la zona traumatizada. Las investigaciones más completas realizadas sobre el comportamiento del blastema se han realizado en los anfibios. De hecho, si el blastema, apenas originado tras el traumatismo, se injerta sobre un muñón de diferente tipo, regenerará un miembro semejante a aquel en que se ha realizado el trasplante, pero si se injerta más tarde, generará un miembro igual al de origen. Tan curioso comportamiento demuestra que mientras el blastema es joven e indiferenciado tiene capacidad plural; en cambio, cuando es menos joven provoca la diferenciación de un tipo de células iguales a las del miembro del que proviene. Así, un blastema recién formado tras cortar la cola de un anfibio e implantado luego en la zona traumatizada de un miembro regenerará éste, pero si se injerta posteriormente regenerará una cola.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morfogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones. Lo primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje

A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

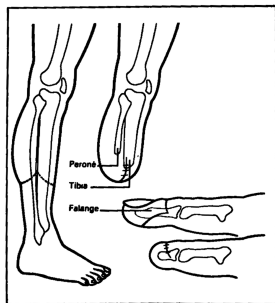


Imagen escalada

La cantidad
La cantidad

Vista cercana del escalado

Figura 4.177: Escalado de la imagen. Fuente: elaboración propia.

Segmentar áreas: Para acelerar el proceso de conversión de imagen a texto utilizando Tesseract-OCR es conveniente dividir la imagen en secciones más pequeñas, para ello se segmentara la imagen en zonas más pequeñas para su posterior procesamiento:

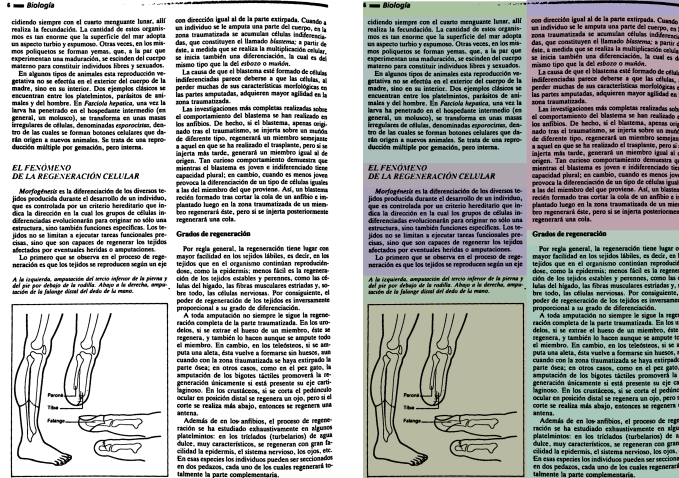


Figura 4.178: Segmentación de la imagen. Fuente: elaboración propia.

Demarcar áreas mínimas de los segmentos: Para acelerar aún más el procesamiento de los segmentos se eliminará el espacio en blanco entre los segmentos:

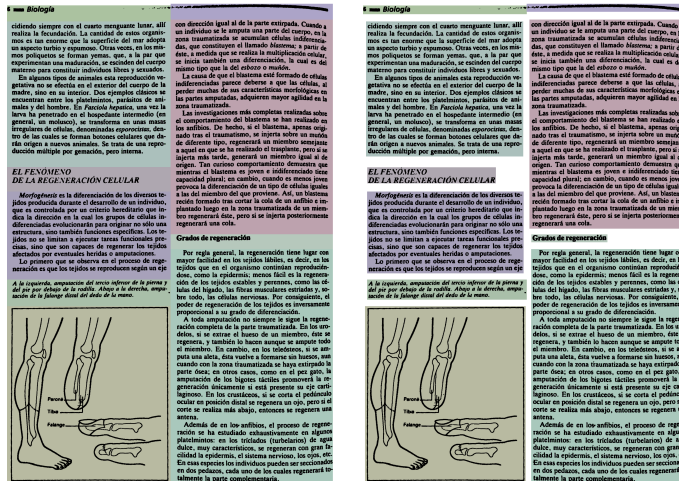


Figura 4.179: Áreas mínimas de los segmentos. Fuente: elaboración propia.

Ordenar segmentos individuales: Una vez definidos los segmentos se procede a ordenarlos para su posterior procesamiento:

6 Biología

ciendo siempre con el cuarto menguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuados.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en unas masas irregulares de células, denominadas *esporocistas*, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

EL FENÓMENO DE LA REGENERACIÓN CELULAR

Morfogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones.

Lo primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje

A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

con dirección igual al de la parte extirpada. Cuando a un individuo se le amputa una parte del cuerpo, en la zona traumatizada se acumulan células indiferenciadas, que constituyen el llamado *blastema*; a partir de éste, a medida que se realiza la multiplicación celular, se inicia también una diferenciación, la cual es del mismo tipo que la del *esbozo* o *muñón*.

La causa de que el blastema esté formado de células indiferenciadas parece deberse a que las células, al perder muchas de sus características morfológicas en las partes amputadas, adquieren mayor agilidad en la zona traumatizada.

Las investigaciones más completas realizadas sobre el comportamiento del blastema se han realizado en los anfibios. De hecho, si el blastema, apenas originado tras el traumatismo, se injerta sobre un muñón de diferente tipo, regenerará un miembro semejante a aquel en que se ha realizado el trasplante, pero si se injerta más tarde, generará un miembro igual al de origen. Tan curioso comportamiento demuestra que mientras el blastema es joven e indiferenciado tiene capacidad plural; en cambio, cuando es menos joven provoca la diferenciación de un tipo de células iguales a las del miembro del que proviene. Así, un blastema recién formado tras cortar la cola de un anfibio e implantado luego en la zona traumatizada de un miembro regenerará éste, pero si se injerta posteriormente regenerará una cola.

Grados de regeneración

Por regla general, la regeneración tiene lugar con mayor facilidad en los tejidos lábiles, es decir, en los tejidos que en el organismo continúan reproduciéndose, como la epidermis; menos fácil es la regeneración de los tejidos estables y perennes, como las células del hígado, las fibras musculares estriadas y, sobre todo, las células nerviosas. Por consiguiente, el poder de regeneración de los tejidos es inversamente proporcional a su grado de diferenciación.

A toda amputación no siempre le sigue la regeneración completa de la parte traumatizada. En los urodélos, si se extrae el hueso de un miembro, éste se regenera, y también lo hacen aunque se ampute todo el miembro. En cambio, en los teleosteos, si se amputa una aleta, ésta vuelve a formarse sin huesos, aun cuando con la zona traumatizada se haya extirpado la parte ósea; en otros casos, como en el pez gato, la amputación de los bigotes táctiles promoverá la regeneración únicamente si está presente su eje cartilaginoso. En los crustáceos, si se corta el pedúnculo ocular en posición distal se regenera un ojo, pero si el corte se realiza más abajo, entonces se regenera una antena.

Además de en los anfibios, el proceso de regeneración se ha estudiado exhaustivamente en algunos platelmintos: en los triclados (turbelarios) de agua dulce, muy característicos, se regeneran con gran facilidad la epidermis, el sistema nervioso, los ojos, etc. En esas especies los individuos pueden ser seccionados en dos pedazos, cada uno de los cuales regenerará totalmente la parte complementaria.

Segmentos mínimos

0 Biología

ciendo siempre con el cuarto menguante lunar, allí realiza la fecundación. La cantidad de estos organismos es tan enorme que la superficie del mar adopta un aspecto turbio y espumoso. Otras veces, en los mismos poliquetos se forman yemas, que, a la par que experimentan una maduración, se escinden del cuerpo materno para constituir individuos libres y sexuados.

En algunos tipos de animales esta reproducción vegetativa no se efectúa en el exterior del cuerpo de la madre, sino en su interior. Dos ejemplos clásicos se encuentran entre los platelmintos, parásitos de animales y del hombre. En *Fasciola hepatica*, una vez la larva ha penetrado en el hospedante intermedio (en general, un molusco), se transforma en unas masas irregulares de células, denominadas *esporocistas*, dentro de las cuales se forman botones celulares que darán origen a nuevos animales. Se trata de una reproducción múltiple por gemación, pero interna.

2. FENÓMENO DE LA REGENERACIÓN CELULAR

Morfogénesis es la diferenciación de los diversos tejidos producida durante el desarrollo de un individuo, que es controlada por un criterio hereditario que indica la dirección en la cual los grupos de células indiferenciadas evolucionarán para originar no sólo una estructura, sino también funciones específicas. Los tejidos no se limitan a ejecutar tareas funcionales precisas, sino que son capaces de regenerar los tejidos afectados por eventuales heridas o amputaciones.

Lo primero que se observa en el proceso de regeneración es que los tejidos se reproducen según un eje

A la izquierda, amputación del tercio inferior de la pierna y del pie por debajo de la rodilla. Abajo a la derecha, amputación de la falange distal del dedo de la mano.

En dirección igual al de la parte extirpada. Cuando a un individuo se le amputa una parte del cuerpo, en la zona traumatizada se acumulan células indiferenciadas, que constituyen el llamado *blastema*; a partir de éste, a medida que se realiza la multiplicación celular, se inicia también una diferenciación, la cual es del mismo tipo que la del *esbozo* o *muñón*.

La causa de que el blastema esté formado de células indiferenciadas parece deberse a que las células, al perder muchas de sus características morfológicas en las partes amputadas, adquieren mayor agilidad en la zona traumatizada.

Las investigaciones más completas realizadas sobre el comportamiento del blastema se han realizado en los anfibios. De hecho, si el blastema, apenas originado tras el traumatismo, se injerta sobre un muñón de diferente tipo, regenerará un miembro semejante a aquel en que se ha realizado el trasplante, pero si se injerta más tarde, generará un miembro igual al de origen. Tan curioso comportamiento demuestra que mientras el blastema es joven e indiferenciado tiene capacidad plural; en cambio, cuando es menos joven provoca la diferenciación de un tipo de células iguales a las del miembro del que proviene. Así, un blastema recién formado tras cortar la cola de un anfibio e implantado luego en la zona traumatizada de un miembro regenerará éste, pero si se injerta posteriormente regenerará una cola.

Grados de regeneración

Por regla general, la regeneración tiene lugar con mayor facilidad en los tejidos lábiles, es decir, en los tejidos que en el organismo continúan reproduciéndose, como la epidermis; menos fácil es la regeneración de los tejidos estables y perennes, como las células del hígado, las fibras musculares estriadas y, sobre todo, las células nerviosas. Por consiguiente, el poder de regeneración de los tejidos es inversamente proporcional a su grado de diferenciación.

A toda amputación no siempre le sigue la regeneración completa de la parte traumatizada. En los urodélos, si se extrae el hueso de un miembro, éste se regenera, y también lo hacen aunque se ampute todo el miembro. En cambio, en los teleosteos, si se amputa una aleta, ésta vuelve a formarse sin huesos, aun cuando con la zona traumatizada se haya extirpado la parte ósea; en otros casos, como en el pez gato, la amputación de los bigotes táctiles promoverá la regeneración únicamente si está presente su eje cartilaginoso. En los crustáceos, si se corta el pedúnculo ocular en posición distal se regenera un ojo, pero si el corte se realiza más abajo, entonces se regenera una antena.

Además de en los anfibios, el proceso de regeneración se ha estudiado exhaustivamente en algunos platelmintos: en los triclados (turbelarios) de agua dulce, muy característicos, se regeneran con gran facilidad la epidermis, el sistema nervioso, los ojos, etc. En esas especies los individuos pueden ser seccionados en dos pedazos, cada uno de los cuales regenerará totalmente la parte complementaria.

Segmentos ordenados

Figura 4.180: Orden de los segmentos. Fuente: elaboración propia.

Una vez llegado a este paso los segmentos serán procesados a través de Tesseract-OCR.

4.5.3.2.2. Preprocesado para OCR.Space API, Azure Computer Vision API y Google Drive OCR API

Las herramientas OCR.Space API, Azure Computer Vision API y Google Drive API por sí mismas realizan una serie de procesos internos para mejorar la precisión del proceso de reconocimiento, por tanto, solo se requiere ajustarse a las limitaciones de la API en cuestión para realizar sus procesos, es decir, convertir la imagen a gris y reducir su peso a 1Mb o menos en caso de OCR.Space, se utilizará la misma modificación para Computer Vision API y Google Drive API, ya que un archivo más ligero reduce el tiempo de subida del contenido de la imagen y acelera el proceso:

Transformar a gris: Esto con fin de mejorar el resultado del siguiente proceso, la reducción de ruido y sobre todo la binarización:

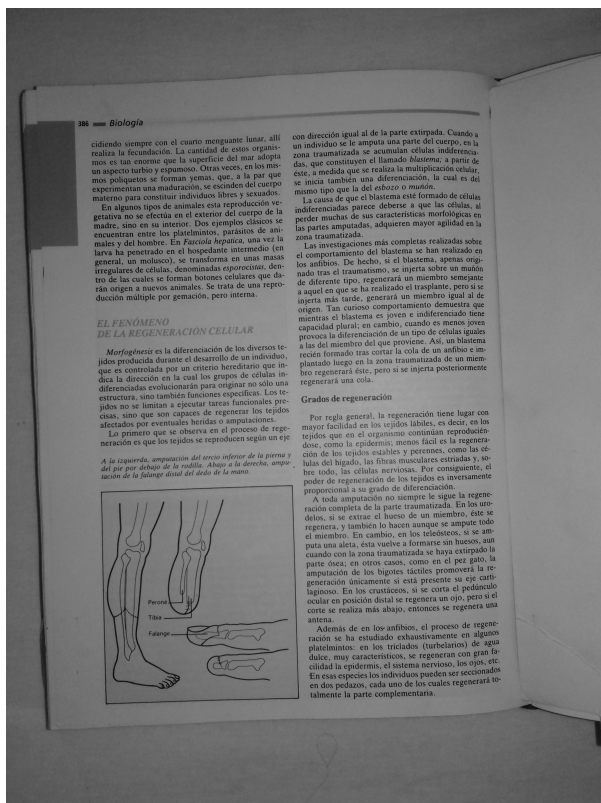
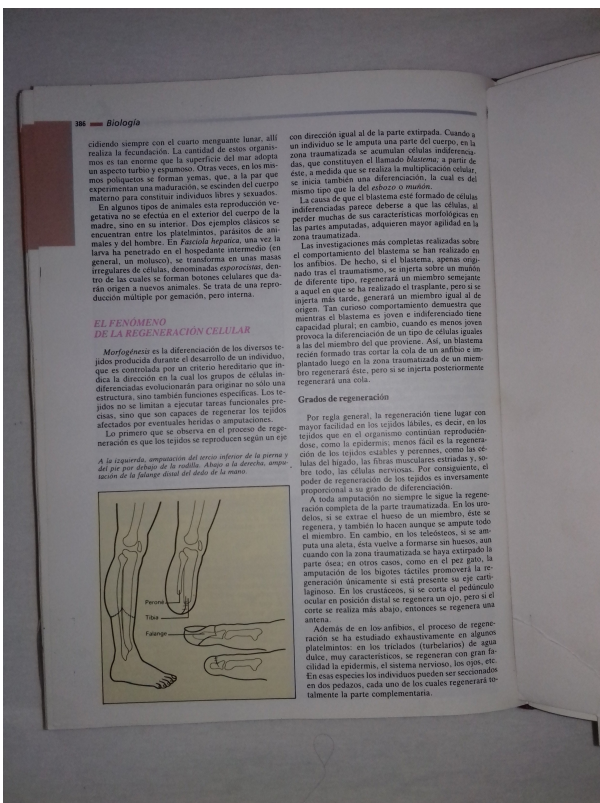


Imagen original

Conversión a gris

Figura 4.181: Conversión de imagen a gris. Fuente: elaboración propia.

Corrección de ángulo: Esto con fin de mejorar la velocidad de respuesta:

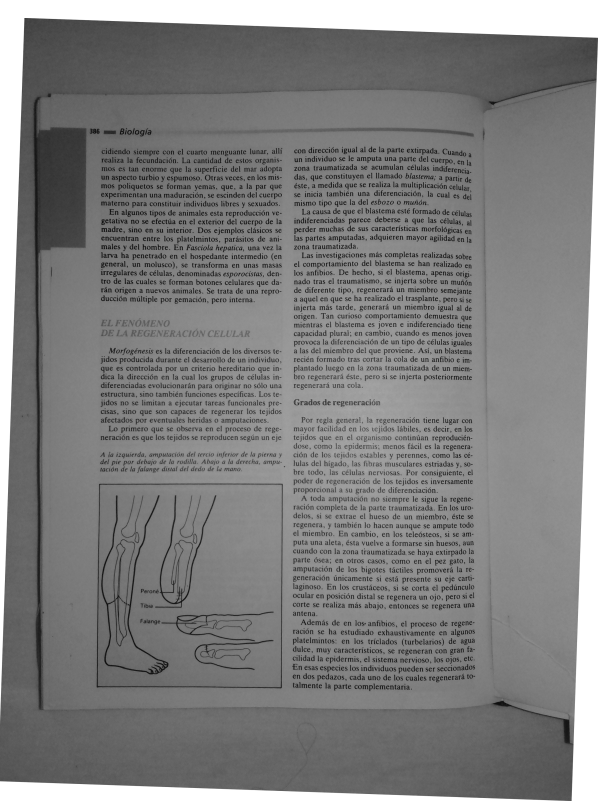
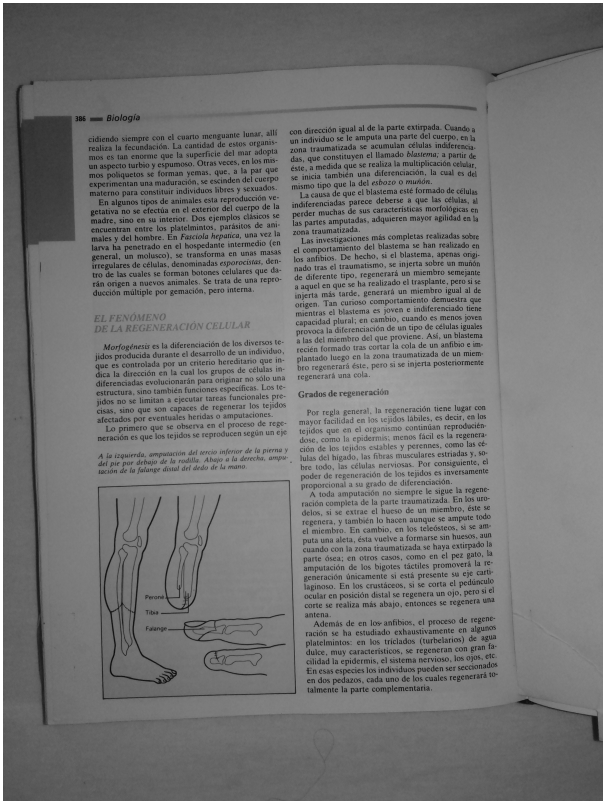


Imagen en gris

Corrección de ángulo

Figura 4.182: Corrección de ángulo. Fuente: elaboración propia.

Una vez realizado el preprocesado la imagen será procesada a través del API que corresponda.

4.5.3.3. Procesamiento

Una vez que la imagen a procesar ha sido mejorada para resaltar sus características es necesario pasarla a través de la herramienta OCR deseada, cada una tiene su manera diferente de funcionar, a continuación, se detalla la manera en la que se efectúa dicho procesamiento:

Tesseract-OCR: Haciendo uso del wrapper para Python “pytesseract” la imagen se procesará de la manera siguiente:

```
Salida_texto = pytesseract.image_to_string(image=imagen,
lang='spa',
config='--psm 1 --oem 1')
```

De lo anterior podemos definir los siguiente:

- **image=imagen:** La imagen a procesar, los formatos aceptados son arrays de imagen de NumPy, imágenes de OpenCV y archivos binarios.
- **lang='spa':** El idioma a utilizar para hacer el reconocimiento en este caso, español (spa).
- **config='-psm 1 -oem 1':** El método de segmentación de página, en este caso detección automática completa y el método de uso de Tesseract-OCR.

OCR.Space API: Haciendo uso del EndPoint y la librería “request” en python:

```
payload = {'scale': True,
          'isOverlayRequired': False,
          'apikey': 'API KEY',
          'language': 'spa'}
respuesta = requests.post(url='https://api.ocr.space/parse/image',
                          files={'imagen.jpg': imagen},
                          data=payload)
```

De lo anterior podemos definir lo siguiente:

- **payload:** La carga de opciones con las que se procesara la imagen:
 - **'scale':** Habilita el escalado para mejorar la respuesta, por defecto es falso (false), en nuestro caso lo habilitaremos para mejorar el resultado.
 - **'apikey':** La clave API de OCR.Space.
 - **'language':** El idioma con el que se reconocerá el texto de la imagen, en este caso utilizaremos español (spa).
- **url:** la dirección URL del EndPoint de la API
- **files:** El contenido binario de la imagen a procesar
- **data:** la “carga” (payload) de configuración definido anteriormente para el procesado de la imagen.

Google Drive OCR API: Haciendo uso del EndPoint y la librería “pydrive” en Python:

```
# Permite la autenticaci\'' de la aplicacion para interactuar con la API
google_auth = GoogleAuth(settings_file='settings.yaml')
# Crea una autenticacion local
google_auth.LocalWebserverAuth()
# Crea un recurso de acceso a la API utilizando la autenticacion
google_drive = GoogleDrive(google_auth)
# Crea un documento vacio
archivo = google_drive.CreateFile({'title': 'imagen a OCR'})
# Define el tipo de documento
archivo['mimeType'] = 'application/vnd.google-apps.document'
# Establece el lenguaje para realizar el OCR
archivo.Upload({'ocrLanguage': 'en'})
archivo.SetContentFile('imagen.jpg')
# Sube la imagen al documento
archivo.Upload()
# Obtiene el texto y transforma la codificacion de UTF-8 con BOM a UTF-8
texto_obtenido = archivo.GetContentString('text/plain').encode('utf-8',
↳ 'replace').decode('utf-8-sig', 'replace')
```

Computer Vision API: Haciendo uso del EndPoint y la librería “request” en Python:

```
headers = {'Ocp-Apim-Subscription-Key': [APY KEY]}
config = '?language={}&detectOrientation=true'.format(language)
respuesta = requests.post(url='{}'.format(self.base_url, config),
    files={'imageFile': open('imagen.jpg', 'rb')},
    headers=headers)
if int(respuesta.status_code) == 200:
    datos_json = respuesta.json()
```

De lo anterior podemos definir lo siguiente:

- **headers:** El encabezado de la petición el cual contiene la clave API
- **config:** Contiene la información sobre el idioma a utilizar en el proceso de OCR así como la información para corregir el ángulo de la imagen en caso que no sea correcto
- **respuesta:** Contiene la petición mediante POST y la respuesta de la API, en ella se define el archivo a procesar junto a los demás datos como el encabezado y la configuración.

Una vez realizado el proceso de extracción de texto el script principal se encargará de convertir el texto obtenido a un formato compatible al utilizado en la conversión de texto a audio utilizando las herramientas adecuadas.

4.5.4. Web scraping o raspado web

4.5.4.1. Herramientas a utilizar

El proceso de web scraping se divide en dos partes, primeramente se obtendrá el contenido de los resultados generados por los motores de búsqueda, para ello se utilizará la librería “request”, ya que su sintaxis simplificada con respecto a librerías más antiguas tales como “urllib” y a su comodidad de uso permiten realizar las peticiones HTTP necesarias para obtener el contenido web de dichas peticiones, luego de ello para realizar el proceso de extracción de los resultados se hará uso de la librería “BeautifulSoup”, la cual permite acceder al contenido HTML retornado de manera más cómoda y orgánica, este proceso se aplicará luego a los resultados seleccionados a partir de la búsqueda previamente realizada.

4.5.4.2. Procesamiento

Para buscar: una vez definido el término o términos a buscar se realizará a través del script del archivo web.py, haciendo uso de la función:

```
resultados = web.buscar_en_linea (busqueda='terminos de busqueda',
                                  pagina=0,
                                  motor='motor de busqueda')
```

De lo anterior se define lo siguiente:

- **Búsqueda:** El o los términos de búsqueda.
- **Página:** la página de resultados a mostrar, a partir de 0.
- **Motor:** El motor de búsqueda a utilizar, los posibles son: “google”, “bing” y “duckduckgo”.

Lo anterior retornará un array bidimensional el cual contendrá el siguiente formato:

| | |
|---------------------|-----------------|
| Título página web 1 | Dirección URL 1 |
| Título página web 2 | Dirección URL 2 |
| ... | ... |
| Título página web N | Dirección URL N |

Tabla 4.34: Ejemplo del array de resultados

A partir de dicho array se enviará formateado título de las páginas al código encargado de transformar el texto a audio. Para lograr lo anterior en el script de Python realizará los siguientes pasos:

Mediante Request se solicita el contenido HTML de los resultados de búsqueda, al utilizar diferentes

buscadores, las URL y parámetros necesarios son diferentes:

| Buscador | URL base | Parámetros |
|------------|-----------------------------------|--|
| Google | https://www.google.com/search | q: El término o términos a buscar start: El número de página de resultados de búsqueda. |
| Bing | https://www.bing.com/search | q: El término o términos a buscar cvid: Cadena de caracteres utilizada para identificar la aplicación, 32 caracteres. firts: El número de página de resultados de búsqueda. |
| DuckDuckGo | https://lite.duckduckgo.com/lite/ | q: El término o términos a buscar. s: el numero de resultados de búsqueda. |

Tabla 4.35: Parámetros necesarios para en raspado web

El HTML obtenido se procesa a través de BeautifulSoup recorriendo sus elementos y extrayendo las características que concuerden con los siguientes paramentos:

| Buscador | Parametros | Elementos seleccionados |
|------------|---|--|
| Google | <ul style="list-style-type: none"> ■ Todos los elementos 'div' cuya clase concuerde con 'jfp3ef'. ■ Dentro de los elementos 'div' anteriores extraer los elementos 'a' y los elementos 'div' cuya clase concuerde únicamente con 'BNeawe vvjwJb AP7Wnd'. ■ De los elementos 'a' anteriores aquellos cuyo elemento 'href' no sea '#' son resultados válidos | <ul style="list-style-type: none"> ■ El elemento 'div' con las clases 'BNeawe vvjwJb AP7Wnd' contiene el título del resultado de búsqueda. ■ El elemento 'a' en su atributo 'href' contiene la dirección URL del resultado de búsqueda |
| Bing | <ul style="list-style-type: none"> ■ Todos los elementos 'li' cuya clase concuerde con 'b_algo' ■ De los elementos 'li' anteriores extraer los elementos 'a' ■ De de los elementos 'a' anteriores aquellos cuyo elemento 'href' no sea '#' son resultados válidos | <ul style="list-style-type: none"> ■ El elemento 'a' contiene en su atributo 'href' la dirección URL del resultado de búsqueda y su texto interno es el título de dicho resultado de búsqueda |
| DuckDuckGo | <ul style="list-style-type: none"> ■ Todos los elementos 'a' cuya clase concuerde con 'result-link' ■ De de los elementos 'a' anteriores aquellos cuyo elemento 'href' no sea '#' son resultados válidos | <ul style="list-style-type: none"> ■ El elemento 'a' contiene en su atributo 'href' la dirección URL del resultado de búsqueda y su texto interno es el título de dicho resultado de búsqueda |

Tabla 4.36: Características a extraer

Una vez extraído los títulos y las URL se deben corregir las URL de 2 de los buscadores, tanto Google como DuckDuckGo:

| Buscador | Modificaciones |
|------------|---|
| Google | Quitar el texto '/url?q=' y recortar la URL hasta el inicio de la siguiente secuencia de caracteres '&sa='. Ejemplo: /url?q= https://example.com&sa=[otros caracteres] Quedaría: https://example.com |
| DuckDuckGo | Quitar el texto '/l/?kh=-1&uddg=' |

Tabla 4.37: Modificaciones a realizar

Para dictar una página: una vez seleccionado el resultado de búsqueda deseado se procederá a extraer el texto de dicha dirección URL a través del script del archivo web.py, haciendo uso de la función:

```
texto = webs.cargar_web('URL')
```

Lo cual nos devolverá el texto del contenido en dicha URL para su posterior proceso de formateo para su envío al código encargado de transformar el texto a audio.

Para lograr lo anterior en el script de Python encargado de cargar el contenido de la dirección URL proporcionada realizará los siguientes pasos:

- Mediante Request se solicita el contenido HTML de la URL seleccionada
- El HTML obtenido se procesa a través de BeautifulSoup recorriendo sus elementos internos y extrayendo el texto con concuerde con las siguientes características:

El tag HTML es uno de los siguientes: h1, h2, h3, h4, h4 y la longitud del texto interno de dicho tag es igual o superior a 20 caracteres

El tag HTML es uno de los siguientes: p, span o article y la longitud del texto interno de dicho tag es igual o superior a los 100 caracteres

- Si las condiciones de selección son válidas, el texto se agrega caso contrario se descarta.

4.5.5. Procesamiento de Texto a voz

4.5.5.1. Herramientas a utilizar

Para el proceso de conversión de texto a voz se dispone de cuatro herramientas: Festival Text to Speech, Cognitive Services Text to Speech API, VoiceRSS API y Google Text to Speech.

Para el proceso de reproducción del audio se hará uso de la aplicación VLC en su modo consola o "cvlc".

4.5.5.2. Preprocesamiento

Las herramientas de texto a voz antes mencionadas requieren diferentes codificaciones de texto, del mismo modo también requiere de diferentes longitudes de cadenas de caracteres para su correcto funcionamiento, el script de Python debe realizar estas conversiones y ajustes para adecuarlos para las respectivas herramientas de acuerdo a la siguiente tabla.

| Herramienta | Codificación soportada | Longitud de cadena máxima |
|----------------------------|------------------------|---------------------------|
| Festival TTS | ISO 8859-1 | N/A |
| Google TTS | Unicode/UTF-8 | <= 700 |
| Cognitive Services TTS API | UTF-8 | <= 750 |
| VoiceRSS TTS API | UTF-8 | <= 900 |

Tabla 4.38: Codificación y longitud de cadenas soportadas

4.5.5.3. Procesamiento

Festival

A través de CLI utilizando la librería estándar "os" de Python, utilizando el parámetro -eval se define la voz a utilizar para realizar la conversión a voz, en este caso español:

```
path = 'text2wave -o audio/festival/{}.wav audio/festival/{}.txt -eval
↳ "{}"'
with open('audio/festival/0.txt', 'wb') as file_audio:
    file_audio.write(texto)
    file_audio.close()
os.system(path.format('0', '0', 'voice_JuntaDeAndalucia_es_sf_diphone'))
```

Lo anterior genera un archivo de audio "WAV" en la ruta audio/festival/0.wav

gTTS (Google-Text to Speech)

A través de una instancia de la librería gTTS, si la longitud del texto a convertir a voz es superior al límite de conversión establecido, el texto se procesará en archivos de audio múltiples iniciando en el número "0" hasta el

número “n” siendo “n” = longitud del texto/700:

```
salida_gtts = gTTS(text=texto_a_convertir, lang='es')
salida_gtts.save('audio/gtts/{}.mp3'.format(str(i)))
```

Lo anterior se utilizaría para crear una colección de archivos “MP3” en la ruta audio/gtts/

Cognitive Services Text to Speech API

A través de una instancia del script de Python preparado para Cognitive Services Text to Speech API, si la longitud del texto a convertir a voz es superior al límite de conversión establecido, el texto se procesará en archivos de audio múltiples iniciando en el número “0” hasta el número “n” siendo “n” = longitud del texto/750:

```
salida = a_voz_bing.speak(language='es-ES',
                          voice='HelenaRUS',
                          file_format='audio-16khz-32kbitrate-mono-mp3',
                          text=texto_a_convertir)
with open('audio/{}/{}.mp3'.format('bing', str(i)), 'wb', 0) as archivo:
    archivo.write(salida)
    archivo.flush()
    archivo.close()
```

Lo anterior se utilizaría para crear una colección de archivos “MP3” en la ruta audio/bing/

VoiceRSS API

A través de una instancia del script de Python preparado para VoiceRSS API, si la longitud del texto a convertir a voz es superior al límite de conversión establecido, el texto se procesará en archivos de audio múltiples iniciando en el número “0” hasta el número “n” siendo “n” = longitud del texto/900:

```
salida = a_voz_vrssl.speak(language='es-es',
                          text=texto_a_convertir,
                          codec='MP3',
                          audio_format='48khz_16bit_stereo')
with open('audio/{}/{}.mp3'.format('voicerss', str(i)), 'wb', 0) as
↪ archivo:
    archivo.write(salida)
    archivo.flush()
    archivo.close()
```

4.5.5.4. Reproducción

La reproducción del audio que contiene la voz se realizará de manera asíncrona, utilizando el script de Python preparado para ello mediante el uso de la librería estándar de Python “subprocess”, de la siguiente manera:

```
id_sub_proc = subprocess.Popen('python player.py -g {} -n
↳ {}.format(motor,
              str(numero)),
                              shell=True)
```

Lo anterior devolverá un id de subprocesso que utilizaremos para esperar la finalización del proceso de reproducción y continuar la ejecución del proceso principal:

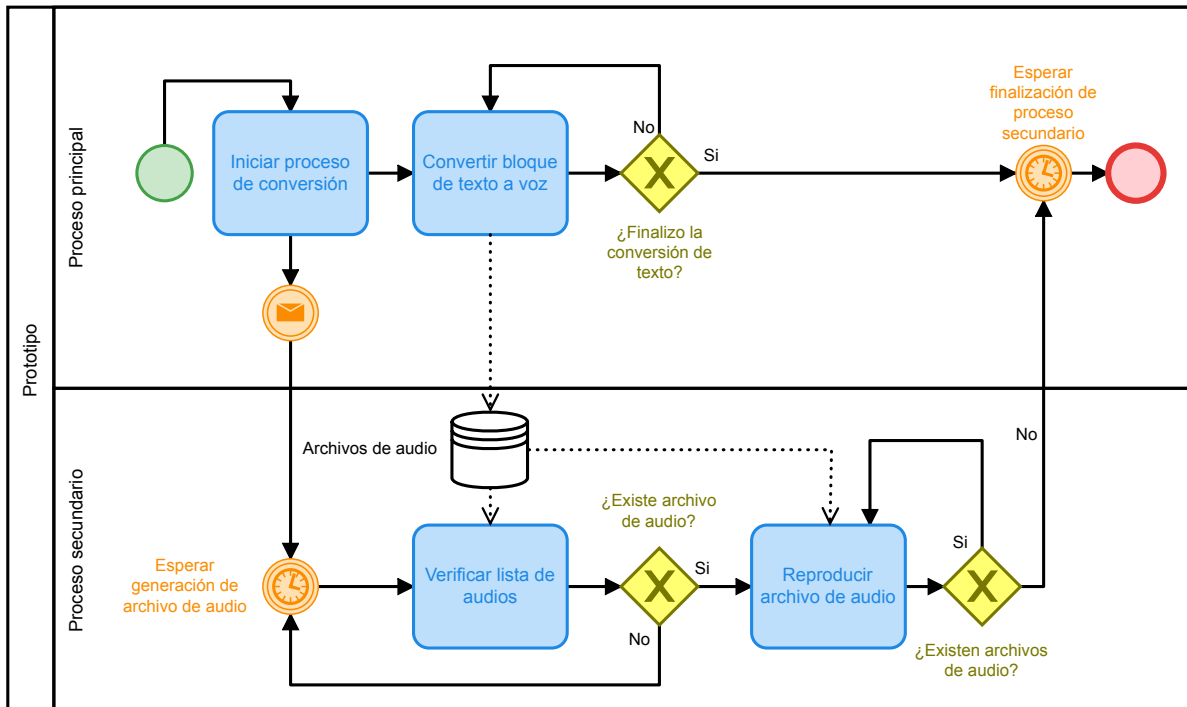


Figura 4.183: Diagrama del reproductor. Fuente: elaboración propia.

Para la reproducción de los archivos de audio se debe tener en cuenta la potencia de funcionamiento de la Raspberry PI así también el hecho de tener diferentes formatos de salida de audio, por ello se hará uso de la aplicación de reproducción de medios multimedia “VLC”, a través de su línea de comando haciendo uso de la librería estándar de Python “os” de la siguiente manera:

```
os.system('cvlc --play-and-exit --rate=[ratio] --gain=[ganancia]
↳ [ruta_archivo]')
```

Los ratios (velocidad de reproducción) y ganancia (volúmenes de reproducción) para las diferentes herramientas son los siguientes, el valor por defecto para ambos atributos es 1 así que los valores mayores o menores representan incremento y decremento respectivamente:

| Herramienta | Ratio | Ganancia |
|----------------------------|-------|----------|
| Festival TTS | 0.95 | 0.7 |
| Google TTS | 1.15 | 0.7 |
| Cognitive Services TTS API | 1.15 | 1 |
| VoiceRSS TTS API | 1.05 | 1 |

Tabla 4.39: Ratios y ganancias.

4.5.6. Script individuales

hunspell.py Proporciona métodos de interacción con la herramienta Hunspell para realizar correcciones ortográficas de los textos obtenidos mediante OCR sobre todo para los resultados de la herramienta Tesseract-OCR

web.py Proporciona los métodos para la interacción del prototipo con contenido web siendo estos resultados de búsqueda de diferentes buscadores y el contenido de dichos resultados caso necesitarse.

tts_voicerss.py Proporciona los métodos para la interacción con la API de texto a voz VoiceRSS.

tts_bing.py Proporciona los métodos para la interacción con la API de texto a voz Cognitive Service Text to Speech.

regex.py Proporciona métodos para la correcciones de errores de texto comunes resultantes del OCR mediante el uso de expresiones regulares.

record.py Proporciona métodos para la grabación de audio a archivos de voz, la creación de filtros y el resampleo de audio.

procesador_imagen.py Proporciona métodos para la limpieza, mejora y segmentación de imágenes, utilizado para realizar OCR.

procesador_tesseract.py Proporciona los métodos necesarios para el procesamiento de imágenes a través de la herramienta Tesseract-OCR, consta de su propio reproductor de archivos de audio.

player.py Proporciona métodos para la reproducción de los archivos de audio de manera secuencial.

ocr_space.py Proporciona los métodos para la interacción con la API de OCR ocr.space.

ocr_google_drive.py Proporciona los métodos para la interacción con la API de OCR Google Drive API.

base.py Es el script encargado de la integración de todos los demás archivos de código para su manejo centralizado, creando así una clase global con todos los métodos necesarios para interactuar con el prototipo, además de ello integra clases de tamaño pequeño tales como: la clase de voz a texto de IBM, la clase de acceso a teclado y la

clase de manipulación de archivos PDF además de ello también se encarga de almacenar las configuraciones y claves API para el funcionamiento de las diferentes herramientas.

run.py Es el encargado de interactuar entre el usuario y el prototipo, hace uso de base.py para realizar todas las interacciones lo denominaremos controlador principal.

4.5.6.1. Controlador principal

El script principal de ejecución es el que se encargará de unir todos los componentes individuales para llevar a cabo las funciones, es el script run.py.

4.5.6.1.1. Configuración de ejecución

‘-m’ o ‘-mode’: define el modo de funcionamiento del prototipo, los valores posibles para esta opción son: ‘mic’ o ‘rec’, el valor establecido por defecto es ‘mic’ con lo cual no es un valor obligatorio.

‘-s’ o ‘-spechtotext’: define la herramienta de conversión de texto a voz a utilizar al utilizar el prototipo, los valores posibles para esta opción son: ‘bing’, ‘google’, ‘hibrido’, ‘ibm’, ‘sphinx’ o ‘wit’, el valor establecido por defecto es ‘hibrido’ con lo cual no es un valor obligatorio.

‘-t’ o ‘-texttospeech’: define la herramienta de conversión de voz a texto a utilizar al utilizar el prototipo, los valores posibles para esta opción son: ‘bing’, ‘festival’, ‘gtts’ o ‘voicerss’, el valor establecido por defecto es ‘gtts’ con lo cual no es un valor obligatorio.

‘-o’ o ‘-ocr’: define la herramienta de OCR a utilizar al utilizar el prototipo, los valores posibles para esta opción son: ‘bing’, ‘cloud’, ‘ocrspace’ o ‘tesseract’, el valor establecido por defecto es ‘ocrspace’ con lo cual no es un valor obligatorio.

‘-w’ o ‘-websearch’: define el motor de búsqueda a utilizar al utilizar el prototipo, los valores posibles para esta opción son: ‘bing’, ‘duckduckgo’ o ‘google’, el valor establecido por defecto es ‘duckduckgo’ con lo cual no es un valor obligatorio.

‘-p’ o ‘-pdf’: define el archivo PDF a utilizar como fuente de lectura, es un valor opcional.

