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

TRABAJO DE GRADO

SISTEMA INTÉRPRETE DE LENGUAS ORALES Y ESCRITAS A LENGUAJE DE SEÑAS SALVADOREÑO UTILIZANDO INTELIGENCIA ARTIFICIAL.

PARA OPTAR AL GRADO DE

INGENIERO(A) DE SISTEMAS INFORMÁTICOS

PRESENTADO POR

ANDREA SOFÍA HERNÁNDEZ CRUZ
WALTER ERNESTO RAMÍREZ CASTILLO
ESTUARDO ALEXANDER RAMOS SALAZAR
JAVIER SANTOS BONILLA

DOCENTE ASESOR

INGENIERO CARLOS ARTURO RUANO MORÁN

NOVIEMBRE, 2019

SANTA ANA, EL SALVADOR, CENTROAMÉRICA

UNIVERSIDAD DE EL SALVADOR

AUTORIDADES

M.Sc. ROGER ARMANDO ARIAS ALVARADO
RECTOR

DR. RAÚL ERNESTO AZCÚNAGA LÓPEZ
VICERRECTOR ACADÉMICO

ING. JUAN ROSA QUINTANILLA QUINTANILLA
VICERRECTOR ADMINISTRATIVO

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL
SECRETARIO GENERAL

LICDO. LUÍS ANTONIO MEJÍA LIPE
DEFENSOR DE LOS DERECHOS UNIVERSITARIOS

LICDO. RAFAEL HUMBERTO PEÑA MARIN
FISCAL GENERAL

FACULTAD MULTIDICIPLINARIA DE OCCIDENTE

AUTORIDADES

M.Ed. ROBERTO CARLOS SIGÜENZA CAMPOS
DECANO

M.Ed. RINA CLARIBEL BOLAÑOS DE ZOMETA
VICEDECANA

LICDO. JAIME ERNESTO SERMEÑO DE LA PEÑA
SECRETARIO

ING. DOUGLAS GARCÍA RODEZNO
JEFA DEL DEPARTAMENTO DE INGENIERA Y ARQUITECTURA

AGRADECIMIENTOS

Muestro mi agradecimiento de gran manera a Dios, mis padres y hermano. A mis abuelos por haber inculcado valores familiares y morales durante toda mi vida. Gracias a todos mis familiares por su apoyo incondicional, palabras de motivación, oraciones, etc.

Gracias a mis compañeros de trabajo de grado por su apoyo en nuestra idea, y su trabajo y esfuerzo día a día a pesar de limitaciones e inconvenientes. Gracias por confiar en mí y permitirme formar parte de este proyecto.

Agradezco a Cristina, mi maestra de Lenguaje de Señas Salvadoreño, a Lucy, intérprete de cada clase y a mis demás compañeros de clases de LESSA: David, Wendy, Harold, Larissa y Betty; por ayudarme en cada clase, y ser apoyo para el proyecto MIA.

Gracias Ingeniero Carlos Ruano por su ayuda en la asesoría de todo el proyecto, sus ideas y conocimientos ayudaron a crecer el proyecto. De igual manera Ingeniero Juan Carlos Peña por su apoyo durante el proceso, y por todos los conocimientos que pudo enseñarme.

Gracias a mis mejores amigos que apoyaron este proyecto: Kevin Chinchilla, Mónica Amaya, Kevin Figueroa, Esperanza Dueñas, Daniel Recinos y Luis Lemus, su apoyo durante todo el proyecto fue un pilar en mi mente para no rendirme.

ANDREA SOFÍA HERNÁNDEZ CRUZ

AGRADECIMIENTOS

Un trabajo de esta magnitud se hace pocas veces en la vida. Es preciso reconocer el esfuerzo de todas las personas involucradas en el proceso de manera directa o indirecta.

Debo agradecer a mi madre, por enseñarme la bondad y el amor hacia los demás. A mi padre, por enseñarme a pensar con precisión y siempre mirar hacia adelante. A mi hermano, por criticar todo lo que hago para ayudarme a siempre mejorar. A mis abuelos, que les dedico mi esfuerzo en la otra vida. Al resto de mi familia, que siempre miran con orgullo todo lo que hago.

Agradecer a los docentes de la universidad que nos ayudaron a caminar a través de la carrera, incluso los que al final no les caía muy bien. Agradecer especialmente al Ingeniero Juan Carlos Peña que al momento de escribir este agradecimiento nos sigue orientando, así como lo hizo durante la carrera. Al Ingeniero Carlos Ruano, que nos orientó como asesor durante este proceso.

A todos mis compañeros de carrera, que me demostraron que no era una competencia sino un logro colectivo.

A Andrea y Javier, que me acompañaron en la carrera desde el primer año y con quienes compartí los mejores proyectos. A Estuardo que, aunque empezamos a trabajar en los últimos años, nos acompañó durante el proyecto más importante.

A Jacque, que estuvo pendiente de todos los problemas que surgieron, me ayudó a tener paciencia e incluso a no dejar tan feo el proyecto. A Henri, que me alentó a entregar algo mejor que él.

A todos los demás que estuvieron pendientes del proceso y que nos dieron ánimos a seguir.
A todos, gracias.

WALTER ERNESTO RAMÍREZ CASTILLO

AGRADECIMIENTOS

Agradecer primeramente a mi madre Ersilda Salazar, mi padre Hector Ramos y mi hermano Hector Ramos por la paciencia y apoyo que mostraron a lo largo de la carrera, apoyándome tanto en las altas como en las bajas, y sin importar las circunstancias siempre me alentaron a seguir adelante.

A mis compañeros de carrera, especialmente a Fátima Escalante, Miguel Solorzano, Kevin Rodríguez, Erick Guirola, Francisco Morales, Rodolfo Olmos, Esperanza Dueñas y José Osorio con quienes con su apoyo y trabajo en equipo me enseñaron que el esfuerzo puede convertirse en grandes resultados.

A mis compañeros de tesis Andrea Hernández, Javier Santos, Walter Castillo, quienes a pesar de conocerlos en los últimos años de la carrera se convirtieron en gran parte de ella, y con quienes aprendí que cualquier problema puede verse desde distintas perspectivas para poder sobrepasarlo.

ESTUARDO ALEXANDER RAMOS SALAZAR

AGRADECIMIENTOS

En primer lugar, quiero agradecer a Dios, quien ha estado en todo momento y ha sido Él quien me ha permitido llegar hasta aquí, quien ha trazado todo este camino y me ha ayudado a salir delante en toda situación, sé muy bien que esta victoria, como otras que vendrán, son gracias a Él.

A mi mamá Sonia, gracias por estar siempre aquí, apoyándome, animándome, ayudándome e incluso empujándome para llegar aún más adelante. Le doy las gracias porque que ha dado todo de usted para lograr esta meta, como su amor, tiempo y paciencia. Una gran parte de lo que soy ahora es gracias a usted, y estoy feliz de lo que he llegado a ser.

A mi papa Moisés, muchas gracias por todo su apoyo en este camino, porque siempre ha estado ahí dispuesto a todo, apoyándome sin importar la situación. Ha sido un gran ejemplo para mí vida, he aprendido mucho de usted y sé que aún hay más. Gracias por esta gran oportunidad.