4.5.6.1.2. Funciones

El controlador principal se encarga de las siguientes funciones:

1. Por línea directa del micrófono

- 1.1. Haciendo uso del script base permanecer en escucha del micrófono a la espera de la entrada de las palabras clave ya sea “prototipo” o “inicio”.
- 1.2. Al recibir la entrada de la palabra clave, el script principal espera la entrada de algún comando de voz, si el audio ingresado no es válido genera una salida por consola y continua a la escucha.
- 1.3. Una vez finalizada la ejecución y procesamiento de un comando continua a la espera de la palabra clave.

2. Mediante grabación

- 2.1. Haciendo uso del script base permanecer en espera de la pulsación de la tecla “espacio” del teclado alfanumérico o la tecla “intro” del teclado numérico.
 - 2.2. Al recibir la pulsación de tecla, el script principal comienza la grabación de audio desde el micrófono en un intervalo de 10 segundos, si el audio ingresado no es válido genera una salida por consola y espera nuevamente la pulsación de tecla.
 - 2.3. Una vez finalizada la ejecución y procesamiento de un comando continua a la espera de la pulsación de tecla.
3. Una vez recibido un comando válido mediante el uso del script base procede a procesar dicho comando
 - 3.1. **Si el comando a procesar es uno de lectura de texto físico:** El controlador principal se encargará de realizar los pasos necesarios para leer una página de texto físico (controlar la cámara, procesar la imagen, enviarla a la herramienta de OCR), luego convertirá el texto obtenido a audio y lo reproducirá, posteriormente se encarga de almacenar el audio para su posterior repetición en caso de necesitarse.
 - 3.2. **Si el comando a procesar es uno de lectura de texto digital:** El controlador principal se encargará de verificar la existencia del documento PDF que se desea leer, si existe y es un documento válido se procederá a realizar los pasos necesarios para leer una página de texto digital luego de ello se encarga de almacenar el audio para su posterior repetición en caso de necesitarse.
 - 3.3. **Si el comando a procesar es uno de búsqueda web:** el controlador principal se encargará de realizar la búsqueda seleccionada haciendo uso de las herramientas definidas al iniciar el prototipo

luego de ello convertirá dichos resultados a audio y los reproducirá, después de ello almacenara el audio con la lista de resultados para su posterior repetición en caso de necesitarse y también de almacenar los términos de búsqueda, los resultados de búsqueda (título y dirección URL) y el número de página de los resultados de búsqueda requerida para los comandos de navegación.

- 3.4. **Si el comando a procesar es un avance o retroceso de navegación:** El controlador principal verificara la caché de términos de búsqueda, si la caché es valida procederá a realizar la búsqueda de la página búsqueda de resultados acorde a lo solicitado, en caso que se solicite un retroceso de navegación estando en la primera página de resultados, el controlador principal generara un mensaje de advertencia y mostrara la primera página de resultados de búsqueda, luego de ello convertirá dichos resultados a audio y los reproducirá, una vez concluido lo anterior almacenara el audio y los resultados de búsqueda (título y dirección URL) y el numero de página de los resultados de búsqueda, si la caché de términos no es valida el controlador principal reproducirá un mensaje de advertencia.
- 3.5. **Si el comando a procesar es una lectura de contenido de una dirección web:** El controlador principal verificara la caché de resultados de búsqueda y se el resultado es válido entonces a través del script base (y este a través del script web) accedera a dicha dirección web extraerá el contenido relevante y procederá a enviarlo a la herramienta para convertir el texto a voz, está a su vez procederá a reproducir el audio obtenido a través del script de reproducción de audio, luego de ello almacenara el audio reproducido para su posterior repetición en caso de necesitarlo.
- 3.6. **Si el comando a procesar es una repetición de texto:** El controlador principal verificara la caché de audio de texto; si es válida volverá a reproducir el texto almacenado como audio, en caso contrario reproducirá un mensaje de advertencia.
- 3.7. **Si el comando a procesar es una repetición de resultados de búsqueda:** El controlador principal verificara la caché de audio de resultados de búsqueda; si es válida volverá a reproducir los resultados almacenados como audio, en caso contrario reproducirá un mensaje de advertencia.

4.5.6.1.3. Diagrama del script principal

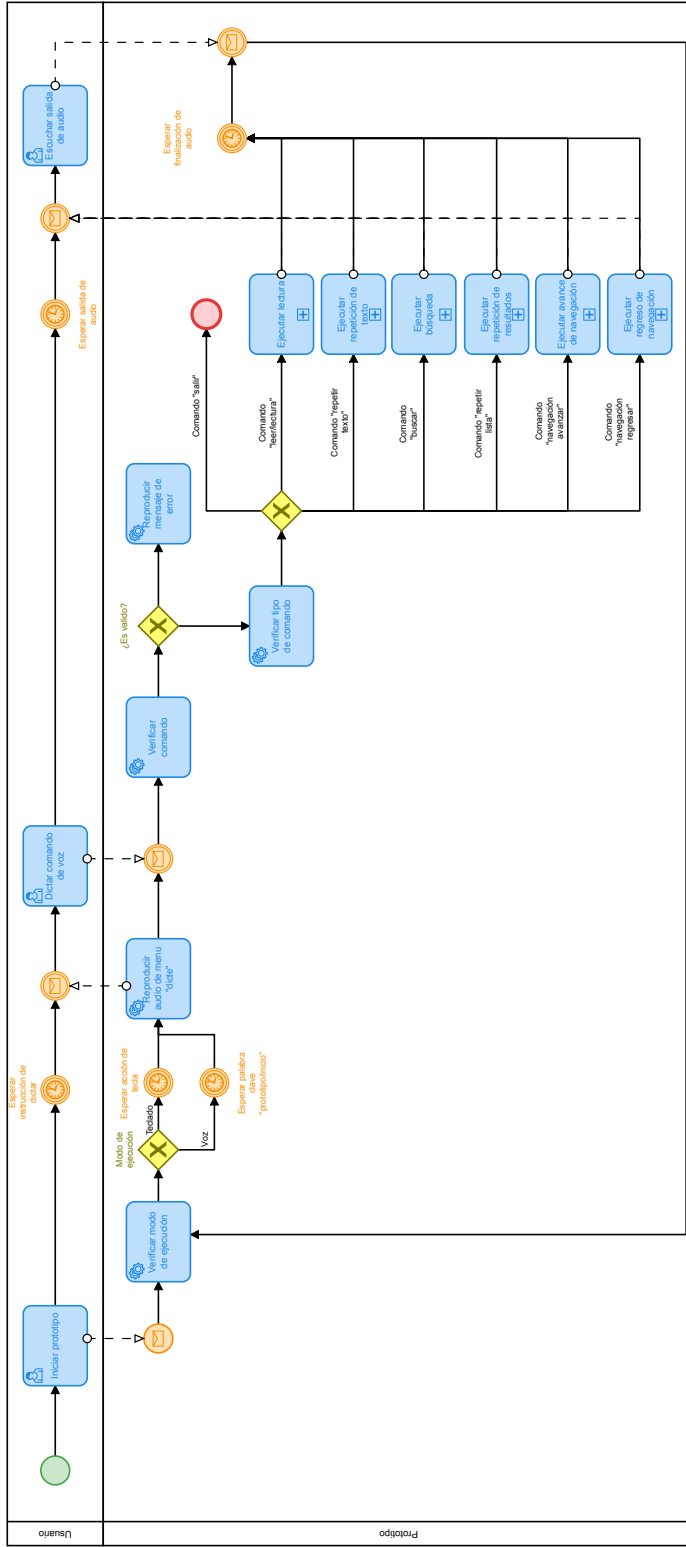


Figura 4.184: Diagrama controlador principal. Fuente: elaboración propia.

4.5.6.1.4. Modos de ejecución definidos

| | Mínimo | API's libres | API's de pago |
|-----------------------|---|---------------------|---------------------------------|
| OCR | Tesseract-OCR | OCR.Space | Computer Vision Google Drive |
| Text-to-Speech | Festival | gTTS VoiceRSS | CSTS |
| Speech-to-Text | CMU Sphinx híbrido (CMU Sphinx y Wit.AI) | Wit.AI | IBM Google CSST |

Tabla 4.40: Modos de ejecución.

4.5.6.1.5. Ventajas y desventajas de los diferentes modos de ejecución

■ mínimo

Ventajas

1. Sin límite de procesos
2. Mayor velocidad de los procesos al estar instaladas las herramientas en el sistema, esto solo aplica si el código está ejecutándose en una PC de potencia media
3. Independencia de Internet
4. Sin necesidad de pago de regalías al tratarse únicamente de herramientas libres

Desventajas

1. Pobre calidad de resultados en la conversión de voz a texto.
2. Pobre calidad de resultados en la conversión de imágenes a texto.
3. Voz muy mecánica/robótica en la conversión de texto a voz.
4. Necesidad de procesos adicionales para el correcto funcionamiento de las herramientas.

■ API's gratuitas

Ventajas

1. Velocidad relativamente rápida en sus procesos.
2. Alta precisión en la conversión de voz a texto.
3. Alta precisión en la conversión de imagen a texto.
4. Voz más natural en el proceso de conversión de texto a voz.
5. Sin necesidad de pago de regalías al utilizar las versiones libres de las librerías.

6. Menor cantidad de procesos.

Desventajas

1. Límites de funcionamiento en el proceso de imágenes a texto.
2. Límites de funcionamiento en el proceso de texto a voz.
3. Dependencia de Internet.

■ **API's de pago**

Ventajas

1. Velocidad relativamente rápida en sus procesos
2. Alta precisión en la conversión de voz a texto
3. Alta precisión en la conversión de imagen a texto
4. Voz más natural en el proceso de conversión de texto a voz
5. Sin necesidad de pago de regalías al utilizar las versiones libres de las librerías
6. Menor cantidad de procesos

Desventajas

1. La necesidad de pago de regalías
2. Dependencia de Internet

4.6. Estándares de código para el proyecto

4.6.1. Nomenclatura

Para la nomenclatura involucrada en el desarrollo del prototipo se hizo uso del estándar de código PEP 8 para Python, a continuación se presentan algunas de las directrices mas importantes para el desarrollo del prototipo.

4.6.1.1. Tabulaciones y espacios

El método de indentación más popular en Python es con espacios. El segundo más popular es con tabulaciones, sin mezclar unos con otros. Cualquier código indentado con una mezcla de espacios y tabulaciones debe ser convertido a espacios exclusivamente. Por tal motivo solo utilizar espacios en lugar de tabulaciones para evitar posibles problemas de ejecución en el futuro.

4.6.1.2. Importaciones

Las importaciones deben estar en líneas separadas.

```
1 import os
2 import sys
```

Las importaciones siempre se colocan al comienzo del archivo, simplemente luego de cualquier comentario o documentación del módulo, y antes de globales y constantes.

4.6.1.3. Convenciones de nombramiento

Nombres para evitar: Nunca se deben usar los caracteres ‘1’ (letra ele en minúscula), ‘O’ (letra o mayúscula), o ‘I’ (letra i mayúscula) como simples caracteres para nombres de variables. En algunas fuentes, estos caracteres son indistinguibles de los números uno y cero. Cuando se quiera usar ‘1’, en lugar usa ‘L’.

Nombre de paquetes y módulos: Los módulos deben tener un nombre corto y en minúscula. Guiones bajos pueden utilizarse si mejora la legibilidad. Los paquetes en Python también deberían tener un nombre corto y en minúscula, aunque el uso de guiones bajos es desalentado. Ya que los nombres de módulos están ligados a los de los archivos, y algunos sistemas operativos distinguen caracteres entre minúsculas y mayúsculas y truncan nombres largos, es importante que sean bastante cortos.

Nombres de Clases: Casi sin excepción, los nombres de clases deben utilizar la convención “CapWords” (palabras que comienzan con mayúsculas). Clases para uso interno tienen un guión bajo como prefijo.

Nombres de constantes: Las constantes son generalmente definidas a nivel módulo, escritas con todas las letras en mayúscula y con guiones bajos separando palabras. Por ejemplo, `CONTENIDO_FINAL` y `TOTAL`.

Nombres de funciones: Las funciones deben ser en minúscula, con las palabras separadas por un guión bajo, aplicándose éstos tanto como sea necesario para mejorar la legibilidad. Siempre usa una sentencia `def` en lugar de una sentencia de asignación que liga a la expresión lambda directamente a un nombre.

```
1 #Si:
2 def f(x): return 2*x
3 #No:
4 f = lambda x: 2*x
```

4.6.1.4. Argumentos de funciones y métodos

Siempre usa `self` para el primer argumento de los métodos de instancia. Si los argumentos de una función coincide con una palabra reservada del lenguaje, generalmente es mejor agregar un guión bajo como sufijo antes de usar una abreviación u ortografía incorrecta.

Evita usar espacios en blanco extraños en las siguientes situaciones:

Inmediatamente dentro de paréntesis, corchetes o llaves:

```
1 #Si:
2 func(ham[1], {eggs: 2})
3 #No:
4 func( ham[ 1 ], { eggs: 2 } )
```

Inmediatamente antes de una coma, un punto y coma o dos puntos:

```
1 #Si:
2 if x == 4: print x, y; x, y = y, x
3 #No:
4 if x == 4 : print x , y ; x , y = y , x
```

Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función:

```
1 #Si:
2 spam(1)
3 #No:
4 spam (1)
```

Inmediatamente antes de un corchete que empieza una indexación o “slicing” (término utilizado tanto en el ámbito de habla inglesa como española):

```
1 #Si:
2 dict['key'] = list[index]
3 #No:
4 dict ['key'] = list [index]
```

Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro:

```
1 #Si:
2 x = 1
3 y = 2
4 long_variable = 3
5 #No:
6 x = 1
7 y = 2
8 long_variable = 3
```

No usar espacios alrededor del = (igual) cuando es utilizado para indicar un argumento en una función (“keyword argument”) o un parámetro con un valor por defecto.

```

1 #Si:
2 def complex(real, imag=0.0):
3     return magic(r=real, i=imag)
4 #No:
5 def complex(real, imag = 0.0):
6     return magic(r = real, i = imag)

```

4.6.1.5. Cadenas de documentación

Los comentarios deben ser oraciones completas. Si un comentario es una frase u oración, su primera palabra debe comenzar con mayúscula, a menos que sea un identificador que comienza con minúscula.

Nunca se deben de cambiar las mayúsculas/minúsculas de los identificadores

4.6.1.6. Comentarios en Bloque

Generalmente consisten en uno o más párrafos compuestos por oraciones completas, por lo que cada una de ellas debe finalizar en un punto. Deberías usar dos espacios luego de una oración que termine con un punto.

Se aplican a algunos (o todos) códigos que los siguen, y están indentados al mismo nivel que ese código. Cada línea de un comentario en bloque comienza con un # (numeral) y un espacio (a menos que esté indentado dentro del mismo comentario). Los párrafos dentro de un comentario en bloque están separados por una línea que contiene únicamente un # (numeral).

```

1 # Comentario por bloque de la funcionalidad del metodo
2 # para obtener el sonido del audio seleccionado,
3 # y posteriormente obtener el texto correspondiente

```

4.6.1.7. Comentarios en Línea

Usa comentarios en línea escasamente. Un comentario en línea es aquel que se encuentra en la misma línea que una sentencia. Los comentarios en línea deberían estar separados por al menos dos espacios de la sentencia. Deberían empezar con un # (numeral) seguido de un espacio.

Los comentarios en línea son innecesarios y de hecho molestos si su acción es obvia. No hacer lo siguiente:

```
1 x = x + 1 # Incrementar x
```

4.6.1.8. Cadenas de Documentación(alias “docstrings”)

Se deben de escribir “docstrings” para todos los módulos, funciones, clases y métodos públicos. Los “docstrings” no son necesarios para los métodos no públicos, pero deberían tener un comentario que describa su función.

Los “docstrings” se utilizan digitando tres comillas dobles (“”) tanto para el principio como el final. En estos se suele describir la funcionalidad de cierto código, a diferencia de los comentarios en bloque, los “docstring” son más extensos ya que cuentan con un mayor grado de descripción.

```
1 """
2     Return a foobang
3     Optional plotz says to frobnicate the bizbaz first.
4 """
```


4.7. Pruebas de rendimiento

4.7.1. Rendimiento con datos controlados

4.7.1.1. Rendimiento de conversión de texto a voz

Condiciones de ejecución:

1. Velocidad de Internet

Bajada: 3.2Mbps/Subida: 0.5Mbps

| Prueba de rendimiento conversión de texto a voz | | | |
|---|----------------------|---------------|---------------------------|
| Caracteres | 4084 | Palabras | 704 |
| Herramienta | Tiempo de generación | # de archivos | Tiempo del audio generado |
| CSTS | 15.19s | 6 | 4m10s |
| gTTS | 2m1.65s | 7 | 5m12s |
| VoiceRSS | 18.44s | 5 | 4m19s |
| Festival | 27.84s | 1 | 3m51s |

Tabla 4.41: Prueba de rendimiento TTS N° 1

| Prueba de rendimiento conversión de texto a voz | | | |
|---|----------------------|---------------|---------------------------|
| Caracteres | 1458 | Palabras | 235 |
| Herramienta | Tiempo de generación | # de archivos | Tiempo del audio generado |
| CSTS | 11.36s | 3 | 1m30s |
| gTTS | 51.34s | 3 | 2m14s |
| VoiceRSS | 9.48s | 2 | 1m33s |
| Festival | 11.35s | 1 | 1m24s |

Tabla 4.42: Prueba de rendimiento TTS N° 2

| Prueba de rendimiento conversión de texto a voz | | | |
|---|----------------------|---------------|---------------------------|
| Caracteres | 40898 | Palabras | 6701 |
| Herramienta | Tiempo de generación | # de archivos | Tiempo del audio generado |
| CSTS | 8m30.51s | 59 | 42m36s |
| gTTS | 11m41.98s | 63 | 1h4m |
| VoiceRSS | 2m52.45s | 48 | 44m24s |
| Festival | 27m39.54s | 1 | 40m10s |

Tabla 4.43: Prueba de rendimiento TTS N° 3

4.7.1.2. Rendimiento de conversión de voz a texto

Condiciones de ejecución:

1. Velocidad de Internet

Bajada: 3.2Mbps / Subida: 0.5Mbps

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|----------------|--------|-----------|--------|
| Comando | Lectura (N° 1) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_leer | [N/A] | 100 | 14.44s |
| IBM | comando_leer | [N/A] | 100 | 3.18s |
| Google | comando_leer | [N/A] | 100 | 3.41s |
| CSST | comando_leer | [N/A] | 100 | 4.15s |
| Sphinx | comando_leer | [N/A] | 100 | 1.55s |
| Híbrido - (Sphinx+Wit) | comando_leer | [N/A] | 100 | 1.53s |

Tabla 4.44: Prueba de rendimiento STT N° 1

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|----------------|--------|-----------|--------|
| Comando | Lectura (N° 2) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_leer | [N/A] | 100 | 11.36s |
| IBM | comando_leer | [N/A] | 100 | 5.97s |
| Google | comando_leer | [N/A] | 100 | 4.05s |
| CSST | comando_leer | [N/A] | 100 | 7.38s |
| Sphinx | comando_leer | [N/A] | 100 | 1.57s |
| Híbrido - (Sphinx+Wit) | comando_leer | [N/A] | 100 | 1.76s |

Tabla 4.45: Prueba de rendimiento STT N° 2

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|------------------------------|------------------------|-----------|--------|
| Comando | Buscar biografía de salarrué | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_buscar | biografía de sala rue | 83.33 | 12.61s |
| IBM | comando_buscar | biografía de sala rue | 83.33 | 4.13s |
| Google | comando_buscar | biografía de salarrue | 100 | 3.84s |
| CSST | comando_buscar | biografía de sala rue | 83.33 | 3.84s |
| Sphinx | comando_buscar | dio gracia de sala rui | 66.67 | 39.19s |
| Híbrido - (Sphinx+Wit) | comando_buscar | biografía de sala rue | 83.33 | 12.07s |

Tabla 4.46: Prueba de rendimiento STT N° 3

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|----------------|---|-----------|--------|
| Comando | Buscar | interacción del oro con otros elementos | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_buscar | interacción del oro con otros elementos | 100 | 11.04s |
| IBM | comando_buscar | interacción del oro con otros elementos | 100 | 4.54s |
| Google | comando_buscar | interacción del oro con otros elementos | 100 | 4.36s |
| CSST | comando_buscar | interacción del oro con otros elementos | 100 | 4.05s |
| Sphinx | comando_buscar | interacción del corro con otros elementos | 91.67 | 39.22s |
| Híbrido - (Sphinx+Wit) | comando_buscar | interacción del oro con otros elementos | 100 | 13.06s |

Tabla 4.47: Prueba de rendimiento STT N° 4

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|--------------------|--------|-----------|--------|
| Comando | Navegación avanzar | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_navegacion | 10 | 100 | 8.45s |
| IBM | comando_navegacion | 10 | 100 | 4.02s |
| Google | comando_navegacion | 10 | 100 | 3.29s |
| CSST | comando_navegacion | 10 | 100 | 3.58s |
| Sphinx | comando_navegacion | 10 | 100 | 1.51s |
| Híbrido - (Sphinx+Wit) | comando_navegacion | 10 | 100 | 1.48s |

Tabla 4.48: Prueba de rendimiento STT N° 5

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|---------------------|--------|-----------|--------|
| Comando | Navegación regresar | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_navegacion | -10 | 100 | 10.58s |
| IBM | comando_navegacion | -10 | 100 | 3.54s |
| Google | comando_navegacion | -10 | 100 | 3.11s |
| CSST | comando_navegacion | -10 | 100 | 3.31s |
| Sphinx | comando_navegacion | -10 | 100 | 1.55s |
| Híbrido - (Sphinx+Wit) | comando_navegacion | -10 | 100 | 1.64s |

Tabla 4.49: Prueba de rendimiento STT N° 6

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|-------------------|--------|-----------|--------|
| Comando | Resultado (1) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_resultado | 1 | 100 | 11.71s |
| IBM | comando_resultado | 1 | 100 | 3.16s |
| Google | comando_resultado | 1 | 100 | 4.89s |
| CSST | comando_resultado | 1 | 100 | 9.98s |
| Sphinx | comando_resultado | 3 | 50 | 1.39s |
| Híbrido - (Sphinx+Wit) | comando_resultado | 3 | 50 | 1.46s |

Tabla 4.50: Prueba de rendimiento STT N° 7

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|-------------------|--------|-----------|--------|
| Comando | Resultado (3) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_resultado | 3 | 100 | 10.33s |
| IBM | comando_resultado | 3 | 100 | 2.78s |
| Google | comando_resultado | 3 | 100 | 2.71s |
| CSST | comando_resultado | 3 | 100 | 3.34s |
| Sphinx | comando_resultado | 6 | 50 | 1.25s |
| Híbrido - (Sphinx+Wit) | comando_resultado | 3 | 100 | 1.47s |

Tabla 4.51: Prueba de rendimiento STT N° 8

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|-------------------|--------|-----------|--------|
| Comando | Resultado (7) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_resultado | 7 | 100 | 12.24s |
| IBM | comando_resultado | 7 | 100 | 4.15s |
| Google | comando_resultado | 7 | 100 | 3.97s |
| CSST | comando_resultado | 7 | 100 | 6.43s |
| Sphinx | comando_resultado | 7 | 100 | 1.29s |
| Híbrido - (Sphinx+Wit) | comando_resultado | 7 | 100 | 1.56s |

Tabla 4.52: Prueba de rendimiento STT N° 9

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|-------------------|--------|-----------|--------|
| Comando | Resultado (10) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_resultado | 10 | 100 | 10.98s |
| IBM | comando_resultado | 10 | 100 | 2.57s |
| Google | comando_resultado | 10 | 100 | 2.55s |
| CSST | comando_resultado | 10 | 100 | 2.99s |
| Sphinx | comando_resultado | 10 | 100 | 1.4s |
| Híbrido - (Sphinx+Wit) | comando_resultado | 10 | 100 | 1.36s |

Tabla 4.53: Prueba de rendimiento STT N° 10

| Prueba de rendimiento conversión de voz a texto | | | | |
|--|-------------------|--------|-----------|--------|
| Comando | Resultado (11) | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_resultado | 11 | 100 | 9.9s |
| IBM | comando_resultado | 11 | 100 | 2.84s |
| Google | comando_resultado | 11 | 100 | 2.76s |
| CSST | comando_resultado | 11 | 100 | 2.99s |
| Sphinx | comando_resultado | 3 | 50.0 | 1.32s |
| Híbrido - (Sphinx+Wit) | comando_resultado | 3 | 50.0 | 2.04s |

Tabla 4.54: Prueba de rendimiento STT N° 11

| Prueba de rendimiento conversión de voz a texto | | | | | |
|--|-------------|--------|-----------|--------|--|
| Comando | Libro (1) | | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo | |
| Wit.AI | comando_pdf | 1 | 100 | 11.03s | |
| IBM | comando_pdf | 1 | 100 | 2.9s | |
| Google | comando_pdf | 1 | 100 | 3.0s | |
| CSST | comando_pdf | 1 | 100 | 2.88s | |
| Sphinx | comando_pdf | 1 | 100 | 36.55s | |
| Híbrido - (Sphinx+Wit) | comando_pdf | 1 | 100 | 7.86s | |

Tabla 4.55: Prueba de rendimiento STT N° 12

| Prueba de rendimiento conversión de voz a texto | | | | | |
|--|-------------|--------|-----------|--------|--|
| Comando | Libro (2) | | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo | |
| Wit.AI | comando_pdf | 2 | 100 | 9.03s | |
| IBM | comando_pdf | 2 | 100 | 2.66s | |
| Google | comando_pdf | 2 | 100 | 2.43s | |
| CSST | comando_pdf | 2 | 100 | 3.28s | |
| Sphinx | comando_pdf | 2 | 100 | 36.81s | |
| Híbrido - (Sphinx+Wit) | comando_pdf | 2 | 100 | 9.14s | |

Tabla 4.56: Prueba de rendimiento STT N° 13

| Prueba de rendimiento conversión de voz a texto | | | | | |
|--|---------------------|--------|-----------|--------|--|
| Comando | Libro avanzar | | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo | |
| Wit.AI | comando_pdf_navegar | 1 | 100 | 10.66s | |
| IBM | comando_pdf_navegar | 1 | 100 | 2.6s | |
| Google | comando_pdf_navegar | 1 | 100 | 2.471s | |
| CSST | comando_pdf_navegar | 1 | 100 | 3.17s | |
| Sphinx | comando_pdf_navegar | 1 | 100 | 37.8s | |
| Híbrido - (Sphinx+Wit) | comando_pdf_navegar | 1 | 100 | 1.54s | |

Tabla 4.57: Prueba de rendimiento STT N° 14

| Prueba de rendimiento conversión de voz a texto | | | | | |
|--|---------------------|--------|-----------|--------|--|
| Comando | Libro regresar | | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo | |
| Wit.AI | comando_pdf_navegar | -1 | 100 | 9.3s | |
| IBM | comando_pdf_navegar | -1 | 100 | 2.76s | |
| Google | comando_pdf_navegar | -1 | 100 | 2.47s | |
| CSST | comando_pdf_navegar | -1 | 100 | 3.8s | |
| Sphinx | comando_pdf_navegar | -1 | 100 | 36.1s | |
| Híbrido - (Sphinx+Wit) | comando_pdf_navegar | -1 | 100 | 9.24s | |

Tabla 4.58: Prueba de rendimiento STT N° 15

| Prueba de rendimiento conversión de voz a texto | | | | |
|---|-----------------------|--------|-----------|--------|
| Comando | Repetir texto | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_repetir_texto | [N/A] | 100 | 10.6s |
| IBM | comando_repetir_texto | [N/A] | 100 | 2.66s |
| Google | comando_repetir_texto | [N/A] | 100 | 2.41s |
| CSST | comando_repetir_texto | [N/A] | 100 | 3.2s |
| Sphinx | comando_repetir_texto | [N/A] | 100 | 38.48s |
| Híbrido - (Sphinx+Wit) | comando_repetir_texto | [N/A] | 100 | 1.54s |

Tabla 4.59: Prueba de rendimiento STT N° 16

| Prueba de rendimiento conversión de voz a texto | | | | |
|---|-----------------------|--------|-----------|--------|
| Comando | Repetir lista | | | |
| Herramienta | Comando | Salida | % acierto | Tiempo |
| Wit.AI | comando_repetir_lista | | 100 | 7.47s |
| IBM | comando_repetir_lista | | 100 | 2.87s |
| Google | comando_repetir_lista | | 100 | 2.8s |
| CSST | comando_repetir_lista | | 100 | 3.34s |
| Sphinx | comando_repetir_lista | | 100 | 38.37s |
| Híbrido - (Sphinx+Wit) | comando_repetir_lista | | 100 | 9.52s |

Tabla 4.60: Prueba de rendimiento STT N° 17

4.7.1.3. Rendimiento de conversión de imagen a texto

Condiciones de ejecución:

1. Velocidad de Internet

Bajada: 3.2Mbps / Subida: 0.5Mbps

| Prueba de rendimiento OCR | | |
|---------------------------|-----------------------------------|----------------|
| Palabras | 320 | |
| Herramienta | Tiempo hasta obtener texto válido | % de precisión |
| Azure Computer Vision | 41.56s | 98 |
| OCR.Space | 38.65s | 98 |
| Tesseract OCR | 2m30.93s | 93 |
| Google Drive OCR | 1m4.51s | 98 |

Tabla 4.61: Prueba de rendimiento OCR N° 1

| Prueba de rendimiento OCR | | |
|---------------------------|-----------------------------------|----------------|
| Palabras | 753 | |
| Herramienta | Tiempo hasta obtener texto válido | % de precisión |
| Azure Computer Vision | 31.16s | 99 |
| OCR.Space | 34.63s | 99 |
| Tesseract OCR | 1m47.39s | 98 |
| Google Drive OCR | 55.39s | 82 |

Tabla 4.62: Prueba de rendimiento OCR N° 2

| Prueba de rendimiento OCR | | | |
|---------------------------|-----------------------------------|----------------|-----|
| Palabras | | | 595 |
| Herramienta | Tiempo hasta obtener texto válido | % de precisión | |
| Azure Computer Vision | 41.01s | 99 | |
| OCR.Space | 34.27s | 98 | |
| Tesseract OCR | 2m2.21s | 93 | |
| Google Drive OCR | 51.22s | 62 | |

Tabla 4.63: Prueba de rendimiento OCR N° 3

| Prueba de rendimiento OCR | | | |
|---------------------------|-----------------------------------|----------------|-----|
| Palabras | | | 680 |
| Herramienta | Tiempo hasta obtener texto válido | % de precisión | |
| Azure Computer Vision | 40.05s | 100 | |
| OCR.Space | 1m38.4s | 99 | |
| Tesseract OCR | 2m2.57s | 96 | |
| Google Drive OCR | 42.02s | 91 | |

Tabla 4.64: Prueba de rendimiento OCR N° 4

| Prueba de rendimiento OCR | | | |
|---------------------------|-----------------------------------|----------------|-----|
| Palabras | | | 885 |
| Herramienta | Tiempo hasta obtener texto válido | % de precisión | |
| Azure Computer Vision | 30.82s | 99 | |
| OCR.Space | 33.64s | 98 | |
| Tesseract OCR | 2m0.57s | 88 | |
| Google Drive OCR | 47.41s | 87 | |

Tabla 4.65: Prueba de rendimiento OCR N° 5

4.7.2. Valoraciones de rendimiento

Tras las pruebas de rendimiento realizadas a las distintas herramientas utilizadas en el desarrollo del prototipo¹, se hace una valoración comparativa de estas en sus diferentes áreas para determinar la combinación idónea para las posteriores pruebas experimentales.

4.7.2.1. Text to Speech (TTS)

Se hace un promedio de todas las pruebas de rendimiento realizadas a las herramientas de conversión de texto a voz¹, según su rendimiento en términos de velocidad de procesamiento, cantidad de archivos de audio producidos y la duración de dichos archivos, obteniéndose la tabla siguiente.

¹Véase sección 4.1 pág. 122

¹Véase sección 4.7.1.1 pág. 242

| Herramienta | Tiempo de generación | # de archivos | Tiempo del audio generado |
|-------------|----------------------|---------------|---------------------------|
| CSTS | 2m59.02s | ±23 | 16m5s |
| gTTS | 14m34.97s | ±24 | 1h7m26s |
| VoiceRSS | 3m20.37s | ±18 | 50m16s |
| Festival | 28m18.73s | 1 | 45m25s |

Tabla 4.66: Resumen de rendimiento de conversión texto a voz

De los resultados obtenidos se sacan las siguientes valoraciones:

1. La herramienta de Text to Speech (TTS) en finalizar sus procesos primero, siendo por mucho la más rápida, es la herramienta CSTS (Cognitive Service Speech Servive).
2. La herramienta de TTS con la salida de audio más clara y con mayor calidad es la herramienta gTTS de Google, siendo también la segunda más lenta.
3. La herramienta con la mejor relación entre calidad y tiempo de procesamiento es la herramienta VoiceRSS siendo considerablemente más rápida que la herramienta gTTS y con mejor calidad de audio que la herramienta Festival.

Con las valoraciones anteriores se define la herramienta VoiceRSS como primera opción para ser la herramienta TTS utilizada en el proceso experimental.

4.7.2.2. Speech to Text (STT)

Se hace un promedio de todas las pruebas de rendimiento realizadas a las herramientas de conversión de voz a texto², según su rendimiento en términos de velocidad de procesamiento y el porcentaje de acierto, es decir que tan bien se convirtió la voz del usuario a texto utilizable y entendible, obteniéndose la tabla siguiente.

| Herramienta | Velocidad | % acierto |
|------------------------|-----------|-----------|
| Wit.AI | 10.69s | 99.0194 |
| IBM | 3.37s | 99.0194 |
| Google | 3.21s | 100.00 |
| CSST | 4.22s | 99.0194 |
| Sphinx | 18.55s | 82.8435 |
| Híbrido - (Sphinx+Wit) | 4.6s | 93.1371 |

Tabla 4.67: Resumen de rendimiento de conversión de voz a texto

De los resultados obtenidos se sacar las siguientes valoraciones:

1. La herramienta de Speech to Text (STT) en finalizar sus procesos primero fue la herramienta Google STT.
2. La herramienta de STT más precisa fue la herramienta de Google STT, interpretando correctamente todas

²Véase sección 4.7.1.2 pág. 243

las palabras del usuario en todos los casos presentados.

3. La herramienta de STT con la mejor relación entre porcentaje de acierto y velocidad de procesamiento es la herramienta de Google STT.

Con las valoraciones anteriores se define la herramienta Google STT como primera opción para ser la herramienta STT utilizada en el proceso experimental.

4.7.2.3. Optical Character Recognition (OCR)

Se hace un promedio de todas las pruebas de rendimiento realizadas a las herramientas de reconocimiento de caracteres¹, según su rendimiento en términos de velocidad de procesamiento y el porcentaje de acierto, es decir que tan bien extraen el texto de un libro físico a texto utilizable y entendible por el ordenador, obteniéndose la tabla siguiente.

| Herramienta | Velocidad | % acierto |
|-----------------------|-----------|-----------|
| Azure Computer Vision | 36.94s | 99.00 |
| OCR Space | 47.99s | 98.40 |
| Tesseract OCR | 2m4.71s | 93.60 |
| Google Drive OCR | 51.91s | 84.00 |

Tabla 4.68: Resumen de rendimiento de conversión de imagen a texto

De los resultados obtenidos se enlistan las siguientes valoraciones:

1. La herramienta de Optical Character Recognition (OCR) en finalizar sus procesos primero fue la herramienta; Tesseract OCR.
2. La herramienta de OCR más precisa fue la herramienta; Azure Computer Vision, interpretando los textos utilizados con un porcentaje de casi el 100 % en todas las pruebas. La herramienta de OCR con la mejor relación entre porcentaje de acierto y velocidad de procesamiento igualmente es la herramienta; Azure Computer Vision.

Con las valoraciones anteriores se define la herramienta Azure Computer Vision OCR como primera opción para ser la herramienta OCR utilizada en el proceso experimental.

¹Véase sección 4.7.1.3 pág. 247

4.8. Experimentación

4.8.1. Tabulación de encuestas experimentales

4.8.1.1. Primera encuesta realizada al grupo experimental

Personas encuestadas = 39

1. ¿Qué tanto se le dificultó realizar la lectura de textos físicos?

Objetivo: Determinar la dificultad con la que el encuestado realizó la lectura de textos físicos en el primer escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 0 | 2 | 2 | 35 |

Tabla 4.69: Respuesta pregunta N° 1

Conclusión: Los resultados reflejan que siendo una persona con discapacidad visual, el leer un libro físico resulta sumamente complicado o hasta imposible según el primer escenario planteado durante el experimento.

2. ¿Qué tanto se le dificultó realizar la lectura de textos digitales?

Objetivo: Determinar la dificultad con la que el encuestado realizó la lectura de textos digitales en el primer escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 0 | 3 | 5 | 31 |

Tabla 4.70: Respuesta pregunta N° 2

Conclusión: Los resultados reflejan que siendo una persona con discapacidad visual, el leer un libro digital resulta sumamente complicado o hasta imposible según el primer escenario planteado durante el experimento.

3. ¿Qué tanto se le dificultó realizar la búsqueda en línea?

Objetivo: Determinar la dificultad con la que el encuestado realizó la búsqueda de información en línea en el primer escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 1 | 3 | 8 | 27 |

Tabla 4.71: Respuesta pregunta N° 3

Conclusión: Los resultados reflejan que siendo una persona con discapacidad visual, el realizar búsquedas de información en línea resulta sumamente complicado o hasta imposible según el primer escenario planteado

durante el experimento.

4. ¿Qué tan familiarizado está con el uso de ordenadores?

Objetivo: Indagar sobre que tan familiar le es al encuestado el uso de ordenadores en su día a día.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 0 | 0 | 4 | 35 |

Tabla 4.72: Respuesta pregunta N° 4

Conclusión: Los resultados reflejan que los encuestados tienen una alta familiaridad con la tecnología, específicamente con los ordenadores.

5. ¿Qué tan útiles opina que son los ordenadores para estudiar?

Objetivo: Saber qué tan útiles considera el encuestado que son los ordenadores en el ámbito académico

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 0 | 0 | 4 | 35 |

Tabla 4.73: Respuesta pregunta N° 5

Conclusión: Los resultados reflejan que los encuestados consideran muy útil el uso de tecnología, específicamente ordenadores, en el desarrollo académico y para la realización de labores educativas.

4.8.1.2. Segunda encuesta realizada al grupo experimental.

1. ¿Qué tanto se le dificultó realizar la lectura de textos físicos?

Objetivo: Determinar la dificultad con la que el encuestado realizó la lectura de textos físicos en el segundo escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 3 | 33 | 0 | 3 | 0 |

Tabla 4.74: Respuesta pregunta N° 1

Conclusión: Los resultados reflejan que para una persona con discapacidad visual, el leer un libro físico, haciendo uso del prototipo, resulta fácil, según el segundo escenario planteado durante el experimento.

2. ¿Qué tanto se le dificultó realizar la lectura de textos digitales?

Objetivo: Determinar la dificultad con la que el encuestado realizó la lectura de textos digitales en el segundo escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 1 | 35 | 3 | 0 | 0 |

Tabla 4.75: Respuesta pregunta N° 2

Conclusión: Los resultados reflejan que para una persona con discapacidad visual, el leer un libro digital, haciendo uso del prototipo, resulta fácil, según el segundo escenario planteado durante el experimento.

3. ¿Qué tanto se le dificultó realizar la búsqueda en línea?

Objetivo: Determinar la dificultad con la que el encuestado realizó la búsqueda de información en línea, en el segundo escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 18 | 21 | 0 | 0 | 0 |

Tabla 4.76: Respuesta pregunta N° 3

Conclusión: Los resultados reflejan que para una persona con discapacidad visual, el hacer búsquedas de información en línea, haciendo uso del prototipo es muy fácil, según el segundo escenario planteado durante el experimento.

4. ¿Cómo considera que fue el tiempo de reacción del prototipo para responder a sus indicaciones?

Objetivo: Conocer la perspectiva del encuestado respecto a la velocidad de reacción del prototipo durante el segundo escenario del experimento verdadero.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 0 | 26 | 9 | 4 |

Tabla 4.77: Respuesta pregunta N° 4

Conclusión: Los resultados reflejan que el encuestado tiene una valoración favorable respecto a la velocidad de respuesta del prototipo, según el segundo escenario planteado durante el experimento.

5. ¿Qué tan difícil resultó usar el prototipo para realizar las actividades planteadas?

Objetivo: Conocer qué tan difícil resulta, para una persona con discapacidad visual, el hacer uso del prototipo.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 17 | 20 | 2 | 0 | 0 |

Tabla 4.78: Respuesta pregunta N° 5

Conclusión: Los resultados reflejan que el prototipo es de fácil uso para personas con discapacidad visual,

según el segundo escenario planteado durante el experimento verdadero.

6. En comparación con el escenario anterior, sin el uso de prototipo, ¿Qué tanto considera que se ha facilitado la realización de las actividades planteadas con el uso de este?

Objetivo: Valorar el beneficio producido por el prototipo, en el mejoramiento de la calidad de estudio que personas con discapacidad visual tienen al hacer uso de este.

| Nada | Poco | Algo | Normal | Mucho |
|------|------|------|--------|-------|
| 0 | 0 | 2 | 5 | 32 |

Tabla 4.79: Respuesta pregunta N° 6

Conclusión: Los resultados reflejan que el uso del prototipo tiene un impacto muy positivo en el mejoramiento de la calidad de estudio de personas con discapacidad visual, según el segundo escenario planteado durante el experimento verdadero.

4.8.2. Análisis de datos experimentales

Se presenta el análisis del nivel de significancia de los dos instrumentos utilizados para el experimento de esta investigación.

Se obtiene el valor de acierto (V_a) de cada instrumento utilizado, se detalla a continuación:

| Primer instrumento aplicado durante el experimento verdadero (población=39) | | | | | |
|---|-----------|-----------|-----------|-----------|-----------|
| Pregunta | 1 | 2 | 3 | 4 | 5 |
| Respuesta esperada | Mucho(35) | Mucho(35) | Mucho(35) | Mucho(35) | Mucho(35) |
| | Normal(4) | Normal(4) | Normal(4) | Normal(4) | Normal(4) |
| Respuesta obtenida | Mucho(35) | Mucho(31) | Mucho(27) | Mucho(35) | Mucho(35) |
| | Normal(2) | Normal(5) | Normal(8) | Normal(4) | Normal(4) |
| | Algo(2) | Algo(3) | Algo(3) | Algo(0) | Algo(0) |
| | Poco(0) | Poco(0) | Poco(1) | Poco(0) | Poco(0) |
| | Nada(0) | Nada(0) | Nada(0) | Nada(0) | Nada(0) |
| % de acierto | 95.0 | 92.21 | 89.43 | 100 | 100 |

Tabla 4.80: Resultados primer instrumento aplicado

$$Va_2 = \frac{\sum(de\ acierto)}{5(100)} = \frac{95,0 + 92,21 + 89,43 + 100 + 100}{500} = \frac{476,64}{500} = 0,95$$

| Segundo instrumento aplicado durante el experimento verdadero (población=39) | | | | | | |
|--|-----------|-----------|---------------|---------------|-----------|-----------|
| Pregunta | 1 | 2 | 3 | 4 | 5 | 6 |
| Respuesta esperada | Poco(35) | Poco(35) | Poco(20) | Normal(25) | Poco(20) | Mucho(35) |
| | 90 % | 90 % | 50 % | 65 % | 50 % | 90 % |
| | Nada(4) | Nada(4) | Nada(19) | Rápido(10) | Nada(19) | Normal(4) |
| | 10 % | 10 % | 50 % | 25 % | 50 % | 10 % |
| Respuesta obtenidas | Mucho(0) | Mucho(0) | Mucho(0) | Muy rápido(4) | Mucho(0) | Mucho(32) |
| | Normal(3) | Normal(0) | Normal(0) | 10 % | Normal(0) | Normal(5) |
| | Algo(0) | Algo(3) | Algo(0) | Muy lento(0) | Algo(2) | Algo(2) |
| | Poco(33) | Poco(35) | Poco(21) | Lento(0) | Poco(20) | Poco(0) |
| | Nada(3) | Nada(1) | Nada(18) | Normal(26) | Nada(17) | Nada(0) |
| | | | | Rápido(9) | | |
| | | | Muy Rápido(4) | | | |
| % de acierto | 92.36 | 92.50 | 99.87 | 100.00 | 94.74 | 90.29 |

Tabla 4.81: Resultados segundo instrumento aplicado

$$Va_2 = \frac{\sum(de\ acierto)}{6(100)} = \frac{92,36 + 92,5 + 99,87 + 100,1 + 94,74 + 90,29}{600} = \frac{569,86}{600} = 0,9497 \simeq 0,95$$

Al obtener el valor de acierto de todos los instrumentos se saca el valor de acierto medio, para así determinar el valor de acierto del experimento.

$$\bar{V}a = \frac{Va_1 + Va_2}{2} = \frac{0,95 + 0,95}{2} = \frac{1,9}{2} = 0,95$$

Para obtener el nivel de significancia de un experimento, se hace uso de la siguiente fórmula:

$$Nivel\ de\ significancia = 1 - Va = 1 - 0,95 = 0,05$$

Según Sampieri¹ el experimento de una investigación social es representativo si su nivel de significancia es menor o igual a 0.05, es decir, tiene un 95 % de veracidad, y es aceptado como herramienta para realizar generalizaciones sobre las variables estudiadas y su comportamiento previamente supuesto en las hipótesis.

El nivel de significancia del experimento desarrollado es de 0.05, es decir, cuenta con un 95 % de veracidad en sus resultados y puede ser utilizado para realizar análisis respecto a las variables de esta investigación.

Para analizar la información obtenida se hace una evaluación de la variable dependiente² (Vd) representada

¹ Sampieri, *Metodología de la investigación*; ed, pág. 350.

² Véase sección 3.1.6.2. Experimento verdadero, pág. 101

por las actividades realizadas (lectura de un texto físico, lectura de un texto digital y la búsqueda de información) durante el experimento en el primer escenario, es decir, donde la variable independiente¹ (Vi) no está presente, a esta etapa se denominará “P”, y luego se evalúan las variables en el segundo escenario, es decir, donde la variable independiente² (Vi) si está presente, a esta etapa se denominará “Q”.

Para realizar un análisis adecuado de los resultados, es necesario hacer una asignación de valores a las respuestas posibles, creando así una escala de dificultad.

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|-------------------|------|------|------|--------|-------|
| Cantidad | 1 | 2 | 3 | 4 | 5 |

Tabla 4.82: Escala de respuestas

4.8.2.1. Análisis de P

Lectura de Textos Físicos

Las respuestas del primer escenario para la lectura de textos físicos son:

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|-------------------|------|------|------|--------|-------|
| Cantidad | 0 | 0 | 2 | 2 | 35 |

Tabla 4.83: Respuesta lectura de textos físicos P

Se multiplica la cantidad de respuestas por el valor numérico que les corresponde, estas después se suman entre sí y se divide el resultado entre la cantidad de encuestas realizadas, obteniendo así el primer valor de las actividades en el escenario uno (P1).

$$P_1 = \frac{(1)(0) + (2)(0) + (3)(2) + (4)(2) + (5)(35)}{39} = \frac{0 + 0 + 6 + 8 + 175}{39} = \frac{189}{39} = 4,846153846 \simeq 4,85$$

Lectura de Textos Digitales

Las respuestas del primer escenario para la lectura de textos digitales son:

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|-------------------|------|------|------|--------|-------|
| Cantidad | 0 | 0 | 3 | 5 | 31 |

Tabla 4.84: Respuesta lectura de textos digitales P

Se multiplica la cantidad de respuestas por el valor numérico que les corresponde, estas después se suman

¹Véase sección 3.1.6.2. Experimento verdadero, pág. 101

²Véase sección 3.1.6.2. Experimento verdadero, pág. 101

entre sí y se divide el resultado entre la cantidad de encuestas realizadas, obteniendo así el segundo valor de las actividades en el escenario uno (P_2).

$$P_2 = \frac{(1)(0) + (2)(0) + (3)(3) + (4)(5) + (5)(31)}{39} = \frac{0 + 0 + 9 + 20 + 155}{39} = \frac{184}{39} = 4,717948718 \simeq 4,72$$

Búsqueda en Línea

Las respuestas del primer escenario para la búsqueda de información en línea son:

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|------------|------|------|------|--------|-------|
| Cantidad | 0 | 0 | 3 | 8 | 27 |

Tabla 4.85: Respuesta búsqueda en línea P

Se multiplica la cantidad de respuestas por el valor numérico que les corresponde, estas después se suman entre sí y se divide el resultado entre la cantidad de encuestas realizadas, obteniendo así el tercer valor de las actividades en el escenario uno (P_3).

$$P_3 = \frac{(1)(0) + (2)(1) + (3)(3) + (4)(8) + (5)(27)}{39} = \frac{0 + 2 + 9 + 32 + 135}{39} = \frac{178}{39} = 4,564102564 \simeq 4,56$$

Se obtiene el valor medio de las actividades del primer escenario (\bar{P}), para determinar el nivel de dificultad que representó a los sujetos evaluados, mientras más alto sea el valor, mayor es la dificultad de “P”.

$$\bar{P} = \frac{\sum P_i}{3} = \frac{P_1 + P_2 + P_3}{3} = \frac{4,85 + 4,72 + 4,56}{3} = \frac{14,13}{3} = 4,71$$

El valor de dificultad obtenido para el escenario “P” es de 4.71, es decir, se acerca mucho al valor máximo de la escala, por lo que se puede determinar que el escenario “P” es muy difícil de realizar para los sujetos evaluados.

4.8.2.2. Análisis de Q

Lectura de Textos Físicos

Las respuestas del segundo escenario para la lectura de textos físicos son:

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|-------------------|------|------|------|--------|-------|
| Cantidad | 3 | 33 | 0 | 6 | 0 |

Tabla 4.86: Respuesta lectura de textos físicos Q

Se multiplica la cantidad de respuestas por el valor numérico que les corresponde, estas después se suman entre sí y se divide el resultado entre la cantidad de encuestas realizadas, obteniendo así el primer valor de las actividades en el escenario dos (Q_1).

$$Q_1 = \frac{(1)(3) + (2)(33) + (3)(0) + (4)(6) + (5)(0)}{39} = \frac{3 + 66 + 0 + 24 + 0}{39} = \frac{93}{39} = 2,384 \simeq 2,38$$

Lectura de Textos Digitales

Las respuestas del segundo escenario para la lectura de textos digitales son:

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|-------------------|------|------|------|--------|-------|
| Cantidad | 1 | 35 | 3 | 0 | 0 |

Tabla 4.87: Respuesta lectura de textos digitales Q

Las respuestas del segundo escenario para la lectura de textos digitales son (Q_2):

$$Q_2 = \frac{(1)(1) + (2)(25) + (3)(3) + (4)(0) + (5)(0)}{39} = \frac{1 + 50 + 9 + 0 + 0}{39} = \frac{60}{39} = 1,538 \simeq 1,54$$

Búsqueda en Línea

Las respuestas del segundo escenario para la búsqueda de información en línea son:

| Respuestas | Nada | Poco | Algo | Normal | Mucho |
|-------------------|------|------|------|--------|-------|
| Cantidad | 18 | 21 | 0 | 0 | 0 |

Tabla 4.88: Respuesta búsqueda en línea Q

Se multiplica la cantidad de respuestas por el valor numérico que les corresponde, estas después se suman entre sí y se divide el resultado entre la cantidad de encuestas realizadas, obteniendo así el tercer valor de las actividades en el escenario dos (Q_3).

$$Q_3 = \frac{(1)(18) + (2)(21) + (3)(0) + (4)(0) + (5)(0)}{39} = \frac{18 + 42 + 0 + 0 + 0}{39} = \frac{60}{39} = 1,538 \simeq 1,54$$

Se obtiene el valor medio de las actividades del segundo escenario (\bar{Q}), para determinar el nivel de dificultad que representó a los sujetos evaluados, mientras más alto sea el valor, mayor es la dificultad de “Q”.

$$\bar{Q} = \frac{\sum Q_i}{3} = \frac{Q_1 + Q_2 + Q_3}{3} = \frac{2,38 + 1,54 + 1,54}{3} = \frac{5,46}{3} = 1,82$$

El valor de dificultad obtenido para el escenario “Q” es de 1.82, es decir, se acerca mucho al valor mínimo de la escala, por lo que se puede determinar que el escenario “Q” es muy fácil de realizar para los sujetos evaluados.

5. Conclusiones y Recomendaciones

El propósito original de esta investigación era resaltar una necesidad presente en la Universidad de El Salvador, específicamente en la Facultad Multidisciplinaria Oriental, respecto a la poca asistencia que reciben personas con discapacidades físicas, puntualmente hablando en el ámbito de formación académica, por lo que esta investigación pretende servir como uno de los pilares que cimienten las bases para un plan más inclusivo y dinámico que permita nivelar las condiciones de todos los estudiantes para acceder a las herramientas que la institución brinda.

Las conclusiones y recomendaciones que se presentan en este documento, se enfocan a dar cierre a la investigación realizada, la cual ha sido desarrollada en base a las problemáticas planteadas al principio de la misma, por lo tanto, es necesario que estas sean valoradas bajo el contexto del servicio y asistencia a personas con discapacidad visual en la Universidad de El Salvador. Los resultados obtenidos en la investigación presentan varios puntos que son el reflejo de la realidad actual para personas con discapacidad visual que se adentran en el mundo académico a nivel superior, así como las necesidades que estos tienen, por lo que el diseño de la propuesta presentada pretende ayudar a que sus condiciones en la universidad sean lo más cercanas posibles a las de un estudiante promedio sin una discapacidad visual notable.

5.1. Conclusiones

Las conclusiones que se derivan del trabajo de investigación que se presenta, enlazan entre sí los temas tratados de uso de inteligencia artificial como asistente académico y la discapacidad visual en estudiantes de nivel superior y son las que se exponen a continuación:

En este documento se demuestra de manera documental la poca asistencia que se le da a estudiantes con discapacidad en el proceso de desarrollo académico hasta su culminación con la obtención de un título en educación superior, esto se refleja en la poca cantidad de estudiantes con discapacidad que asisten y culminan una carrera en las universidades en comparación con el número de estudiantes sin discapacidad que lo hacen. Se debe recalcar que la diferencia en la cantidad de estudiantes universitarios entre aquellos que no tienen una discapacidad y aquellos que sí, no es producto de la diferencia en el número de personas discapacitadas que están en el rango de edad para asistir a la universidad (estadísticamente hablando entre 18 y 60 años) y el número de personas que no sufren discapacidad en ese mismo rango de edad; ya que como se explica en el capítulo 2, al presentar las mismas condiciones de vida generales, dejando únicamente el factor de la discapacidad como variable se obtiene una diferencia muy notoria,

siendo un 4.9836 % de la población sin discapacidad estudiantes universitarios para el 2016 y solamente un 0.7637 % de la población con discapacidad era estudiante universitario ese mismo año (dicho porcentaje es el valor de discapacidad sensorial que incluye más que solo discapacidad visual, siendo el porcentaje de la discapacidad visual mucho menor), dejando en evidencia la necesidad de programas de asistencia en las universidades que permitan nivelar esos porcentajes.

Se plantea el uso de diversas tecnologías cuyo enfoque principal es la asistencia a personas con discapacidades, directamente en la facilitación de actividades y simplificación de procesos que normalmente a una persona no discapacitada le representaría un reto menor.

Esta investigación se centró en la discapacidad visual, por lo que las tecnologías orientadas en esta área conocidas como Tiflotecnologías son las que se utilizaron para crear un intermediario o “prototipo” que sirva como puente de unión entre un estudiante con discapacidad visual y la información que necesita obtener. Se hizo uso de múltiples tecnologías como lo son el Reconocimiento Óptico de Caracteres (OCR, Optical Character Recognition) que permitieron al asistente capturar y hacer un escaneo completo de las páginas de libros físicos, extraer el contenido de estas y procesarlo hasta convertirlo en texto digital, para ser entregado al usuario a través de una segunda herramienta llamada Procesador de Texto a Voz (TTS, Text to Speech) que recibe el texto digital y genera un audio que contiene la información requerida, siendo leída por una voz predeterminada al usuario. Se utilizó una herramienta conocida como Procesador de Voz a Texto (STT, Speech to Text) que es una herramienta que cumple la función inversa a la que realiza el TTS, ya que su labor es capturar un comando vocal (un audio) dicho por el usuario y procesarlo hasta convertirlo en texto digital; Debido a que la lengua española es muy amplia y variada en sus fonemas para la formación de palabras y frases que tiene el mismo significado, se definieron secuencias fonéticas específicas o “palabras clave”, que al ser capturadas por el STT servirían como activadores de las diversas funciones del asistente, simplificando la interacción humano-máquina. Una última tecnología utilizada fue el Raspado Web o (WS, Web Scraping) que permite por medio de comandos en lenguaje de programación Python adentrarse en la web y extraer el contenido de etiquetas HTML específicas, por lo que es posible adentrarse en una página web de un tema en específico y extraer el contenido de esta para que mediante el TTS sea procesado y escuchado por el usuario, dejando al prototipo como un intermediario que transforma la información desde un medio visual a un medio auditivo.

Cabe resaltar que en el capítulo 3 se detalla cómo se realizaron pruebas experimentales, donde, en un

ambiente controlado, se evaluó el desempeño de dos grupos de personas (G1 y G2) en realizar una serie de tareas que pretendían simular las actividades de uso del prototipo, la única diferencia entre ambos grupos fue el hecho que en uno los integrantes podían usar la vista para realizar las tareas y en el otro grupo no podían usarla. En total, la prueba se realizó tres veces la prueba en total, una vez en el grupo que podía usar la vista y dos veces en el grupo que no podía usar la vista. La prueba al primer grupo sirvió como referencia para determinar la dificultad que estas tareas representan a una persona promedio y tener un valor para comparaciones al final. Las pruebas al segundo grupo se realizaron de manera que, en la primera prueba se hicieran las tareas sin la capacidad de ver y sin ayuda del prototipo para que comparado con los resultados del primer grupo se determinará la diferencia en dificultad; la segunda prueba contemplaba la realización de las tareas sin tener la capacidad de ver y haciendo uso del prototipo, logrando determinar la diferencia de dificultad comparado con el primer grupo. En la etapa de análisis de los resultados del experimento se concluyó que la brecha de dificultad en realizar las tareas planteadas en comparación con el grupo de referencia es menor haciendo uso del prototipo que cuando no se utiliza, por lo que se puede determinar que el uso del prototipo ayuda a nivelar considerablemente las condiciones de una persona con discapacidad visual en materia de acceso a la información comparado con una persona que no sufre discapacidad visual.

5.2. Recomendaciones

Las recomendaciones planteadas en este documento tienen como propósito sugerir condiciones ideales para la implementación del prototipo, las condiciones de uso, mantenimiento y posibles mejoras que surjan en el futuro, ya sea en la puesta en marcha de un proyecto de inclusión que contemple el uso del prototipo, o en otra investigación que pretenda profundizar en el tema e incluya nueva información o variables que para el día de hoy, por limitantes de tiempo, capital, subdesarrollo de tecnologías, condiciones sociales o políticas universitarias no lograron ser incluidas en esta investigación.

Existen diversos proveedores o desarrolladores de las distintas tecnologías utilizadas para el desarrollo del prototipo, por ende, existen múltiples combinaciones entre estas. Se tiene la libertad de utilizar cualquier combinación según sea el criterio de quien inicie el prototipo. Las pruebas de rendimiento realizadas en esta investigación evaluaron criterios como claridad, velocidad y precisión en cada una de las tecnologías; al final se determinó una combinación idónea por lo que se recomienda sea la principal opción al momento de utilizar el prototipo.

Al implementarse este prototipo se recomienda el seguimiento preciso de las indicaciones planteadas en el Manual de Armado del prototipo, y que se siga la guía de instalación de software que se encuentra en este documento. Aunque en un futuro las herramientas de hardware disponibles puedan tener especificaciones superiores a las requeridas en este documento, se recomienda el uso de hardware que se asemeje al utilizado en esta investigación por motivos de sinergia entre el código fuente y el hardware.

Se recomienda que el mantenimiento del equipo, tanto físico como lógico (hardware y software) sea realizado por personal debidamente capacitado para hacerlo, y que cuente con las herramientas adecuadas para la tarea.

Para un adecuado funcionamiento del prototipo se recomienda que no sea expuesto al aire libre, que la temperatura ambiente no sea demasiado alta, que se tenga una conexión estable a internet y que se indique el adecuado uso de este a los usuarios según los parámetros definidos en el documento.

6. Anexos

6.1. Anexo: Censo de Escuelas de Educación Especial para el año 2006

| CÓDIGO | INSTITUCIÓN EDUCATIVA |
|--------|---|
| 10047 | ESCUELA DE EDUCACIÓN ESPECIAL "DE AHUACHAPAN" |
| 10072 | ESCUELA DE EDUCACIÓN ESPECIAL "DE ATQUIZAYA" |
| 10232 | ESCUELA DE EDUCACIÓN ESPECIAL ÇHALCHUAPA" |
| 10304 | ESCUELA DE EDUCACIÓN ESPECIAL "DE METAPAN" |
| 10406 | ESCUELA DE EDUCACIÓN ESPECIAL . ^{SA} ISA ALVAREZ DE DIAZ" |
| 10547 | ESCUELA DE EDUCACIÓN ESPECIAL "DE ACAJUTLA" |
| 10584 | ESCUELA DE EDUCACIÓN ESPECIAL "DE ARMENIA" |
| 10642 | ESCUELA DE EDUCACIÓN ESPECIAL "DE JUAYUA" |
| 10731 | ESCUELA DE EDUCACIÓN ESPECIAL "DE SONSONATE" |
| 10821 | ESCUELA DE EDUCACIÓN ESPECIAL "DE CHALATENANGO" |
| 11125 | ESCUELA DE EDUCACIÓN ESPECIAL "NUEVA SAN SALVADOR" |
| 11415 | ESCUELA DE EDUCACIÓN ESPECIAL "DE MEJICANOS" |
| 11553 | ESCUELA DE EDUCACIÓN ESPECIAL "DEL CENTRO DE CAPACITACION LABORAL EL PROGRESO " |
| 11573 | ESCUELA DE EDUCACIÓN ESPECIAL "DEL BARRIO SAN JACINTO" |
| 11791 | ESCUELA DE EDUCACIÓN ESPECIAL "GENERAL ADOLFO O. BLANDON " |
| 12090 | ESCUELA DE EDUCACIÓN ESPECIAL "DE ZACATECOLUCA" |
| 12155 | ESCUELA DE EDUCACIÓN ESPECIAL "DE ILOBASCO" |
| 12219 | ESCUELA DE EDUCACIÓN ESPECIAL "SEÑORA PATRICIA VÁSQUEZ DE AMAYA" |
| 12405 | ESCUELA DE EDUCACIÓN ESPECIAL "DE SAN VICENTE" |
| 12590 | ESCUELA DE EDUCACIÓN ESPECIAL "DE JUCUAPA" |
| 12721 | ESCUELA DE EDUCACIÓN ESPECIAL "MONSEÑOR RODRIGO ORLANDO CABRERA" |
| 12739 | ESCUELA DE EDUCACIÓN ESPECIAL "DE USULUTAN" |
| 12997 | ESCUELA DE EDUCACIÓN ESPECIAL "DE LA COLONIA SANTA INES" |
| 12999 | ESCUELA DE EDUCACIÓN ESPECIAL "DE SAN MIGUEL |
| 13254 | ESCUELA DE EDUCACIÓN ESPECIAL "SAN FRANCISCO DE ASIS" |
| 13422 | ESCUELA DE EDUCACIÓN ESPECIAL "LA UNIÓN" |
| 14707 | ESCUELA DE EDUCACIÓN ESPECIAL "PROFESOR NAPOLEÓN EFRAIN GONZALEZ" |
| 14823 | ESCUELA DE EDUCACIÓN ESPECIAL "MARIA ESPERANZA ESCOBAR VIUDA DE GUANDIQUE" |
| 14865 | ESCUELA DE EDUCACIÓN ESPECIAL REINALDO BORJAS PORRAS" |
| 88138 | ESCUELA DE EDUCACIÓN ESPECIAL ÇATÓLICA "PADRE VITO GUARATO" |

Tabla 6.1: Anexo: Censo de escuelas de educación especial para el año 2006.

6.2. Anexo: Instalación del sistema operativo

- **Descarga del Sistema Operativo Raspbian:** Ingresar a la página oficial de Raspbian y seleccionar la versión del S.O que deseamos instalar en nuestra PI.

<https://www.raspberrypi.org/downloads/raspbian/>

Se escogerá la versión “Raspbian Buster With Desktop”, y se descarga la versión ZIP. Se selecciona la versión con escritorio integrado ya que tendremos que hacer uso de la interfaz gráfica para poder interactuar con el hardware y no por consola.

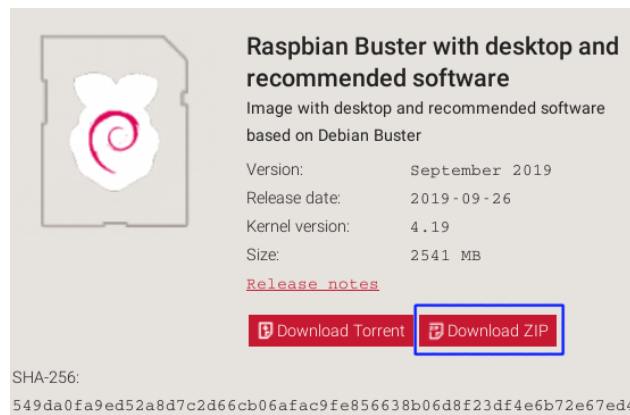


Figura 6.1: Selección del sistema operativo

- **Descarga del programa “Etcher”:** Etcher permite hacer booteable una memoria microSD, dejando instalado el S.O en la memoria, en este caso el archivo “.img” descargado en el paso N° 1.

Descargar el instalador de dicho programa, para la versión del sistema operativo que estés utilizando, instalarlo y posteriormente ejecutarlo para continuar con los demás pasos.

<https://etcher.io/>

- **Formateo de la tarjeta SD:** Formatear la tarjeta microSD para la instalación del Sistema Operativo, para ello hay que insertar la tarjeta en una máquina, en el listado de dispositivos reconocidos dar click derecho sobre el dispositivo que se desea formatear, en este caso la microSD, en el menú desplegado seleccionar “Formatear”, se mostrará una ventana donde seleccionaremos “Iniciar”, esto para que el formateo de nuestra tarjeta inicie, una vez terminado el proceso, en la misma ventana emergente aparecerá el mensaje de “Finalizado”, esto quiere decir que el proceso finalizó exitoso.

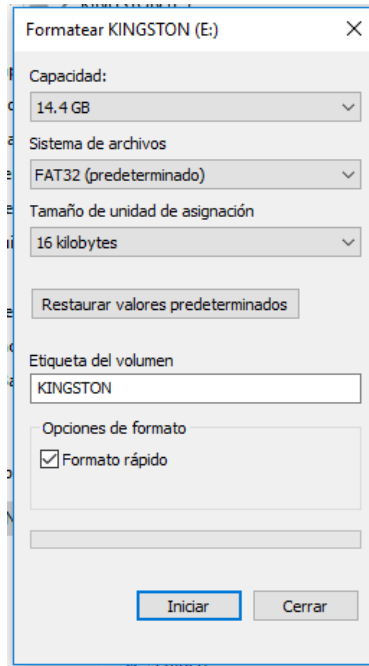


Figura 6.2: Formateo de la tarjeta SD

- Instalación del S.O en la tarjeta microSD:** Se inserta la tarjeta ya formateada a la máquina, se inicia el software “NOOBS”, una vez el programa ha reconocido a la microSD se selecciona del listado de dispositivos reconocidos, y se escoge la dirección de la ruta del archivo con extensión .iso/.img, se seleccionar “Iniciar” y así empieza el proceso de instalación del S.O, se tarda unos cuantos minutos dicho proceso. Una vez terminado se mostrará el mensaje con su respectivo estado, en dado caso ocurra un error, el programa nos notificará, si fue exitoso también de la misma forma.

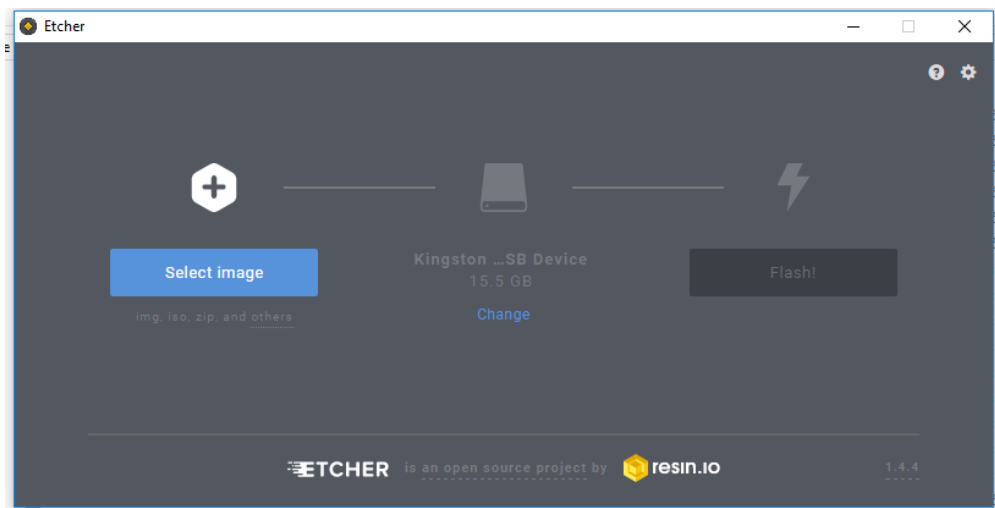
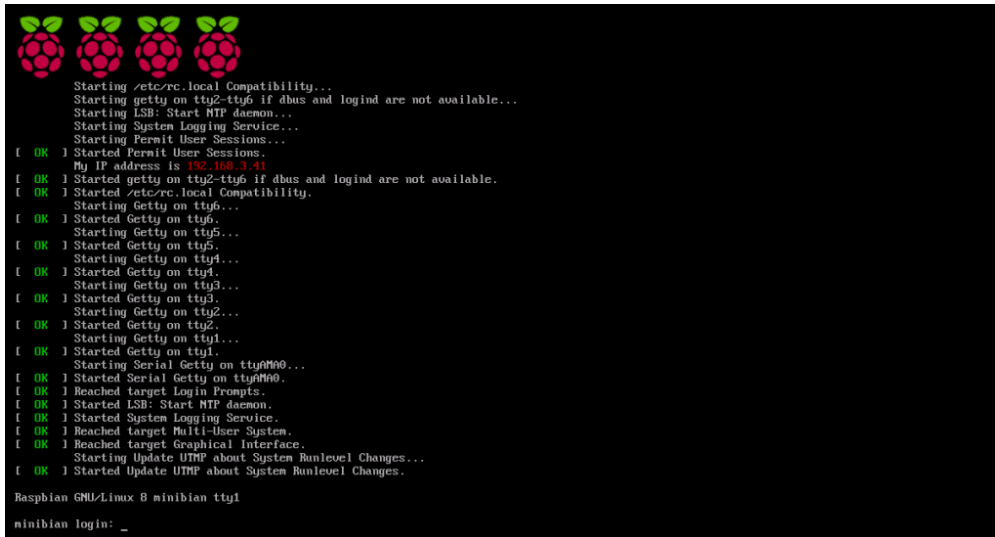


Figura 6.3: Interfaz del programa Etcher

- Accediendo por primera vez a la Raspberry PI:** Ingresamos en la Raspberry PI la tarjeta SD ya con el Sistema Operativo instalado en el paso anterior, la conectamos a una fuente de poder y conectada a un monitor para poder visualizar el proceso de arranque de esta, recordando además de los otros periféricos, tales como el teclado y un mouse, para que la Raspberry pueda funcionar no es necesario que tenga estos periféricos al encenderse, pero sí que esté con un monitor, ya que así podremos saber si el arranque está funcionando y no hay error en la tarjeta microSD.



```

Starting /etc/rc.local Compatibility...
Starting getty on tty2-tty6 if dbus and logind are not available...
Starting LSB: Start NTP daemon...
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started Permit User Sessions.
      My IP address is 192.168.3.41
[ OK ] Started getty on tty2-tty6 if dbus and logind are not available.
[ OK ] Started /etc/rc.local Compatibility.
      Starting Getty on tty6...
[ OK ] Started Getty on tty6.
      Starting Getty on tty5...
[ OK ] Started Getty on tty5.
      Starting Getty on tty4...
[ OK ] Started Getty on tty4.
      Starting Getty on tty3...
[ OK ] Started Getty on tty3.
      Starting Getty on tty2...
[ OK ] Started Getty on tty2.
      Starting Getty on tty1...
[ OK ] Started Getty on tty1.
      Starting Serial Getty on ttyAMA0...
[ OK ] Started Serial Getty on ttyAMA0.
[ OK ] Reached target Login Prompts.
[ OK ] Started LSB: Start NTP daemon.
[ OK ] Started System Logging Service.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
      Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Raspbian GNU/Linux 8 minibian tty1
minibian login: _

```

Figura 6.4: Arranque de Raspbian Buster

- Configuraciones de la Raspberry PI luego del primer arranque:** Se mostrará ciertas ventanas donde se tiene que configurar la zona horaria y demás configuraciones básicas, terminar esa configuración y proceder con los demás pasos.

6.3. Anexo: Instalación de Librerías para el funcionamiento del prototipo

Para la instalación de todas las dependencias requeridas en el proyecto, se prepararon ficheros de instalación:

Ejecutar la instalación:

```
chmod a+x install.sh venv.sh
sudo ./install.sh
```

Archivo install.sh

```
#!/bin/bash
echo -e [Instalacion de librerias requeridas]\\n
sudo apt-get -y install python3 python-dev python3-dev python3-venv build-
↳ essential python-pip
sudo apt-get -y install libasound2-data libasound2 libasound2-dev tesseract-
↳ ocr tesseract-ocr-spa libtesseract-dev
sudo apt-get -y install libopencv-dev flac autoconf libtool automake bison
↳ python-dev swig portaudio19-dev libpulse-dev vlc festival sox ffmpeg lame
↳ mpg321 fswebcam
sudo apt-get -y install pkg-config python-dev zlib1g zlib1g-dev hunspell
↳ hunspell-es
mkdir prototipo && python3 -m venv prototipo
#!/usr/bin/env bash
cd prototipo/
echo -e [Instalacion de librerias con PIP y venv]\\n
source bin/activate
pv=$(python --version 2>&1)
pv=$(echo ${pv[1]} | cut -d '.' -f 1,2 2>&1)
carpeta=lib/python${pv}/site-packages/speech_recognition/pocketsphinx-data
wget https://mupdf.com/downloads/archive/mupdf-1.16.0-source.tar.gz
tar -zxvf mupdf-1.16.0-source.tar.gz && cd mupdf-1.16.0-source
export CFLAGS="-fPIC"
make HAVE_X11=no HAVE_Glfw=no HAVE_Glut=no prefix=/usr/local
sudo make HAVE_X11=no HAVE_Glfw=no HAVE_Glut=no prefix=/usr/local install
cd ..
mkdir sphinx && cd sphinx
git clone https://github.com/cmuspinx/sphinxbase && git clone https://github.
↳ com/cmuspinx/pocketsphinx
cd sphinxbase && sudo ./autogen.sh && sudo make && sudo make install
cd ../pocketsphinx && sudo ./autogen.sh && sudo make && sudo make install &&
↳ cd ../ && mkdir esp && cd esp
echo -e [Instalando paquetes de PIP]\\n
pip3 install wheel
pip3 install --upgrade numpy Pillow beautifulsoup4 pytesseract requests opencv
↳ -python gTTS pyaudio pynput pocketsphinx SpeechRecognition watson-developer
↳ -cloud PyDrive pyMuPDF termcolor html2text
echo -e [Descargando paquetes de idiomas]\\n
wget https://sourceforge.net/projects/cmuspinx/files/Acoustic%20and%20
↳ Language%20Models/Spanish/cmuspinx-es-5.2.tar.gz
wget https://sourceforge.net/projects/cmuspinx/files/Acoustic%20and%20
↳ Language%20Models/Spanish/es-20k.lm.gz
wget https://sourceforge.net/projects/cmuspinx/files/Acoustic%20and%20
↳ Language%20Models/Spanish/es.dict
wget http://tiny.cc/es_hotwords -O es-ES_hotwords.zip
```

```

git clone https://github.com/guadalinex-archive/hispavoces
echo -e [Instalando idiomas]\\n
sudo mkdir /usr/share/festival/voices/spanish/
sudo dpkg -i hispavoces/packages/festvox-palpc16k_1.0-1_all.deb && sudo dpkg -
↳ i hispavoces/packages/festvox-sflpc16k_1.0-1_all.deb
tar -xzvf cmusphinx-es-5.2.tar.gz && gzip -d es-20k.lm.gz && unzip es-
↳ ES_hotwords.zip
mv es-20k.lm language-model.lm.bin
mv cmusphinx-es-5.2/model_parameters/voxforge_es_sphinx.cd_ptm_4000 acoustic-
↳ model
mv es.dict pronunciation-dictionary.dict
cd ../../
echo -e [Instalando paquetes de CMU sphinx - version ${pv}]\n
mkdir ${carpeta}/es-ES
cp sphinx/esp/language-model.lm.bin ${carpeta}/es-ES/
cp sphinx/esp/pronunciation-dictionary.dict ${carpeta}/es-ES/
cp -r sphinx/esp/acoustic-model ${carpeta}/es-ES/
cp -r sphinx/esp/es-ES_hotwords ${carpeta}/
echo -e [Actualizar ficheros]\\n
cp lib/python${pv}/site-packages/speech_recognition/__init__.py lib/python${pv}
↳ }/site-packages/speech_recognition/__init__old.py
cp lib/python${pv}/site-packages/speech_recognition/__init__.py in.py
url1a="https://api.cognitive.microsoft.com/sts/v1.0/issueToken"
url1b="https://westus.api.cognitive.microsoft.com/sts/v1.0/issueToken"
url2a="https://speech.platform.bing.com/speech/recognition/interactive/
↳ cognitiveservices/v1?{}".format(urllencode({'
url2b="https://westus.stt.speech.microsoft.com/speech/recognition/
↳ conversation/cognitiveservices/v1?language={}&format=simple".format (
↳ language)'
url3a="language": language,'
url3b='# "language": language,'
url4a="locale": language,'
url4b='# "locale": language,'
url5a="requestid": uuid.uuid4(),%%%)'
url5b='# "requestid": uuid.uuid4(),)})'
sed -i -e "s!${url1a}!${url1b}!" in.py
sed -i -e "s!${url2a}!${url2b}!" in.py
sed -i -e "s!${url2a}!${url2b}!" in.py
sed -i -e "s!${url3a}!${url3b}!" in.py
sed -i -e "s!${url4a}!${url4b}!" in.py
sed -i -z "s/\n/%%%/g" in.py
sed -i -e "s!${url5a}!${url5b}!" in.py
sed -i -e "s/%%%/\\n/g" in.py
cp -f in.py lib/python${pv}/site-packages/speech_recognition/__init__.py
rm -f b.txt c.txt d.txt in.py
echo -e [Instalando el proyecto]\\n
mkdir proyecto
cd proyecto
wget http://tiny.cc/prototipo_release -O prototipo.zip && unzip prototipo.zip

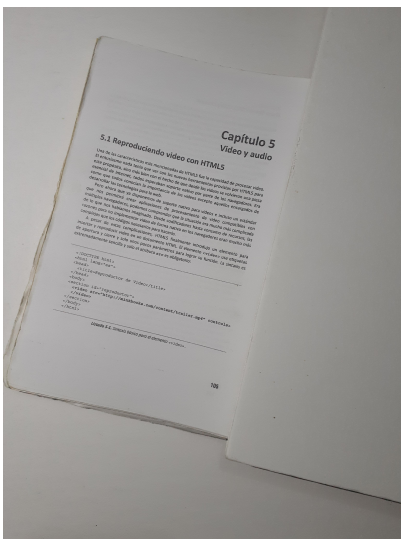
```

6.4. Anexo: Manual de requerimientos para el funcionamiento de la toma de fotografías a libros de texto

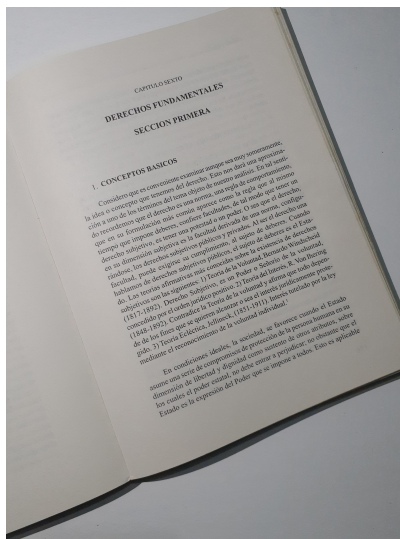
Debido a que el dispositivo tiene como agregado el módulo de extracción de texto en imágenes, dichas imágenes son al respecto de libros de texto, estos libros de texto deben de tener ciertas características o condicionamientos para tener una fotografía con una calidad decente y pueda ser utilizada por el dispositivo que conlleva a una extracción exitosa de información. Las condiciones a tomar en cuenta son las siguientes:

- Tamaño del libro
- Posición del contenido
- Campo de visión del contenido
- Tipo de papel
- Tipo de encuadernado

Para una mejor comprensión del tipo de imagen idónea y una no muy idónea, a continuación se muestran dos imágenes con la cuales se explicará cada uno de las condiciones y el cómo puede significar en los resultados de extracción de texto.