A mi hermana Annie, por seguirme de cerca en esta aventura, viviendo las alegrías, las tristezas y los enojos que surgieron, pero que sin importar la situación que sea, has estado apoyándome. Me has enseñado como las cosas pueden lograrse, aunque el camino se vea difícil. Siempre nos hemos apoyado y sé que seguirá siendo así para todo el trayecto que aún nos queda por delante, gracias hermanita.

Nuevamente agradezco Dios, por la familia que ha puesto en mi vida, no solo a mis padres y hermana, sino también a mis abuelos, tíos y primos, que de una u otra manera han estado ahí para mí. Sé que es Dios quien los ha puesto aquí, y que sin ustedes este camino no sería el mismo. Esta victoria es de todos, los amo.

Es igual de importante agradecer a mis compañeros de trabajo de grado, por el apoyo y las experiencias que pudimos vivir, no fue fácil y los problemas no faltaron, pero tampoco los buenos momentos y las risas, sin más que decir les doy las gracias y un “Lo logramos”.

JAVIER SANTOS BONILLA

INDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El presente proyecto describe la metodología y desarrollo de un sistema intérprete LESSA para ayudar a la comunicación hacia personas con discapacidad auditiva, a través del uso de tecnologías informáticas utilizando inteligencia artificial con base en el procesamiento natural del lenguaje. Desarrollando una herramienta de ayuda que pueda interpretar voz y texto, para reproducir su significado en Lenguaje de Señas Salvadoreño LESSA por medio de un modelo 3D animado.

El sistema en desarrollo pretende apoyar para la interpretación del Lenguaje de Señas Salvadoreño. Se pretende presentar el diseño de una herramienta básica destinada a interpretar el lenguaje, basado en las señas usadas en LESSA. Así mismo presentar señas reconocidas y utilizadas por las personas que se comunican mediante el lenguaje de señas, para sentar un precedente que incentive a que más proyectos sean iniciados y puedan seguir creciendo en relevancia.

A continuación, se describen los pasos seguidos en la investigación aplicada, ya que el principal propósito es caracterizar las dificultades de comunicación que afrontan las personas con discapacidad auditiva, ante la limitante del conocimiento de Lenguaje de Señas Salvadoreño (LESSA) en el país. Dicho esto, se describe de las diferentes tecnologías informáticas seleccionadas para poder desarrollar la herramienta, por qué fueron seleccionadas y cómo trabajan en conjunto para poder llevar a cabo la interpretación.

También, se presenta documentación sobre el análisis y la clasificación de los resultados obtenidos, y, finalmente, se plantean conclusiones y recomendaciones que el grupo investigador considera pertinentes, para que la investigación funcione como guía y sea de interés para los lectores y futuras investigaciones que se realicen en áreas afines.

CAPITULO I
GENERALIDADES

ANTECEDENTES

Aplicaciones de Informática en el área de Lengua de Señas

Manitas 1.0. En el campo de la informática, en el año 2013 se dio a conocer una aplicación móvil llamada “Manitas 1.0” también conocida como “LESSA móvil” desarrollada para dispositivos Android. Dicha aplicación buscaba entablar una conversación con personas con discapacidad auditiva y, así mismo usarse como una plataforma para aprender la base del sistema LESSA. La aplicación consiste en un teclado virtual que muestra cada uno de los gestos que se necesitan para comunicarse con una persona con discapacidad auditiva. Aunque se expresó que se desea continuar con el desarrollo de la aplicación agregando un modelo 3D para realizar las señas.

Sign’n. Un grupo de jóvenes mexicanos del estado de Jalisco en 2017, comenzaron la creación de una aplicación para dispositivos móviles que permitiría traducir texto y voz a la Lengua de Señas Mexicana. Briana Osorio, fue la joven que inició el proyecto como una empresa emergente, explica que el objetivo es que la aplicación funcione como herramienta para cualquier persona que desee aprender lenguaje de señas o establecer una conversación con una persona con discapacidad auditiva. El funcionamiento consiste que desde un dispositivo móvil el usuario podrá dictar o escribir una palabra o frase que será interpretada y traducida mediante inteligencia artificial con la ayuda de un personaje virtual, se mostrará una imagen de una persona realizando las señas que corresponda a la frase o palabra.

La app facilita la inclusión de las personas con discapacidad auditiva, en su entorno social y familiar, pero también en el ámbito laboral pues, cualquier persona podría utilizar la plataforma de manera gratuita para facilitar el diálogo con ese sector de la población. Este proyecto cuenta con la colaboración de expertos en modelado y animación 3D, un lingüista y personas con discapacidad auditiva como asesores. El desarrollo de la aplicación es apoyado por miembros de la Asociación de Silentes de Jalisco y continúa en desarrollo a fecha de realización de este documento.

HETAH.net. Hetah.net es un sitio traductor a lenguaje de señas y braille. El sitio es administrado por la Fundación Hetah (Herramientas Tecnológicas para la Ayuda Humanitaria) de Colombia. El sitio web cuenta con un personaje animado llamado Iris, que se encarga de hacer visibles los textos que se ingresan al traductor. El software cuenta con un gran número de palabras que va incrementando gradualmente para mejorar su calidad.

Para el año 2013, contaba con 2949 señas. También el sitio ha incorporado una aplicación para traducir de textos a lenguaje braille que se pueden imprimir para realizar las perforaciones típicas de este tipo de escritura. Esta aplicación realiza una búsqueda de señas analizando oraciones en su estructura gramatical, haciendo énfasis en el país origen y realizando un deletreo en las palabras de las cuales no encuentra equivalencia. La fundación asegura que la herramienta en ningún momento pretende reemplazar al intérprete humano, sin embargo, es una ayuda más.

Signalo. Signalo es una aplicación multiplataforma gratuita que traduce de manera automática texto y voz a Lengua de Señas Argentina a través de un personaje diseñado con animación computada al que bautizaron “Sigu”. El proyecto fue diseñado por la Asociación Civil Señas en Acción, que se dedica a difundir la lengua de señas para minimizar las barreras de exclusión que dificultan la comunicación para las personas sordas o con dificultades auditivas. La herramienta se basa en un diccionario virtual, en donde se asocia cada palabra con su representación en la lengua de señas. Con el aporte de la comunidad sorda, el diccionario se enriquecerá de manera colaborativa.

La iniciativa fue seleccionada en el marco del Desafío Google 2017 y estaba sometida la elección de la ciudadanía, que tenía que elegir a través de su voto, si participaría de la final regional que se llevó a cabo en México. Por llegar a la final nacional recibieron 350,000 dólares, más el apoyo de Google. Esta organización está conformada por un equipo de profesionales interdisciplinarios compuesto por personas con discapacidad auditiva, intérpretes de Lengua de Señas, docentes y psicólogos. Es la primera vez que se realiza en América Latina el Desafío Google.org, que tiene el objetivo de fomentar proyectos de innovación con impacto social.

PLANTEAMIENTO DEL PROBLEMA

Según la Unión Europea, la inclusión social se define como el proceso que asegura que aquellas personas que están en exclusión social, tengan las oportunidades y recursos necesarios para participar en la vida económica, social y cultural disfrutando un nivel de vida y bienestar que se considere normal en la sociedad en la que ellos viven.