A



B



C

Figura 6.5: Imágenes de muestra de requerimientos de libros. Parte A



D



E

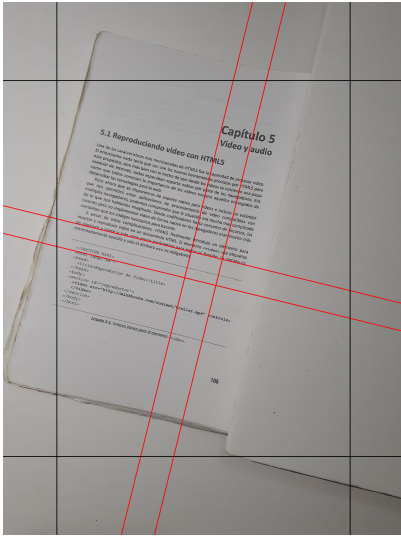
Figura 6.6: Imágenes de muestra de requerimientos de libros. Parte B

Tamaño del libro: El dispositivo tiene un determinado espacio (a.cm x b.cm) espacio para poder colocar libros, por lo tanto el libro no debe de superar lo alto ni lo ancho de dicha superficie.

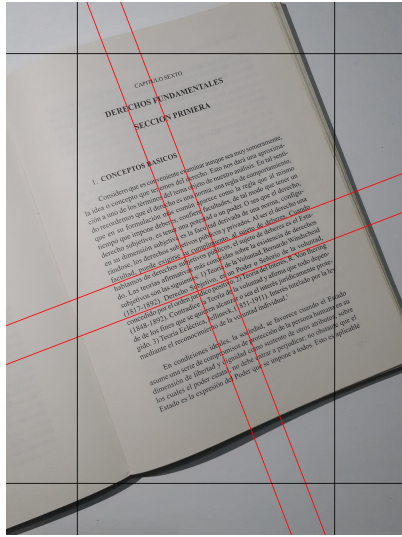
Tiene que quedar en el encuadre de la fotografía, así que el tamaño algunas veces puede ser relativamente pequeño pero mientras pueda ser incorporado en la fotografía todo está bien, la cámara tiene una resolución muy buena para tomar captura del texto aunque el libro no abarque por completo el encuadre de la fotografía.

En el caso de todas las fotografías anteriores se captan con encuadre de la fotografía, pese a que las imágenes A y B no cubran un área grande.

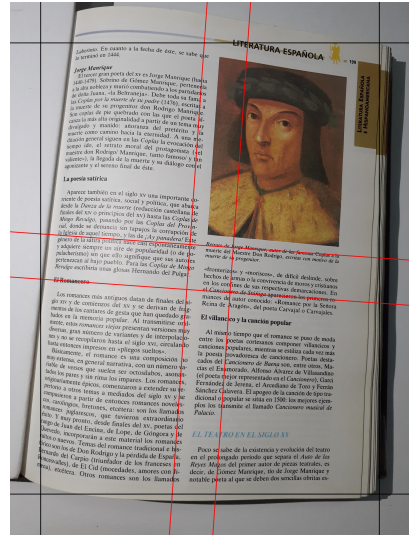
Posición del contenido: Para que el dispositivo pueda ejecutar una correcta extracción de texto del contenido fotografiado deben cumplirse determinadas condiciones para que este sea posible, la primera es la posición, el libro a procesar debe estar alineado con el centro de la cámara, lo segundo es que el texto no debe de tener inclinaciones mayores a 45° con respecto a la línea central de la toma.



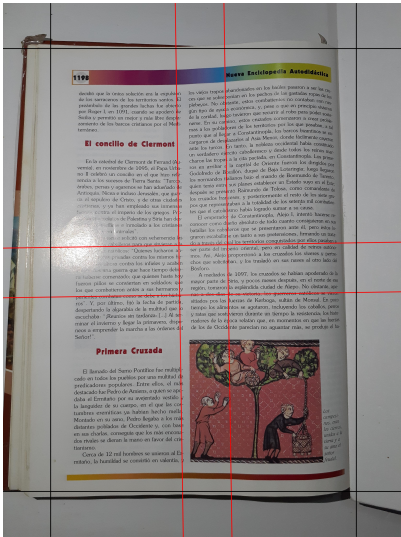
A



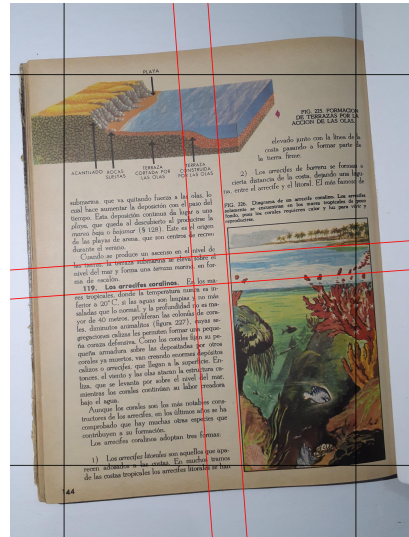
B



C



D



E

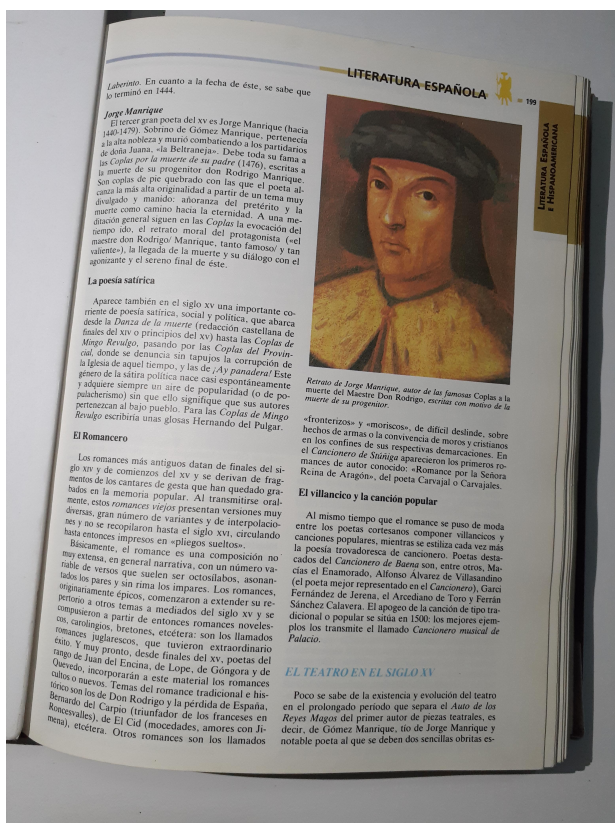
Figura 6.7: Representación del ángulo del texto en las imágenes, las líneas rojas indican la inclinación con respecto al encuadre.

Campo de visión del contenido: Como se ha mencionado en puntos anteriores el encuadre de la cámara tiene una gran influencia en la calidad de la fotografía, pero hay que tener en cuenta el campo de visión del texto que se quiere extraer.

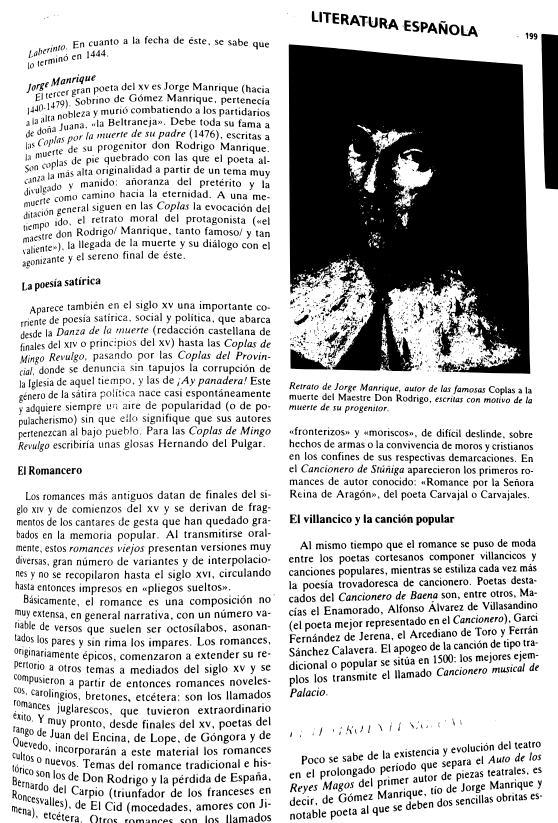
Un ejemplo de problema con el campo de visión puede ser que al desear extraer el contenido de solo una página, pero si el libro esta abierto y la cámara percibe parte de la hoja que no nos interesa extraer el texto resultante puede contener errores de secuencia, para ello se sugiere el uso de un bloqueador de páginas de preferencia de color blanco. El caso en el que se tenga que intervenir con las manos para posicionar el libro es probable que en la

fotografía aparezcan manos, ésto no resulta un inconveniente importante, siempre y cuando la mano no interfiera o esté superpuesta en el recuadro principal de texto que se desea extraer.

Tipo de papel: El tipo de papel tiene influencia en los resultados de la extracción de texto, ya que hay algunas veces que se querrá extraer texto de una revista. Las revistas tienen un tipo de papel muy particular comparada con un libro, esta diferencia es el material del papel, el papel de la revista suele ser reflectante y cuando recibe luz en ciertos puntos el texto es imposible de poder diferenciar aún para el ser humano. Se recomienda no utilizar libros cuyo papel que sea reflectante o afectado por la iluminación, y mejor utilizar el tipo de papel habitual que no refleja lo que es la luz. En el caso de las imágenes C se cuenta con buen tipo de papel ya que el tipo de papel no refleja la luz de manera directa y en la imagen D el papel es del tipo reflexivo.

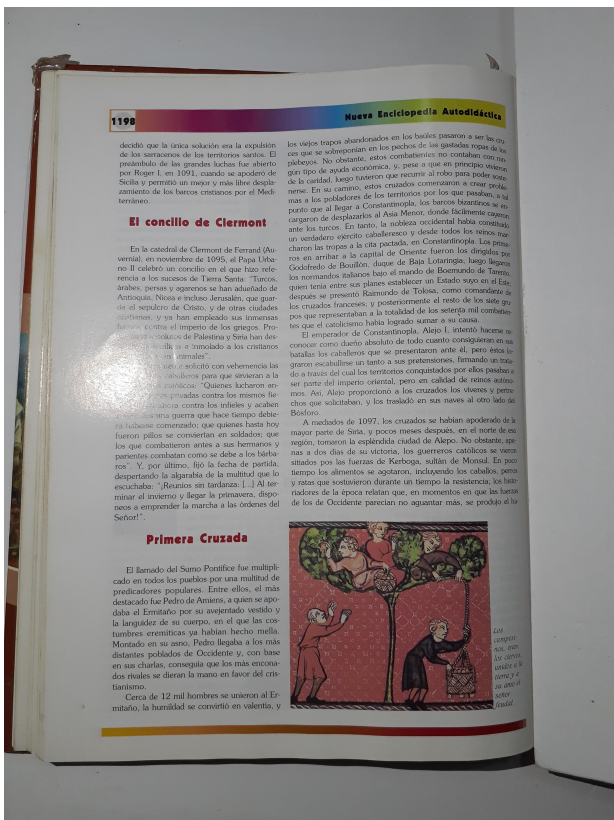


C.1

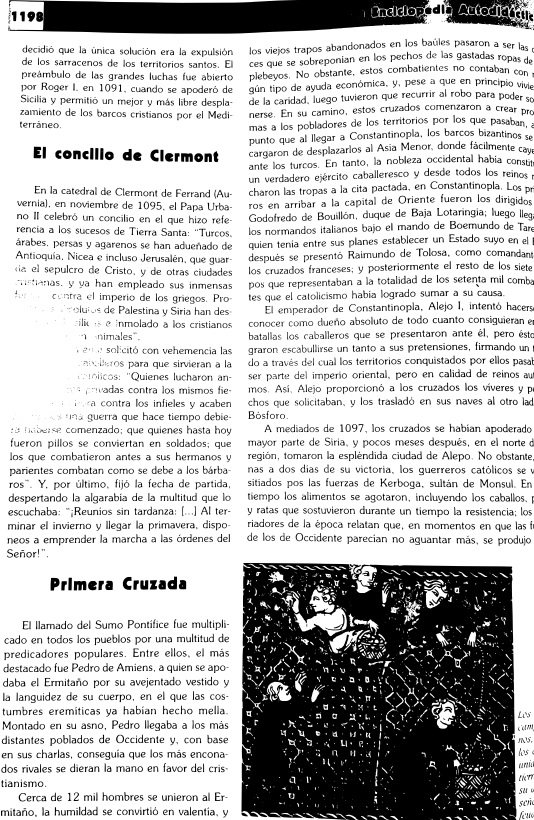


C.2

Figura 6.8: Ejemplo de página reflectante y no reflectante. Parte A. Ejemplo de página no reflectante (C.1)



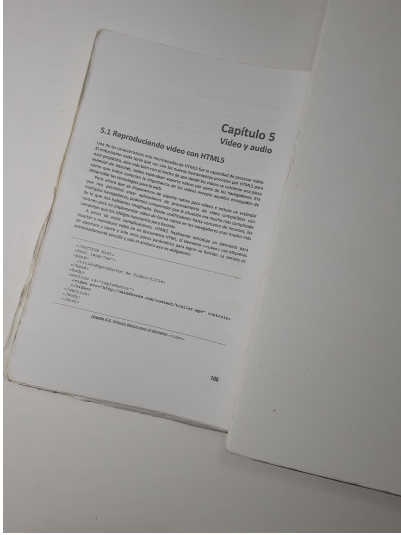
D.1



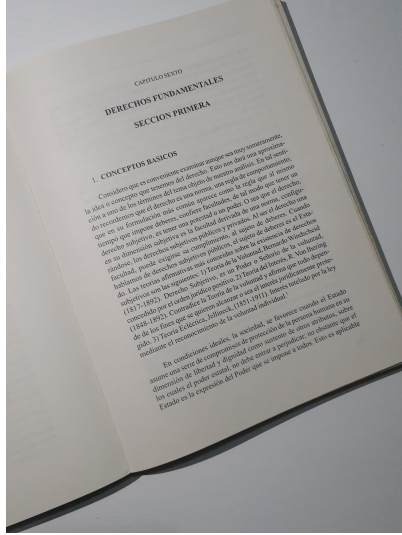
D.2

Figura 6.9: Ejemplo de página reflectante y no reflectante. Parte B. Página reflectante (D.1), se puede notar la perdida de texto (D.2).

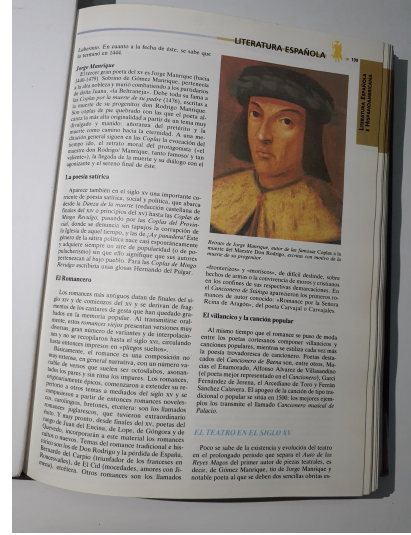
Tipo de encuadernado: Para poder colocar el libro en la superficie y tomar la fotografía el libro tiene que ser manipulado, hay libros que por su encuadernado son más fáciles de manipular que otros, ésta manipulación esta intrínsecamente relacionada con el tipo de libro que es. En la imagen A el libro tiene un encuadernado en espiral lo cual facilita su colocación en la base, la imagen B se puede apreciar un tipo de empastado de tapa suave que puede ser manipulado y dejarlo expandido por sí solo o usar intervención con las manos para que este siga manteniendo su posición y poder realizar la fotografía, pero también presenta el problema de que las páginas no se sostienen perfectamente planas lo cual puede generar deformaciones en el texto al obtener la imagen, en la imagen C el libro cuenta con un encuadernado de tapa dura y sus páginas son lo suficientemente grandes como para mantenerse planas por su propio peso.



A



B.1



C

CAPÍTULO SEXTO
DERECHOS FUNDAMENTALES
SECCION PRIMERA

1. CONCEPTOS BASICOS

Considero que es conveniente examinar aunque sea muy someramente, la idea o concepto que tenemos del derecho. Esto nos dará una aproximación a uno de los términos del tema objeto de nuestro análisis. En tal sentido recordemos que el derecho es una norma, una regla de comportamiento, que en su formulación más común aparece como la regla que al mismo tiempo que impone deberes, confiere facultades, de tal modo que tener un derecho subjetivo, es tener una potestad o un poder. O sea que el derecho en su dimensión subjetiva es la facultad derivada de una norma, configurándose, los derechos subjetivos públicos o privados. Al ser el derecho una facultad, puede exigirse su cumplimiento al sujeto de deberes. Cuando hablamos de derechos subjetivos públicos, el sujeto de deberes es el Estado. Las teorías afirmativas más conocidas sobre la existencia de derechos subjetivos son las siguientes: 1) Teoría de la Voluntad; Bernardo Windscheid (1817-1892). Derecho Subjetivo, es un Poder o Señorío de la voluntad, concedido por el orden jurídico positivo; 2) Teoría del Interés; R. Von Ihering (1848-1892). Contradice la Teoría de la voluntad y afirma que todo depende de los fines que se quieren alcanzar o sea el interés jurídicamente protegido; 3) Teoría Ecléctica; Jellinek, (1851-1911). Interés tutelado por la ley mediante el reconocimiento de la voluntad individual.

En condiciones ideales, la sociedad se favorece cuando el Estado asume una serie de compromisos de protección de la persona humana en su dimensión de libertad y dignidad como sustento de otros atributos, sobre los cuales el poder estatal, no debe entrar a perjudicar; no obstante que el Estado es la expresión del Poder que se impone a todos. Esto es aplicable

B.2

Figura 6.10: Ejemplo del desajuste causado por las páginas de libros pequeños y medianos con un empastado de tapa suave o tapa dura, las líneas rojas son de referencia, las secciones en azul son para resaltar el desajuste.

Es muy probable que en alguna ocasión se necesite escanear un libro con un empastado difícil de poder doblar y posicionar en la superficie, en esos casos bastará con cumplir con los otros requerimientos antes mencionados para poder realizar una captura de fotografía con calidad decente y poder procesarla.

6.5. Anexo: Código fuente

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: base.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 # genericas
6 import os
7 import re
8 import sys
9 import requests
10 import threading
11 import subprocess
12 from time import time as stime, sleep as pause
13 from termcolor import colored, cprint
14 # PDF
15 import fitz
16 # scraping web
17 from web import WebS
18 # teclado
19 from pynput import keyboard
20 # reconocimiento de voz - speech-to-text
21 from record import Record
22 import speech_recognition as sp
23 from gtts import gTTS, gTTSException
24 from watson_developer_cloud import SpeechToTextV1, WatsonApiException
25 # texto a voz - text-to-speech
26 from tts_bing import BingTTS, BingBadRequestException, BingAuthException
27 from tts_voicerss import VoiceRSS, VrBadRequestException
28 # imagenes y OCR
29 import cv2
30
31 from picamera import PiCamera
32 from io import BytesIO
33 from PIL import Image
34 from procesador_tesseract import ProcesarConTesseract
35 from ocr_google_drive import OCRGoogleDrive
36 from regex import RegexTexto
37 from ocr_space import OCRSpace, OCRSpaceNoKeyException,
   ↳ OCRSpaceBadRequestException, OCRSpaceRequestException
38 from ocr_bing import OCRBing, OCRBingNoKeyException,
   ↳ OCRBingBadRequestException, OCRBingRequestException
39
40 def print_error(x): cprint(x, 'red', attrs=['bold', 'underline'], end='
   ↳ \n')
41 def print_info(x): cprint(x, 'yellow', attrs=['bold'], end=' \n')
42 def print_log(x): cprint(x, 'cyan', attrs=['bold'], end=' \n')
43 def print_log_a(x): cprint(x, 'magenta', attrs=['underline'], end=' \n')
44 def print_info_nb(x): cprint(x, 'yellow', attrs=['bold'], end='')
45 def print_info_rv(x): cprint(x, 'magenta', attrs=['bold'], end=' \n')
46
47 class Prototipo:
48
49     -def __init__(self, arranque=True):
50     -"""

```

```

51 --Inicializa y establece los parametros de configuracion necesarios para
    ↳ el funcionamiento del prototipo
52
53 --:param arranque:
54 --"""
55 --# inicializacion y configuracion de variables globales
56 --self.reconocedor = sp.Recognizer()
57 --self.config_pyteseract = '-l spa --oem 1 --psm 3'
58 --self.api_key = {'tts_voicerss': '[API KEY]',
59 -----'tts_bing': '[API KEY]',
60 -----'stt_wit': '[API KEY]',
61 -----'stt_google': '[API KEY]',
62 -----'ocr_space': '[API KEY]',
63 -----'stt_ibm': '[API KEY]',
64 -----'ocr_bing': '[API KEY]'}
65 --self.numeros = ['uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis',
    ↳ 'siete', 'ocho', 'nueve', 'diez', 'once']
66 --self.diccionario = [('buscar', 1.0), ('lectura', 1.0), ('repetir
    ↳ lista', 1.0), ('repetir texto', 1.0),
67 -----('navegacion avanzar', 1.0), ('navegacion regresar', 1.0),
    ↳ ('libro', 1.0),
68 -----('libro avanzar', 1.0), ('libro regresar', 1.0),
69 -----('resultado uno', 1.0), ('resultado dos', 1.0), ('resultado tres',
    ↳ 1.0),
70 -----('resultado cuatro', 1.0), ('resultado cinco', 1.0), ('resultado
    ↳ seis', 1.0),
71 -----('resultado siete', 1.0), ('resultado ocho', 1.0), ('resultado
    ↳ nueve', 1.0),
72 -----('resultado diez', 1.0), ('resultado once', 1.0)]
73 --self.__ibm_sp =
    ↳ SpeechToTextV1(url='https://stream.watsonplatform.net/speech-to-text/api',
74 ----- iam_apikey=self.api_key['stt_ibm'])
75 --self.ip_cam = None
76 --self.__from_cam = False
77 --self.__grabando = False
78 --self.__cola_grabacion = False
79 --self.webs = WebS()
80 --self.indice_navegacion = 0
81 --self.__en_proceso = False
82 --self.__proce = None
83 --self.pro_tesseract = ProcesarConTesseract()
84 --self.__regex = RegexTexto()
85 --self.imdir = 'debug'
86 --self.global_n = 0
87 --self.__device = 2
88 --self.__debug = False
89 --self.__rec = Record(filename='para_filtro')
90 --self.__rec.generar_filtro('filtro_temporal')
91 --self.__rec.cerrar()
92 --if arranque:
93 ---self.id_sub_proceso_player = None
94 ---self.id_sub_proceso_record = None
95 ---self.hilo_teclado = threading.Thread(target=self.__hilo_grabacion)
96 ---self.hilo_teclado.daemon = True
97 ---self.hilo_teclado.start()
98
99

```

```

100 """
101
102 --Funciones de imagenes y PDF
103
104 """
105
106 # cambiar por la funcion de camara y tratar de convertir a blanco y negro
107
108 def obtener_imagen_camara(self, archivo, from_cam=False):
109 """
110 --Obtiene una imagen a partir de la camara por defecto en el sistema
111 --:param from_cam: Define si el archivo de imagen se leera desde la camara
112 --:param archivo:
113 --:return: El archivo de imagen obtenido, None en caso de ocurrir un error
114 """
115 if from_cam or self.__from_cam:
116 stream = BytesIO()
117 camera = PiCamera()
118 camera.resolution = (2592, 1944)
119 camera.framerate = 30
120 camera.shutter_speed = camera.exposure_speed
121 camera.exposure_mode = "off"
122
123 try:
124 camera.capture(stream, "jpeg")
125 camera.close()
126 except:
127 print_error("[dev][error] Error en captura de imagen")
128 exit(0)
129
130 im = Image.open(stream).rotate(-90, expand=1)
131 im.save(archivo, 'JPEG', quality=100)
132 try:
133 imagen = cv2.imread(archivo)
134 except cv2.error:
135 return None
136
137 return imagen
138
139
140 def texto_de_imagen(self, imagen, motor='ocrspace'):
141 """
142 --Obtiene el texto de la imagen ingresada
143
144 --:param imagen: El archivo de imagen a procesar
145 --:param motor: El archivo de imagen a procesar
146 --:return: El texto de la imagen, None en caso de error
147 """
148 if isinstance(imagen, str):
149 imagen = self.obtener_imagen_camara(imagen, from_cam=self.__from_cam)
150 if motor == 'ocrspace':
151 ocr_space_ex = (OCRSpaceNoKeyException, OCRSpaceRequestException,
152                OCRSpaceBadRequestException)
153 try:
154 ocr_space = OCRSpace(self.api_key['ocr_space'])
155 texto = ocr_space.texto_de_imagen(imagen, language='spa')
156 texto = self.__regex.regex(texto, hard=False)

```

```

156 ----return texto
157 ---except ocr_space_ex as ex:
158 ----print_info('[dev][info][ocr_space] Error {}'.format(ex))
159 --if motor == 'bing':
160 ---ocr_bing_ex = (OCR BingNoKeyException, OCR BingRequestException,
    ↪ OCR BingBadRequestException)
161 ---try:
162 ----ocr_bing = OCR Bing(self.api_key['ocr_bing'])
163 ----texto = ocr_bing.texto_de_imagen(imagen)
164 ----texto = self.__regex.regex(texto, hard=False)
165 ----return texto
166 ---except ocr_bing_ex as ex:
167 ----print_info('[dev][info][ocr_bing] Error {}'.format(ex))
168 --if motor == 'google':
169 ---ocr_google = OCR GoogleDrive()
170 ---texto = ocr_google.texto_de_imagen(image_path=imagen)
171 ---texto = self.__regex.regex(texto, hard=True)
172 ---return texto
173
174 -def leer_pagina_pdf(self, archivo, pagina):
175 -"""
176 --Lee la pagina seleccionada del archivo PDF seleccionado
177
178 --:param archivo: La ruta al archivo PDF a leer, obligatorio
179 --:param pagina: La pagina del archivo PDF a leer, obligatorio
180 --:return: El texto de la pagina si es valido, la imagen de la pagina si
    ↪ es valido, Falso si no es valido
181 -"""
182 --if os.path.exists(archivo):
183 ---pdf = fitz.Document(archivo)
184 ---n_paginas = pdf.pageCount
185 ---if int(pagina) > n_paginas:
186 ----return [False]
187 ---else:
188 ----texto_pagina = pdf.loadPage(int(pagina) -
    ↪ 1).getText('text').strip().replace('-\n', '').replace('\n',
189 -----' ')
190 ----if len(texto_pagina) < 20:
191 -----matrix = fitz.Matrix(4, 4)
192 -----pix = pagina.getPixmap(matrix=matrix, colorspace='RGB', alpha=False)
193 -----out = Image.frombytes('RGB', [pix.width, pix.height],
    ↪ pix.samples).convert('L')
194 -----out.save('images/pdf.jpg', 'JPEG', quality=80)
195 ----return ['Imagen']
196 ---else:
197 ----return [True, texto_pagina]
198 --else:
199 ---return [False]
200
201 -"""
202 -
203 --Funciones de audio
204
205 -"""
206
207 -@staticmethod
208 -def reproducir_audio_menu(archivo):

```

```

209 --"""
210 --Reproduce un archivo de audio del menu
211
212 --:param archivo: El archivo del menu seleccionado
213 --"""
214 --direccion = 'menu/{}.mp3'.format(archivo)
215 --if os.path.exists(direccion):
216 ---t1 = stime()
217 ---print_log(['log][info] Reproducir audio - menu/{}.mp3'.format(archivo))
218 ---os.system('cvlc -A=alsa --play-and-exit {} 2>
↳ /dev/null'.format(direccion))
219 ---t2 = str(round(stime() - t1, 2))
220 ---print_log(['log][info] Audio reproducido - menu/{}.mp3 - duracion: {}
↳ segundos'.format(archivo, t2))
221
222 -def texto_a_audio(self, texto, tts='gtts', only_gen=False, reintento=0,
↳ id_anterior=-1):
223 --"""
224 --Convierte el texto ingresado a voz haciendo uso de la herramienta
↳ seleccionada
225
226 --:param texto: El texto que se desea convertir a voz
227 --:param tts: La herramienta de texto a voz seleccionada para la
↳ conversion, por defecto 'gtts'
228 --:param only_gen:
229 --:param reintento:
230 --:param id_anterior:
231 --:return: Si el proceso es correcto devuelve un array de la forma [True,
↳ n] con n siendo el numero de archivos de audio, caso contrario
↳ devuelve [False]
232 --"""
233 --asynchr = False
234 --a_voz_bing = BingTTS(self.api_key['tts_bing'])
235 --a_voz_vrssl = VoiceRSS(self.api_key['tts_voicerss'])
236 --tts_exceptions = (gTTSError, BingAuthException,
↳ BingBadRequestException, VrBadRequestException)
237 --tts_types = ['gtts', 'bing', 'voicerss', 'festival']
238 --if tts not in tts_types:
239 ---print_error(['dev][error] tts no valido')
240 ---return [False]
241 --texto_preparado = self.preparar_texto(texto=texto, motor=tts)
242 --marca_total = stime()
243 --n_audio = 0
244 --bing_contador = 0
245 --id_proc = None
246 --for i in range(0, len(texto_preparado[0])):
247 ---if id_anterior != -1:
248 ----i = id_anterior
249 ---try:
250 ----marca = stime()
251 ----salida = None
252 ----if tts == 'festival':
253 -----temp_texto = texto_preparado[0][0]
254 ----else:
255 -----temp_texto = texto_preparado[0][i]
256 ----try:

```

```

257 -----print_log('[E: {}]-[C: {}]-[L: {}][T: {}]'.format(tts, str(i),
↳ str(len(str(temp_texto))),
258 -----str(temp_texto.replace('\n', ' '))))
259 -----except TypeError:
260 -----print_log('[E: {}]-[C: {}]-[L: {}][T: {}]'.format(tts, str(i),
↳ str(len(str(temp_texto))),
261 -----str(str(temp_texto).replace('\n', ' '))))
262 -----if tts == 'festival':
263 -----print_log('[dev][log][IO] Escribiendo archivo -
↳ audio/festival/{}.txt'.format(str(i)))
264 -----with open('audio/festival/0.txt', 'wb') as file_audio:
265 -----file_audio.write(temp_texto)
266 -----file_audio.close()
267 -----print_log('[dev][log][IO] Archivo escrito -
↳ audio/festival/{}.txt'.format(str(i)))
268 -----print_log('[dev][log][IO] Escribiendo archivo -
↳ audio/festival/{}.wav'.format(str(i)))
269 -----os.system('text2wave -o audio/festival/{}.wav audio/festival/{}.txt
↳ -eval "{}"'.format(
270 -----'0', '0', '(voice_JuntaDeAndalucia_es_sf_diphone)'))
271 -----print_log('[dev][log][IO] Archivo escrito -
↳ audio/festival/{}.wav'.format(str(i)))
272 -----elif tts == 'gtts':
273 -----print_log('[dev][log][IO] Escribiendo archivo -
↳ audio/gtts/{}.mp3'.format(str(i)))
274 -----salida_gtts = gTTS(text=temp_texto, lang='es')
275 -----salida_gtts.save('audio/gtts/{}.mp3'.format(str(i)))
276 -----print_log('[dev][log][IO] Archivo escrito -
↳ audio/gtts/{}.mp3'.format(str(i)))
277 -----elif tts == 'bing':
278 -----error_conversion = False
279 -----try:
280 -----salida = a_voz_bing.speak(language='es-ES',
281 -----voice='HelenaRUS',
282 -----file_format='audio-16khz-32kbitrate-mono-mp3',
283 -----text=temp_texto)
284 -----print_log('[dev][log][IO] Escribiendo archivo -
↳ audio/{}/{}.mp3'.format(tts, str(i)))
285 -----except (BingBadRequestException, BingAuthException):
286 -----error_conversion = True
287 -----print_error('[dev][error][bing] Error al convertir texto a voz')
288 -----if error_conversion:
289 -----reintento += 1
290 -----if reintento < 10:
291 -----print_info('[dev][info][bing] Reintentando...
↳ {}'.format(str(reintento)))
292 -----self.texto_a_audio(texto=texto, tts=tts, only_gen=True,
↳ reintento=reintento, id_anterior=i)
293 -----else:
294 -----return print_error('[dev][error][bing] La conversion de texto a
↳ voz a fallado demasiado,'
↳ 'saliendo')
295 -----
296 -----else:
297 -----with open('audio/{}/{}.mp3'.format(tts, str(i)), 'wb', 0) as
↳ archivo:
298 -----archivo.write(salida)
299 -----archivo.flush()

```



```

300 -----archivo.close()
301 -----print_log('[dev][log][IO] Archivo escrito -
↳ texto/{}/{}.mp3'.format(tts, str(i)))
302 -----bing_contador = bing_contador + 1
303 -----if bing_contador is 15:
304 -----bing_contador = 0
305 -----print_log('[E: Bing]-[Pausa: 120 seg]')
306 -----pause(120)
307 ----elif tts == 'voicerss':
308 ----error_conversion = False
309 ----try:
310 -----salida = a_voz_vrss.speak(language='es-es',
311 -----text=temp_texto,
312 -----codec='OGG',
313 -----audio_format='24khz_16bit_mono')
314 ----except (VrBadRequestException):
315 ----error_conversion = True
316 ----print_error('[dev][error] Error al convertir texto a voz')
317 ----if error_conversion:
318 ----reintento += 1
319 ----if reintento < 10:
320 ----print_info('[dev][info][voicerss] Reintentando...
↳ {}'.format(str(reintento)))
321 ----else:
322 ----return print_error('[dev][error][voicerss] La conversion de texto
↳ a voz a fallado demasiado,'
323 -----'saliendo')
324 ----else:
325 ----print_log('[dev][log][IO] Escribiendo archivo -
↳ audio/{}/{}.ogg'.format(tts, str(i)))
326 ----with open('audio/{}/{}.ogg'.format(tts, str(i)), 'wb', 0) as
↳ archivo:
327 -----archivo.write(salida)
328 -----archivo.flush()
329 -----archivo.close()
330 ----print_log('[dev][log][IO] Archivo escrito -
↳ texto/{}/{}.ogg'.format(tts, str(i)))
331 ----trans = round(stime() - marca, 3)
332 ----print_log_a('[E: {}]-[T: {} segundos]'.format(tts, trans))
333 ----if asyncr is False and only_gen is False:
334 ----asyncr = True
335 ----id_proc = self.reproducir_texto(tts, len(texto_preparado[0]))
336 ----print_log('[dev][info] PID obtenido: {}'.format(str(id_proc.pid)))
337 ----n_audio = len(texto_preparado[0])
338 ----except tts_exceptions as err:
339 ----print_error('[log][error] Error al procesar la conversion de texto a
↳ voz - {} - {}'.format(tts, err))
340 --trans_total = round(stime() - marca_total, 3)
341 --print_log('[E: {}]-[A: {} segundos]'.format(tts, trans_total))
342 --if id_proc is not None:
343 ---id_proc.wait()
344 ---id_proc.kill()
345 ---print_log('Subproceso reproduccion finalizado:
↳ {}'.format(str(id_proc.pid)))
346 --return [True, n_audio]
347
348 -@staticmethod

```

```

349 -def reproducir_texto(motor='gtts', numero=0, espera=False):
350     """
351     --Reproduce los archivos de audio de la herramienta proporcionada,
    ↪ siguiendo el numero a partir de 0
352
353     --:param motor: El motor de conversion de donde se generan los archivos
    ↪ de audio, por defecto 'gtts'
354     --:param numero: El numero de archivos de audio a reproducir, a partir de
    ↪ 0
355     --:return: El id del subprocesso de reproduccion
356     """
357     id_sub_proc = subprocess.Popen('exec python player.py -g {} -n
    ↪ {}'.format(motor, str(numero)),
358     ----- shell=True)
359     print_log('[dev][info] Subproceso de reproduccion iniciado:
    ↪ {}'.format(str(id_sub_proc.pid)))
360     if espera:
361         id_sub_proc.wait()
362     else:
363         return id_sub_proc
364
365     """
366
367     -Grabacion
368
369     """
370
371     -def __hilo_grabacion(self):
372         """
373         --Mantiene en segundo plano una espera de teclas presionadas si se
    ↪ captura alguna lo envia a __escuchar_teclado
374
375         """
376         print_info('[dev][info] Iniciado servicio de escucha de teclado')
377         with keyboard.Listener(on_press=self.__escucha_teclado) as escucha:
378             escucha.join()
379
380         -def __escucha_teclado(self, tecla):
381             """
382             --Procesa la tecla proporcionada y si es "Espacio" o "Intro" y no hay
    ↪ otro proceso en espera inicia una nueva
383             --grabacion
384
385             --:param tecla:
386             """
387             if tecla == keyboard.Key.space and self.__cola_grabacion is False:
388                 self.iniciar_grabacion(instruccion='dicte')
389
390
391         -def iniciar_grabacion(self, instruccion=None):
392             """
393             --Inicia un nuevo proceso de grabacion de audio
394
395             --:param instruccion: El archivo de audio del menu a reproducir al iniciar
    ↪ la grabacion
396             """
397             if self.__en_proceso:

```

```

398 ---print_info('[dev][info] Proceso bloqueado')
399 ---else:
400 ---self.__en_proceso = True
401 ---try:
402 ----print_info('[dev][info] Ejecutar subprocesso grabar')
403 ----if instruccion is not None:
404 -----self.reproducir_audio_menu(instruccion)
405 -----rec = Record()
406 -----rec.grabar()
407 -----rec.cerrar()
408 -----self.__cola_grabacion = True
409 ---print_info('[dev][info] Grabacion finalizada')
410 ---except subprocess.CallProcessError:
411 ----print_error('[dev][error] Error subprocesso grabacion')
412
413 -def cola_proceso(self):
414 --"""
415 --Establece la cola de proceso si existe o no una grabacion previa
416
417 --:return: Verdadero si existe una grabacion previa a procesar, Falso en
↳ el caso contrario
418 --"""
419 --if self.__cola_grabacion is True:
420 ---self.__cola_grabacion = False
421 ---self.grabando = False
422 ---return True
423 ---else:
424 ---return False
425
426 -def reestablecer_proceso(self):
427 --"""
428 --Reestablece el estado del proceso de grabacion
429
430 --"""
431 --if self.__en_proceso:
432 ---self.__en_proceso = False
433
434 -"""
435
436 --Procesos
437
438 -"""
439
440 -@staticmethod
441 -def preparar_texto(texto, motor='gtts'):
442 --"""
443 --Se encarga de formatear el texto de entrada para adecuarla a las
↳ diferentes herramientas de conversion de texto a voz
444
445 --:param texto: El texto a formatear, obligatorio
446 --:param motor: La herramienta en la cual se desea procesar el texto, por
↳ defecto 'gtts'
447 --:return: El texto formateado en un array de la forma [bloque, bloque,
↳ ..., bloque]
448 --"""
449 --if motor != 'festival':
450 ---texto = texto.lower()

```

```

451 --if motor == 'festival':
452 ---return [[texto.encode('latin-1', 'ignore')]]
453 --if motor == 'bing':
454 ---texto = texto.replace('oesi', 'oesi')
455 --largo_base = {'gtts': 700, 'voicerss': 900, 'bing': 750}
456 --largo_espa = {'gtts': 650, 'voicerss': 850, 'bing': 700}
457 --print_info('[dev][info] Largo texto original:
↳ {}'.format(str(len(texto))))
458 --arr_texto = []
459 --if len(texto) < largo_base[motor]:
460 ---arr_texto.append(texto)
461 --else:
462 ---ou = True
463 ---while ou:
464 ----if len(texto) < largo_base[motor]:
465 -----ou = False
466 -----arr_texto.append(texto)
467 ----else:
468 -----pos = texto.find(' ', largo_espa[motor])
469 -----arr_texto.append(texto[0:pos])
470 -----texto = texto[pos + 1:]
471 --return [arr_texto]
472
473
474 -def comandos_microfono(self, motor, ajuste_ruido=False, mensaje=False):
475 --"""
476 --Reconoce un comando a partir de la linea directa del microfono
477
478 --:param motor: La herramienta utilizada para el reconocimiento de voz a
↳ texto
479 --:param ajuste_ruido: Si es verdadero se realizara el ajuste de umblar
↳ de ruido para la finalizacion de la deteccion, por defecto falso
480 --:param mensaje: Si es verdadero se reproduce un mensaje que indica el
↳ dictado del comando, por defecto falso
481 --:return: Si el comando es valido lo retorna de la manera [True,
↳ comando, valor], caso contrario [False, None, None]
482 --"""
483 --with sp.Microphone() as fuente:
484 ---fuente.device_index = self.__device
485 ---if ajuste_ruido:
486 ----pre = self.reconocedor.energy_threshold
487 ----self.reconocedor.adjust_for_ambient_noise(fuente, 0.5)
488 ----nue = self.reconocedor.energy_threshold
489 ----print_info('[dev][info] Ajustando ruido de fondo -
↳ {}--->{}'.format(str(int(pre)), str(int(nue))))
490 ---if mensaje:
491 ----self.reproducir_audio_menu('dicte')
492 ----grabacion_old = self.reconocedor.listen(fuente)
493 ---self.reproducir_audio_menu('espere')
494 --if motor == 'sphinx' or motor == 'hibrido':
495 ---alterno = False
496 ---try:
497 ----resultado = self.reconocedor.recognize_sphinx(grabacion_old,
↳ language='es-ES_hotwords',
498 ----- keyword_entries=self.diccionario)
499 ----print_info_nb('[dev][info][mic] salida: ')
500 ----print_info_rv(resultado)

```

```

501 ----if resultado.find('buscar') > -1 or resultado.find('libro') > -1:
502 -----self.__rec.suprimir_ruido(grabacion_old.get_wav_data(), 'wav',
    ↳ 'filtro_temporal', 0.17)
503 -----comando = self.leer_comando_archivo('records/out.wav', motor=motor,
    ↳ original='records/temp_voz.wav')
504 ----else:
505 -----comando = self.estructurar_comando(texto=resultado)
506 ---except sp.UnknownValueError:
507 ---print('[dev][warn] Texto no reconocido')
508 ----alternativo = True
509 ---except sp.RequestError:
510 ---print("[dev][warning] error Sphinx")
511 ----alternativo = True
512 ----if alternativo:
513 ----self.__rec.suprimir_ruido(grabacion_old.get_wav_data(), 'wav',
    ↳ 'filtro_temporal', 0.17)
514 ----comando = self.leer_comando_archivo('records/out.wav', motor='wit',
    ↳ original='records/temp_voz.wav')
515 ---else:
516 ---extension = 'wav'
517 ---self.__rec.suprimir_ruido(grabacion_old.get_wav_data(), extension,
    ↳ 'filtro_temporal', 0.17)
518 ---comando =
    ↳ self.leer_comando_archivo('records/out.{}'.format(extension),
    ↳ motor=motor)
519 ---return comando
520
521 -def lectura_microfono(self, ajuste_ruido=False):
522 -"""
523 --Se encarga de detectar el inicio del reconocimiento de comando a traves
    ↳ de microfono
524
525 --:param ajuste_ruido: Define si se debe ajustar el umbral de ruido para
    ↳ la finalizacion de deteccion de voz
526 --:return: True si es valido, False en caso contrario
527 -"""
528 --with sp.Microphone() as fuente:
529 ---fuente.device_index = self.__device
530 ---if ajuste_ruido:
531 ----pre = self.reconocedor.energy_threshold
532 ----self.reconocedor.adjust_for_ambient_noise(fuente, 0.5)
533 ----nue = self.reconocedor.energy_threshold
534 ----print_info('[dev][info] Ajuste de ruido ambiental:
    ↳ {}--->{}'.format(str(int(pre)), str(int(nue))))
535 ---try:
536 ----grabacion = self.reconocedor.listen(fuente, timeout=5)
537 ---except sp.WaitTimeoutError:
538 ----print_error('[dev][error] Expiro tiempo de escucha')
539 ----return False
540 ---try:
541 ----resultado = self.reconocedor.recognize_sphinx(grabacion,
    ↳ language='es-ES_hotwords',
542 ----- keyword_entries=[('prototipo', 1.0), ('inicio', 1.0)])
543 ----print_info_nb('[dev][info][mic] salida: ')
544 ----print_info_rv(resultado)
545 ----if resultado.find('prototipo') > -1 or resultado.find('inicio') > -1:
546 -----return True

```

```

547 ----else:
548 ----print_error('[dev][error] Salida erronea: {}'.format(resultado))
549 ----return False
550 ---except sp.UnknownValueError:
551 ----print_error('[dev][error] No se detecto nada')
552 ----return False
553 ---except sp.RequestError:
554 ----print("[dev][warning] error Sphinx")
555 ----return False
556
557 -@staticmethod
558 -def __letras_a_numero(texto):
559 --"""
560 --Convierte el texto ingresado a su formato de digitos
561
562 --:param texto: El texto numerico a convertir
563 --:return: Los numeros convertidos a digitos
564 --"""
565 --numero = 'mcd'
566 --cientos = [['doscientos', '2'], ['trescientos', '3'], ['cuatrocientos',
↵ '4'],
567 ---- ['quinientos', '5'], ['seiscientos', '6'], ['setecientos', '7'],
↵ ['ochocientos', '8'],
568 ---- ['novecientos', '9'], ['ciento', '1'], ['cien', '1']]
569 --decenas = [['diez', '1b'], ['once', '11'], ['doce', '12'], ['trece',
↵ '13'], ['catorce', '14'], ['quince', '15'],
570 ---- ['dieciseis', '16'], ['dieciseis', '16'], ['diecisiete', '17'],
↵ ['dieciocho', '18'],
571 ---- ['diecinueve', '19'], ['veinte', '2b'], ['veintiuno', '21'],
↵ ['veintidos', '22'],
572 ---- ['veintitres', '23'],
573 ---- ['veinticuatro', '24'], ['veinticinco', '25'], ['veintiseis', '26'],
↵ ['veintisiete', '27'],
574 ---- ['veintiocho', '28'], ['veintinueve', '29'], ['treinta', '3b'],
↵ ['cuarenta', '4b'],
575 ---- ['cincuenta', '5b'],
576 ---- ['sesenta', '6b'], ['setenta', '7b'], ['ochenta', '8b'], ['noventa',
↵ '9b']]
577 --unidades = [['uno', '1'], ['dos', '2'], ['tres', '3'], ['cuatro', '4'],
↵ ['cinco', '5'], ['seis', '6'],
578 ---- ['siete', '7'], ['ocho', '8'], ['nueve', '9']]
579 --miles = False
580 --ciento = False
581 --for i in range(1, len(unidades)):
582 ---if texto.find(unidades[i][0] + ' mil') != -1:
583 ----texto = texto.replace(unidades[i][0] + ' mil', '')
584 ----numero = numero.replace('m', unidades[i][1])
585 ----miles = True
586 --if texto.find('mil') > -1:
587 ---numero = numero.replace('m', '1')
588 ---texto = texto.replace('mil', '')
589 ---miles = True
590 ---else:
591 ---numero = numero.replace('m', '')
592 --for i in range(0, len(cientos)):
593 ---if texto.find(cientos[i][0]) > -1:
594 ----texto = texto.replace(cientos[i][0], '')

```

```

595 ----numero = numero.replace('c', cientos[i][1])
596 ----ciento = True
597 ----break
598 --for i in range(0, len(decenas)):
599 ---if texto.find(decenas[i][0]) > -1:
600 ----texto = texto.replace(decenas[i][0], '')
601 ----if decenas[i][1].find('b') > -1:
602 ----numero = numero.replace('d', decenas[i][1])
603 ----else:
604 ----numero = numero.replace('d', decenas[i][1]).replace('u', '')
605 ----break
606 --for i in range(0, len(unidades)):
607 ---if texto.find(unidades[i][0]) > -1:
608 ----numero = numero.replace('u', unidades[i][1])
609 ----break
610 --if miles or ciento:
611 ---numero = numero.replace('c', '0').replace('d', '0')
612 --else:
613 ---numero = numero.replace('c', '').replace('d', '')
614 --if numero.find('bu') > -1:
615 ---numero = numero.replace('bu', '0')
616 --else:
617 ---numero = numero.replace('c', '').replace('b', '').replace('u', '0')
618 --return numero
619
620 -def estructurar_comando(self, texto):
621 --"""
622 --Genera una comando a partir del texto ingresado a partir del
623 ↪ reconocimiento de voz
624
625 --:param texto: El texto obtenido a partir del reconocimiento de voz
626 --:return: Si el comando es valido retorna dicho comando de la forma
627 ↪ [True, Comando, Valor] caso contrario retorna [False, None, None]
628 --"""
629 ---Bloque de coincidencias
630 ---"""
631 --texto = texto.lower()
632 --if texto.find('repetir ') != -1:
633 ---if texto.find('lista') != -1:
634 ----comando = [True, 'comando_repetir_lista', ' ']
635 ---if texto.find('text') != -1:
636 ----comando = [True, 'comando_repetir_texto', ' ']
637 --elif texto.find('navegacion ') != -1:
638 ---if texto.find('avanzar') != -1:
639 ----self.indice_navegacion += 10
640 ----comando = [True, 'comando_navegacion', self.indice_navegacion]
641 ---elif texto.find('regresar') != -1:
642 ----if self.indice_navegacion > 0:
643 ----self.indice_navegacion -= 10
644 ----comando = [True, 'comando_navegacion', self.indice_navegacion]
645 ----else:
646 ----comando = [True, 'comando_limite', ' ']
647 --elif texto.find('resultado ') != -1:
648 ---arr_texto = texto.split(' ')
649 --for bloque in arr_texto:

```

```

650 ----if bloque.find('resultado') != -1:
651 -----texto = bloque
652 -----break
653 ---x = 1
654 ---for i in self.numeros:
655 ----texto = texto.replace(' ' + i, str(x))
656 ----x += 1
657 ---texto = re.sub('[^0-9]', '', texto)
658 ---comando = [True, 'comando_resultado', int(texto)]
659 --elif texto.find('buscar') != -1:
660 ---pos = texto.find('buscar')
661 ---comando = [True, 'comando_buscar', texto[pos + 7:len(texto)]]
662 --elif texto.find('libro ') != -1:
663 ---if texto.find('avanzar') != -1:
664 ----comando = [True, 'comando_pdf_navegar', 1]
665 ---elif texto.find('regresar') != -1:
666 ----comando = [True, 'comando_pdf_navegar', -1]
667 ---else:
668 ----texto = texto.strip()
669 ----pos = texto.find('libro')
670 ----numeros = texto[pos + 6:len(texto)]
671 ----try:
672 -----prueba = int(numeros.replace(' ', ''))
673 -----arr_numeros = numeros.split(' ')
674 -----final = 0
675 -----for i in range(0, len(arr_numeros)):
676 -----final += int(arr_numeros[i])
677 -----numeros = str(final)
678 ----except ValueError:
679 ----numeros = self.__letras_a_numero(texto)
680 ----comando = [True, 'comando_pdf', numeros]
681 --elif texto.find('lectura') != -1:
682 ---comando = [True, 'comando_leer', ' ']
683 --return comando
684
685 -def leer_comando_archivo(self, archivo, motor='sphinx', original=None):
686 --"""
687 --Genera un comando a partir de un archivo de audio que contenga voz
688 ↪ utilizando la herramienta de voz a texto
689 --proporcionada
690 --:param archivo: El archivo de audio con el contenido de voz
691 --:param motor: La herramienta de voz a texto para el reconocimiento de
692 ↪ voz, por defecto 'sphinx'
693 --:param original: El archivo de audio original sin procesar
694 --:return: El comando en caso de encontrarse alguno de la forma [True,
695 ↪ comando, valor], [False, None, None] en caso contrario
696 --"""
697 --# Nuevo sistema de comandos
698 --texto = ''
699 --no_repetir_hibrido = False
700
701 ---Bloque de procesos
702 --"""
703 --if motor == 'ibm':
704 ---with open(archivo, 'rb') as audio_file:

```



```

704 ----try:
705 -----print ('Procesando: {}'.format (archivo))
706 -----salida = self.__ibm_sp.recognize (audio=audio_file,
    ↪ content_type='audio/wav',
707 ----- model='es-ES_BroadbandModel')
708 -----salida = salida.get_result ()
709 -----transcripcion = []
710 -----palabras = None
711 -----for palabras in salida["results"]:
712 -----if "alternatives" not in palabras:
713 -----print ('No salida')
714 -----try:
715 -----for hipotesis in palabras["alternatives"]:
716 -----if "transcript" in hipotesis:
717 -----transcripcion.append (hipotesis['transcript'])
718 -----texto = ' '.join (transcripcion)
719 -----except TypeError:
720 -----texto = ''
721 -----print_info_nb (' [dev][info][stt][ibm] Salida base: ')
722 -----print_info_rv (texto)
723 -----except WatsonApiException:
724 -----print_error (' [dev][error][stt][ibm] No se encontro texto valido')
725 --else:
726 --if motor == 'sphinx':
727 --with sp.AudioFile (original) as fuente_audio:
728 --self.reconocedor.adjust_for_ambient_noise (fuente_audio, 0.5)
729 --audio = self.reconocedor.record (source=fuente_audio)
730 --else:
731 --with sp.AudioFile (archivo) as fuente_audio:
732 --audio = self.reconocedor.record (source=fuente_audio)
733 --if motor == 'sphinx' or motor == 'hibrido':
734 --try:
735 -----texto = self.reconocedor.recognize_sphinx (audio_data=audio,
    ↪ language='es-ES_hotwords',
736 ----- keyword_entries=self.diccionario)
737 -----texto = texto.strip ()
738 -----except sp.UnknownValueError:
739 -----print_error (' [dev][error][stt][{}] No se encontro texto
    ↪ valido'.format (motor))
740 -----if motor == 'hibrido':
741 -----print_info (' [dev][info][stt][hibrido] Cambiando a wit.ai')
742 -----try:
743 -----texto = self.reconocedor.recognize_wit (audio_data=audio,
    ↪ key=self.api_key['stt_wit'])
744 -----if texto.find ('buscar') != -1:
745 -----no_repetir_hibrido = True
746 -----except (sp.UnknownValueError, sp.RequestError):
747 -----print_error (' [dev][error][stt][hibrido] No se encontro texto
    ↪ valido')
748 -----else:
749 -----print_info (' [dev][info][stt][sphinx] Cambiando a modo completo')
750 -----try:
751 -----texto = self.reconocedor.recognize_sphinx (audio_data=audio,
    ↪ language='es-ES')
752 -----texto = texto.strip ()
753 -----except sp.UnknownValueError:

```

```

754 -----print_error(' [dev][error][stt][sphinx] No se encontro texto
      ↳ valido')
755 ----print_info_nb(' [dev][info][stt][sphinx] Salida base: ')
756 ----print_info_rv(texto)
757 ----if motor == 'wit':
758 ----try:
759 -----texto = self.reconocedor.recognize_wit(audio_data=audio,
      ↳ key=self.api_key['stt_wit'])
760 ----except (sp.UnknownValueError, sp.RequestError):
761 ----print_error(' [dev][error][stt][wit] No se encontro texto valido')
762 ----print_info_nb(' [dev][info][stt][wit] Salida base: ')
763 ----print_info_rv(texto)
764 ----if motor == 'google':
765 ----try:
766 -----texto = self.reconocedor.recognize_google(audio_data=audio,
      ↳ language='es-ES',
767 ----- key=self.api_key['stt_google'])
768 ----except sp.UnknownValueError:
769 ----print_error(' [dev][error][stt][google] No se encontro texto valido')
770 ----print_info_nb(' [dev][info][stt][google] Salida base: ')
771 ----print_info_rv(texto)
772 ----if motor == 'bing':
773 ----try:
774 -----texto = self.reconocedor.recognize_bing(audio_data=audio,
      ↳ language='es-ES',
775 -----key=self.api_key['tts_bing']).replace(',',' ')
776 ----except sp.UnknownValueError:
777 ----print_error(' [dev][error][stt][bing] No se encontro texto valido')
778 ----texto = texto[0:len(texto) - 1]
779 ----print_info_nb(' [dev][info][stt][bing] Salida base: ')
780 ----print_info_rv(texto)
781 ----if texto.find('buscar') > -1:
782 ----if motor == 'sphinx':
783 ----print_info(' [dev][info][stt][sphinx] Mejorar busqueda - modo sphinx')
784 ----texto = self.reconocedor.recognize_sphinx(audio_data=audio,
      ↳ language='es-ES')
785 ----texto = texto.strip()
786 ----texto = texto.replace(' ', ' ')
787 ----print_info(' [dev][info][stt][sphinx] Salida base: {}'.format(texto))
788 ----if motor == 'hibrido' and no_repetir_hibrido is False:
789 ----print_info(' [dev][info][stt][hibrido] Mejorar busqueda - modo
      ↳ hibrido')
790 ----texto = self.reconocedor.recognize_wit(audio_data=audio,
      ↳ key=self.api_key['stt_wit'])
791 ----print_info_nb(' [dev][info][stt][hibrido] Salida base: ')
792 ----print_info_rv(texto)
793 ----pos = texto.find(' ')
794 ----texto = 'buscar {}'.format(texto[pos + 1:len(texto)])
795 --texto = texto.lower()
796 --return self.estructurar_comando(texto)

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: ocr_bing.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 """
6
7     Una libreria para utilizar la API REST del servicio Azure Computer
8     ↪ Vision Service
9
10 """
11 import cv2
12 import requests
13 from PIL import Image
14 from procesador_imagen import ProcesarImagen
15
16 class OCRBing:
17     -def __init__(self, api_key=None):
18     --self.base_url =
19     ↪ 'https://westcentralus.api.cognitive.microsoft.com/vision/v2.0/ocr'
20     --self.api_key = api_key
21     --self.__nnn = 0
22     --self.__debug = False
23     --self.__texto_1 = ''
24     --self.__texto_2 = ''
25
26     -def __guardar_imagen(self, img, texto=''):
27     --if self.__debug:
28     ---if len(str(self.__nnn)) < 2:
29     ----numero = '0' + str(self.__nnn)
30     ---else:
31     ----numero = str(self.__nnn)
32     ---cv2.imwrite('images/debug/{}.png'.format(numero, str(texto)), img)
33     ---self.__nnn += 1
34
35     -@staticmethod
36     -def __pre_procesar_imagen(imagen):
37     --"""
38     --Realiza un proceso de mejora de la calidad de la imagen para el
39     ↪ procesamiento posterior de la misma
40     --:param imagen: La imagen a procesar
41     --:return: La imagen procesada, La imagen original en caso de error
42     --"""
43     --procesador = ProcesarImagen()
44     --procesador.imdir = 'temp_vision'
45     --if isinstance(imagen, str):
46     ---base = cv2.imread(imagen)
47     ---else:
48     ---base = imagen
49     ---del imagen
50     ---base = procesador.a_gris(imagen=base)
51     ---binaria = procesador.binarizado(imagen=base,
52     ↪ tipo=procesador.tipo_binarizacion.GAUSS, v1=255, v2=16, bil=1)
53     ---giro = procesador.corregir_angulo_new(imagen=binaria, minimo=0.01,
54     ↪ borde=1, escalado=1,
55     ----- metodo=procesador.tipo_corrreccion_angulo.LINEAS,

```

```

    ↪ procesado=False)
52 --print(giro)
53 --if giro:
54 ---base = cv2.warpAffine(base, giro[0], (giro[1], giro[2]),
    ↪ flags=cv2.INTER_CUBIC,
55 ----- borderMode=cv2.BORDER_CONSTANT, borderValue=[255])
56 --media = Image.fromarray(cv2.cvtColor(base,
    ↪ cv2.COLOR_GRAY2RGB)).convert(mode='L')
57 --media.save('images/ocr_vision/final_1.jpg', 'JPEG')
58 --return True
59
60 -def texto_de_imagen(self, file_path, api_key=None, language='unk'):
61 --"""
62 --Extrae el texto de la imagen proporcionada
63 --:param file_path: Ruta de la imagen
64 --:param api_key: Key API
65 --:param language: El lenguaje principal en la imagen
66 --:return: El texto de la imagen
67 --"""
68 --texto_imagen = ''
69 --temp_path = 'images/ocr_vision/final_1.jpg'
70 --if api_key is not None:
71 ---self.api_key = api_key
72 --if self.api_key is None:
73 ---raise OCRBingNoKeyException('[dev][error][ocr_vision] No se
    ↪ proporciono ninguna api key')
74 --headers = {'Ocp-Apim-Subscription-Key': self.api_key}
75 --config = '?language={}&detectOrientation=true'.format(language)
76 --try:
77 ---self.__pre_procesar_imagen(file_path)
78 ---r = requests.post(url='{}'.format(self.base_url, config),
79 ----- files={'imageFile': open(temp_path, 'rb')},
80 ----- headers=headers)
81 --except requests.exceptions.RequestException as ex:
82 ---raise OCRBingRequestException('[dev][error][ocr_vision] Error request:
    ↪ {}'.format(ex))
83 --if int(r.status_code) != 200:
84 ---if int(r.status_code) != 403:
85 ----raise OCRBingBadRequestException('[dev][error][ocr_vision] Error bad
    ↪ request: {}'.format(r.status_code))
86 --if int(r.status_code) == 200:
87 ---json_data = r.json()
88 ---out = ''
89 ---for boxes in json_data['regions']:
90 ----for lines in boxes['lines']:
91 -----for words in lines['words']:
92 -----out = '{} {}'.format(out, words['text'])
93 ----texto_imagen = out.strip()
94 ---return texto_imagen
95
96
97 class OCRBingNoKeyException(Exception):
98 -def __init__(self, message):
99 --self.message = '{}'.format(message)
100 --super(OCRBingNoKeyException, self).__init__(self.message)
101
102

```

```
103 class OCRBingBadRequestException(Exception):
104     def __init__(self, message):
105         self.message = '{}'.format(message)
106         super(OCRBingBadRequestException, self).__init__(self.message)
107
108
109 class OCRBingRequestException(Exception):
110     def __init__(self, message):
111         self.message = '{}'.format(message)
112         super(OCRBingRequestException, self).__init__(self.message)
```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: ocr_google_drive.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 import cv2
6 import sys
7 import numpy as np
8 from PIL import Image
9 from time import time
10 from pydrive.auth import GoogleAuth
11 from pydrive.drive import GoogleDrive
12 from threading import Thread
13 from procesador_imagen import ProcesarImagen
14
15 class OCRGoogleDrive:
16     def __init__(self):
17         self.__nnn = 0
18         self.__debug = False
19         self.__texto_1 = ''
20         self.__texto_2 = ''
21     -
22     def __guardar_imagen(self, img, texto=''):
23         if self.__debug:
24             if len(str(self.__nnn)) < 2:
25                 numero = '0' + str(self.__nnn)
26             else:
27                 numero = str(self.__nnn)
28             cv2.imwrite('images/debug/{}.png'.format(numero, str(texto)), img)
29             self.__nnn += 1
30
31     def __pre_procesar_imagen(self, imagen):
32         """
33         --Realiza un proceso de mejora de la calidad de la imagen para el
34         ↪ procesamiento posterior de la misma
35         --:param imagen: La imagen a procesar
36         --:return: La imagen procesada, La imagen original en caso de error
37         """
38         procesador = ProcesarImagen()
39         procesador.imdir = 'temp_drive'
40         if isinstance(imagen, str):
41             base = cv2.imread(imagen)
42         else:
43             base = imagen
44         del imagen
45         base = procesador.a_gris(imagen=base)
46         binaria = procesador.binarizado(imagen=base,
47             ↪ tipo=procesador.tipo_binarizacion.GAUSS, v1=255, v2=16, bil=1)
48         giro = procesador.corregir_angulo_new(imagen=binaria, minimo=0.5,
49             ↪ borde=1, escalado=1,
50             ----- metodo=procesador.tipo_correccion_angulo.LINEAS,
51             ↪ procesado=False)
52         if giro:
53             base = cv2.warpAffine(base, giro[0], (giro[1], giro[2]),
54                 ↪ flags=cv2.INTER_CUBIC,
55                 ----- borderMode=cv2.BORDER_CONSTANT, borderValue=[255])
56         media = Image.fromarray(cv2.cvtColor(base,

```

```

    ↪ cv2.COLOR_GRAY2RGB)).convert(mode='L')
52 --media.save('images/ocr_drive/final_1.jpg', 'JPEG')
53 --return True
54
55 -def texto_de_imagen(self, image_path):
56 --"""
57 --Extrae el texto de la imagen proporcionada
58
59 --:param image_path: La ruta de la imagen a procesar
60 --:return: El texto de la imagen obtenido, False en caso de error
61 --"""
62 --self.__pre_procesar_imagen(imagen=image_path)
63 --gauth = GoogleAuth(settings_file='settings.yaml')
64 --gauth.LocalWebserverAuth()
65 --g = GoogleDrive(gauth)
66 --t1 = time()
67 --file_g = g.CreateFile({'title': 'ocr image 1'})
68 --file_g['mimeType'] = 'application/vnd.google-apps.document'
69 --file_g.Upload({'ocrLanguage': 'es'})
70 --file_g.SetContentFile('images/ocr_drive/final_1.jpg')
71 --file_g.Upload()
72 --texto_get = file_g.GetContentString('text/plain').encode('utf-8',
    ↪ 'replace').decode('utf-8-sig', 'replace')
73 --file_g.Trash()
74 --return texto_get

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: ocr_space.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 """
6
7 -Una libreria para utilizar la API REST del servicio ocr.space
8
9 """
10 import os
11 import cv2
12 import sys
13 import json
14 import math
15 from time import time
16 import requests
17 import numpy as np
18 from PIL import Image
19 from procesador_imagen import ProcesarImagen
20
21 class OCRSpace:
22     -def __init__(self, api_key=None):
23     -"""
24     -A
25
26     -:param api_key:
27     -"""
28     --self.base_url = 'https://api.ocr.space/parse/image'
29     --self.api_key = api_key
30     --self.__nnn = 0
31     --self.__debug = False
32     --self.__texto_1 = ''
33     --self.__texto_2 = ''
34
35     -def __guardar_imagen(self, img, texto=''):
36     --if self.__debug:
37     ---if len(str(self.__nnn)) < 2:
38     ----numero = '0' + str(self.__nnn)
39     ---else:
40     ----numero = str(self.__nnn)
41     ---cv2.imwrite('images/debug/{}.{}.png'.format(numero, str(texto)), img)
42     ---self.__nnn += 1
43
44     -@staticmethod
45     -def __pre_procesar_imagen(imagen):
46     -"""
47     --Realiza un proceso de mejora de la calidad de la imagen para el
48     ↪ procesamiento posterior de la misma
49
50     --:param imagen: La imagen a procesar
51     --:return: La imagen procesada, La imagen original en caso de error
52     -"""
53     --procesador = ProcesarImagen()
54     --procesador.imdir = 'temp_space'
55     --if isinstance(imagen, str):
56     ---base = cv2.imread(imagen)

```



```

56 --else:
57 ---base = imagen
58 --del imagen
59 --base = procesador.a_gris(imagen=base)
60 --binaria = procesador.binarizado(imagen=base,
    ↳ tipo=procesador.tipo_binarizacion.GAUSS, v1=255, v2=16, bil=1)
61 --giro = procesador.corregir_angulo_new(imagen=binaria, minimo=0.01,
    ↳ borde=1, escalado=1,
62 ----- metodo=procesador.tipo_corrreccion_angulo.LINEAS,
    ↳ procesado=False)
63 --if giro:
64 ---base = cv2.warpAffine(base, giro[0], (giro[1], giro[2]),
    ↳ flags=cv2.INTER_CUBIC,
65 ----- borderMode=cv2.BORDER_CONSTANT, borderValue=[255])
66 --media = Image.fromarray(cv2.cvtColor(base,
    ↳ cv2.COLOR_GRAY2RGB)).convert(mode='L')
67 --media.save('images/ocr_space/final_1.jpg', 'JPEG')
68 --return True
69
70 -def corregir_tamano(self, imagen):
71 --path = 'images/ocr_space/final_1.jpg'
72 --if isinstance(imagen, str):
73 ---path = imagen
74 ---imagen = Image.open(imagen)
75 --else:
76 ---imagen = Image.fromarray(cv2.cvtColor(imagen, cv2.COLOR_GRAY2RGB))
77 --x_y = imagen.size
78 --if x_y[0] * x_y[1] > 900000:
79 ---imagen.save(path, 'JPEG', quality=70)
80
81 -def texto_de_imagen(self, file_path, api_key=None, language='spa',
    ↳ overlay=False, scale=False):
82 --"""
83 --Extrae el texto de la imagen proporcionada
84
85 --:param file_path: La ruta de la imagen a procesar
86 --:param api_key: Key API
87 --:param language: Idioma del texto de la imagen
88 --:param overlay:
89 --:param scale:
90 --:return: El texto de la imagen obtenido, False en caso de error
91 --"""
92 --texto_imagen = ''
93 --temp_path = 'images/ocr_space/final_1.jpg'
94 --if api_key is not None:
95 ---self.api_key = api_key
96 --if self.api_key is None:
97 ---raise OCRSpaceNoKeyException('[dev][error][ocr_space] No se
    ↳ proporciono ninguna api key')
98 --payload = {'scale': scale, 'isOverlayRequired': overlay, 'apikey':
    ↳ self.api_key, 'language': language,
99 ----- 'detectOrientation': True}
100 --try:
101 ---self.__pre_procesar_imagen(file_path)
102 ---self.corregir_tamano('images/ocr_space/final_1.jpg')
103 ---with open(temp_path, 'rb') as contenido:
104 -----r = requests.post(url=self.base_url,

```

```

105 ----- files={'imagen.jpg': contenido},
106 ----- data=payload)
107 --except requests.exceptions.RequestException as ex:
108 ---raise OCRSpaceRequestException('[dev][error][ocr_space] Error request:
    ↪ {}'.format(ex))
109 --if int(r.status_code) != 200:
110 ---if int(r.status_code) != 403:
111 ----raise OCRSpaceBadRequestException('[dev][error][ocr_space] Error bad
    ↪ request: {}'.format(r.status_code))
112 --if int(r.status_code) == 200:
113 ---json_data = r.json()
114 ---for x in json_data['ParsedResults']:
115 ----texto_imagen = x['ParsedText']
116 ---texto_imagen = ' '.join(texto_imagen.split())
117 ---return texto_imagen
118
119
120 class OCRSpaceNoKeyException(Exception):
121 -def __init__(self, message):
122 --self.message = '{}'.format(message)
123 --super(OCRSpaceNoKeyException, self).__init__(self.message)
124
125
126 class OCRSpaceBadRequestException(Exception):
127 -def __init__(self, message):
128 --self.message = '{}'.format(message)
129 --super(OCRSpaceBadRequestException, self).__init__(self.message)
130
131
132 class OCRSpaceRequestException(Exception):
133 -def __init__(self, message):
134 --self.message = '{}'.format(message)
135 --super(OCRSpaceRequestException, self).__init__(self.message)

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: player.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 import os
6 import sys
7 import argparse
8 from time import sleep, time as time
9 from termcolor import cprint
10
11 def print_error(x): cprint(x, 'red', attrs=['bold', 'underline'], end='
↳ \n')
12 def print_info(x): cprint(x, 'yellow', attrs=['bold'], end=' \n')
13 def print_log(x): cprint(x, 'cyan', attrs=['bold'], end=' \n')
14 def print_log_a(x): cprint(x, 'magenta', attrs=['underline'], end=' \n')
15 def print_info_nb(x): cprint(x, 'yellow', attrs=['bold'], end='')
16 def print_info_rv(x): cprint(x, 'magenta', attrs=['bold'], end=' \n')
17
18 if __name__ == '__main__':
19 -parser = argparse.ArgumentParser(description='Background audio player')
20 -parser.add_argument('--engine', '-g', help='Motor', default='gtts',
↳ required=False, choices=['gtts',
21 ----- 'bing',
22 ----- 'voicerss',
23 ----- 'festival'])
24 -parser.add_argument('--numero', '-n', help='Numero de archivos (a partir
↳ de cero)', default=0, required=False)
25 -argu = parser.parse_args()
26 -motor = argu.engine
27 -numero = int(argu.numero)
28 -path = 'audio/{}/'.format(motor)
29 -print_info('[dev][info][player] Engine: {}'.format(motor))
30 -print_info('[dev][info][player] Medios: {}'.format(str(numero)))
31 -duracion_total = 0
32 -intentos = 20
33 -rate_arr = {'gtts': '1.05',
34 ----'bing': '0.95',
35 ----'voicerss': '1.05',
36 ----'festival': '0.95'}
37 -gain_arr = {'gtts': '0.9',
38 ----'bing': '1',
39 ----'voicerss': '1',
40 ----'festival': '0.9'}
41 -while True:
42 --file_path_e = '{}{}.mp3'.format(path, 0)
43 --if motor == 'festival':
44 ---file_path_e = '{}{}.wav'.format(path, 0)
45 --if motor == 'voicerss':
46 ---file_path_e = '{}{}.ogg'.format(path, 0)
47 --intentos -= 1
48 --sleep(0.2)
49 --if intentos == 0:
50 ---print_info('[dev][info][player] No se encontraron medios')
51 ---sys.exit(1)
52 --else:
53 ---break

```

```

54 for i in range(0, numero):
55 --file_path = '{}{}.mp3'.format(path, str(i))
56 --if motor == 'voicerss':
57 ---file_path = '{}{}.ogg'.format(path, str(i))
58 ---sleep(0.1)
59 --if os.path.exists(file_path):
60 ---print_info('[dev][info][player] Inicio reproduccion:
    ↳ {}'.format(file_path))
61 ---d1 = time()
62 ----vlc_args = '-A=alsa --play-and-exit --gain={} --rate={}'
63 ----print_log('cvlc {}
    ↳ /home/pi/prototipo/proyecto/proto/{}'.format(vlc_args.format(
64 -----gain_arr[motor],
65 -----rate_arr[motor]),
66 ----- file_path))
67 ----os.system('cvlc {} /home/pi/prototipo/proyecto/proto/{} 2>
    ↳ /dev/null'.format(vlc_args.format(
68 ----gain_arr[motor], rate_arr[motor]), file_path))
69 ---d2 = round(time() - d1, 2)
70 ---print_info('[dev][info][player] Fin reproduccion: {} - Duracion: {}
    ↳ segundos'.format(file_path, str(d2)))
71 ---duracion_total = duracion_total + d2
72 --else:
73 ---print_info('[dev][info][player] No encontrado: {}'.format(file_path))
74 -horas = 0
75 -minutos = 0
76 -segundos = 0
77 --if duracion_total > 60:
78 --minutos = int(duracion_total // 60)
79 --segundos = round(duracion_total - (minutos * 60), 2)
80 --else:
81 --segundos = round(duracion_total, 2)
82 --if minutos > 60:
83 --horas = int(minutos // 60)
84 --minutos = minutos - (horas * 60)
85 -print_log('[dev][info][player] Total reproducido:
    ↳ {}:{}:{}'.format(str(int(horas)), str(int(minutos)), str(segundos)))
86 -sys.exc_info()

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: procesador_imagen.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 import os
6 import cv2
7 import sys
8 import math
9
10 import random
11 import numpy as np
12 from enum import Enum
13 from collections import Counter
14
15
16 class TipoBinarizado(Enum):
17     -MEAN = 1
18     -GAUSS = 2
19     -MEAN_GAUSS = 3
20
21
22 class TipoCorte(Enum):
23     -PROCESADO = 1
24     -POSICIONES = 2
25
26
27 class TipoSegmentacion(Enum):
28     -GENERAR = 1
29     -POSICIONES = 2
30
31
32 class TipoCorreccionAngulo(Enum):
33     -LINEAS = 1
34     -AREA_MINIMA = 2
35
36
37 class TipoLimpieza(Enum):
38     -PROCESADO = 1
39     -POSICIONES = 2
40
41
42 class ProcesarImagen:
43
44     -def __init__(self):
45         --self.imdir = 'images/'
46         --self.__debug = True
47         --self.__global_n = 0
48         --self.tipo_binarizacion = TipoBinarizado
49         --self.tipo_corte = TipoCorte
50         --self.tipo_segmentacion = TipoSegmentacion
51         --self.tipo_corrreccion_angulo = TipoCorreccionAngulo
52         --self.tipo_limpieza = TipoLimpieza
53     --pass
54
55     -def a_gris(self, imagen):
56         --"""

```

```

57 --Convierte la imagen proporcionada a escala de grises
58 --:param imagen: La imagen a convertir, obligatorio
59 --:return: La imagen convertida
60 --"""
61 --imagen = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
62 --return imagen
63
64 -def escalar(self, imagen, proporcion):
65 --"""
66 --Escala la imagen de acuerdo a la proporcion ingresada
67
68 --:param imagen: La imagen a escalar, obligatorio
69 --:param proporcion: La proporcion a la que se escalara la imagen, en
  ↪ formato decimal
70 --:return: La imagen escalada
71 --"""
72 --a1, a2 = map(int, imagen.shape)
73 --if a1 < 20 or a2 < 20:
74 ---return imagen
75 --if proporcion > 1:
76 ---imagen = cv2.resize(imagen, None, fx=proporcion, fy=proporcion,
  ↪ interpolation=cv2.INTER_CUBIC)
77 --elif proporcion < 1:
78 ---imagen = cv2.resize(imagen, None, fx=proporcion, fy=proporcion,
  ↪ interpolation=cv2.INTER_LINEAR)
79 --return imagen
80
81 -def demarcar(self, imagen, radio, iteraciones):
82 --"""
83 --Se encarga de demarcar los caracteres contenidos en una imagen
84
85 --:param imagen: La imagen a procesar, obligatorio
86 --:param radio: El radio de demarcacion, al menos 2
87 --:param iteraciones: El numero de veces que se aplicara el proceso de
  ↪ demarcado, al menos 1
88 --:return: La imagen procesada
89 --"""
90 --imagen = cv2.erode(imagen, np.ones((radio, radio), np.uint8),
  ↪ iterations=iteraciones)
91 --return imagen
92
93 -def brillo(self, imagen, brillo=1.05):
94 --"""
95 --Se encarga de modificar el brillo de la imagen proporcionada
96
97 --:param imagen: La imagen a procesar, obligatorio
98 --:param brillo: El ajuste de brillo a realizar, en formato decimal, por
  ↪ defecto 1.05
99 --:return: La imagen procesada
100 --"""
101 --gamma = np.array([(i / 255.0) ** brillo) * 255 for i in np.arange(0,
  ↪ 256)]).astype("uint8")
102 --imagen = cv2.LUT(imagen, gamma)
103 --return imagen
104
105 -def binarizado(self, imagen, tipo=TipoBinarizado.MEAN, v1=0, v2=0,
  ↪ bil=12):

```

```

106 --"""
107 --Convierte una imagen a blanco y negro
108
109 --:param imagen: La imagen a procesar, obligatorio
110 --:param tipo: El tipo de metodo de binarizacion a utilizar, MEAN o
    ↪ GAUSS, por defecto MEAN
111 --:param v1: El rango minimo de ajuste, por defecto 0
112 --:param v2: El rango maximo de ajuste, por defecto 0
113 --:param bil: El valor de filtrado bilateral a pasar a la imagen antes de
    ↪ binarizar, por defecto 12
114 --:return: La imagen procesada
115 --"""
116 --if not isinstance(tipo, TipoBinarizado):
117 ---pass
118 ---metodo = cv2.ADAPTIVE_THRESH_MEAN_C
119 ---if tipo == TipoBinarizado.MEAN:
120 ----metodo = cv2.ADAPTIVE_THRESH_MEAN_C
121 ----if v1 == 0 and v2 == 0:
122 -----v1 = 37
123 -----v2 = 2
124 ---elif tipo == TipoBinarizado.GAUSS:
125 ----metodo = cv2.ADAPTIVE_THRESH_GAUSSIAN_C
126 ----if v1 == 0 and v2 == 0:
127 -----v1 = 11
128 -----v2 = 2
129 ---if tipo != TipoBinarizado.MEAN_GAUSS:
130 ----imagen = cv2.adaptiveThreshold(cv2.bilateralFilter(imagen, bil, 75,
    ↪ 75), 255, metodo,
131 ----- cv2.THRESH_BINARY, v1, v2)
132 ---if tipo == TipoBinarizado.MEAN:
133 ----imagen = cv2.dilate(imagen, np.ones((3, 3), np.uint8), iterations=1)
134 ---else:
135 ----imagen_mean = self.binarizado(imagen, TipoBinarizado.MEAN, v1, v2)
136 ----imagen_gaus = self.binarizado(imagen, TipoBinarizado.GAUSS)
137 ----formal = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15, 1))
138 ----forma2 = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15, 15))
139 ----imagen_mean =
    ↪ cv2.bitwise_not(cv2.morphologyEx(cv2.bitwise_not(imagen_mean),
    ↪ cv2.MORPH_OPEN, formal))
140 ----imagen_mean =
    ↪ cv2.bitwise_not(cv2.morphologyEx(cv2.bitwise_not(imagen_mean),
    ↪ cv2.MORPH_OPEN, forma2))
141 ----imagen = cv2.bitwise_and(imagen_mean, imagen_gaus)
142 ---return imagen
143
144 -def corregir_angulo_new(self, imagen, minimo, maximo=45,
    ↪ alto_minimo=0.001, linea_minima=0.25, linea_maxima=0.01,
145 -----angulo=180, borde=80, escalado=0.5,
    ↪ metodo=TipoCorreccionAngulo.LINEAS, procesado=True):
146 --"""
147 --Realiza el ajuste de angulo a la imagen proporcionada
148
149 --:param imagen: La imagen a procesar, obligatorio
150 --:param minimo: El angulo minimo valido, obligatorio
151 --:param maximo: El angulo maximo valido, por defecto 45
152 --:param alto_minimo: El alto minimo de la linea con respecto al alto de
    ↪ la imagen, por defecto 0.001