En El Salvador, la Ley de Equiparación de Oportunidades para las Personas con Discapacidad, aprobada según Decreto Legislativo No. 888 por la Asamblea Legislativa y publicada en el Diario Oficial No. 226 del 1 de diciembre del año 2000, en su capítulo IV sobre Educación, se contempla ampliamente la responsabilidad del Estado y sus instituciones de facilitar, en condiciones de igualdad, la integración de la población con alguna discapacidad. De igual modo, está contemplada en la Línea de Acción No. 4.3.4 la Medida de la Política 3, que insta a "promover que las Universidades destinen recursos de investigación y enriquezcan sus cátedras en el ámbito de la atención de personas con Necesidades Educativas Especiales (NEE)".

A pesar que en El Salvador existen diversas leyes y organizaciones que buscan asegurar y mejorar la integración de la población con alguna discapacidad, que incluye a las personas con discapacidad auditiva, son pocas las instituciones que realmente invierten esfuerzos en su cumplimiento. Aún más notorio es la escasez del desarrollo de proyectos que utilicen las tecnologías de la información para crear herramientas y recursos que puedan utilizarse como parte de soluciones para beneficiar de alguna manera la situación actual de esta población minoritaria.

Es por estas razones que se vuelve relevante el uso de las tecnologías de la información para la creación de herramientas que puedan ser utilizadas como recursos para la implementación de distintos proyectos en beneficios hacia las personas con discapacidad auditiva, a fin de mejorar su integración a la sociedad.

De esta forma hacer notar las ventajas y oportunidades que presenta el apoyar en el uso y desarrollo de las tecnologías informáticas y sentar bases que abran camino e interés hacia la realización más proyectos y herramientas de esta índole. Todo esto a su vez puede facilitar y por ende aumentar la aplicación y mejora de proyectos en las distintas instituciones país.

Por lo tanto, lo que se busca con el proyecto MIA, es probar que las nuevas tecnologías son la herramienta más adecuada para brindar ayuda a personas con discapacidad auditiva, potencialmente para ser implementada a futuro en lugares donde para ellos sea un beneficio neto.

OBJETIVOS

General

- Diseñar software con la capacidad de interpretar el habla y texto en español, para traducirlos al lenguaje de señas salvadoreño, y mostrar la traducción en gestos a través de un modelo 3D en tiempo real.

Específicos

- Demostrar que se puede hacer uso de las tecnologías de la información para ayudar en la inclusión de las personas con discapacidad auditiva.

- Diseñar una solución que pueda ser utilizada tanto en procesos que sirvan para mejorar la inclusión de las personas con discapacidad auditiva como también una solución que permita apoyar el aprendizaje del lenguaje de señas salvadoreño.

- Ayudar al proceso de interacción en el ambiente cotidiano para personas con discapacidad auditiva y oral, y así poder establecer una vía rápida y sencilla de comunicación en una dirección hacia las personas con discapacidad auditiva y oral.

JUSTIFICACIÓN

En El Salvador la comunidad sorda como minoría lingüística, todavía no es incluida de manera concreta, incluso entre las mismas comunidades de personas sordas. A esto se le suma las necesidades educativas, así como las necesidades de desenvolvimiento de cualquier persona en el ambiente cotidiano, dentro un sistema social que les asigna una etiqueta de "discapacitados".

Igualmente se debe resaltar como en los artículos N° 9, 11, 12, 44 y 90 de la Ley de Cultura, en las cuales en resumen se excluye a la minoría lingüística sorda. Dichos artículos hacen caso omiso de esta minoría como si se tratase de una presencia inexistente dentro del país. Inclusive a pesar de la existencia de leyes que han sido dictadas para abogar a favor de estas comunidades, son pocas las instituciones que realmente tratan de cumplirlas.

Las instituciones que sí dirigen esfuerzos por tratar de cumplirlas, lo logran de manera limitada y esto es debido a que la mayor parte del tiempo los recursos disponibles son escasos. Un ejemplo son los hospitales que poseen un intérprete de lenguaje de señas, al igual que los demás empleados tiene su jornada laboral definida y solo pueden atender personas sordas dentro de horas específicas, una vez fuera del horario, se hace un problema muy difícil poder dar la debida atención a un paciente sordo.

A pesar de todos estos obstáculos presentes, la comunidad lingüística sorda no se ha mantenido absenta de mostrar esfuerzos para ser reconocidos y aceptados, organizaciones como FUNDASORDO, Asociación Salvadoreña de Sordos, Fundación Manos Mágicas, que han surgido de la unión de los esfuerzos para luchar por los derechos de todas las personas con discapacidad auditiva, siendo uno de sus mayores esfuerzos el fomentar la aceptación y hacer crecer la capacitación en lenguas de Señas Salvadoreña LESSA.

Es por estas razones descritas que como grupo de trabajo de grado de la Universidad de El Salvador hemos propuesto el diseño de un intérprete de voz al lenguaje de señas salvadoreño LESSA. Con este proyecto pretendemos proponer una herramienta, la cual pueda facilitar la interacción con las personas con dificultades auditivas, servir como apoyo en la difusión y entendimiento del Lenguaje de Señas Salvadoreño. Siendo una de las razones más fundamentales del desarrollo de este proyecto, el poder contribuir con los esfuerzos invertidos por los diferentes grupos de comunidades sordas por conseguir desarrollo pleno en la vida cotidiana en el país.

Se ha decidido seguir un enfoque de trabajo para establecer cimientos en los cuales puede expandirse por los interesados en apoyar la causa, de igual forma, poder ser utilizado en favor de la ayuda de otras comunidades de personas que serán beneficiadas con las tecnologías usadas.

Dentro de las razones más importantes que impulsa al desarrollo del proyecto, se busca demostrar la aplicación de las tecnologías informáticas para su implementación, y beneficiar a las personas con discapacidad auditiva, incitando al desarrollo de aplicaciones del área de la informática y de la misma manera la educación.

ALCANCES

Con este proyecto se busca diseñar una herramienta que ayude a demostrar cómo el uso de las tecnologías informáticas pueden utilizarse en beneficio hacia las personas con personas con discapacidad auditiva y vocal, lo cual se pretende lograr mediante la interpretación de voz o texto y realizar su respectiva traducción hacia un Lenguaje de Señas, esto se logra mediante el ingreso de datos (voz o texto), el cual es procesado y enviado hacia MIA (Modelo Intérprete Animado) para proveer la salida con la interpretación visual correspondiente (LESSA).

Esta herramienta pretende ser únicamente una alternativa y/o apoyo para la interpretación hacia el Lenguaje de Señas, mas no el sustituir a los intérpretes humanos. Este proyecto busca presentar una solución básica hacia la tarea de interpretación rápida del lenguaje y establecer estándares para futuros desarrollos y mejoras.

Los alcances del proyecto comprenden:

- ✓ Cubrir un mínimo de 250 señas del Lenguaje de Señas Salvadoreño básico, mediante la realización de animaciones sobre un modelo 3D.
- ✓ El módulo NLP será capaz de procesar 50 palabras por segundo como mínimo.
- ✓ El 85% de las señas sean comprendidas por el usuario.
- ✓ Integración de la solución con un software de mensajería de terceros.
- ✓ Ofrecer servicio de NLP para un mínimo de 5 clientes simultáneos.
- ✓ Presentar una aplicación funcional.

LIMITACIONES

- ✓ Debido a la diversidad de señas empleadas que aún no se encuentran registradas en el Lenguaje de Señas Salvadoreñas (LESSA) es difícil recopilar todas estas, solamente se contará con un anexo de las palabras aún no registradas más utilizadas.

- ✓ No se puede contar con el software para la enseñanza particular o compleja, de material académico ya que cada área de enseñanza posee sus señas y expresiones propias para ser impartidos, el software está basado en diccionario LESSA, enfocado a una interacción en un ambiente cotidiano.

- ✓ El Lenguaje de Señas Salvadoreño está sujeto a adaptaciones debido a los diversos cambios culturales y tecnológicos en el transcurso del tiempo.

- ✓ Estará diseñado como un traductor de una sola vía, de voz o texto hacía lenguaje de señas.

- ✓ Debido a la fuente del diccionario LESSA es posible no poder encontrar equivalencias con palabras de carácter moderno, así como neologismos.

- ✓ No se toma como un sustituto a un intérprete humano propio del Lenguaje de Señas Salvadoreña.

DISEÑO METODOLÓGICO

Investigación aplicada

La expresión "Investigación Aplicada" se popularizó durante el siglo XX para referirse al tipo de estudios científicos orientados a resolver problemas de la vida cotidiana y a controlar situaciones prácticas. Este tipo de investigación se centra en la solución de problemas delimitados por un contexto especifico, en donde se busca la adquisición y utilización de los conocimientos de una o varias áreas especializadas con el propósito principal de su aplicación a fin de mejorar la problemática. Para Murillo (2008), la investigación aplicada recibe el nombre de “investigación práctica o empírica”, que se caracteriza principalmente, por la búsqueda de aplicación o utilización de los conocimientos adquiridos, a la vez que se adquieren otros, durante la aplicación de estos mismos.

Este tipo de investigación se centra principalmente en el análisis y solución de problemas cotidianos, con el especial énfasis en áreas de los social. Por lo general se parte de necesidades sociales y/o de otras áreas, se procede a la implementación de la investigación y finalmente se obtiene un resultado o solución, que puede brindarse en distintas formas como innovación tecnológica, un producto y otras.

El presente proyecto pretende utilizar tecnologías e inteligencia artificial, para aplicarlas sobre problemas de comunicación de las personas con discapacidad auditiva y oral. En base a los diferentes problemas observados, tanto sociales como personales que las personas con discapacidad auditiva enfrentan en lo cotidiano, se parte con la iniciativa de brindar una herramienta, que, mediante una correcta aplicación, pueda ser implementada como ayuda en el día a día de estas personas. De la idea general de diseñar un intérprete que facilite la comunicación de una vía, comenzar con la abstracción de las partes esenciales que servirán como las bases principales para que el proyecto pueda llevarse a cabo.

Según la necesidad del proyecto se hará uso de las siguientes técnicas:

Entrevistas

- Entrevistas con persona con discapacidad auditiva.
- Entrevista con intérpretes del lenguaje de señas LESSA.

Fuentes de información

- Publicaciones hechas por los diversos grupos que apoyan a la comunidad lingüística sorda.
- Documentos oficiales publicados por las organizaciones mencionadas.
- Investigaciones realizadas en las comunidades lingüísticas sordas del país.

Método de desarrollo

Para el desarrollo del sistema, se utilizarán las primeras cinco fases del ciclo de vida para el desarrollo de sistemas informáticos, siendo de esta manera las siguientes:

- Investigación preliminar.
- Determinación de los requerimientos.
- Diseño del sistema.
- Desarrollo del sistema.
- Pruebas.

Investigación preliminar.

Esta primera fase abarca toda la información recopilada por lo antecedentes, marco teórico y el proporcionado por las entrevistas y las distintas fuentes de información recolectada.

Determinación de requerimientos.

Para la determinación de requerimientos se hará uso de la información de la investigación preliminar, sumando a esta los objetivos, alcances y limitantes planteadas, para que el producto resultante cumpla con las características propuestas. La determinación de los requerimientos se realizará mediante el uso de una matriz de toma de decisiones.

Diseño del sistema.

Para el diseño del sistema se utilizarán esquemas de casos de uso que guiarán posteriormente en la parte del desarrollo de los diseños de interfaces y reglas de negocios, tomando en cuenta todos los requisitos anteriormente establecidos.

Desarrollo del sistema.

Para el desarrollo del sistema se utilizarán técnicas de desarrollo estructurado, el cual, al descomponer el problema en partes funcionales pequeñas, permitirán el desarrollo por medios de módulos que posteriormente conformarán el proyecto completo en conjunto.

Pruebas.

Para la prueba del sistema se hará uso de las pruebas de usuario.

CAPITULO II
MARCO TEÓRICO

COMUNICACIÓN

El término comunicación procede del latín *communicare* que significa “hacer a otro partícipe de lo que uno tiene”. La comunicación es la acción de comunicar o comunicarse, es el proceso por el que se transmite y recibe una información. Pero, para que un proceso de comunicación se lleve a cabo, es indispensable la presencia de algunos elementos: emisor, receptor y un canal de comunicación, que puede ser oral o escrito.

La comunicación es un modo de intercambio de información entre un emisor y un receptor, en el cual el primero transmite el mensaje y el segundo interpreta y produce una respuesta, de ser necesario. En lo que se refiere a los seres humanos, la comunicación es una actividad psíquica propia, derivada del pensamiento, el lenguaje y del desenvolvimiento de las capacidades psicosociales de las relaciones.

A través de la palabra, comunicamos nuestros pensamientos y sentimientos, y establecemos relaciones personales con nuestros familiares, amigos, en la escuela, en el trabajo, y en la comunidad.

Por proceso de comunicación se conoce, al conjunto de actividades asociadas a un intercambio de datos. En el marco de esta actividad existen códigos compartidos entre el emisor y el receptor, es decir, combinaciones de reglas que permiten que se concrete la comunicación, y que a través de canales transmiten el mensaje, éstos pueden ser físicos o a través de un soporte digital. Los obstáculos que se presentan en el proceso de comunicación se les denomina ruidos.

Elementos de la Comunicación

Existen una serie de elementos que hacen posible la comunicación:

- Emisor: Es el encargado de iniciar la acción de comunicar y emitir cierta información.

- Receptor: Es quien recibe el mensaje o información, ajustándolo e interpretando según los signos lingüísticos. Luego de su interpretación está en capacidad de emitir una respuesta al emisor.
- Mensaje: Es el contenido que se transmite.
- Canal: Es el medio a través del cual es transmitida la información o mensaje.
- Código: Son signos del idioma, combinados por el emisor según sea el mensaje que desea transmitir.
- Contexto: Es el medio o entorno que rodea al emisor y receptor al momento de realizarse la comunicación.

Tipos de Comunicación

Comunicación Verbal. La comunicación verbal es el tipo de comunicación en la que se utilizan signos lingüísticos en el mensaje. Los signos son en su mayoría arbitrarios y/o convencionales, ya que expresan lo que se transmite y además son lineales; cada símbolo va uno detrás de otro. La comunicación verbal está formada esencialmente por:

Comunicación Oral. Esencialmente se realiza a través del habla, bien sea de una manera personal o por medio de dispositivos como, teléfonos, videos, voz por internet, radio y televisión.

Comunicación Escrita. Se utilizan signos o símbolos escritos, bien sea a mano o impresos, pueden ser transmitidos a través de correos electrónicos, cartas, notas, entre otros.