```

```

153 --:param linea_minima: El ancho de la linea minima con respecto al ancho
    ↪ de la imagen, por defecto 0.25
154 --:param linea_maxima: El alto maximo de la linea con respecto al alto de
    ↪ la imagen, por defecto 0.01
155 --:param angulo: El valor de angulo de deteccion valido, por defecto 180
156 --:param borde: El borde a aplicar a la imagen antes de realizar el
    ↪ ajuste de angulo
157 --:param escalado: El valor del escalado para realizar el proceso de
    ↪ deteccion de lineas, para mejorar el rendimiento, por defecto 0.5
158 --:param metodo: Tipo de metodo para la deteccion del angulo, pueden ser
    ↪ LINEAS o AREA_MINIMA, por defecto LINEAS
159 --:param procesado: Si se va a procesar la imagen o solo se realizara la
    ↪ deteccion, por defecto True
160 --:return: Si procesado es True, la imagen con la correccion de angula
    ↪ realizada, caso contrario un array con la forma [centro_de_la_imagen,
    ↪ nuevo_ancho, nuevo_alto, -angulo]
161 --"""
162 --alto, ancho = map(int, imagen.shape)
163 --angulo_medio = 0
164 --ag = 0
165 --para_rotacion = cv2.bitwise_not(imagen)
166 --cv2.rectangle(para_rotacion, (0, 0), (ancho, alto), (0, 0, 0), borde)
167 --cv2.rectangle(imagen, (0, 0), (ancho, alto), (255, 255, 255), borde)
168 --para_rotacion = self.escalar(imagen=para_rotacion, proporcion=escalado)
169 --p_r_alto, p_r_ancho = map(int, para_rotacion.shape)
170 --if metodo == self.tipo_correccion_angulo.LINEAS:
171 ---para_rotacion = cv2.morphologyEx(para_rotacion, cv2.MORPH_CLOSE,
    ↪ np.ones((12, 30), np.uint8))
172 ---para_rotacion = cv2.morphologyEx(para_rotacion, cv2.MORPH_OPEN,
    ↪ np.ones((10, 35), np.uint8))
173 ---mi_alto = int(p_r_alto * alto_minimo)
174 ---mi_line = int(p_r_ancho * linea_minima)
175 ---ma_line = int(p_r_ancho * linea_maxima)
176 ---ang_line = np.pi / angulo
177 ---try:
178 ----liness = cv2.HoughLinesP(para_rotacion, mi_alto, ang_line, 1000,
    ↪ minLineLength=mi_line,
179 ----- maxLineGap=ma_line)
180 ----if liness is None:
181 -----para_rotacion = cv2.bitwise_not(para_rotacion)
182 -----angulo_min = self.corregir_angulo_new(imagen=para_rotacion,
    ↪ minimo=minimo, maximo=maximo,
183 ----- alto_minimo=alto_minimo, linea_minima=linea_minima,
184 ----- linea_maxima=linea_maxima, angulo=angulo, borde=borde,
185 ----- escalado=escalado, metodo=TipoCorreccionAngulo.AREA_MINIMA,
186 ----- procesado=False)
187 ----if angulo_min is not False:
188 -----angulo_medio = angulo_min[3]
189 ----else:
190 -----n_lineas = int(len(liness) * 0.005)
191 ----if n_lineas < 1:
192 -----n_lineas = int(len(liness) // 100) + 1
193 ----blank_1 = np.zeros((p_r_alto, p_r_ancho), np.uint8)
194 ----proporcion_alto = int(p_r_alto * 0.25)
195 ----proporcion_ancho = int(p_r_ancho * 0.01)
196 ----angulos = []
197 ----for i in range(0, len(liness), n_lineas):

```



```

198 -----x1, y1, x2, y2 = liness[i][0]
199 -----xx = x2 - x1
200 -----yy = y2 - y1
201 -----m = 0
202 -----if xx != 0:
203 -----m = yy / xx
204 -----angulo_linea = round(math.degrees(math.atan(m)), 4)
205 -----ratio1 = abs(y2 - y1)
206 -----ratio2 = abs(x2 - x1)
207 -----if ratio1 <= proporcion_alto and ratio2 >= proporcion_ancho and \
208 -----y1 >= int(p_r_alto * 0.1) and y2 <= int(p_r_alto * 0.9) and abs(
209 -----angulo_linea) < 40:
210 -----angulos.append(angulo_linea)
211 -----cv2.line(blank_1, (x1, y1), (x2, y2), 255, 1)
212 -----veces = Counter(angulos)
213 -----frecuentes = veces.most_common(5)
214 -----suma = 0
215 -----repetidos = 0
216 -----if len(frecuentes) == 0:
217 -----para_rotacion = cv2.bitwise_not(para_rotacion)
218 -----angulo_min = self.corregir_angulo_new(imagen=para_rotacion,
    ↪ minimo=minimo, maximo=maximo,
219 -----alto_minimo=alto_minimo, linea_minima=linea_minima,
220 -----linea_maxima=linea_maxima, angulo=angulo, borde=borde,
221 -----escalado=escalado,
222 -----metodo=TipoCorreccionAngulo.AREA_MINIMA, procesado=False)
223 -----if angulo_min is not False:
224 -----angulo_medio = angulo_min[3]
225 -----elif len(frecuentes) == 1:
226 -----angulo_medio = frecuentes[0][0]
227 -----else:
228 -----for a in frecuentes:
229 -----suma += a[0] * a[1]
230 -----if a[0] == 0.0:
231 -----repetidos += int(math.sqrt(a[1])) + 1
232 -----else:
233 -----repetidos += a[1]
234 -----angulo_medio = suma / repetidos
235 -----if angulo_medio == 0:
236 -----if procesado:
237 -----return imagen
238 -----else:
239 -----return False
240 ---except cv2.error:
241 ---para_rotacion = cv2.bitwise_not(para_rotacion)
242 ---angulo_min = self.corregir_angulo_new(imagen=para_rotacion,
    ↪ minimo=minimo, maximo=maximo,
243 -----alto_minimo=alto_minimo, linea_minima=linea_minima,
244 -----linea_maxima=linea_maxima, angulo=angulo, borde=borde,
245 -----escalado=escalado, metodo=TipoCorreccionAngulo.AREA_MINIMA,
246 -----procesado=False)
247 ---if angulo_min is not False:
248 ---angulo_medio = angulo_min[3]
249 ---ag = -angulo_medio
250 ---elif metodo == self.tipo_corrreccion_angulo.AREA_MINIMA:
251 ---blank_1 = para_rotacion
252 ---try:

```

```

253 ----cc = np.column_stack(np.where(blank_1 > 0))
254 ----ag = cv2.minAreaRect(cc)[-1]
255 ---except TypeError:
256 ----if procesado:
257 -----return imagen
258 ----else:
259 -----return False
260 --if abs(ag) > 45:
261 ---nag = ag + 90
262 --else:
263 ---nag = ag
264 --# si el angulo necesita un ajuste se realiza, caso contrario no)
265 --if minimo <= abs(nag) < maximo:
266 ---nag = -nag
267 ---ct = (ancho // 2, alto // 2)
268 ---m = cv2.getRotationMatrix2D(ct, nag, 1.0)
269 ---seno = np.abs(m[0, 0])
270 ---cose = np.abs(m[0, 1])
271 ---nuevo_alto = int((alto * seno) + (ancho * cose))
272 ---nuevo_ancho = int((alto * cose) + (ancho * seno))
273 ---m[0, 2] += (nuevo_ancho / 2) - ct[0]
274 ---m[1, 2] += (nuevo_alto / 2) - ct[1]
275 ---imagen = cv2.warpAffine(imagen, m, (nuevo_ancho, nuevo_alto),
    ↪ flags=cv2.INTER_CUBIC,
276 -----borderMode=cv2.BORDER_CONSTANT, borderValue=255)
277 ---if procesado:
278 ----return imagen
279 ---else:
280 ----return [m, nuevo_ancho, nuevo_alto, -ag]
281 --if procesado:
282 ---return imagen
283 --else:
284 ----return False
285
286
287 -def limpieza_new(self, imagen, proporcion=0.095,
    ↪ tipo=TipoLimpieza.PROCESADO):
288 --"""
289 --Realiza la limpieza de una imagen binaria
290
291 --:param imagen: La imagen a procesar
292 --:param proporcion:
293 --:param tipo:
294 --:return:
295 --"""
296 --alto, ancho = map(int, imagen.shape)
297 --base = cv2.bitwise_not(imagen)
298 --base = cv2.morphologyEx(base, cv2.MORPH_CLOSE, np.ones((150, 25),
    ↪ np.uint8))
299 --base = cv2.morphologyEx(base, cv2.MORPH_CLOSE, np.ones((250, 25),
    ↪ np.uint8))
300 --base = cv2.erode(base, np.ones((10, 10), np.uint8))
301 --base = cv2.medianBlur(base, 15)
302 --base = cv2.bitwise_not(base)
303 --ff = cv2.bitwise_or(base, imagen)
304 --punto_y_1 = 0
305 --punto_y_2 = alto

```

```

306 --punto_x_1 = 0
307 --punto_x_2 = ancho
308 --for i in range(0, alto):
309 ---if (base[i:i + 1, 0:ancho] < 255).sum() == 0:
310 ----punto_y_1 = i
311 ----else:
312 ----break
313
314 --for i in range(alto - 1, 0, -1):
315 ---if (base[i:i + 1, 0:ancho] < 255).sum() == 0:
316 ----punto_y_2 = i
317 ----else:
318 ----break
319
320 --for i in range(0, ancho):
321 ---if (base[0:alto, i:i + 1] < 255).sum() == 0:
322 ----punto_x_1 = i
323 ----else:
324 ----break
325
326 --for i in range(ancho - 1, 0, -1):
327 ---if (base[0:alto, i:i + 1] < 255).sum() == 0:
328 ----punto_x_2 = i
329 ----else:
330 ----break
331
332 --m_alto = punto_y_2 - punto_y_1
333 --m_ancho = punto_x_2 - punto_x_1
334
335 --array_alto = []
336 --for i in range(0, alto):
337 ---ny = (base[i:i + 1, punto_x_1 + int(m_ancho * proporcion):punto_x_2 -
↪ int(m_ancho * proporcion)]
338 ---- < 255).sum()
339 ---array_alto.append(ny)
340
341 --array_ancho = []
342 --for i in range(0, ancho):
343 ---nx = (base[punto_y_1 + int(m_alto * (proporcion / 2)):punto_y_2 -
↪ int(m_alto * (proporcion / 2)), i:i + 1]
344 ---- < 255).sum()
345 ---array_ancho.append(nx)
346
347 --punto_y_1_n = 0
348 --punto_y_2_n = alto
349 --punto_x_1_n = 0
350 --punto_x_2_n = ancho
351
352 --for i in range(punto_y_1 + int(m_alto * proporcion), 0, -1):
353 ---if array_alto[i] <= 1:
354 ----if tipo == TipoLimpieza.POSICIONES:
355 ----punto_y_1_n = i - alto // 100
356 ----else:
357 ----punto_y_1_n = i - alto // 100
358 ----if punto_y_1_n < 0:
359 ----punto_y_1_n = 0
360 ----break

```

```

361 --for i in range(punto_y_2 - int(m_alto * proporcion), alto):
362 ---if array_alto[i] <= 1:
363 ----if tipo == TipoLimpieza.POSICIONES:
364 -----punto_y_2_n = i - alto // 100
365 -----else:
366 -----punto_y_2_n = i - alto // 100
367 ----if punto_y_2_n > alto:
368 -----punto_y_2_n = alto
369 ----break
370 --for i in range(punto_x_1 + int(m_ancho * proporcion), 0, -1):
371 ---if array_ancho[i] <= 15:
372 ----punto_x_1_n = i - array_ancho[i] - 5
373 ----if punto_x_1_n < 0:
374 -----punto_x_1_n = 0
375 ----break
376 --for i in range(punto_x_2 - int(m_ancho * proporcion), ancho):
377 ---if array_ancho[i] <= 15:
378 ----punto_x_2_n = i + array_ancho[i] + 5
379 ----if punto_x_2_n > ancho:
380 -----punto_x_2_n = ancho
381 ----break
382 --if tipo == TipoLimpieza.PROCESADO:
383 ---salida = imagen[punto_y_1:punto_y_2, punto_x_1_n:punto_x_2_n]
384 --else:
385 ---salida = [[punto_y_1_n, punto_y_2_n, punto_x_1_n, punto_x_2_n],
386 ----- [punto_y_1, punto_y_2, punto_x_1, punto_x_2]]
387 --return salida
388
389 -def recortar(self, imagen, modo=TipoCorte.PROCESADO, blur=15,
↪ correccion=0):
390 --alto, ancho = map(int, imagen.shape)
391 --try:
392 ---acrop = np.where(cv2.bitwise_not(cv2.medianBlur(imagen, blur)) > 0)
393 ---acrop = np.column_stack(acrop)
394 --except TypeError:
395 ---if modo == TipoCorte.PROCESADO:
396 -----return imagen
397 ---else:
398 -----return [0, alto, 0, ancho]
399 --bordes = cv2.boxPoints(cv2.minAreaRect(acrop))
400 --bordes = np.intp(bordes)
401 --x0 = bordes[2][1]
402 --y0 = bordes[2][0]
403 --x1 = bordes[3][1]
404 --y1 = bordes[3][0]
405 --x2 = bordes[1][1]
406 --y2 = bordes[1][0]
407 --x3 = bordes[0][1]
408 --y3 = bordes[0][0]
409 --xf0 = x0 if (x0 <= x1) else x1
410 --yf0 = y0 if (y0 <= y2) else y2
411 --xf1 = x2 if (x2 >= x3) else x3
412 --yf1 = y1 if (y1 >= y3) else y3
413
414 --xf0 -= ancho // 200 - int(blur / 2)
415 --xf1 += ancho // 200 + int(blur / 2)
416 --yf0 -= alto // 200 - int(blur / 2)

```

```

417 --yf1 += alto // 200 + int(blur / 2)
418
419 --if xf0 < 0:
420 ---xf0 = 0
421 --if xf1 > ancho:
422 ---xf1 = ancho
423 --if yf0 < 0:
424 ---yf0 = 0
425 --if yf1 > alto:
426 ---yf1 = alto
427 --if modo == TipoCorte.PROCESADO:
428 ---imagen = imagen[yf0:yf1, xf0:xf1]
429 ---return imagen
430 --elif modo == TipoCorte.POSICIONES:
431 ---return [yf0, yf1, xf0, xf1]
432
433
434 -def segmentar_test(self, imagen, modo=TipoSegmentacion.POSICIONES):
435 --s_fragments = []
436 --segmentos_1 = self.__seg_horizontal(imagen, 0, 0.008, 0.5)
437 --alto_base, ancho_base = map(int, imagen.shape)
438 --regio2 = (imagen == 0).sum()
439 --calcul2 = regio2 / (alto_base * ancho_base)
440 --point = 0
441 --n1 = -1
442 --x = 1
443 --totali = 0
444 --para_seg = imagen.copy()
445 --# Recorrido primera segmentacion
446 --for segmento_1 in segmentos_1:
447 ---n1 += 1
448 ---segmento_a = para_seg[point:segmento_1[0], 0:ancho_base]
449 ---# Salto si segmento no valido
450 ---if point == segmento_1[0] or (segmento_a < 255).sum() <= 625:
451 ----point = segmento_1[0]
452 ----continue
453 ---segmentos_2 = self.__seg_vertical(segmento_a, alto_base, ancho_base,
↳ n1, 0, 0.045, 0.2, 0.0035)
454 ---point_2 = 0
455 ---nh = 0
456 ---alto_seg, ancho_seg = map(int, segmento_a.shape)
457 ---peso = calcul2 + ((alto_seg * ancho_seg) / (alto_base * ancho_base))
458 ---# Si la segmentacion es posible y es valido
459 ---if segmentos_2 and peso >= 0.055:
460 ----if (segmento_1[0] - point) < (alto_base * 0.05):
461 -----if (segmento_a < 255).sum() <= 625:
462 -----continue
463 -----totali += 1
464 -----else:
465 -----# Segunda segmentacion
466 -----for segmento_2 in segmentos_2:
467 -----# Salto si segmento no es valido
468 -----if point_2 == segmento_2[0] or (segmento_2[0] - point_2) < 6:
469 -----point_2 = segmento_2[0]
470 -----nh += 1
471 -----continue
472 -----segmento_b = para_seg[point:segmento_1[0], point_2:segmento_2[0]]

```

```

473 -----nh += 1
474 -----if (segmento_1[0] - point) > ((segmento_2[0] - point_2) * 0.1):
475 -----# 0.05, 0.008, 0.3
476 -----segmentos_3 = self.__seg_horizontal(segmento_b, 0, 0.005, 0.2)
477 -----if segmentos_3:
478 -----point_3 = 0
479 -----for segmento_3 in segmentos_3:
480 -----if point_3 == segmento_3[0]:
481 -----continue
482 -----segmento_c = para_seg[point + point_3:point + segmento_3[0],
↪ point_2:segmento_2[0]]
483 -----if (segmento_c < 255).sum() <= 625:
484 -----point_3 = segmento_3[0]
485 -----continue
486 -----if modo == TipoSegmentacion.GENERAR:
487 -----s_fragments.append(segmento_c)
488 -----else:
489 -----s_fragments.append([point + point_3, point + segmento_3[0],
↪ point_2,
490 -----segmento_2[0]])
491 -----totali += 1
492 -----point_3 = segmento_3[0]
493 -----else:
494 -----if (segmento_b < 255).sum() <= 625:
495 -----continue
496 -----if modo == TipoSegmentacion.GENERAR:
497 -----s_fragments.append(segmento_b)
498 -----else:
499 -----s_fragments.append([point, segmento_1[0], point_2,
↪ segmento_2[0]])
500 -----totali += 1
501 -----else:
502 -----totali += 1
503 -----if (segmento_b < 255).sum() <= 625:
504 -----point_2 = segmento_2[0]
505 -----nh += 1
506 -----continue
507 -----if modo == TipoSegmentacion.GENERAR:
508 -----s_fragments.append(segmento_b)
509 -----else:
510 -----s_fragments.append([point, segmento_1[0], point_2,
↪ segmento_2[0]])
511 -----point_2 = segmento_2[0]
512 -----nh += 1
513 -----else:
514 -----# Si no hay segmentos se trata de segmentar nuevamente de manera
↪ horizontal solo si el alto es > 25%
515 -----if modo == TipoSegmentacion.GENERAR:
516 -----s_fragments.append(segmento_a)
517 -----else:
518 -----s_fragments.append([point, segmento_1[0], 0, ancho_base])
519 -----totali += 1
520 -----point = segmento_1[0]
521 -----x += 1
522 -----return s_fragments

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: procesador_tesseract.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 import os
6 import cv2
7 import sys
8 import tempfile
9 import glob
10 import subprocess
11 import pytesseract
12 from PIL import Image
13 from gtts import gTTS
14 from regex import RegexTexto
15 from time import time, sleep
16 from threading import Thread
17 from termcolor import cprint
18 # from procesador_imagen_bkp import ProcesarImagen
19 from procesador_imagen import ProcesarImagen
20 from hunspell import Whunspell as hunspell
21
22
23 def print_error(x): cprint(x, 'red', attrs=['bold', 'underline'], end='
↳ \n')
24 def print_info(x): cprint(x, 'yellow', attrs=['bold'], end=' \n')
25 def print_log(x): cprint(x, 'cyan', attrs=['bold'], end=' \n')
26 def print_log_a(x): cprint(x, 'magenta', attrs=['underline'], end=' \n')
27 def print_info_nb(x): cprint(x, 'yellow', attrs=['bold'], end='')
28 def print_info_rv(x): cprint(x, 'magenta', attrs=['bold'], end=' \n')
29
30
31 class ProcesarConTesseract:
32     """
33     -Clase encargada de interactuar con tesseract-ocr y controlar los
↳ procesos de conversion y reproduccion de audio
34     """
35     -def __init__(self):
36     """
37     --Inicializa variables y define los valores iniciales para Tesseract-ocr,
↳ Hunspell y el Speech-to-Text
38     """
39     --self.__p = ProcesarImagen()
40     --self.__r = RegexTexto()
41     --self.__d = hunspell.HunSpell('es_ES', 'es_ES')
42     --self.__acumulador_imagenes = []
43     --self.__acumulador_texto = []
44     --self.__audios = []
45
46     --self.__cache_audios = []
47
48     --self.__pre_acumulado = ''
49
50     --self.__n_hilos_tesseract = 0
51     --self.__final_tesseract = False
52     --self.__final_texto = False
53     --self.__final_reproductor = False

```

```

54
55 --self.__f_t = False
56 --self.timer = time()
57 --self.fin_tex = False
58
59 --self.__proceso_terminado = False
60 --self.__tts = 'festival'
61 --self.texto_acumulado = ''
62 --self.texto_acumulado_regex = ''
63 --self.texto_acumulado_no_corregido = ''
64 --self.stop = False
65 --os.environ['OMP_THREAD_LIMIT'] = '1'
66
67 -def __controlador_tesseract(self):
68     """
69     --Se encarga de controlar los procesos individuales de Tesseract-ocr para
    ↪ la conversion de imagenes a texto
70     """
71     --print_log('[Controlador tesseract] Iniciado')
72     --while True:
73         ---sleep(0.05)
74         ---if len(self.__acumulador_imagenes) > 0:
75             ----break
76         --while not self.__final_tesseract:
77             ---if self.__n_hilos_tesseract < 2 and len(self.__acumulador_imagenes) >
    ↪ 0:
78                 ----self.__n_hilos_tesseract += 1
79                 ----imagen = self.__acumulador_imagenes.pop(0)
80                 ----nuevo_hilo = Thread(target=self.__hilo_tesseract, args=(imagen[0],
    ↪ imagen[1],))
81                 ----nuevo_hilo.start()
82                 ---if len(self.__acumulador_imagenes) == 0 and self.__n_hilos_tesseract
    ↪ == 0:
83                     ----print_log('[Controlador tesseract] Finalizado')
84                     ----self.__final_tesseract = True
85                     ----break
86
87 -def __controlador_texto(self):
88     """
89     --Se encarga de controlar el texto acumulado en el proceso de conversion
    ↪ y su envio a conversion para texto a voz
90     """
91     --print_log('[Controlador texto] Iniciado')
92     --puntero = 0
93     --puntero_temp = 0
94     --valido = True
95     --m = 0
96     --while valido:
97         ---sleep(0.01)
98         ---m += 1
99         ---acumulado = ''
100     ---for i in range(puntero, len(self.__acumulador_texto)):
101         ----if self.__acumulador_texto[i] != '':
102             ----puntero_temp = i
103             ----acumulado += str(self.__acumulador_texto[i]) + ' '
104         ----else:
105             ----break

```



```

106 ---a_procesar = ''
107 ---if len(str(self.__pre_acumulado)) + len(str(accumulado)) > 700:
108 ----if puntero < puntero_temp:
109 -----puntero = puntero_temp + 1
110 -----a_procesar = str(self.__pre_acumulado) + str(accumulado)
111 -----bloques = []
112 -----while len(a_procesar) > 700:
113 -----puntero_espacio_1 = a_procesar.find(' ', 650)
114 -----puntero_espacio_2 = a_procesar.find(' ', puntero_espacio_1 + 1)
115 -----if puntero_espacio_2 < puntero_espacio_1:
116 -----puntero_espacio_2 = puntero_espacio_1
117 -----bloque = a_procesar[0:puntero_espacio_1]
118 -----if bloque[:2] != '- ' and bloque[:3] != '- ':
119 -----bloques.append(bloque)
120 -----a_procesar = a_procesar[puntero_espacio_1 + 1:]
121 -----else:
122 -----bloques.append(a_procesar[0:puntero_espacio_2])
123 -----a_procesar = a_procesar[puntero_espacio_2 + 1:]
124 ----self.__pre_acumulado = a_procesar
125 ----for b in bloques:
126 ----self.__convertir_a_voz(b)
127 ----sleep(0.1)
128 ---else:
129 ----maximo = len(self.__acumulador_texto) - 1
130 ----if puntero_temp >= maximo:
131 -----if len(self.__pre_acumulado) + len(str(accumulado)) > 0:
132 -----self.__convertir_a_voz(self.__pre_acumulado + ' ' + str(accumulado))
133 -----valido = False
134 --self.__final_texto = True
135 --self.fin_tex = True
136 --print_log('[Controlador texto] Finalizado')
137
138 -def __convertir_a_voz(self, texto):
139 ---"""
140 --Convierte el segmento de texto proporcionado a voz y almacena la
141   ↪ direccion del archivo generado a la lista
142 --de ficheros de audio
143 --:param texto: El texto que se desea convertir a voz
144 --:return: Falso si el largo del texto obtenido es cero, Verdadero en
145   ↪ caso contrario
146 ---"""
146 --texto = str(texto).strip()
147 --if len(str(texto)) == 0:
148 ---return False
149 --archivo = ''
150 --self.texto_acumulado_no_corregido += ' ' + str(texto)
151 --texto = self.__r.regex(input_text=texto, hard=True)
152 --self.texto_acumulado_regex += ' ' + str(texto)
153 --array_texto = texto.split(' ')
154 --self.__d.load_text(texto)
155 --salida_texto = ''
156 --for i in range(0, len(array_texto)):
157 ---if len(str(array_texto[i])) > 5:
158 ----salida_texto += ' ' + str(array_texto[i])
159 ----continue
160 ---if self.__d.spell_new(array_texto[i]):

```

```

161 ----salida_texto += ' ' + array_texto[i]
162 ----else:
163 ----sugerencia = self.__d.suggest_new(array_texto[i])
164 ----if sugerencia is None:
165 -----salida_texto += ' ' + array_texto[i]
166 ----else:
167 -----salida_texto += ' ' + sugerencia[0]
168 --texto = salida_texto
169 --self.texto_acumulado += ' ' + str(texto)
170 --self.texto_acumulado = self.texto_acumulado.strip()
171 --if self.__f_t is False:
172 ---self.__f_t = True
173 ---self.timer = time()
174 --if self.__tts == 'festival':
175 ---archivo_id = str(time())[1:14].replace('.', '')
176 ---archivo = 'audio/festival/temp_' + archivo_id
177 ---texto = str(texto)
178 ---texto = texto.replace('"', '').replace('\n', ' ').strip()
179 ---texto = texto.encode('latin-1', 'ignore')
180 ---with open(archivo + '.txt', 'wb') as file_audio:
181 ----file_audio.write(texto)
182 ----file_audio.close()
183 ---os.system('text2wave -o {}.wav {}.txt -eval "{}".format(archivo,
↳ archivo,
184 ----- ' (voice_JuntaDeAndalucia_es_sf_diphone)'))
185 ---archivo += '.wav'
186 --if self.__tts == 'gtts':
187 ---archivo = 'audio/gtts/temp_' + str(time())[1:14].replace('.', '') +
↳ '.mp3'
188 ---salida_gtts = gTTS(text=texto, lang='es')
189 ---salida_gtts.save(archivo)
190 --self.__audios.append(archivo)
191 --return True
192
193 -def __hilo_tesseract(self, imagen, id):
194 -"""
195 --Se encarga de convertir un segmento de imagen individual a texto y
↳ almacenar el resultado en el acumulador
196 --de texto
197
198 --:param imagen: El segmento de imagen a procesar
199 --:param id: El identificador numerico utilizado para el almacenamiento
↳ del resultado de texto
200 -"""
201 --t1 = time()
202 --os.environ['OPM_THREAD_LIMIT'] = '1'
203 --texto = pytesseract.image_to_string(image=imagen, lang='spa',
↳ config='--psm 3', nice=10)
204 --texto = ' '.join(texto.split())
205 --texto = self.__r.regex(texto, True)
206 --if texto == '':
207 ---texto = ' '
208 --self.__acumulador_texto[id] = texto
209 --self.__n_hilos_tesseract -= 1
210
211 -def __controlador_reproductor(self):
212 -"""

```

```

213 --Se encarga de controlar la reproduccion de los ficheros de audio
    ↳ almacenados en la lista de ficheros de audio
214 --"""
215 --print_log(' [Controlador reproductor] Iniciado')
216 --valido = True
217 --vlc_args = '-A=alsa --play-and-exit --rate=1.1 --gain=0.9'
218 --while valido:
219 ---if self.stop is True:
220 ----valido = False
221 ----self.__final_reproductor = True
222 ----break
223 ---if len(self.__audios) > 0:
224 ----sleep(0.01)
225 ----archivo = self.__audios.pop(0)
226 ----self.__cache_audios.append(archivo)
227 ----print_info(' [Reproduciendo] ' + str(archivo))
228 ----if self.__tts == 'festival':
229 -----os.system('cvlc {} {} 2> /dev/null'.format(vlc_args, archivo))
230 ----if self.__tts == 'gtts':
231 -----os.system('cvlc {} {} 2> /dev/null'.format(vlc_args, archivo))
232 ----print_log(' [Fin] ' + str(archivo))
233 ----else:
234 ----if self.__final_texto:
235 -----print_log(' [Controlador reproductor] Finalizado')
236 -----valido = False
237 -----self.__final_reproductor = True
238 ----sleep(0.01)
239
240 -def procesar(self, imagen, motor_tts='gtts'):
241 --"""
242 --Se encarga de procesar y segmentar la imagen proporcionada para su
    ↳ posterior conversion a texto y despues a voz
243 --:param imagen: En contenido binario de la imagen a procesar
244 --:param motor_tts: La herramienta de conversion de texto a voz
    ↳ seleccionada, por defecto 'gtts'
245 --:return:
246 --"""
247 --# Validacion
248 --archivos = glob.glob('audio/{}/*.*'.format(motor_tts))
249 --for i in archivos:
250 ---os.remove(i)
251 --hilo_tesseract = Thread(target=self.__controlador_tesseract)
252 --hilo_texto = Thread(target=self.__controlador_texto)
253 --hilo_reproductor = Thread(target=self.__controlador_reproductor)
254 --hilo_tesseract.start()
255 --hilo_texto.start()
256 --hilo_reproductor.start()
257 --if isinstance(imagen, str):
258 ---imagen = cv2.imread(imagen)
259 --self.__p.imdir = 'test'
260 --self.__tts = motor_tts
261 --gris = self.__p.a_gris(imagen=imagen)
262 --binaria = self.__p.binarizado(imagen=gris,
    ↳ tipo=self.__p.tipo_binarizacion.GAUSS, v1=255, v2=11, bil=1)
263 --demarcado = self.__p.demarcar(imagen=binaria, radio=2, iteraciones=1)
264 --corregida = self.__p.corregir_angulo_new(imagen=demarcado, minimo=0.01,
    ↳ maximo=200, borde=1, escalado=1,

```

```

265 ----- metodo=self.__p.tipo_corrrecion_angulo.LINEAS)
266 --limpia = self.__p.limpieza_new(imagen=corregida, proporcion=0.3)
267 --recortada = self.__p.recortar(imagen=limpia,
    ↪ modo=self.__p.tipo_corte.PROCESADO, blur=19)
268 --cv2.imwrite('debug/aaa.jpg', recortada)
269 --try:
270 ---corregida_2 = self.__p.corregir_angulo_new(imagen=recortada,
    ↪ minimo=0.5, borde=1, escalado=1,
271 ----- metodo=self.__p.tipo_corrrecion_angulo.LINEAS)
272 --except (AttributeError):
273 ---# Error de atributo, probablemente la imagen no contiene informacion
274 ---return None
275 --escalada = self.__p.escalar(imagen=corregida_2, proporcion=2)
276 --posiciones = self.__p.limpieza_new(imagen=escalada, proporcion=0.1,
    ↪ tipo=self.__p.tipo_limpieza.POSICIONES)
277 --nx = 0
278 --minima = None
279 --if posiciones[0][0] > 2:
280 ---minima = escalada[0:posiciones[0][0],
    ↪ posiciones[1][2]:posiciones[1][3]]
281 ---limpia_2 = escalada[posiciones[0][0]:posiciones[1][1],
    ↪ posiciones[1][2]:posiciones[1][3]]
282 ---nx = 1
283 --else:
284 ---limpia_2 = escalada
285 --segmentado = self.__p.segmentar_test(imagen=limpia_2,
    ↪ modo=self.__p.tipo_segmentacion.POSICIONES)
286 --self.__acumulador_texto = [''] * (len(segmentado) + (1 + nx))
287 --self.__acumulador_texto[len(self.__acumulador_texto) - 1] = ','
288 --lista_imagenes = []
289 --n = 0
290 --if minima is not None:
291 ---minima = self.__p.escalar(minima, 0.5)
292 ---lista_imagenes.append([minima, 0])
293 ---n = 1
294 --for i in segmentado:
295 ---corte = limpia_2[i[0]:i[1], i[2]:i[3]]
296 ---recorte = self.__p.recortar(imagen=corte,
    ↪ modo=self.__p.tipo_corte.PROCESADO, blur=9)
297 ---corregido = self.__p.corregir_angulo_new(imagen=recorte, minimo=0.05,
    ↪ maximo=10, borde=0, escalado=1,
298 ----- metodo=self.__p.tipo_corrrecion_angulo.AREA_MINIMA)
299 ---corregido = self.__p.escalar(corregido, 0.5)
300 ---lista_imagenes.append([corregido, n])
301 ---n += 1
302 --self.__acumulador_imagenes = lista_imagenes
303 --while not self.__proceso_terminado:
304 ---sleep(0.01)
305 ---if self.__final_tesseract and self.__final_texto and
    ↪ self.__final_reproductor:
306 ----self.__proceso_terminado = True
307 --print_info('[Fin del proceso]')
308 --self.__init__()
309 --return self.__cache_audios
310
311 --def repetir_cache(self, cache=None):
312 --if cache is None:

```

```
313 ---self.__audios = self.__cache_audios
314 --else:
315 ---self.__audios = cache
316 --self.__cache_audios = []
317 --self.__final_tesseract = True
318 --self.__final_texto = False
319 --self.__final_reproductor = False
320 --self.__proceso_terminado = False
321 --hilo_reproductor = Thread(target=self.__controlador_reproductor)
322 --hilo_reproductor.start()
323 --sleep(5)
324 --self.__final_texto = True
325 --while not self.__proceso_terminado:
326 ---sleep(0.01)
327 ---if self.__final_tesseract and self.__final_texto and
    ↪ self.__final_reproductor:
328 ----self.__proceso_terminado = True
329 --print_info('[Fin del proceso]')
330 --return self.__cache_audios
```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: record.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 import os
6 import io
7 import sys
8 import wave
9 import math
10 import numpy
11 import pyaudio
12 import threading
13 from time import sleep
14 from termcolor import cprint
15 from pynput import keyboard
16 if sys.platform != 'win32':
17     -from ctypes import *
18     -from contextlib import contextmanager
19
20 -ERROR_HANDLER_FUNC = CFUNCTYPE(None, c_char_p, c_int, c_char_p, c_int,
    ↪ c_char_p)
21
22 -def py_error_handler(filename, line, function, err, fmt):
23 --del filename
24 --del line
25 --del function
26 --del err
27 --del fmt
28
29 -c_error_handler = ERROR_HANDLER_FUNC(py_error_handler)
30
31
32 -@contextmanager
33 -def noalsaerroutput():
34 --asound = cdll.LoadLibrary('libasound.so')
35 --asound.snd_lib_error_set_handler(c_error_handler)
36 --yield
37 --asound.snd_lib_error_set_handler(None)
38 assert numpy
39
40 def print_error(x): cprint(x, 'red', attrs=['bold', 'underline'], end='
    ↪ \n')
41 def print_info(x): cprint(x, 'yellow', attrs=['bold'], end=' \n')
42 def print_log(x): cprint(x, 'cyan', attrs=['bold'], end=' \n')
43 def print_log_a(x): cprint(x, 'magenta', attrs=['underline'], end=' \n')
44 def print_info_nb(x): cprint(x, 'yellow', attrs=['bold'], end='')
45 def print_info_rv(x): cprint(x, 'magenta', attrs=['bold'], end=' \n')
46
47 # Modulo de grabacion de audio enlazado desde base.py
48
49 class Record:
50     -"""
51     -Clase encargada de la grabacion de audio
52     -"""
53     -def __init__(self, filename='voz'):
54     -"""

```

```

55 --Establece los parametros de funcionamiento e inicializa variables
56 """
57 --self.__id_record = 2
58 --self.__framerate = 44100
59 --self.__buffer = 2048
60 --self.__filename = filename
61 --self.__filtro = 'noise'
62 --self.__r_frames = -1
63 --self.__filtrar = False
64 --self.__resamplear = True
65 --self.__proporcion = 0.21
66 --with noalsaerroutput():
67 ----self.record = pyaudio.PyAudio()
68 --self.__stream = self.record.open(format=pyaudio.paInt16, channels=1,
    ↪ rate=self.__framerate, input=True,
69 ----- frames_per_buffer=self.__buffer,
    ↪ input_device_index=self.__id_record)
70 --self.__frames = io.BytesIO()
71 --self.__f_count = 0
72 --self.__n_count = 0
73 --self.__cerrado = False
74 --self.finalizar = False
75
76
77 -def generar_filtro(self, filename):
78 --temp = self.__filename
79 --self.__filename = 'para_filtro'
80 --self.grabar(resamplear=True, no_keyboard=True, n_frames=40)
81 --os.system('sox records/para_filtro.wav -n noiseprof
    ↪ records/{}.prof'.format(str(filename)))
82 --print_info('[dev][info][rec] Filtro generado: {}'.format(str(filename)))
83 --self.__filename = temp
84
85 -def suprimir_ruido(self, binary_data, data_format, filter, proportion,
    ↪ resamplear=-1):
86 --formatos = ['wav', 'flac', 'aiff']
87 --arg = ''
88 --if resamplear != -1:
89 ---arg = '-r {}'.format(str(resamplear))
90 --if data_format not in formatos:
91 ---return binary_data
92 --else:
93 ---with open('records/temp_voz.{}'.format(str(data_format)), 'wb') as
    ↪ archivo_temporal:
94 ----archivo_temporal.write(binary_data)
95 ----archivo_temporal.close()
96 --print_log('sox records/temp_voz.{}{} records/out.{} noisered
    ↪ records/{}.prof {} '.format(
97 ----str(data_format), str(arg), str(data_format), str(filter),
    ↪ str(proportion)
98 ----))
99 ---os.system('sox records/temp_voz.{}{} records/out.{} noisered
    ↪ records/{}.prof {} '.format(str(data_format),
100 ----- str(arg),
101 ----- str(data_format),
102 ----- str(filter),
103 ----- str(proportion)

```

```

104 ----)
105 ---with open('records/out.{}'.format(str(data_format)), 'rb') as
    ↪ archivo_salida:
106 ----data_out = archivo_salida.read()
107 ----archivo_salida.close()
108 ---return data_out
109
110 -def grabar(self, resamplear=True, filtrar=False, filtro='noise',
    ↪ proporcion=0.21, no_keyboard=False, n_frames=-1):
111 ---"""
112 --Inicia la grabacion de audio y la escucha de teclado mediante hilos
113 ---"""
114 --if self.__cerrado:
115 ---self.__init__(self.__filename)
116 --self.__filtro = filtro
117 --self.__proporcion = proporcion
118 --self.__filtrar = filtrar
119 --self.__r_frames = n_frames
120 --self.__resamplear = resamplear
121 --self.__frames = io.BytesIO()
122 --hilol = threading.Thread(target=self.__grabando)
123 --hilol.daemon = True
124 --hilol.start()
125 --if no_keyboard is False:
126 ---hilo2 = threading.Thread(target=self.escucha_teclado)
127 ---hilo2.daemon = True
128 ---hilo2.start()
129 --hilol.join()
130
131 -def escucha_teclado(self):
132 ---"""
133 --Se encarga de interceptar las teclas precionadas y enviar el valor
    ↪ obtenido a la funcion control_tecla
134 ---"""
135 --with keyboard.Listener(on_press=self.control_tecla) as escucha:
136 ---escucha.join()
137
138 -def control_tecla(self, tecla):
139 ---"""
140 --Se encarga de finalizar la grabacion si la tecla ingresada coincide con
    ↪ "espacio" o "intro"
141
142 --:param tecla: El parametro de tecla recibido desde escucha_teclado
143 --:return:
144 ---"""
145 --if self.finalizar:
146 ---return False
147 --if tecla == keyboard.Key.space:
148 ---self.finalizar = True
149 ---return False
150
151 -def __grabando(self):
152 ---"""
153 --Se encarga de escribir en disco en archivo de audio obtenido desde el
    ↪ microfono
154 ---"""
155 --max_frames = int(math.ceil((self.__framerate / self.__buffer) * 10))

```