Comunicación no Verbal. Esta forma de comunicación se realiza sin la emisión de palabras o mensajes, solo utilizando el lenguaje corporal, las expresiones faciales, la postura. Es esencialmente el lenguaje corporal del hablante. Este tipo de lenguaje está formado por ciertos elementos:

- Apariencia.
- Altavoz: se refiere a la ropa, pulcritud, etc.

- Entorno: iluminación, tamaño de la habitación, mobiliario y decoración.
- Expresiones: posturas, faciales del lenguaje corporal y gestos.
- Sonidos: tonos de voz, velocidad de voz y volumen.

Un mensaje puede estar codificado en lenguajes diferentes como:

- Sistemas de signos como el lenguaje corporal de gestos, sonidos, proximidad y sonidos sin palabras,
- Idiomas como español, portugués, inglés, francés, etc.,
- Códigos con significado como por ejemplo los colores del semáforo.

DISCAPACIDAD

La discapacidad es una condición bajo la cual algunas personas presentan alguna deficiencia física, mental, intelectual o sensorial que a largo plazo afectan la forma de interactuar y participar plenamente en la sociedad.

La Convención Internacional sobre los Derechos de las Personas con Discapacidad, aprobada por la ONU en 2006, define de manera general a quien o quienes poseen una o más discapacidades como persona con discapacidad. Términos como “discapacitados”, “ciegos”, “sordos”, etc., aun siendo correctamente empleados, pueden ser considerados despectivos o peyorativos, ya que para algunas personas estos términos “etiquetan” a quien padece la discapacidad, lo que puede interpretarse como discriminación. Por lo tanto, para evitar conflictos de tipo semántico, se usan a veces las formas: “personas con diversidad funcional” o “personas con discapacidad”.

La Clasificación Internacional del Funcionamiento de la Discapacidad y de la Salud (CIF) de la Organización Mundial de la Salud, distingue entre las funciones del cuerpo (fisiológico o psicológico, visión) y las estructuras del cuerpo (piezas anatómicas, ojo y estructuras relacionadas). La debilitación en estructura o la función corporal se define como

participación de la anomalía, del defecto, de la pérdida o de otra desviación significativa de ciertos estándares generalmente aceptados de la población, que pueden fluctuar en un cierto plazo. La actividad se define como la ejecución de una tarea o de una acción.

El CIF enumera 9 amplios dominios del funcionamiento que pueden verse afectados:

- Aprendiendo y aplicando conocimiento
- Tareas y demandas generales
- Comunicación
- Movilidad
- Cuidado en sí mismo
- Vida doméstica
- Interacciones y relaciones interpersonales
- Áreas importantes de la vida
- Vida de la comunidad, social y cívica

En el aspecto médico se ve a la discapacidad como una enfermedad, causando directamente una deficiencia, el trauma, o la otra condición de la salud que por lo tanto requiere la asistencia médica sostenida proporcionada bajo la forma de tratamiento individual por los profesionales.

Ciertos organismos relacionados con la diversidad funcional han intentado acuñar nuevos términos, en busca de una nueva visión social de este colectivo, la Organización Mundial de la Salud (OMS), promocionó la Clasificación Internacional del Funcionamiento de la Discapacidad y de la Salud (CIF) Suiza (2001), sustituyendo la anterior Clasificación Internacional de Deficiencias, Discapacidades y Minusvalías (CIDDM) de 1980.

- Deficiencia por déficit en el funcionamiento, la pérdida o anormalidad de una parte del cuerpo o de una función fisiológica o mental, la contextualiza a una desviación significativa de la mediana estandarizada de la población;

- Discapacidad por limitación en la actividad referida a las dificultades en la ejecución calificadas en distintos grados que supongan una desviación importante en cantidad y calidad en relación en una persona sin alteración de salud;

- Minusvalía por la restricción en la participación, son problemas que un individuo puede experimentar en su implicación en situaciones vitales, la participación está determinada por lo esperado de un individuo sin discapacidad en una determinada cultura o sociedad.

Tipos de discapacidad

Motriz. Se caracteriza por la disminución parcial o total de la movilidad de uno o más miembros del cuerpo, lo que se traduce en una dificultad o impedimento a la hora de realizar diversas tareas motoras, en especial las de la motricidad fina. Esto producto de que la discapacidad puede llegar a generar en la persona movimientos incontrolados, temblores, dificultad de coordinación, fuerza reducida, entre otros.

Discapacidad visual. En el mundo existen aproximadamente 280 millones de personas que sufren de discapacidad visual, siendo casi 40 millones ciegas y más de 240 de baja visión.

De los tipos de discapacidad la visual se divide en dos, y la primera y más popular es la pérdida total de la visión o ceguera, y la menos conocida es la disminución parcial, que es de hecho, la más frecuente.

Discapacidad intelectual. La discapacidad intelectual implica una serie de limitaciones en las habilidades que la persona aprende para funcionar en su vida diaria y que le permiten responder ante distintas situaciones y lugares. La discapacidad intelectual se expresa en la relación con el entorno. Por tanto, depende tanto de la propia persona como de las barreras u obstáculos que tiene alrededor. Es importante señalar que la discapacidad intelectual no es una enfermedad mental.

Auditiva. Pérdida total o parcial de la percepción de los sonidos. Se dice que una persona es sorda cuando su deficiencia auditiva es total o profunda, hipoacúsica si su pérdida de la audición es parcial y su audición puede mejorar con el uso de dispositivos electrónicos como los audífonos.

Tipos de discapacidad auditiva. Normalmente las causas de la discapacidad auditiva son: genética, adquirida (es decir, que se adquiere en algún momento de la vida, como un accidente), congénitas, que son las prenatales, perinatales, por traumatismo, enfermedad, larga exposición al ruido o por toma de medicamentos demasiado combativos para el nervio auditivo, lo que lleva a la pérdida de audición.

La discapacidad se determina según el momento de la adquisición de la pérdida auditiva, la localización de la lesión y el grado de pérdida auditiva. La importancia de determinar ciertas características de la persona con discapacidad auditiva permite comprender mejor el porqué de su sordera.

- Según la adquisición de la pérdida auditiva.

    - Sordera prelocutiva: Es antes de que se haya desarrollado el lenguaje, es decir antes de los tres años de vida.
    - Sordera postlocutiva: Las personas que adquirieron la sordera luego del desarrollo del lenguaje oral.

- Según la localización de la lesión.

    - Conducción o de transmisión: Alteraciones causadas por enfermedades u obstrucciones en el oído externo y medio, frenando el paso de las ondas sonoras.
    - Percepción o neurosensorial: Lesiones en el oído interno o en la vía nerviosa auditiva y los daños son irreversibles.
    - Mixta: La causa es conductiva y de percepción, los problemas se presenten en el oído externo o medio como en el interno. (Mendieta, 2016)

Desigualdad para las personas con discapacidad

La persona con discapacidad es un sujeto con derecho. El ejercicio de sus derechos se ve dificultado y vulnerado por su condición. En la mayoría de países, las personas con discapacidad pueden solicitar el reconocimiento de su condición y, a partir de cierto grado, un certificado de discapacidad, que les permite acceder a una serie de ventajas. El término minusvalía se considera peyorativo fuera del ámbito legal.