```

156 --print_info('[dev][info][rec] Grabacion iniciada')
157 --print_info('[dev][info][rec] Frames maximos {}'.format(str(max_frames)))
158 --try:
159 ---while not self.finalizar:
160 ----buffer = self.__stream.read(self.__buffer,
    ↳ exception_on_overflow=False)
161 ----if self.__r_frames > -1 and self.__f_count < 3:
162 -----self.__f_count += 1
163 -----continue
164 ----if len(buffer) > 0:
165 -----self.__frames.write(buffer)
166 ----if self.__f_count < max_frames:
167 -----self.__f_count += 1
168 -----else:
169 -----self.finalizar = True
170 ----if -1 < self.__r_frames <= self.__f_count:
171 -----self.finalizar = True
172 ----elif self.finalizar:
173 ----break
174 ----else:
175 ----break
176 --except (pyaudio.paInternalError, pyaudio.paUnanticipatedHostError):
177 -- print_error('[dev][error] Error record')
178 --try:
179 ---all_frames = self.__frames.getvalue()
180 ---wf = wave.open('records/{}.wav'.format(self.__filename), 'wb')
181 ---wf.setnchannels(1)
182 ---wf.setsampwidth(pyaudio.PyAudio().get_sample_size(pyaudio.paInt16))
183 ---wf.setframerate(self.__framerate)
184 ---wf.writeframes(all_frames)
185 ---wf.close()
186 ---resample = ''
187 ---agregado = ''
188 ---if self.__resamplear:
189 ---resample = '-r 20000 '
190 ---if self.__filtrar:
191 ---agregado = ' noisiered records/{}.prof {}'.format(
192 ----str(self.__filtro), str(self.__proporcion))
193 ---if resample != '' or agregado != '':
194 ---os.system('sox records/{}.wav {}-c 1 records/{}_out.wav{}'.format(
195 ----str(self.__filename),
196 ----str(resample),
197 ----str(self.__filename),
198 ----str(agregado)
199 ----))
200 --print_info('[dev][info][rec] Fin grabacion')
201 --except wave.Error:
202 ---pass
203
204 -def cerrar(self):
205 --if self.__cerrado is False:
206 ---self.__stream.stop_stream()
207 ---self.__stream.close()
208 ---self.record.terminate()
209 ---self.__cerrado = True

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: regex.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 import re
6
7
8 class RegexTexto:
9     """
10 -Clase encargada de corregir errores y cortes de palabras debido a los
    ↪ guiones
11 """
12 -def __init__(self):
13     """
14 --Carga los diccionarios a utilizar
15 """
16 --self.__dicc_soft = [['(\S-\s\S)', '- ', ''], [(\S\s\S)', ' ', ''],
    ↪ ['(\s{3})', ' ', ' ']]
17 --self.__dicc_hard = [['(\S;\S)', ';', 'i'], [(\S;\s)', ';', 'i'],
    ↪ ['(\S!\S)', '!', 'l'],
18 -----['(\S!\s\S)', '!', 'l'], [(\S-\s\S)', '- ', ''], [(\S\s\S)', ' ',
    ↪ ' '],
19 -----['(\s{3})', ' ', ' '], [(\s{2})', ' ', ' '], [(\.\{3})', '...',
    ↪ ''],
20 -----['(-{3})', '- ', ''], [(\S;\S)', ';', 'i'], [(\S:\S)', ':', 'i'],
    ↪ ['(\s\.\s)+', ' ', ' '],
21 -----['(\s_\s)', ' _ ', ' '], [(\s-\s)', ' - ', ' '], [(\S_\S)', ' _ ',
    ↪ ''],
22 -----['(c°\S)', '°', 'o'], [(p°\S)', '°', 'o'], [(\D0\D)', '0', 'o'],
    ↪ ['(dcs)', 'dcs', 'des'],
23 -----['(\S°\S)', '°', 'o'], [(\S°\S)', '°', 'o'], [(n)', 'n', 'en'],
    ↪ ['(«)', '«', ''],
24 -----['(»)', '»', '']]
25 -def regex(self, input_text, hard=False):
26     """
27 --Realiza un proceso de correccion de texto
28
29 --:param input_text: El texto ingresado para su procesamiento
30 --:param hard: El tipo de procesamiento a realizar
31 --:return: El texto corregido
32 """
33 --diccionario = self.__dicc_soft
34 --if hard:
35 ---diccionario = self.__dicc_hard
36 --for i in diccionario:
37 ---n = 0
38 ---while n < 2:
39 ----results = re.findall(i[0], input_text)
40 ----if results:
41 -----for j in results:
42 -----new_term = j.replace(i[1], i[2])
43 -----input_text = input_text.replace(j, new_term)
44 ----n += 1
45 --input_text = input_text.replace('\n', ' ')
46 --return ' '.join(input_text.split()).strip()

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: run.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 import os
6 import sys
7 import argparse
8 from time import sleep
9 from base import Prototipo
10 from termcolor import colored, cprint
11
12 def print_bold(x): cprint(x, 'green', attrs=['bold'], end=' \n')
13 def espaciado(x): print_bold((((' ' * (47 - int(len(x) / 2))) + x + (' ' *
    ↪ (47 - int(len(x) / 2))))[0:94])
14
15 def print_error(x): cprint(x, 'red', attrs=['bold', 'underline'], end='
    ↪ \n')
16 def print_info(x): cprint(x, 'yellow', attrs=['bold'], end=' \n')
17 def print_log(x): cprint(x, 'cyan', attrs=['bold'], end=' \n')
18 def print_log_a(x): cprint(x, 'magenta', attrs=['underline'], end=' \n')
19 def print_info_nb(x): cprint(x, 'yellow', attrs=['bold'], end='')
20 def print_info_rv(x): cprint(x, 'magenta', attrs=['bold'], end=' \n')
21
22 if __name__ == '__main__':
23     -parser = argparse.ArgumentParser(description='Prototipo')
24     -parser.add_argument('-s', '--speechtotoext', help='Motor de conversion de
    ↪ voz a texto', default='hibrido',
25     -----required=False, choices=['wit', 'google', 'sphinx', 'bing', 'ibm',
    ↪ 'hibrido'])
26     -parser.add_argument('-t', '--texttospeech', help='Motor de conversion de
    ↪ texto a voz', default='gtts',
27     -----required=False, choices=['gtts', 'bing', 'voicerss', 'festival'])
28     -parser.add_argument('-o', '--ocrengine', help='Motor de conversion de
    ↪ imagen a texto', default='ocrspace',
29     -----required=False, choices=['ocrspace', 'tesseract', 'google', 'bing'])
30     -parser.add_argument('-w', '--websearch', help='Motor de busqueda en
    ↪ linea', default='duckduckgo',
31     -----required=False, choices=['google', 'bing', 'duckduckgo'])
32     -parser.add_argument('-i', '--imagefile', help='Imagen a procesar',
    ↪ default='web.jpg', required=False)
33     -parser.add_argument('-p', '--pdf', help='PDF a procesar', default=None,
    ↪ required=False)
34     -parser.add_argument('-m', '--mode', help='Modo de ejecucion',
    ↪ default='mic', required=False, choices=['mic', 'rec'])
35     -args = parser.parse_args()
36     -motor_stt = args.speechtotoext
37     -motor_tts = args.texttospeech
38     -motor_ocr = args.ocrengine
39     -motor_web = args.websearch
40     -img_file = args.imagefile
41     -pdf_file = args.pdf
42     -run_mode = args.mode
43     -pagina_pdf = 1
44     -cache_resultados = None
45     -cache_texto = ''
46     -is_cache_resultados = False

```

```

47 -is_cache_texto = False
48 -cache_audio = 0
49 -save_busqueda = ''
50 -procesar = False
51 -ayuda_navegacion = False
52 -teclado = True
53 -if run_mode == 'mic':
54 --teclado = False
55 -if run_mode == 'rec':
56 --# FILTRO
57 --pass
58 -separador = '=' * 94
59 -print_bold(separador)
60 -espaciado('+++ Configuracion de inicio +++')
61 -print_bold('=' * 94)
62 -print_bold('[dev][info] Modo de ejecucion: {}'.format(run_mode))
63 -print_bold('[dev][info] Motor speech-to-text: {}'.format(motor_stt))
64 -print_bold('[dev][info] Motor text-to-speech: {}'.format(motor_tts))
65 -print_bold('[dev][info] Motor busqueda web: {}'.format(motor_web))
66 -print_bold('[dev][info] Motor OCR: {}'.format(motor_ocr))
67 -print_bold('[dev][info] Img OCR: {}'.format(img_file))
68 -print_bold('=' * 94) + '\n' + ('-' * 94))
69 -espaciado('+++ Modo alternativo: Solo voz - dicte: \'prototipo\' para
↳ iniciar el reconocimiento +++')
70 -print_bold('=' * 94) + '\n' +('=' * 94) + '\n' + ('+' * 94))
71 -espaciado('+++ Comandos disponibles +++')
72 -print_bold('+' * 94)
73 -print_bold('+++ - lectura: lee una pagina fisica----- +++')
74 -print_bold('+++ - buscar <terminos>: Busca los terminos dictados en
↳ internet----- +++')
75 -print_bold('+++ - resultado <numero>: Carga el texto del resultado
↳ escogido de una busqueda----+++')
76 -print_bold('+++ - repetir lista: Repite la lista de busqueda si se ha
↳ realizado alguna---- +++')
77 -print_bold('+++ - repetir texto: Repite el texto de una pagina fisica
↳ leida o una pagina web cargada +++')
78 -print_bold('+++ - navegacion avanzar: Carga la siguiente lista de
↳ resultados de busqueda--- +++')
79 -print_bold('+++ - navegacion regresar: Carga la lista anterior de
↳ resultados de busqueda--- +++')
80 -print_bold('+++ - libro <numero>: Lee la pagina correspondiente del
↳ archivo PDF proporcionado-- +++')
81 -print_bold('+++ - libro avanzar: Lee la pagina sigiente del archivo PDF
↳ proporcionado---- +++')
82 -print_bold('+++ - libro regresar: Lee la pagina anterior del archivo PDF
↳ proporcionado---- +++')
83 -print_bold('+' * 94)
84 -p = Prototipo(arranque=teclado)
85 -# menu: Bienvenido
86 -p.reproducir_audio_menu('bienvenido')
87 -# menu: ayuda de inicio
88 -p.reproducir_audio_menu('introduccion')
89 -# menu: dicte su comando
90 -print_info('[dev][info][main] en espera')
91 -paso = False
92 -mensaje = True
93 -contador = 0

```

```

94 while True:
95     sleep(0.1)
96     if run_mode == 'rec':
97         procesar = p.cola_proceso()
98         if procesar:
99             procesar = False
100            # iniciar reconocimiento
101            p.reproducir_audio_menu('espera')
102            print_info(' [dev][info][main] procesando')
103            comando = p.leer_comando_archivo('records/voz_out.wav', motor_stt)
104            else:
105                comando = [False, False]
106                # procesar = p.cola_proceso()
107            else:
108                if paso is False:
109                    if contador == 0:
110                        print_info(' [dev][info] Escuchando microfono')
111                        contador += 1
112                        paso = p.lectura_microfono(True)
113                if paso is True:
114                    comando = p.comandos_microfono(motor_stt, True, mensaje)
115                    mensaje = False
116                else:
117                    comando = [False, False]
118                if comando[0]:
119                    print_info(' [dev][info] Comando reconocido: {}'.format(comando[1]))
120                    p.reproducir_audio_menu('espera')
121                    if comando[1] == 'comando_pdf' or comando[1] == 'comando_pdf_navegar':
122                        if comando[1] == 'comando_pdf':
123                            n_pagina = int(comando[2])
124                            print_info(' [dev][info] Intentando cargar pagina:
125                                ↪ {}'.format(str(n_pagina)))
126                        else:
127                            n_pagina = pagina_pdf + int(comando[2])
128                            if n_pagina < 1:
129                                n_pagina = 1
129                            salida_pdf = p.leer_pagina_pdf(pdf_file, n_pagina)
130                            if salida_pdf[0] == True:
131                                salida = p.texto_a_audio(salida_pdf[1], motor_tts)
132                                cache_texto = salida_pdf[1]
133                                pagina_pdf = int(comando[2])
134                                if salida[0]:
135                                    cache_audio = salida[1]
136                                    is_cache_texto = True
137                                elif salida_pdf[0] == 'Imagen':
138                                    if motor_ocr == 'tesseract':
139                                        p.pro_tesseract.procesar(imagen='images/pdf.jpg',
140                                            ↪ motor_tts=motor_tts)
141                                        pagina_pdf = int(salida_pdf[2])
142                                        is_cache_texto = True
143                                    else:
144                                        texto_imagen = p.texto_de_imagen(imagen='images/pdf.jpg',
145                                            ↪ motor=motor_ocr)
146                                        cache_texto = texto_imagen
147                                        is_cache_texto = True
148                                        pagina_pdf = int(salida[2])
149                                        salida = p.texto_a_audio(texto_imagen, motor_tts)

```

```

148 -----if salida[0]:
149 -----cache_audio = salida[1]
150 ----else:
151 ----print_error('[dev][error] No existe la pagina seleccionada')
152 ----if comando[1] == 'comando_leer':
153 ----print_info('[dev][info] Leer texto en fisico - {}'.format(motor_ocr))
154 ----p.obtener_imagen_camara(archivo='imagenes/{}'.format(img_file),
    ↪ from_cam=True)
155 ----if motor_ocr == 'tesseract':
156 -----salida =
    ↪ p.pro_tesseract.procesar(imagen='imagenes/{}'.format(img_file),
    ↪ motor_tts=motor_tts)
157 -----if salida == None:
158 -----#Error
159 -----print_error('[dev][error] Error al procesar la imagen')
160 ----else:
161 -----is_cache_texto = True
162 ----else:
163 -----texto_imagen =
    ↪ p.texto_de_imagen(imagen='imagenes/{}'.format(img_file), motor=motor_ocr)
164 -----cache_texto = texto_imagen
165 -----is_cache_texto = True
166 -----salida = p.texto_a_audio(texto_imagen, motor_tts)
167 -----if salida[0]:
168 -----cache_audio = salida[1]
169 ----elif comando[1] == 'comando_repetir_texto':
170 ----if is_cache_texto:
171 ----if motor_ocr == 'tesseract':
172 ----p.pro_tesseract.repetir_cache()
173 ----else:
174 -----id_proceso = p.reproducir_texto(motor=motor_tts,
    ↪ numero=cache_audio, espera=True)
175 ----else:
176 ----p.reproducir_audio_menu('no_texto')
177 ----print_info('[dev][info] No existe cache de texto') # falta respuesta
    ↪ de audio
178 ----elif comando[1] == 'comando_buscar':
179 ----cache_resultados = p.webs.buscar_en_linea(busqueda=comando[2],
    ↪ pagina=0, motor=motor_web)
180 ----if cache_resultados[0] is not None:
181 ----texto_busqueda = 'Resultado de la busqueda de {},
    ↪ '.format(comando[2])
182 ----for i in range(0, cache_resultados[2]):
183 ----texto_busqueda = '{} resultado numero {}'.format(texto_busqueda,
    ↪ str(i + 1))
184 ----texto = cache_resultados[0][i]
185 ----texto_busqueda = '{}', {}, '.format(texto_busqueda, texto)
186 ----texto_busqueda = '{}', {}'.format(texto_busqueda, 'fin de resultados
    ↪ de busqueda')
187 ----if motor_tts == 'festival' and sys.platform == 'win32':
188 ----texto_busqueda = texto_busqueda.replace('"', '')
189 ----is_cache_resultados = True
190 ----resultado = p.texto_a_audio(texto_busqueda, motor_tts)
191 ----if resultado[0]:
192 ----cache_audio = resultado[1]
193 ----save_busqueda = comando[2]
194 ----if not ayuda_navegacion:

```

```

195 -----ayuda_navegacion = True
196 -----p.reproducir_audio_menu('navegacion')
197 ----else:
198 ----p.reproducir_audio_menu('error_busqueda')
199 ----comando = [False, False]
200 ----elif comando[1] == 'comando_repetir_lista':
201 ----if is_cache_resultados:
202 ----p.reproducir_texto(motor=motor_tts, numero=cache_audio, espera=True)
203 ----else:
204 ----print_error('[dev][error] No existe cache de busqueda') # falta
    ↪ respuesta de audio
205 ----elif comando[1] == 'comando_resultado':
206 ----if save_busqueda is not None:
207 ----texto_web = [None]
208 ----try:
209 ----texto_web =
    ↪ p.webs.cargar_web_alter(cache_resultados[1][int(comando[2]) - 1])
210 ----except IndexError:
211 ----print_error('[dev][error] Error de indice')
212 ----p.reproducir_audio_menu('error_indice')
213 ----if texto_web[0] is not None:
214 ----cache_texto = texto_web[0]
215 ----is_cache_texto = True
216 ----cache_audio = p.texto_a_audio(cache_texto, motor_tts)[1]
217 ----else:
218 ----print('[dev][warn] Error de request')
219 ----else:
220 ----print_info('[dev][info] No hay resultados de busqueda previa')
221 ----p.reproducir_audio_menu('no_previa')
222 ----elif comando[1] == 'comando_navegacion':
223 ----if save_busqueda != '':
224 ----cache_resultados = p.webs.buscar_en_linea(busqueda=save_busqueda,
    ↪ pagina=comando[2])
225 ----if cache_resultados[0] is not None:
226 ----texto_busqueda = 'Resultados de la busqueda de {},
    ↪ '.format(save_busqueda)
227 ----for i in range(0, cache_resultados[2]):
228 ----texto_busqueda = '{} resultado numero {}'.format(texto_busqueda,
    ↪ str(i + 1))
229 ----texto = cache_resultados[0][i]
230 ----texto_busqueda = '{} {}, {}'.format(texto_busqueda, texto)
231 ----texto_busqueda = '{} {}, {}'.format(texto_busqueda,
232 ----- 'fin de resultados de busqueda,'
233 ----- 'seleccione la opcion deseada')
234 ----if motor_tts == 'festival' and sys.platform == 'win32':
235 ----texto_busqueda = texto_busqueda.replace('"', '')
236 ----is_cache_resultados = True
237 ----resultado = cache_audio = p.texto_a_audio(texto_busqueda, motor_tts)
238 ----if resultado[0]:
239 ----cache_audio = resultado[1]
240 ----else:
241 ----p.reproducir_audio_menu('error_busqueda')
242 ----elif comando[1] == 'comando_limite':
243 ----if is_cache_resultados:
244 ----p.reproducir_audio_menu('limite')
245 ----p.reproducir_texto(motor=motor_tts, numero=cache_audio, espera=True)
246 ----paso = False

```

```
247 ---mensaje = True
248 ---comando = [False, True]
249 ---print_info('[dev][info] Saliendo...')
250 ---contador = 0
251 ---else:
252 ---if run_mode == 'mic':
253 ----if comando[1]:
254 ----print_info('[dev][info] En espera - microfono')
255 ----comando = [False, False]
256 ----elif comando[1] == False and paso == True:
257 ----print_info('[dev][info] No se reconoce el texto ingresado')
258 ----p.reproducir_audio_menu('no_comando')
259 ----paso = False
260 ---else:
261 ----# print_info('[dev][info][main] en espera')
262 ----p.reestablecer_proceso()
```



```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: tts_bing.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 """
6
7 -Un modulo para utilizar la API REST del Azure Cloud Services -
  ↳ Text-To-Speech
8
9 """
10 import requests
11
12
13 class BingTTS:
14
15     -def __init__(self, api_key=None):
16     -"""
17     --Establece los parametros necesarios
18
19     --:param key_api: La clave API necesaria para acceder a la API
20     -"""
21     --self.base_url =
22     ↳ 'https://westus.tts.speech.microsoft.com/cognitiveservices/v1'
23     --self.token_url =
24     ↳ 'https://westus.api.cognitive.microsoft.com/sts/v1.0/issueToken'
25     --self.payload = '<speak version=\'1.0\'
26     ↳ xmlns="http://www.w3.org/2001/10/synthesis" xml:lang=\'{}\>' \
27     ----- '<voice name=\'Microsoft Server Speech Text to Speech Voice ({}
28     ↳ {})\>' \
29     ----- '{}</voice></speak>'
30     --self.api_key = api_key
31     --self.token = None
32     --self.voice_list = {
33     ---"ar-EG": ["Hoda"],
34     ---"ar-SA": ["Naayf"],
35     ---"bg-BG": ["Hoda"],
36     ---"ca-ES": ["HerenaRUS"],
37     ---"cs-CZ": ["Jakub"],
38     ---"da-DK": ["HelleRUS"],
39     ---"de-AT": ["Michael"],
40     ---"de-CH": ["Karsten"],
41     ---"de-DE": ["Hedda", "HeddaRUS", "Stefan, Apollo"],
42     ---"el-GR": ["Stefanos"],
43     ---"en-AU": ["Catherine", "HayleyRUS"],
44     ---"en-CA": ["Linda", "HeatherRUS"],
45     ---"en-GB": ["Susan, Apollo", "HazelRUS", "George, Apollo"],
46     ---"en-IE": ["Sean"],
47     ---"en-IN": ["Heera, Apollo", "PriyaRUS", "Ravi, Apollo"],
48     ---"en-US": ["ZiraRUS", "JessaRUS", "BenjaminRUS"],
49     ---"es-ES": ["Laura, Apollo", "HelenaRUS", "Pablo, Apollo"],
50     ---"es-MX": ["HildaRUS", "Raul, Apollo"],
51     ---"fi-FI": ["HeidiRUS"],
52     ---"fr-CA": ["Caroline", "HarmonieRUS"],
53     ---"fr-CH": ["Guillaume"],
54     ---"fr-FR": ["Julie, Apollo", "HortenseRUS", "Paul, Apollo"],
55     ---"he-IL": ["Asaf"],

```

```

52 ---"hi-IN": ["Kalpana, Apollo", "Kalpana", "Hemant"],
53 ---"hr-HR": ["Matej"],
54 ---"hu-HU": ["Szabolcs"],
55 ---"id-ID": ["Andika"],
56 ---"it-IT": ["Cosimo, Apollo"],
57 ---"ja-JP": ["Ayumi, Apollo", "Ichiro, Apollo", "HarukaRUS", "LuciaRUS",
↪ "EkaterinaRUS"],
58 ---"ko-KR": ["HeamiRUS"],
59 ---"ms-MY": ["Rizwan"],
60 ---"nb-NO": ["HuldaRUS"],
61 ---"nl-NL": ["HannaRUS"],
62 ---"pl-PL": ["PaulinaRUS"],
63 ---"pt-BR": ["HeloisaRUS", "Daniel, Apollo"],
64 ---"pt-PT": ["HeliaRUS"],
65 ---"ro-RO": ["Andrei"],
66 ---"ru-RU": ["Irina, Apollo", "Pavel, Apollo"],
67 ---"sk-SK": ["Filip"],
68 ---"sl-SI": ["Lado"],
69 ---"sv-SE": ["HedvigRUS"],
70 ---"ta-IN": ["Valluvar"],
71 ---"th-TH": ["Pattara"],
72 ---"tr-TR": ["SedaRUS"],
73 ---"vi-VN": ["An"],
74 ---"zh-CN": ["HuihuiRUS", "Yaoyao, Apollo", "Kangkang, Apollo"],
75 ---"zh-HK": ["Tracy, Apollo", "TracyRUS", "Danny, Apollo"],
76 ---"zh-TW": ["Yating, Apollo", "HanHanRUS", "Zhiwei, Apollo"]
77 --}
78 --self.formats_list = [
79 ---'raw-8khz-8bit-mono-mulaw',
80 ---'raw-16khz-16bit-mono-pcm',
81 ---'raw-24khz-16bit-mono-pcm',
82 ---'riff-8khz-8bit-mono-mulaw',
83 ---'riff-16khz-16bit-mono-pcm',
84 ---'riff-24khz-16bit-mono-pcm',
85 ---'audio-16khz-32kbitrate-mono-mp3',
86 ---'audio-16khz-64kbitrate-mono-mp3',
87 ---'audio-16khz-128kbitrate-mono-mp3',
88 ---'audio-24khz-48kbitrate-mono-mp3',
89 ---'audio-24khz-96kbitrate-mono-mp3',
90 ---'audio-24khz-160kbitrate-mono-mp3'
91 --]
92
93 -def get_token(self):
94 --"""
95 --Genera un token de autenticacion de 10 minutos de duracion para acceder
↪ a la API
96
97 --:return: El token de autenticacion
98 --"""
99 --headers = {'Ocp-Apim-Subscription-Key': self.api_key}
100 --response_token = requests.post(self.token_url, headers=headers)
101 --if int(response_token.status_code) != 200:
102 ---raise BingAuthException(response_token.content)
103 --else:
104 ---return response_token.content
105
106 -def speak(self, language='en-US', voice='ZiraRUS',

```

```

    ↪ file_format='riff-8khz-8bit-mono-mulaw', text=None):
107 --"""
108 --Convierte el texto ingresado a voz utilizando los parametros de
    ↪ configuracion proporcionados
109
110 --:param language: El idioma a utilizar en el formato xx-XX, por defecto
    ↪ 'es-US'
111 --:param voice: La voz a utilizar del idioma seleccionado, por defecto
    ↪ 'ZiraRUS'
112 --:param file_format: El formato de archivo de salida seleccionado, por
    ↪ defecto 'riff-8khz-8bit-mono-mulaw'
113 --:param text: El texto que se desea convertir a audio, no mayor a 950
    ↪ caracteres
114 --:return: El contenido binario del archivo de voz generado
115 --"""
116 --if len(text) > 950:
117 ---raise BingTextLongException('[dev][error][tts_bing] Input text is too
    ↪ long')
118 --elif not text:
119 ---raise BingNoTextException('[dev][error][tts_bing] No input text')
120 --if not self.token:
121 ---self.token = self.get_token()
122 --if language not in self.voice_list:
123 ---raise BingLanguageException('[dev][error][tts_bing] {} language not
    ↪ available'.format(language))
124 --if voice not in self.voice_list[language]:
125 ---raise BingLanguageException('[dev][error][tts_bing] {} language does
    ↪ not have {} voice'.format(language,
126 ----- voice))
127 --if file_format not in self.formats_list:
128 ---raise BingFormatException('[dev][error][tts_bing] Format not valid')
129 --headers = {'Content-type': 'application/ssml+xml',
130 ----- 'X-Microsoft-OutputFormat': file_format,
131 ----- 'X-Search-AppId': 'D9BD1436F82A7F8F5C775E37E80C8790',
132 ----- 'X-Search-ClientID': 'EF828F621DEE0B05CF5A35D35472AE10',
133 ----- 'User-Agent': 'BingTTS',
134 ----- 'Authorization': 'Bearer {}'.format(self.token.decode('utf-8'))}
135 --body = self.payload.format(language,
136 ----- language,
137 ----- voice,
138 ----- text).encode('utf-8', 'ignore')
139 --try:
140 ---response = requests.post(self.base_url, headers=headers, data=body)
141 --except requests.exceptions.RequestException as err:
142 ---raise BingRequestException('[dev][error][tts_bing] Request error:
    ↪ {}'.format(err))
143 --if int(response.status_code) != 200:
144 ---if int(response.status_code) == 401:
145 -----self.token = None
146 -----return self.speak(language=language,
147 ----- voice=voice,
148 ----- file_format=file_format,
149 ----- text=text)
150 --else:
151 -----raise BingBadRequestException(response.status_code)
152 --elif int(response.status_code) == 200:
153 ---header = response.headers.get('content-type').lower()

```

```

154 ---if 'text' in header or 'html' in header:
155 ----raise BingFileException(' [dev][error][tts_bing] Audio file can\'t
↳ read')
156 ---else:
157 ----return response.content
158
159
160 """
161
162 -Definicion de las clases encargadas del manejo de las excepciones
163 -
164 """
165
166
167 class BingAuthException(Exception):
168 -def __init__(self, message):
169 --self.message = '{}'.format(message)
170 --super(BingAuthException, self).__init__(self.message)
171
172
173 class BingLanguageException(Exception):
174 -def __init__(self, message):
175 --self.message = '{}'.format(message)
176 --super(BingLanguageException, self).__init__(self.message)
177
178
179 class BingFormatException(Exception):
180 -def __init__(self, message):
181 --self.message = '{}'.format(message)
182 --super(BingFormatException, self).__init__(self.message)
183
184
185 class BingRequestException(Exception):
186 -def __init__(self, message):
187 --self.message = '{}'.format(message)
188 --super(BingRequestException, self).__init__(self.message)
189
190
191 class BingFileException(Exception):
192 -def __init__(self, message):
193 --self.message = '{}'.format(message)
194 --super(BingFileException, self).__init__(self.message)
195
196
197 class BingTextLongException(Exception):
198 -def __init__(self, message):
199 --self.message = '{}'.format(message)
200 --super(BingTextLongException, self).__init__(self.message)
201
202
203 class BingBadRequestException(Exception):
204 -def __init__(self, message):
205 --self.message = '{}'.format(message)
206 --super(BingBadRequestException, self).__init__(self.message)
207
208
209 class BingNoTextException(Exception):

```

```
210 def __init__(self, message):  
211     self.message = '{}'.format(message)  
212     super(BingNoTextException, self).__init__(self.message)
```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: tts_voicerss.py - cada (-) al principio de cada linea
4 #   representa 1 tabulacion o 4 espacios
5 """
6
7 -Una libreria para utilizar la API REST del servicio voiceRSS
8
9 """
10 import requests
11
12
13 class VoiceRSS:
14
15     def __init__(self, api_key=None):
16         """
17         ----Establece los parametros necesarios
18
19         ----:param api_key: La clave API necesaria para acceder a la API
20         ----"""
21         --self.base_url = 'http://api.voicerss.org/?{}'
22         --self.payload = 'key={}&hl={}&f={}&c={}&src={}'
23         --self.api_key = api_key
24         --self.lang_list = ['ca-es', 'zh-cn', 'zh-hk', 'zh-tw', 'da-dk', 'nl-nl',
25             ↪ 'en-au', 'en-ca', 'en-gb', 'en-in',
26             ----- 'en-us', 'fi-fi', 'fr-ca', 'fr-fr', 'de-de', 'it-it', 'ja-jp',
27             ↪ 'ko-kr', 'nb-no', 'pl-pl',
28             ----- 'pt-br', 'pt-pt', 'ru-ru', 'es-mx', 'es-es', 'sv-se']
29         --self.format_list = ['8khz_8bit_mono', '8khz_8bit_stereo',
30             ↪ '8khz_16bit_mono', '8khz_16bit_stereo',
31             -----'11khz_8bit_mono', '11khz_8bit_stereo', '11khz_16bit_mono',
32             ↪ '11khz_16bit_stereo',
33             -----'12khz_8bit_mono', '12khz_8bit_stereo', '12khz_16bit_mono',
34             ↪ '12khz_16bit_stereo',
35             -----'16khz_8bit_mono', '16khz_8bit_stereo', '16khz_16bit_mono',
36             ↪ '16khz_16bit_stereo',
37             -----'22khz_8bit_mono', '22khz_8bit_stereo', '22khz_16bit_mono',
38             ↪ '22khz_16bit_stereo',
39             -----'24khz_8bit_mono', '24khz_8bit_stereo', '24khz_16bit_mono',
40             ↪ '24khz_16bit_stereo',
41             -----'32khz_8bit_mono', '32khz_8bit_stereo', '32khz_16bit_mono',
42             ↪ '32khz_16bit_stereo',
43             -----'44khz_8bit_mono', '44khz_8bit_stereo', '44khz_16bit_mono',
44             ↪ '44khz_16bit_stereo',
45             -----'48khz_8bit_mono', '48khz_8bit_stereo', '48khz_16bit_mono',
46             ↪ '48khz_16bit_stereo',
47             -----'alaw_8khz_mono', 'alaw_8khz_stereo', 'alaw_11khz_mono',
48             ↪ 'alaw_11khz_stereo',
49             -----'alaw_22khz_mono', 'alaw_22khz_stereo', 'alaw_44khz_mono',
50             ↪ 'alaw_44khz_stereo',
51             -----'ulaw_8khz_mono', 'ulaw_8khz_stereo', 'ulaw_11khz_mono',
52             ↪ 'ulaw_11khz_stereo',
53             -----'ulaw_22khz_mono', 'ulaw_22khz_stereo', 'ulaw_44khz_mono',
54             ↪ 'ulaw_44khz_stereo']
55         --self.codec_list = ['MP3', 'WAV', 'AAC', 'OGG', 'CAF']
56
57
58
59
60
61

```

```

42 def speak(self, language='es-mx', text=None, codec='MP3',
↳ audio_format='32khz_16bit_stereo'):
43 """
44 --Convierte el texto ingresado a voz utilizando los parametros de
↳ configuracion proporcionados
45
46 --:param language: El idioma a utilizar en el formato xx-xx, por defecto
↳ 'es-mx'
47 --:param text: El texto que se desea convertir a audio, no mayor a 950
↳ caracteres
48 --:param codec: El codec de archivo de salida seleccionado, por defecto
↳ 'MP3'
49 --:param audio_format: El formato de archivo de salida seleccionado, por
↳ defecto '32khz_16bit_stereo'
50 --:return: El contenido binario del archivo de voz generado
51 """
52 if len(text) > 950:
53 raise VrTextLongException(' [dev][error][tts_voicerSS] El texto
↳ ingresado es muy largo')
54 elif not text:
55 raise VrNoTextException(' [dev][error][tts_voicerSS] No se ingreso
↳ texto')
56 if language not in self.lang_list:
57 raise VrLangException(' [dev][error][tts_voicerSS] Lenguaje no valido')
58 if codec not in self.codec_list:
59 raise VrCodecException(' [dev][error][tts_voicerSS] Codec no valido')
60 if audio_format not in self.format_list:
61 raise VrFormatException(' [dev][error][tts_voicerSS] Formato no valido')
62 tts_request = self.base_url.format(self.payload.format(self.api_key,
↳ language, audio_format, codec, text))
63 try:
64 response = requests.get(tts_request)
65 except requests.exceptions.RequestException:
66 raise VrRequestException(' [dev][error][tts_voicerSS] Request error')
67 if int(response.status_code) != 200:
68 raise VrBadRequestException(response.status_code)
69 elif int(response.status_code) == 200:
70 header = response.headers.get('content-type').lower()
71 if 'text' in header or 'html' in header:
72 raise VrFileException(' [dev][error][tts_voicerSS] Archivo de audio no
↳ legible')
73 else:
74 return response.content
75
76
77 class VrRequestException(Exception):
78 def __init__(self, message):
79 self.message = '{}'.format(message)
80 super(VrRequestException, self).__init__(self.message)
81
82
83 class VrFileException(Exception):
84 def __init__(self, message):
85 self.message = '{}'.format(message)
86 super(VrFileException, self).__init__(self.message)
87
88

```

```
89 class VrTextLongException(Exception):
90     -def __init__(self, message):
91         --self.message = '{}'.format(message)
92         --super(VrTextLongException, self).__init__(self.message)
93
94
95 class VrBadRequestException(Exception):
96     -def __init__(self, message):
97         --self.message = '{}'.format(message)
98         --super(VrBadRequestException, self).__init__(self.message)
99
100
101 class VrNoTextException(Exception):
102     -def __init__(self, message):
103         --self.message = '{}'.format(message)
104         --super(VrNoTextException, self).__init__(self.message)
105
106
107 class VrLangException(Exception):
108     -def __init__(self, message):
109         --self.message = '{}'.format(message)
110         --super(VrLangException, self).__init__(self.message)
111
112
113 class VrFormatException(Exception):
114     -def __init__(self, message):
115         --self.message = '{}'.format(message)
116         --super(VrFormatException, self).__init__(self.message)
117
118
119 class VrCodecException(Exception):
120     -def __init__(self, message):
121         --self.message = '{}'.format(message)
122         --super(VrCodecException, self).__init__(self.message)
```



```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: web.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 """
6 web.py:
7 -Proporciona metodos para la interaccion web
8 """
9
10 import requests
11 from bs4 import BeautifulSoup
12 import html2text
13 from termcolor import cprint
14 from urllib.parse import urlencode, unquote
15
16 def print_error(x): cprint(x, 'red', attrs=['bold', 'underline'], end='
↳ \n')
17 def print_info(x): cprint(x, 'yellow', attrs=['bold'], end=' \n')
18 def print_log(x): cprint(x, 'cyan', attrs=['bold'], end=' \n')
19 def print_log_a(x): cprint(x, 'magenta', attrs=['underline'], end=' \n')
20 def print_info_nb(x): cprint(x, 'yellow', attrs=['bold'], end='')
21 def print_info_rv(x): cprint(x, 'magenta', attrs=['bold'], end=' \n')
22
23 class WebS:
24 -def __init__(self):
25 --pass
26
27 -@staticmethod
28 -def buscar_en_linea(busqueda, pagina, motor='google'):
29 --""" Realiza una busqueda de uno o varios terminos en internet y genera
↳ una lista de resultados con la
30 --forma [[lista_texto], [lista_url], numero_resultados]
31 --:param busqueda: El termino o terminos de busqueda
32 --:param pagina: El numero de pagina de busqueda en base a 10
33 --:param motor: Motor de busqueda, por defecto google.com
34 --:return: La lista con los resultados de busqueda, None en caso de error
35 --"""
36 --lista_urls = []
37 --lista_texto = []
38 --n_urls = 0
39 --headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64;
↳ rv:54.0) Gecko/20100101 Firefox/54.0'}
40 --query = busqueda
41 --if motor == 'google':
42 ---if pagina == 0:
43 ----pagina = 1
44 ---url_google = 'https://www.google.com/search?{}'.format(urlencode({'q':
↳ query, 'start': pagina}))
45 ---print_info(' [dev] [info] [web] E: {} - URL: {}'.format(motor,
↳ url_google))
46 ---try:
47 ----r_google = requests.get(url_google)
48 ----code_req = r_google.status_code
49 ----if code_req == 200:
50 -----# Nuevo scraping
51 -----s = BeautifulSoup(markup=r_google.content, features='html.parser')

```

```

52 -----# print(s)
53 -----lnk = s.find_all(name='a')
54 -----for ln in lnk:
55 -----divs = ln.find_all('div')
56 -----if divs != []:
57 -----if len(divs) == 2:
58 -----url = ln['href']
59 -----texto = divs[0].get_text()
60 -----if url and texto:
61 -----if url != '#' and texto.find('Imágenes de ') == -1 and
↳ texto.find('\n') == -1:
62 -----nurl = url.replace('/url?q=', '')
63 -----pos = nurl.find('&sa=')
64 -----nurl = nurl[0:pos]
65 -----lista_urls.append(nurl)
66 -----lista_texto.append(texto)
67 -----n_urls += 1
68 -----url = '#'
69 -----texto = False
70 ---print_info(['log][info] Resultados de búsqueda encontrados') # dev
71 ---except requests.RequestException:
72 ---print_error(['log][error] Error al ejecutar la búsqueda en línea') #
↳ dev
73 ---return None
74 ---if motor == 'bing':
75 ---params = None
76 ---if pagina == 0:
77 ---params = {'q': query, 'cvid': '9C5FE5E49ADF00723D6FDF4F0A0AC5DD',
↳ 'FORM': 'PERE'}
78 ---else:
79 ---pag = (pagina * 10) + 1
80 ---if pagina == 1:
81 ---params = {'q': query, 'FORM': 'PERE', 'first': str(pag), 'cvid':
↳ '9C5FE5E49ADF00723D6FDF4F0A0AC5DD'}
82 ---elif pagina > 1:
83 ---params = {'q': query, 'FORM': 'PERE' + str(pagina - 1), 'first':
↳ str(pag),
84 ----- 'cvid': '9C5FE5E49ADF00723D6FDF4F0A0AC5DD'}
85 ---url_bing = 'https://www.bing.com/search?{}'.format(urllib.urlencode(params))
86 ---print_info(['dev][info][web] E: {} - URL: {}'.format(motor, url_bing))
87 ---try:
88 ---r_bing = requests.get(url_bing, headers=headers)
89 ---code_req = r_bing.status_code
90 ---if code_req == 200:
91 ---s = BeautifulSoup(markup=r_bing.content, features='html.parser')
92 ---divs = s.find_all(name='li', attrs={'class': 'b_algo'})
93 ---nnn = 0
94 ---for div in divs:
95 ---nnn += 1
96 ---url = div.find('a')['href']
97 ---texto = div.find('a').get_text()
98 ---if url != '#':
99 ---lista_urls.append(url)
100 ---lista_texto.append(texto)
101 ---n_urls += 1
102 ---print_info(['log][info] Resultados de búsqueda encontrados') # dev
103 ---except requests.RequestException:

```

```

104 ----print_error('[log][error] Error al ejecutar la búsqueda en línea') #
    ↪ dev
105 ----return None
106 --if motor == 'duckduckgo':
107 ---params = None
108 ---url_duckgo = 'https://duckduckgo.com/lite/'
109 ---if pagina > 8:
110 ----pagina = 0
111 ---if pagina > 2:
112 ----nn = pagina // 3
113 ----n = pagina - (nn * 3)
114 ----if 3 <= pagina <= 5:
115 ----params = {'q': query, 's': '30', 'v': '1', 'dc': '30', 'kl':
    ↪ 'en-us', 'api': '/d.js', 'o': 'json'}
116 ----if 6 <= pagina <= 8:
117 ----params = {'q': query, 's': '80', 'v': '1', 'dc': '80', 'kl':
    ↪ 'en-us', 'api': '/d.js', 'o': 'json'}
118 ---else:
119 ----n = pagina
120 ----params = {'q': query, 's': 0}
121 ---pos1 = (n * 10)
122 ---pos2 = (n * 10) + 9
123 ---if n == 2:
124 ----pos2 = (n * 10) + 8
125 ---print_info('[dev][info][web] E: {} - URL: {}?{}'.format(motor,
    ↪ url_duckgo, urlencode(params)))
126 ---try:
127 ----r_duckgo = requests.post(url_duckgo, data=params, headers=headers)
128 ----code_req = r_duckgo.status_code
129 ----if code_req == 200:
130 -----# Nuevo scraping
131 -----s = BeautifulSoup(markup=r_duckgo.content, features='html.parser')
132 -----urls = s.find_all(name='a', attrs={'class': 'result-link'})
133 -----arr_pos = 0
134 -----for url in urls:
135 -----if url['href'] != '#':
136 -----if pos1 <= arr_pos <= pos2:
137 -----nurl = url['href']
138 -----nurl = unquote(nurl)
139 -----lista_urls.append(nurl)
140 -----lista_texto.append(url.get_text())
141 -----n_urls += 1
142 -----arr_pos += 1
143 ----print_info('[log][info] Resultados de búsqueda encontrados') # dev
144 ---except requests.RequestException:
145 ----print_error('[log][error] Error al ejecutar la búsqueda en línea') #
    ↪ dev
146 ----return None
147 --return [lista_texto, lista_urls, n_urls]
148
149 -# cargar contenido de una web
150 -@staticmethod
151 -def cargar_web(url):
152 --"""
153 --Obtiene el contenido de texto a partir de la URL proporcionada
154 --:param url: La URL a la que se desea acceder
155 --:return: El texto contenido en la URL seleccionada, None en caso de

```

```

↪ error
156 """
157 --requests_exceptions = (requests.RequestException,
↪ requests.ConnectionError)
158 --try:
159 ---print_info('[dev][info] Intento de scraping - URL: {}'.format(url))
160 ---req_scraping = requests.get(url)
161 ---req_scraping.encoding = 'utf-8'
162 ---code_response_scraping = req_scraping.status_code
163 ---if code_response_scraping == 200:
164 ----# return req_scraping.text
165 ----content_scraping = BeautifulSoup(req_scraping.text, 'html.parser')
166 ----# article da resultados duplicados
167 ----listado_tags = ['p', 'span', 'li']
168 ----listado_heads = ['h1', 'h2', 'h3', 'h4', 'h5']
169 ----divs_information = content_scraping.find_all(listado_tags)
170 ----divs_text = ''
171 ----y = 0
172 ----prev = ''
173 ----for div in divs_information:
174 -----header = False
175 -----for i in listado_heads:
176 -----if str(div).find(i) > -1:
177 -----header = True
178 -----break
179 -----if len(div.get_text()) > 10 or (len(div.get_text()) > 5 and header):
180 -----texto_item = div.get_text()
181 -----if prev != texto_item:
182 -----prev = texto_item
183 -----divs_text = '{} {}'.format(divs_text, texto_item)
184 -----y += 1
185 ----divs_text = ' '.join(divs_text.split())
186 ----print_info('[log][info] Scraping finalizado') # dev
187 ----return [divs_text, None]
188 ----elif 404 == code_response_scraping:
189 ----print_error('[log][error] Error 404')
190 ----return [None, '404']
191 ----else:
192 ----print_error('[log][error] Error request scraping') # dev
193 ----return [None, 'request']
194 --except TypeError:
195 ---print_error('[log][error] Error en el Scraping') # dev
196 ---return [None, 'connect']
197
198 -@staticmethod
199 -def cargar_web_alter(url):
200 --h = html2text.HTML2Text()
201 --# h.ignore_tables = True
202 --h.ignore_images = True
203 --h.ignore_links = True
204 --h.images_to_alt = True
205 --h.ignore_emphasis = True
206 --requests_exceptions = (requests.exceptions.RequestException,
↪ requests.exceptions.ConnectionError, TypeError)
207 --try:
208 ---print_info('[dev][info] Intento de scraping - URL: {}'.format(url))
209 ---req_scraping = requests.get(url)

```

```

210 ---req_scraping.encoding = 'utf-8'
211 ---code_response_scraping = req_scraping.status_code
212 ---print(str(code_response_scraping))
213 ---if code_response_scraping == 200:
214 ----# content_scraping = BeautifulSoup(req_scraping.text, 'html.parser')
215 ----divs_text = h.handle(req_scraping.text)
216 ----list = divs_text.split('\n\n')
217 ----acumulado = ''
218 ----for i in range(0, len(list)):
219 -----sublist = list[i].split('\n')
220 -----for j in range(0, len(sublist)):
221 -----sub_texto = sublist[j].strip().replace(' ', ' ')
222 -----if len(sub_texto) == 0:
223 -----continue
224 -----for i in range(1, 5):
225 -----if sub_texto[0:i + 1] == (('#' * i) + ' '):
226 -----sub_texto = sub_texto[i + 1:]
227 -----break
228 -----if sub_texto[0:2] == '* ':
229 -----sub_texto = sub_texto[2:]
230 -----acumulado += sub_texto.strip() + '\n'
231 -----acumulado = acumulado.replace('[editar]', '').replace('\n', ' ')
232 -----acumulado = str(acumulado)
233 ----return [acumulado, None]
234 ---elif 404 == code_response_scraping:
235 ---print_error('[log][error] Error 404')
236 ---return [None, '404']
237 ---else:
238 ---print_error('[log][error] Error request scraping') # dev
239 ---return [None, 'request']
240 --except (requests_exceptions):
241 ---print_error('[log][error] Error en el Scraping') # dev
242 ---return [None, 'connect']

```

```

1 # -*- coding: utf-8 -*-
2 #
3 # Archivo: hunspell.py - cada (-) al principio de cada linea
4 # representa 1 tabulacion o 4 espacios
5 import os
6 import sys
7 import tempfile
8 import subprocess
9 from time import sleep
10
11
12 class Whunspell:
13
14 -class HunSpell:
15 --def __init__(self, dictionary='es_US.dic', affic='es_US.aff'):
16 ---"""
17 ---A
18
19 ---:param dictionary:
20 ---:param affic:
21 ---"""
22 ---self.__contenido = []
23 ---self.__array_palabras = []
24 ---self.__temp_file = None
25 ---self.__path = 'hunspell'
26 ---dictionary = dictionary.split('.')[0]
27 ---self.__dictionary = dictionary.replace('.dic', '')
28 ---self.__spell_check = subprocess.Popen('{} -i utf-8 -G -d {}
↳ -a'.format(self.__path,
29 -----self.__dictionary),
30 ----- stdout=subprocess.PIPE,
31 ----- stdin=subprocess.PIPE,
32 ----- bufsize=1,
33 ----- shell=True)
34 ---self.__spell_check.stdout.readline()
35 ---assert affic
36
37 --def stop(self):
38 ---"""
39 ---A
40
41 ---"""
42 ---self.__spell_check.communicate()
43 ---self.__spell_check.terminate()
44 ---self.__spell_check.kill()
45
46 --def load_text(self, text):
47 ---"""
48 ---A
49
50 ---:param text:
51 ---"""
52 ---arr_palabras = text.split(' ')
53 ---lineas = []
54 ---self.__temp_file = tempfile.NamedTemporaryFile(mode='w',
↳ encoding='utf-8', suffix='.txt', delete=False)

```

```

55 ---self.__temp_file.write(text.replace(' ', '\n'))
56 ---self.__temp_file.close()
57 ---sleep(0.05)
58 ---mp = subprocess.Popen('{} -i utf-8 -a -d {} <"{}"'.format(self.__path,
↳ self.__dictionary,
59 ----- self.__temp_file.name),
60 ----- stdout=subprocess.PIPE,
61 ----- shell=True)
62 ---mp.stdout.readline()
63 ---lineas = mp.stdout.readlines()
64 ---n = 0
65 ---for i in range(0, len(lineas)):
66 ----if lineas[i] != b'\r\n':
67 -----linea = lineas[i].decode('utf-8', 'replace').replace('\r\n', '')
68 ----if linea[0:2] == '& ':
69 -----self.__contenido.append(linea)
70 -----n += 1
71 ----else:
72 -----if (i + 1) < len(lineas) - 1:
73 -----if lineas[i] == b'\r\n' and lineas[i+1] == b'\r\n':
74 -----self.__contenido.append('# ' + str(arr_palabras[n]))
75 -----n += 1
76 ---mp.communicate(0)
77 ---mp.terminate()
78 ---mp.wait(timeout=0.2)
79 ---mp.kill()
80
81 --def spell_new(self, word):
82 ---"""
83 ---A
84
85 ---:param word:
86 ---:return:
87 ---"""
88 ---word = word.replace(' ', '')
89 ---if self.__temp_file is not None:
90 ----try:
91 -----pass
92 -----os.unlink(self.__temp_file.name)
93 ----except (PermissionError, FileNotFoundError):
94 -----pass
95 ---out = True
96 ---for i in range(0, len(self.__contenido)):
97 ----if self.__contenido[i].find(word) > -1:
98 -----out = False
99 ----break
100 ---return out
101
102 --def suggest_new(self, word):
103 ---"""
104 ---A
105
106 ---:param word:
107 ---:return:
108 ---"""
109 ---word = word.replace(' ', '')
110 ---if self.__temp_file is not None:

```

```

111 ----try:
112 ----pass
113 -----os.unlink(self.__temp_file.name)
114 ----except (PermissionError, FileNotFoundError):
115 ----pass
116 ---salida = None
117 ---for i in range(0, len(self.__contenido)):
118 ----if self.__contenido[i][0:1] != '#':
119 -----if self.__contenido[i].find('& ' + word + ' ') > -1:
120 -----salida = self.__contenido[i].split(': ')
121 -----if salida[1].find(', ') > -1:
122 -----salida = salida[1].split(', ')
123 -----else:
124 -----salida = [salida[1]]
125 -----break
126 ---return salida
127
128 --def spell(self, word):
129 ---"""
130 ---A
131
132 ---:param word:
133 ---:return:
134 ---"""
135 ---if not isinstance(word, str):
136 ----word = str(word)
137 ---word += '\n'
138 ---word = word.encode('utf-8', 'ignore')
139 ---self.__spell_check.stdin.write(word)
140 ---self.__spell_check.stdin.flush()
141 ---sleep(0.001)
142 ---eof = True
143 ---out = ''
144 ---while eof:
145 ----line = self.__spell_check.stdout.readline()
146 ----if line == b'\r\n':
147 -----eof = False
148 ----else:
149 -----out = line.decode('utf-8', 'replace').replace('\r\n', '')
150 ---if out == '*' or out == '+':
151 ----return True
152 ---else:
153 ----return False
154
155 --def suggest(self, word):
156 ---sleep(0.1)
157 ---word += '\n'
158 ---word = word.encode('utf-8', 'ignore')
159 ---self.__spell_check.stdin.write(word)
160 ---self.__spell_check.stdin.flush()
161 ---eof = True
162 ---out = ''
163 ---while eof:
164 ----line = self.__spell_check.stdout.readline()
165 ----if line == b'\r\n':
166 -----eof = False
167 ----else:

```



```
168 -----out += line.decode('utf-8', 'replace').replace('\r\n', '')
169 ---lista = []
170 ---if out.find(': ') > -1:
171 ----listado = out.split(': ')
172 ----lista = listado[1].split(',')
173 ---return lista
```

Archivo settings.yaml necesario para crear el token de acceso de Google Drive

```
1 client_config_backend: settings
2 client_config:
3   client_id: [id_cliente].apps.googleusercontent.com
4   client_secret: [API cliente]
5
6 save_credentials: True
7 save_credentials_backend: file
8 save_credentials_file: credentials.json
9
10 get_refresh_token: True
11
12 oauth_scope:
13 - https://www.googleapis.com/auth/drive.file
14 - https://www.googleapis.com/auth/drive.install
```

6.6. Anexo: Encuesta dirigida a personal de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador



**Universidad de El Salvador
Facultad Multidisciplinaria Oriental
Departamento de Ingeniería y Arquitectura**

Encuesta dirigida a personal de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

Objetivo: Determinar la disposición del personal de la Unidad Bibliotecaria en la implementación del prototipo "Asistente para lectura de textos físicos y digitales utilizando Inteligencia Artificial" dentro de la Unidad Bibliotecaria en la Facultad multidisciplinaria Oriental de la Universidad de El Salvador.

Indicación: Responda según su criterio o experiencia las interrogantes presentadas a continuación, marcando con una X la respuesta que elija.

1. ¿En alguna ocasión a atendido a estudiantes que sufran discapacidad visual, en el contexto de su trabajo?

Si _____

No _____

2. ¿Como considera la atención de la Facultad a los estudiantes con discapacidad respecto a las facilidades de acceso a material de estudio que les brinda?

Insuficiente _____

Adecuada _____

Óptima _____

Excelente _____

3. ¿Considera el uso de la tecnología como una herramienta útil en el proceso de aprendizaje?

Si _____

No _____

4. ¿A tenido experiencia con el uso de ordenadores u otra herramienta tecnológica en el contexto de su trabajo?

Si _____

No _____

5. ¿Considera que el dispositivo podría ser de utilidad para que estudiantes que sufren discapacidad visual utilicen las instalaciones de la Unidad Bibliotecaria para estudiar?

Si _____

No _____

6. ¿Considera que existe un espacio físico dentro de la Unidad Bibliotecaria que pueda ser dedicado para el uso del dispositivo? ¿Donde sería?

Si _____

No _____

7. ¿Estaría a favor con la implementación del dispositivo dentro de las instalaciones de la Unidad Bibliotecaria?

Si _____

No _____

8. ¿Participaría en los procesos de capacitación en el uso del dispositivo?

Si _____

No _____

9. ¿Brindaría asistencia en el uso del dispositivo a estudiantes con discapacidad visual si llega a ser requerido?

Si _____

No _____

10. ¿Estaría de acuerdo en adecuar material bibliográfico a un formato que facilite su uso en el dispositivo?

Si _____

No _____

6.7. Anexo: Cuestionario dirigido a Jefatura de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador



**Universidad de El Salvador
Facultad Multidisciplinaria Oriental
Departamento de Ingeniería y Arquitectura**

Cuestionario dirigido a Jefatura de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

Objetivo: Determinar la viabilidad en la implementación del prototipo "Asistente para lectura de textos físicos y digitales utilizando Inteligencia Artificial" dentro de la Unidad Bibliotecaria en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

Indicación: Responda según su criterio o experiencia las interrogantes presentadas a continuación, marcando con una X la respuesta que elija.

1. ¿Desde una perspectiva Administrativa, considera posible la implementación del dispositivo dentro de las instalaciones de la Unidad Bibliotecaria?

Si _____ No _____ Tal vez _____

¿Por qué? _____

2. ¿La Unidad Bibliotecaria tomaría a bien gestionar la asignación de los recursos necesarios para la implementación del dispositivo?

Si _____ No _____

3. ¿La Unidad Bibliotecaria cuenta con la capacidad de dedicar y adecuar un espacio físico para la implementación del dispositivo?

Si _____ No _____

¿Dónde? _____

4. ¿La Unidad Bibliotecaria cuenta con personal dedicado al área de restauración y mantenimiento de libros?

Si _____ No _____

5. Para un mejor uso del dispositivo es necesario que los libros que se utilicen tengan un formato determinado, por ejemplo, hojas individuales tamaño carta, texto en color negro justificado, una sola columna, borde no menor a un centímetro en ancho y alto, etc... ¿Es posible adecuar libros a un formato similar al presentado?

Si _____ No _____

6. ¿Se cuenta con personal al que se le pueda asignar la asistencia de estudiantes en el uso del dispositivo?

Si _____ No _____

7. ¿Es posible apartar tiempo del personal en la Unidad Bibliotecaria para su capacitación en el uso del dispositivo?
- Si _____ No _____
8. ¿Considera de utilidad, para la comunidad universitaria, la implementación del dispositivo en las instalaciones de la Unidad Bibliotecaria?
- Si _____ No _____
9. ¿Según su opinión, cree que la Universidad Nacional debería implementar soluciones para facilitar el estudio de estudiantes con discapacidad?
- Si _____ No _____
10. ¿La Unidad Bibliotecaria cuenta en la actualidad con un plan de asistencia o apoyo que facilite el uso del material de estudio a personas con este tipo de discapacidad?
- Si _____ No _____

6.8. Anexo: Cuestionario dirigido a la Jefatura de la Unidad de Sistemas Informáticos de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador



**Universidad de El Salvador
Facultad Multidisciplinaria Oriental
Departamento de Ingeniería y Arquitectura**

Cuestionario dirigido a la jefatura de la Unidad de Sistemas Informáticos de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

Objetivo: Sondar la disposición que tiene la Unidad de Sistemas Informáticos en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador en la implementación del prototipo "Asistente para lectura de textos físicos y digitales utilizando Inteligencia Artificial"

Indicación: Responda según su criterio o experiencia las interrogantes presentadas a continuación, marcando con una X la respuesta que elija.

1. ¿Cuál es el tipo de conexión al servicio de internet con la que se cuenta en la Unidad Bibliotecaria para uso administrativo?

2. ¿Cuál es la velocidad en el servicio de internet que se tiene en la Unidad Bibliotecaria para uso administrativo?

3. Es posible designar una conexión dedicada independiente para el funcionamiento del dispositivo con al menos 2.0 Mbps de velocidad?

Si _____

No _____

4. ¿Es viable la instalación de una estructura de red en la Unidad Bibliotecaria adaptada al dispositivo?

Si _____

No _____

5. ¿La Unidad de Sistemas Informáticos tomaría a bien gestionar la asignación de los recursos necesarios para la implementación del dispositivo?

Si _____

No _____

6. ¿Con que tipo de servidores se cuenta en la Unidad de Sistemas Informáticos?

7. ¿Los servidores de la Unidad de Sistemas Informáticos interpreta el lenguaje de programación Python como lenguaje de servidor?

Si _____

No _____

8. ¿Se cuenta con personal técnico para dar el mantenimiento necesario al dispositivo?

Si _____

No _____

9. ¿Considera de utilidad, para la comunidad universitaria, la implementación del dispositivo en las instalaciones de la facultad?

Si _____

No _____

10. ¿Según su opinión, cree que la Universidad Nacional debería implementar soluciones para facilitar el estudio de estudiantes con discapacidad?

Si _____

No _____

Bibliografía

2to3 - Automated Python 2 to 3 code translation — Python 3.7.2 documentation. URL:

<https://docs.python.org/3/library/2to3.html> (visitado 10-10-2018).

4.1. Numpy — Computación científica con Python para módulos de evaluación continua en asignaturas de ciencias aplicadas. URL:

http://webs.ucm.es/info/aocg/python/modulos_cientificos/numpy/index.html
(visitado 10-10-2018).

A Python Book: Beginning Python, Advanced Python, and Python Exercises. URL:

https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html (visitado 10-10-2018).

About CMUSphinx – CMUSphinx Open Source Speech Recognition. URL:

<https://cmusphinx.github.io/wiki/about/> (visitado 10-10-2018).

About Python™ | Python.org. URL: <https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place> (visitado 10-10-2018).

Adobe Photoshop: Summary from the Encyclopedia of Graphics File Formats. URL:

<https://www.fileformat.info/format/psd/egff.htm> (visitado 10-10-2018).

An overview of the SPHINX speech recognition system - Acoustics, Speech and Signal Processing. URL:

https://www.ri.cmu.edu/pub_files/pub2/lee_k_f_1990_1/lee_k_f_1990_1.pdf
(visitado 10-10-2018).

Announcing Tesseract OCR - The official Google Code blog. URL:

<http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html>
(visitado 10-10-2018).

API REST de servicios de Voz: servicios de Voz - Azure Cognitive Services | Microsoft Docs. URL:

<https://docs.microsoft.com/es-es/azure/cognitive-services/speech-service/rest-apis> (visitado 10-10-2018).

Authenticate your users | Drive REST API | Google Developers. URL:

<https://developers.google.com/drive/api/v3/about-auth> (visitado 10-10-2018).

Basic OCR in OpenCV | Damiles Blog. URL:

<http://blog.damiles.com/2008/11/20/basic-ocr-in-opencv.html> (visitado 10-10-2018).

Bitmap | GIMP, aplicaciones didácticas. URL:

<http://www.ite.educacion.es/formacion/materiales/86/cd/m2/bitmap.html> (visitado 10-10-2018).

Boeing, Geoff y Paul Waddell. “New insights into rental housing markets across the united states: web scraping and analyzing craigslist rental listings”. En: *Journal of Planning Education and Research* 37.4 (2017), págs. 457-476.

Bourne, Rupert RA et al. “Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis”. En: *The Lancet Global Health* 5.9 (2017), e888-e897.

Bradski, Gary y Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, Inc., 2008.

Chiariglione, Leonardo. *RIDING THE MEDIA BITS*. URL:

https://web.archive.org/web/20111101091827/http://ride.chiariglione.org/MPEG%27s_1st_steps.php (visitado 10-08-2018).

Coalson, Josh. *FLAC - format*. URL: <https://xiph.org/flac/format.html#stream> (visitado 10-08-2018).

Code and Data to evaluate OCR accuracy, originally from UNLV/ISRI. URL:

<https://code.google.com/archive/p/isri-ocr-evaluation-tools/> (visitado 10-10-2018).

Contracts for Python. URL: <http://www.nongnu.org/pydbc/> (visitado 10-10-2018).

Cámara Web - Webcam , características y capacidades .:: www.informaticamoderna.com ::. URL:

http://www.informaticamoderna.com/Camara_web.htm (visitado 10-08-2018).

Cámaras IP. URL: http://www.rnds.com.ar/articulos/031/RNDS_084W.pdf (visitado 10-08-2018).

d'Albe, EE Fournier. "On a type-reading optophone". En: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 90.619 (1914), págs. 373-375.

Data model — Python v3.0.1 documentation. URL: <https://docs.python.org/3.0/reference/datamodel.html#special-method-names>

(visitado 10-10-2018).

Debian – Details of package python-imaging in sid. URL:

<https://packages.debian.org/sid/python-imaging> (visitado 10-10-2018).

DIGESTYC. *Encuesta de Hogares de Propósitos Múltiples 2016.* Dirección General de Estadística y Censos, República de El Salvador, 2016.

effbot / pil-2009-raclette — Bitbucket. URL: <https://web.archive.org/web/20150315041249/http://hg.effbot.org/pil-2009-raclette>

(visitado 10-10-2018).

El píxel | GIMP, aplicaciones didácticas. URL:

http://www.ite.educacion.es/formacion/materiales/86/cd/m2/el_pxel.html

(visitado 10-10-2018).

Enable the Google Drive API | Drive REST API | Google Developers. URL:

<https://developers.google.com/drive/api/v3/enable-drive-api> (visitado

10-10-2018).

Entornos virtuales en Python con virtualenv | Oficina de Software Libre. URL: <https://osl.ugr.es/2016/10/17/entornos-virtuales-en-python-con-virtualenv/>

(visitado 10-10-2018).

FAQs - Raspberry Pi Documentation. URL: <https://www.raspberrypi.org/documentation/faqs/>

(visitado 10-10-2018).

Farrell, Gabriel. *The story of blindness.* Harvard University Press, 1956.

Festival. URL: <http://www.cstr.ed.ac.uk/projects/festival/> (visitado 10-10-2018).

Flaxman, Seth R et al. "Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis". En: *The Lancet Global Health* 5.12 (2017), e1221-e1234.

Foster, A y S Resnikoff. "The impact of Vision 2020 on global blindness". En: *Eye* 19.10 (2005), pág. 1133.

Free OCR API. URL: <https://ocr.space/ocrapi> (visitado 10-10-2018).

General Python FAQ — Python 3.7.2 documentation. URL:

<https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place> (visitado 10-10-2018).

Getting started tutorial. URL: <https://cloud.ibm.com/docs/services/speech-to-text?topic=speech-to-text-gettingStarted#gettingStarted> (visitado 10-10-2018).

GIF: Summary from the Encyclopedia of Graphics File Formats. URL:

<https://www.fileformat.info/format/gif/egff.htm> (visitado 10-10-2018).

GitHub - tesseract-ocr/langdata: Source training data for Tesseract for lots of languages. URL:

<https://github.com/tesseract-ocr/langdata> (visitado 10-10-2018).

GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository). URL:

<https://github.com/tesseract-ocr/tesseract/> (visitado 10-10-2018).

Gonzalez, Rafael C y Richard E Woods. “Digital Image Processing, Prentice Hall”. En: *New Jersey* (2002).

Google Code - Updates: Announcing Tesseract OCR. URL:

<https://web.archive.org/web/20061026075310/http://google-code-updates.blogspot.com/2006/08/announcing-tesseract-ocr.html> (visitado 10-10-2018).

Google's Tesseract OCR engine is a quantum leap forward | Linux.com | The source for Linux information. URL:

<https://www.linux.com/news/googles-tesseract-ocr-engine-quantum-leap-forward> (visitado 10-10-2018).

Grumpy: Go running Python! | Google Open Source Blog. URL:

<https://opensource.googleblog.com/2017/01/grumpy-go-running-python.html> (visitado 10-10-2018).

gTTS — gTTS documentation. URL: <https://gtts.readthedocs.io/en/latest/> (visitado 10-10-2018).

Guido van Rossum Stepping Down from Role as Python's Benevolent Dictator For Life | Linux Journal. URL:

<https://www.linuxjournal.com/content/guido-van-rossum-stepping-down-role-pythons-benevolent-dictator-life> (visitado 10-10-2018).

- Gupta, Maya R, Nathaniel P Jacobson y Eric K Garcia. “OCR binarization and image pre-processing for searching historical documents”. En: *Pattern Recognition* 40.2 (2007), págs. 389-397.
- Hernández, Luis et al. “Estado del arte en tecnología del habla”. En: *Comunicaciones de Telefónica I+ D* 5 (1994), págs. 3-27.
- History and License — Python 3.7.2 documentation*. URL: <https://docs.python.org/3/license.html> (visitado 10-10-2018).
- Holley, Rose. “How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs”. En: *D-Lib Magazine* 15.3/4 (2009).
- How does OCR document scanning work? - Explain that Stuff*. URL: <https://www.explainthatstuff.com/how-ocr-works.html> (visitado 10-10-2018).
- Huang, Xuedong et al. *Spoken language processing: A guide to theory, algorithm, and system development*. Vol. 1. Prentice hall PTR Upper Saddle River, 2001, págs. 4-5, 27, 793, 796, 803-805, 807, 809.
- Image Optimization with Grooper*. URL: <https://grooper.com/image-optimization.html> (visitado 10-10-2018).
- Image Processing with the Computer Vision API | Microsoft Azure*. URL: <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/> (visitado 10-10-2018).
- Installation — pip 19.1.dev0 documentation*. URL: <https://pip.pypa.io/en/latest/installing/> (visitado 10-10-2018).
- Introduction to Google Drive API | Drive REST API | Google Developers*. URL: <https://developers.google.com/drive/api/v3/about-sdk> (visitado 10-10-2018).
- ISO/IEC 11172-3:1993 - Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio*. URL: <https://www.iso.org/standard/22412.html> (visitado 10-08-2018).
- ISRI. *Memorándum UAIP ISRI, El Salvador*. República de El Salvador, 2014.
- Issue 263 - tesseract-ocr - patch to enable hOCR output - An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google. - Google Project Hosting*. URL:

- <https://web.archive.org/web/20121113065732/http://code.google.com/p/tesseract-ocr/issues/detail?id=263> (visitado 10-10-2018).
- itertools* — *Functions creating iterators for efficient looping* — *Python 3.7.2 documentation*. URL: <https://docs.python.org/3/library/itertools.html> (visitado 10-10-2018).
- JPEG File Interchange Format: Summary from the Encyclopedia of Graphics File Formats*. URL: <https://www.fileformat.info/format/jpeg/egff.htm> (visitado 10-10-2018).
- Kapidakis, Sarantos, Cezary Mazurek y Marcin Werla. *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14-18, 2015, Proceedings*. Vol. 9316. Springer, 2015.
- Macintosh PICT: Summary from the Encyclopedia of Graphics File Formats*. URL: <https://www.fileformat.info/format/macpict/egff.htm> (visitado 10-10-2018).
- Makholm, Henning. *henning.makholm.net/xcftools/xcfspec-saved*. URL: <http://henning.makholm.net/xcftools/xcfspec-saved> (visitado 10-10-2018).
- Microsoft Windows Bitmap: Summary from the Encyclopedia of Graphics File Formats*. URL: <https://www.fileformat.info/format/bmp/egff.htm> (visitado 10-10-2018).
- Milyaev, Sergey et al. "Image binarization for end-to-end text understanding in natural images". En: *2013 12th International Conference on Document Analysis and Recognition*. IEEE. 2013, págs. 128-132.
- MINED. *Resultados de la Información Estadística de Instituciones de Educación Superior 2016*. Ministerio de Educación, República de El Salvador, 2016.
- Modos de color | GIMP, aplicaciones didácticas*. URL: http://www.ite.educacion.es/formacion/materiales/86/cd/m2/modos_de_color.html (visitado 10-10-2018).
- MPEG Press Release*. URL: https://web.archive.org/web/20110503174827/http://mpeg.chiariglione.org/meetings/kurihama91/kurihama_press.htm (visitado 10-08-2018).
- MPEG Press Release*. URL: https://web.archive.org/web/20100812034709/http://mpeg.chiariglione.org/meetings/london/london_press.htm (visitado 10-08-2018).

Más información acerca de los formatos de audio. URL:

https://online-audio-converter.com/es/help/audio_formats (visitado 10-08-2018).

OCR - Community Help Wiki. URL: <https://help.ubuntu.com/community/OCR> (visitado 10-10-2018).

OCR Introduction. URL: <http://www.dataid.com/aboutocr.htm> (visitado 10-10-2018).

ONU. *Convención sobre los Derechos de las Personas con Discapacidad.* Artículo 1. Organización de las Naciones Unidas, 2006.

OpenCV library. URL: <https://opencv.org> (visitado 10-10-2018).

Optical Character Recognition (OCR) – How it works. URL: <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/> (visitado 10-10-2018).

Opus Codec. URL: <https://www.opus-codec.org> (visitado 10-08-2018).

Pasamontes, José Colás. “Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión de habla continua en español.” En: *Estudios de Lingüística del Español* 12 (2001), pág. 1.

Pati, Peeta Basa y AG Ramakrishnan. “Word level multi-script identification”. En: *Pattern Recognition Letters* 29.9 (2008), págs. 1218-1229.

PEP 20 – The Zen of Python | Python.org. URL: <https://www.python.org/dev/peps/pep-0020/> (visitado 10-10-2018).

PEP 289 – Generator Expressions | Python.org. URL: <https://www.python.org/dev/peps/pep-0289/> (visitado 10-10-2018).

PEP 3000 – Python 3000 | Python.org. URL: <https://www.python.org/dev/peps/pep-3000/> (visitado 10-10-2018).

PEP 373 – Python 2.7 Release Schedule. URL: <https://legacy.python.org/dev/peps/pep-0373/> (visitado 10-10-2018).

PEP 466 – Network Security Enhancements for Python 2.7.x | Python.org. URL:

<https://www.python.org/dev/peps/pep-0466/> (visitado 10-10-2018).

Pillow — Pillow (PIL Fork) 6.0.0.dev0 documentation. URL:

<https://pillow.readthedocs.io/en/latest/> (visitado 10-10-2018).

pip - The Python Package Installer — pip 19.0.3 documentation. URL: <https://pip.pypa.io/en/stable/>
(visitado 10-10-2018).

pip · PyPI. URL: <https://pypi.org/project/pip/0.2/> (visitado 10-10-2018).

Platforms - OpenCV library. URL: <https://opencv.org/platforms/> (visitado 10-10-2018).

PNAIPD. Política Nacional de Atención Integral a las Personas con Discapacidad, República de El Salvador.
República de El Salvador, 2014.

PNG: Summary from the Encyclopedia of Graphics File Formats. URL:

<https://www.fileformat.info/format/png/egff.htm> (visitado 10-10-2018).

PortAudio - an Open-Source Cross-Platform Audio API. URL: <http://www.portaudio.com> (visitado
10-10-2018).

Pricing - Pricing | IBM. URL: <https://www.ibm.com/cloud/watson-speech-to-text/pricing>
(visitado 10-10-2018).

PrintToBraille Tool. ocr-gui-frontend. MILE Lab, Dept of EE, IISc. URL:

<https://web.archive.org/web/201412251115650/https://code.google.com/p/ocr-gui-frontend/> (visitado 10-10-2018).

PyAudio Documentation — PyAudio 0.2.11 documentation. URL:

<https://people.csail.mit.edu/hubert/pyaudio/docs/> (visitado 10-10-2018).

pyDatalog. URL: <https://sites.google.com/site/pydatalog/> (visitado 10-10-2018).

Python 3.0 Release | Python.org. URL: <https://www.python.org/download/releases/3.0/> (visitado
10-10-2018).

Python boss Guido van Rossum is stepping down from leading the project. URL:

<https://www.theinquirer.net/inquirer/news/3035842/python-boss-guido-van-rossum-steps-down-after-30-years> (visitado 10-10-2018).

[Python-Dev] SETL (was: Lukewarm about range literals). URL:

<https://mail.python.org/pipermail/python-dev/2000-August/008881.html>
(visitado 10-10-2018).

Python Imaging Library (PIL). URL: <http://www.pythonware.com/products/pil/> (visitado 10-10-2018).

Python Metaclasses: Who? Why? When? URL:

<https://www.webcitation.org/5lubkaJRc?url=http://www.python.org/community/pycon/dc2004/papers/24/metaclasses-pycon.pdf>
(visitado 10-10-2018).

Quilis, Antonio y Joseph A. Fernández. *Antonio Quilis y Joseph A. Fernández. Curso de fonética y fonología españolas. Para estudiantes anglo-americanos*. Consejo superior de investigaciones científicas, 1966.

Quilis, Antonio y Joseph A Fernández. “Curso de fonética y fonología españolas: para estudiantes angloamericanos”. En: (1969).

Raspberry Pi 3 Model B+ - Raspberry Pi. URL:

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (visitado 10-10-2018).

Ray Smith (2007). An Overview of the Tesseract OCR Engine. URL:

<https://github.com/tesseract-ocr/docs/blob/master/tesseractcdar2007.pdf>
(visitado 10-10-2018).

Recognize printed, handwritten text - Computer Vision - Azure Cognitive Services | Microsoft Docs. URL:

<https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-recognizing-text> (visitado 10-10-2018).

Reconocimiento de voz con Transcripción de voz de Cloud | API de Cloud Speech-to-Text | Google Cloud. URL:

<https://cloud.google.com/speech-to-text/> (visitado 10-08-2018).

Requests: HTTP para Humanos — documentación de Requests - 1.1.0. URL:

<http://docs.python-requests.org/es/latest/> (visitado 10-10-2018).

Resolución de una imagen | GIMP, aplicaciones didácticas. URL: http://www.ite.educacion.es/formacion/materiales/86/cd/m2/resolucin_de_una_imagen.html (visitado

10-10-2018).

- Riedl, Christoph et al. "Detecting figures and part labels in patents: competition-based development of graphics recognition algorithms". En: *International Journal on Document Analysis and Recognition (IJ DAR)* 19.2 (2016), págs. 155-172.
- RNPN. *Datos Estadísticos de Personas con Discapacidad a Marzo del 2016*. Registro Nacional de las Personas Naturales, República de El Salvador, 2016.
- Sampieri, Roberto Hernández. *Metodología de la investigación*; ed. McGraw-Hill: México, 1991.
- Schantz, Herbert F. *The history of OCR, optical character recognition*. Recognition Technologies Users Association Manchester, VT, 1982.
- Sezgin, Mehmet y Bülent Sankur. "Survey over image thresholding techniques and quantitative performance evaluation". En: *Journal of Electronic imaging* 13.1 (2004), págs. 146-166.
- Speech to Text API | Microsoft Azure*. URL: <https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/> (visitado 10-10-2018).
- SpeechRecognition · PyPI*. URL: <https://pypi.org/project/SpeechRecognition/> (visitado 10-10-2018).
- Suen, CY et al. "Future challenges in handwriting and computer applications". En: *3rd International Symposium on Handwriting and Computer Applications*. 1987.
- Tappert, Charles C., Ching Y. Suen y Toru Wakahara. "The state of the art in online handwriting recognition". En: *IEEE Transactions on pattern analysis and machine intelligence* 12.8 (1990), págs. 787-808.
- Tesseract: an Open-Source Optical Character Recognition Engine | Linux Journal*. URL: <https://www.linuxjournal.com/article/9676> (visitado 10-10-2018).
- Text to Speech API | Microsoft Azure*. URL: <https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/> (visitado 10-10-2018).
- The basic patter recognition and classification with openCV | Damiles Blog*. URL: <http://blog.damiles.com/2008/11/14/the-basic-patter-recognition-and-classification-with-opencv.html> (visitado 10-10-2018).

The Fourth Annual Test of OCR Accuracy. URL: http://www.expervision.com/wp-content/uploads/2012/12/1995.The_Fourth_Annual_Test_of_OCR_Accuracy.pdf (visitado 10-10-2018).

The History of Python: A Brief Timeline of Python. URL: <https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html> (visitado 10-10-2018).

The Making of Python. URL: <https://www.artima.com/intv/pythonP.html> (visitado 10-10-2018).

TIFF: Summary from the Encyclopedia of Graphics File Formats. URL: <https://www.fileformat.info/format/tiff/egff.htm> (visitado 10-10-2018).

Tiflotecnología – Web de la ONCE. 2018. URL: <https://www.once.es/servicios-sociales/tecnologiyrecursosadaptados/tiflotecnologia> (visitado 20-05-2018).

Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix. 2008. URL: <https://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/> (visitado 01-07-2018).

Train your Tesseract. URL: <http://trainyourtesseract.com> (visitado 10-10-2018).

Trier, Øivind Due y Anil K Jain. “Goal-directed evaluation of binarization methods”. En: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 12 (1995), págs. 1191-1201.

Vargiu, Eloisa y Mirko Urru. “Exploiting web scraping in a collaborative filtering-based approach to web advertising.” En: *Artif. Intell. Research* 2.1 (2013), págs. 44-54.

Vectorial | GIMP, aplicaciones didácticas. URL: <http://www.ite.educacion.es/formacion/materiales/86/cd/m2/vectorial.html> (visitado 10-10-2018).

Voice RSS - Free Text-to-speech (TTS) online service. URL: <http://www.voicerss.org/default.aspx> (visitado 10-10-2018).

Voice RSS - Text-to-speech (TTS) API documentation. URL: <http://www.voicerss.org/api/documentation.aspx> (visitado 10-10-2018).

Voice RSS - Text-to-speech (TTS) API Overview. URL: <http://www.voicerss.org/api/default.aspx>
(visitado 10-10-2018).

Watson Speech to Text - Overview | IBM Cloud. URL:

<https://www.ibm.com/cloud/watson-speech-to-text> (visitado 10-10-2018).

WAVE Audio File Format. URL:

<https://www.loc.gov/preservation/digital/formats/fdd/fdd000001.shtml>

(visitado 10-08-2018).

What is the Computer Vision API? - Computer Vision | Microsoft Docs. URL: [https:](https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home)

[//docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home](https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home)

(visitado 10-10-2018).

What is the point of an online interactive OCR text editor? | Fenno-Ugrica. URL:

<https://blogs.helsinki.fi/fennougrica/2014/02/21/ocr-text-editor/> (visitado

10-10-2018).

What's New in Python 2.0 — Python 3.7.2 documentation. URL:

<https://docs.python.org/3/whatsnew/2.0.html> (visitado 10-10-2018).

Xiph.org. URL: <https://xiph.org/vorbis/> (visitado 10-08-2018).

Xiph.org: Ogg. URL: <https://www.xiph.org/ogg/> (visitado 10-08-2018).