La mayor desigualdad se da en la desinformación de la discapacidad que tiene enfrente las personas sin discapacidad y el no saber cómo desenvolverse con la persona con discapacidad, logrando un distanciamiento no querido. La sociedad debe eliminar las barreras para lograr la equidad de oportunidades entre personas con discapacidad y personas sin discapacidad. Para lograr esto, tenemos las tecnologías de apoyo.

La Convención Internacional sobre los Derechos de las Personas con Discapacidad (CRPD, por su sigla en inglés) es un instrumento internacional de derechos humanos de las Naciones Unidas destinado a proteger los derechos y la dignidad de las personas con discapacidad. Partes en la Convención tienen la obligación de promover, proteger y garantizar el pleno disfrute de los derechos humanos de las personas con discapacidad y garantizar que gocen de plena igualdad ante la ley.

TECNOLOGÍAS DE APOYO

Se llama tecnología de apoyo o de ayuda a todo tipo de equipo, objeto, sistema, producto, máquina, instrumento, programa o servicio que puede ser usado para suplir, aumentar, mantener, compensar o mejorar las capacidades funcionales de las personas con impedimento o discapacidad (motriz, sensorial o cognitiva). También es llamada tecnología de adaptación o de ayuda para la vida independiente, ya que les facilita a los individuos que las utilizan, llevar a cabo tareas que antes eran incapaces de cumplir o tenían grandes dificultades para realizarlas. Algunos ejemplos son el bastón blanco, el andador, la silla de ruedas, perros de asistencia, etc.

El uso de las tecnologías como medio para incrementar, mantener o mejorar las capacidades funcionales de los individuos es una práctica común en el ámbito de la intervención con personas con discapacidad. García Viso y Puig de la Bellacasa (1988) definen las ayudas técnicas como utensilios para que el individuo pueda compensar una deficiencia o discapacidad sustituyendo una función o potenciando los restos de las mismas.

La naturaleza de las tecnologías de apoyo es tan variada que se han propuesto, para ello, distintas filosofías de clasificación. La Convención Internacional sobre los Derechos de las Personas con Discapacidad de Naciones Unidas, que entró en vigor el 3 de mayo de 2008, establece que hay que garantizar a las personas con discapacidad la igualdad de acceso a las tecnologías de la información y la comunicación (TIC).

Las tecnologías de la información y las comunicaciones afectan en gran manera las formas en que las personas pueden aprender, realizar tareas o actividades, recrearse, comunicarse entre muchos otros aspectos. Con el rápido avance en el desarrollo de nuevas tecnologías es cada vez más común ver el uso de estas en todos los ámbitos, incluso en actividades las cotidianas, con el objetivo de facilitar, automatizar y mejorar las maneras en cómo realizamos las actividades.

Por todo esto es muy importante adaptar el uso de las tecnologías actuales y crear nuevas, que ayuden a satisfacer las necesidades de personas con discapacidad y utilizarlas como tecnologías asistivas en ámbitos educacionales, cotidianos y terapéuticos, tal como lo dicen Albert Cook y Janice Polgar (Assistive Technologies: Principles & Practices, 2015).

LENGUA DE SEÑAS

Lengua de señas o lengua de signos, es una lengua natural de expresión y configuración gesto-espacial y percepción visual, por la cual las personas con discapacidad auditiva pueden establecer un canal de comunicación con su entorno social, ya sea conformado por otros individuos sordos o por cualquier persona que conozca la lengua de señas. Mientras que con el lenguaje oral la comunicación se establece en un canal vocal-auditivo, el lenguaje de señas lo hace por un canal gesto-viso-espacial.

Las lenguas de señas modernas, al igual que las lenguas orales, están sujetas al proceso universal de cambio lingüístico, que hace que evolucionen con el tiempo y eventualmente, una misma lengua puede evolucionar en lugares diferentes hacia variedades diferentes. De hecho, muchas de las lenguas modernas de señas pueden ser clasificadas en familias:

- Lenguas originadas en la antigua lengua de señas de Kent, usada durante el siglo XVII, que dio lugar a la lengua de señas usada en Martha's Vineyard (Massachusetts), que influyó de manera importante en la lengua de señas americana (ASL).
- Lenguas originadas en la antigua lengua de señas francesa. Estas lenguas se remontan a las formas estandarizadas de las lenguas de señas usadas en España, Italia y Francia desde el siglo XVIII en la educación de los sordos. En concreto, la antigua lengua de señas francesa se desarrolló en el área de París, gracias a los esfuerzos del abad Charles Michel de l'Épée en su escuela de sordos. En tiempos modernos esta lengua ha dado lugar a otras varias, como la lengua de señas americana (ASL), la

lengua de señas mexicana (LSM), la moderna lengua de señas francesa (LSF), la lengua de señas italiana (LIS), la lengua de señas de Irlanda (IRSL) y las lenguas de señas ibéricas, derivandose a dos lenguas diferentes con cierta inteligibilidad mutua: La lengua de señas española (LSE), la lengua de señas catalana (LSC).

- Lenguas originadas en la lengua de señas británica (BSL), que se diversificó durante el siglo XIX dando lugar a la lengua de señas australiana (Auslan), la lengua de señas de Nueva Zelanda (NZSL) y la lengua de señas de Irlanda del Norte (NIRSL).

- Lenguas originadas en la lengua de señas alemana (DGS), se considera que está relacionada con la lengua de señas de la Suiza alemana (DSGS), la lengua de señas austríaca (ÖGS) y probablemente la lengua de señas israelí (ISL).

Lengua y lenguaje

En muchos estudios sobre lenguas de comunidades sordas es posible encontrar cualquiera de las dos palabras Lengua o Lenguaje al inicio del nombre de las lenguas de los Sordos. Así, por ejemplo, se habla de "Lengua de Señas Venezolana", pero también de "Lenguaje de Señas Nicaragüense".

Ambas palabras son términos de la teoría lingüística: "Lengua" designa un específico sistema de signos que es utilizado por una comunidad concreta para resolver sus situaciones comunicativas. "Lenguaje", por su parte, designa una capacidad única de la especie humana para comunicarse a través de sistemas de signos. Según ello, "lenguaje" refiere a una habilidad que heredamos genéticamente y que nos permite constituir sistemas lingüísticos y usarlos en la estructuración nuestra cultura. Tales sistemas, que no son aportados por la naturaleza, sino por la evolución de las culturas humanas, son las "lenguas".

El lenguaje, definido entonces como capacidad humana de crear y usar las lenguas de modo natural, es patrimonio común de sordos y oyentes, y subyace tanto a las lenguas habladas como a las señas. El vocablo "idioma" podría sustituir a "lengua" en este sentido, ya que poseen

valores muy similares, y es con “idioma” como suele codificarse popularmente un cercano equivalente al sentido que “lengua” tiene en la teoría lingüística. Pero atendiendo al hecho de que la mayoría de investigadores occidentales hayan optado por “lengua”, se nos impone seguir en español esa tendencia general para nombrar a los códigos viso-espaciales de las comunidades sordas.

Lingüísticamente, pareciera resultar más apropiado usar el término “lengua” que “lenguaje” para designar la lengua de una comunidad de sordos particulares, ya que esta es una versión más, otra actualización histórica de la capacidad universal del lenguaje.

Gestual

Este adjetivo está relacionado etimológicamente con la idea de expresar ciertos significados con la cara, las manos o el cuerpo. Eso es, en términos generales, aplicable a las lenguas de los Sordos. No obstante, su uso para designar las lenguas visuo-espaciales de estas comunidades tropieza con el hecho del sentido moderno de “gestos” refiere al conjunto de expresiones no lingüísticas que acompañan al habla a modo de apoyo de lo dicho y que no conforman un código productivo, esto es, carecen de la posibilidad de construir significados complejos. Las lenguas de las comunidades de sordos, en cambio, que usan la cara, las manos y el cuerpo como articuladores, son sistemas lingüísticos, capaces de codificar cualquier clase de información.

Manual

Este adjetivo, cuyo uso ha conocido una extensión aún mayor que el anterior en la denominación de las lenguas de los Sordos, tiene el inconveniente de que tipifica estos sistemas como basados en la actividad de las manos, obviando la importancia que para tales lenguas tiene la actividad no manual.

Muchos estudios desarrollados sobre la gramática de las lenguas de los Sordos demuestran que la cohesión del discurso se articula más en la actividad no manual que en la manual. Otros ejemplos de la importancia gramatical de los rasgos no manuales son los estudios que acerca de la Lengua de Señas Norteamericana (ASL) ha realizado Liddell (1980) sobre las funciones de la posición de la cabeza en operaciones gramaticales como topicalización o subordinación, y los estudios de Wilbur (1994) sobre el papel que en esa misma lengua tiene el parpadeo como marcador de la estructura de las frases.

Señas y signos

Estas dos palabras españolas tienen un origen común, pero una historia diferente. “Signo” es una versión romance del latín “signum”, que pasó al español por vía culta, mientras que “seña”, que deriva de “signa”, el plural de “signum”, llegó hasta nosotros por vía de la lengua hablada, por lo que sufrió las transformaciones fonéticas a las que debe su forma actual.

La diferencia en el origen hace que “seña” tenga hoy una amplia gama de usos en la lengua hablada, mientras que “signo” se circunscribe más bien a las ciencias sociales para designar específicamente el producto de una convención social según la cual a una cierta señal física (un sonido, una imagen visual, etc.) se vincula un cierto significado. De acuerdo con eso, todas las palabras de una lengua, ya sea hablada o señada, son signos, por lo que, para un lingüista, parecería redundante la frase “lengua de signos”.

Al no ser "señas" y "signos" términos estrictamente sinónimos, algunos expertos opinan que la denominación "lengua de signos", mayoritaria en España, es terminológicamente incorrecta, argumentando que, según Saussure, todas las lenguas son en rigor "sistemas de signos". No obstante, al margen de la terminología estrictamente empleada en el campo de la lingüística, ambas palabras son utilizadas en el uso común, dependiendo del país.

El abecedario de acuerdo al Lenguaje de Señas

El alfabeto es un sistema de escritura simbólica de las letras del alfabeto de manera oral-escrita mediante las manos que se observa en la figura 3. El alfabeto o sistemas de escritura dependen de cada país o comunidad en donde se encuentre, ya que tendrá señas y códigos propios del sitio. En algunos países hispanos existen algunas variaciones en el alfabeto.

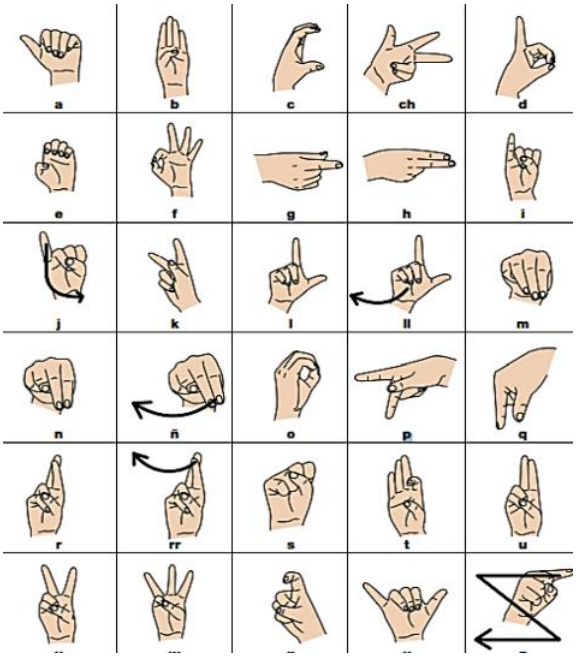

Figura 1. Alfabeto ASL

Las personas sordas instruidas (que sepan leer y escribir) de casi todo el mundo usan un grupo de señas para representar las letras del alfabeto con el que se escribe la lengua oral del país. Es esto lo que se denomina alfabeto manual o alfabeto dactilológico. En el caso de los países de habla hispana, donde se usa el alfabeto latino, las personas sordas usan alfabetos con algunas similitudes pero que difieren según la fonética de la lengua oral con la que coexisten, como también del país. En Inglaterra se usa un alfabeto bimanual. En los países que usan alfabetos distintos al latino (alfabetos hebreo, árabe, amhárico, etc.) existen otras formas de

representación entre las personas sordas. Lo mismo se aplica a los países donde se usan sistemas de escritura no alfabéticos (como es el caso de Japón, China, etc.).

Lengua de señas salvadoreña

En El Salvador se utiliza un sistema de señas y gestos conocido como LESSA (Lengua de Señas Salvadoreña) oficializado por la Asamblea Legislativa de la República de El Salvador en el decreto No. 716 del 14 de julio 2014 en el cual se "reconoce La Lengua de Señas Salvadoreña, LESSA como lengua natural y oficial de las personas sordas salvadoreñas".

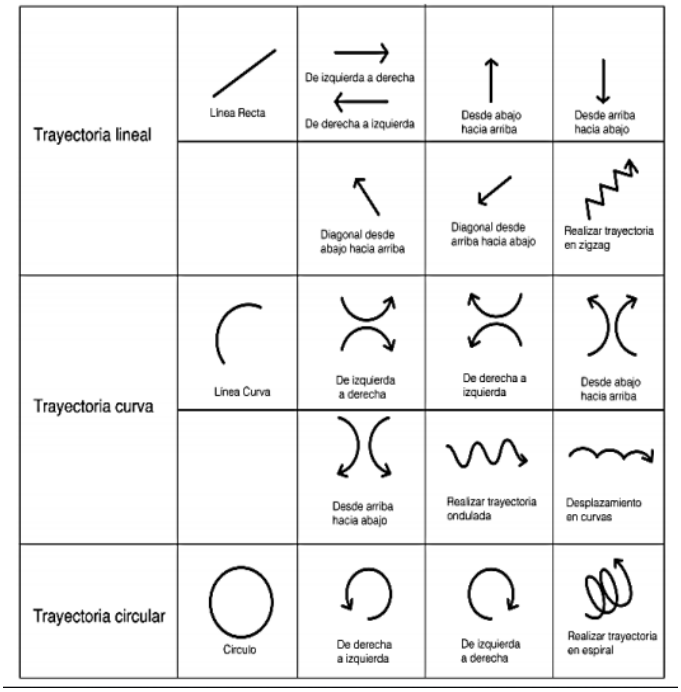

Figura 2. Trayectorias de movimiento

La Lengua de Señas Salvadoreño es un idioma utilizado por la comunidad de sordos o personas con discapacidad auditiva en El Salvador. Su principal objetivo es proporcionar educación a las personas sordas del país. Por motivos de culturización en el país hay tres formas distintas de lenguaje de señas.

Primero, el lenguaje de señas estadounidense (ASL) fue traído a El Salvador desde los Estados Unidos por misioneros que establecieron pequeñas escuelas comunales para sordos. Segundo, el gobierno también ha creado una escuela para sordos, que enseña a través de su propio lenguaje de señas salvadoreño. El tercer tipo de lenguaje de señas utilizado es una combinación de lenguaje de señas estadounidense y lenguaje de señas salvadoreño. La mayoría de los sordos entienden y confían en ambos. Su propio lenguaje de señas salvadoreño único, se basa en su idioma y es más útil en encuentros regulares.

Siendo El Salvador un pueblo muy expresivo con las manos y gestos, la mayoría de las señas utilizadas por la comunidad sorda son gestos naturales de la población oyente, adoptadas por las personas sordas, teniendo así una misma raíz cultural, y aclaran en sí el concepto que se expresa o la realidad que representan las señas.

LESSA está compuesto de gestos y señas las cuales se encuentran agrupadas por afinidades llamadas clasificadores (geografía, partes del cuerpo, familia, zona específica en la que se vive, etc.), dichas señales se pueden encontrar recopiladas en varios diccionarios, pero ninguno oficial, los cuales poseen más de 500 palabras acompañadas de su expresión correspondiente. Debido a que el sistema LESSA no cuenta con su diccionario oficial, aún se encuentra con el problema de contar con una documentación muy limitada en comparación con la totalidad de señas que se utilizan dentro del país.

Esto se debe a que el lenguaje es bastante empírico. Hay instituciones que trabajan para fomentar el lenguaje como la Asociación Salvadoreña de Sordos, el Centro de Audición y Lenguaje y la Escuela de Sordos. Ahí se instruye y explica a detalle por qué se hace una seña, cuál es el significado, para qué se utiliza. Para poder establecer una seña es necesario que las personas sordas entiendan de qué se está hablando, debido a que el lenguaje de señas es pobre en cuanto a gramática como las lenguas verbales, ya que ellos tienen su propia gramática.

Gramática LESSA. LESSA está fundamentado en el lenguaje español salvadoreño, esto significando que sus señas están basadas en la interpretación física de acciones, objetos o habiendo algo esencial que caracteriza a una persona, objeto, animal, etc. Por ejemplo, la seña

de la palabra Papá simula como tener un bigote, como se menciona, una representación física de una persona. Siendo en otra lengua de señas diferente, dependiendo del contexto cultural del país, puede cambiar.

Así también, lo que se busca con LESSA es comunicarse de la manera más sencilla posible, esto indicando que hay palabras conocidas como algunos determinantes, artículos y algunas preposiciones que no cuentan con una seña, porque son omitidas en LESSA. Como, por ejemplo: la, el, los, a, de, etc.

Por ejemplo, la oración: *Mi mamá va a la casa de mi tía.* Esta oración expresada en LESSA es así: *Mi mamá ir casa mi tía.* Lo que se busca es simplificar la oración para una comunicación más rápida y efectiva, sin perder tiempo en señas innecesarias. Por lo tanto, se remueven las palabras *a, la* y *de,* que no se signan por lo mencionado anteriormente.

Otro aspecto a tomar en cuenta es la conjugación de verbos, como se ve en el ejemplo anterior el verbo de la oración original *va,* presente en la segunda persona singular del verbo ir, es corregido a su infinitivo, la forma raíz del verbo. Cuando el verbo está en tiempo presente o futuro, como, por ejemplo, *Mi mamá fue a la casa de mi tía.* El verbo *fue* siendo la forma pasado en la segunda persona del singular, pasa a *ir,* pero con la característica que por medio de una seña se establece que es en pasado, presente o futuro, según sea el caso.

En este ejemplo se concluye que la oración signada sería: *Mi mamá pasado ir casa mi tía.* Si es necesario establecer día por ejemplo ayer, la oración sería: *Mi mamá ayer ir casa mi tía.*

Lo mismo ocurre con las preguntas, por ejemplo, una pregunta sencilla *¿Cuántos años tienes?,* una traducción literal de la oración sería *Cuántos años tener,* lo cual es incorrecto. En LESSA en la mayoría de ocasiones se cambia la estructura gramatical de las preguntas, en este caso la oración signada es: ¿*Tu cumpleaños cuántos?* Ahora probablemente la oración en español gramatical no tenga sentido, pero en LESSA es la manera correcta de preguntar la edad de alguien. Explicando, el verbo *tienes,* en la segunda persona del presente singular, siendo su

infinitivo *tener,* se obvia, porque las personas no “tienen” años, sino en LESSA, cumplen años, por lo tanto, se utiliza la seña de Cumpleaños, para formar la oración.

Como mencionado, en la mayoría de preguntas, la estructura cambia y he aquí más ejemplos:

- Yo voy a la escuela en bus los lunes. Cómo se interpreta en LESSA es: Yo ir bus escuela lunes.
- Mi abuelo quiere a mi mamá. En LESSA: Mi abuelo querer mi mamá.
- Tu estudiaste 5 años en la universidad. En LESSA: Tu antes estudiar 5 años universidad.

Números en LESSA. Los números son signados con la mano derecha o izquierda, del 0 al 5, con ambas manos del 6 al 10. Al llegar al número 11 se utilizan ambas manos, indicando primero el número 10 y luego con una mano el dígito menos significativo por ejemplo 1, para formar el número 11. Al llegar al número 16, se indica el número 10 y luego el 6 con ambas manos también, y de esa manera con los números del 16 al 19.

De los números de 20 en adelante solo se indican con una mano, haciendo primero 2 y luego el 0, moviendo un poco la mano hacia la derecha o izquierda, en el cambio de dígito, dependiendo de cual mano se esté utilizando. Los números de 3 dígitos o más de igual manera, se hacen los tres números con la misma mano, solo moviendo la mano un poco hacia un lado en el cambio de cada dígito.

INTELIGENCIA ARTIFICIAL

El inicio de la inteligencia artificial puede ubicarse durante la segunda guerra mundial, en la cual el célebre científico informático británico Alan Turing trabajó para descifrar el código "Enigma", que fue utilizado por las fuerzas alemanas para enviar mensajes de forma segura. Alan Turing y su equipo crearon la máquina Bombe que se utilizó para descifrar los mensajes de Enigma.

Las máquinas Enigma y Bombe sentaron las bases del aprendizaje automático. Según Turing, una máquina que podría conversar con humanos sin que los humanos sepan que es una máquina ganaría el "juego de imitación" y podría decirse que es "inteligente". Fue igualmente los inicios para que las computadoras podían “entender” el lenguaje escrito, y es que muy bien enigma lograba descifrar las palabras a base de cálculos probabilísticos y un conjunto base de de letras en el mensaje a reemplazar cada una cantidad de letras recorridas y formar el mensaje decodificado de manera entendible respetando las reglas del lenguaje.

En 1956, el científico informático estadounidense John McCarthy organizó la Conferencia de Dartmouth, en la que se adoptó por primera vez el término "Inteligencia Artificial". Los centros de investigación aparecieron en todo Estados Unidos para explorar el potencial de la IA. Los investigadores Allen Newell y Herbert Simon fueron instrumentales en la promoción de la IA como un campo de la informática que podría transformar el mundo.

En 1951, una máquina conocida como Ferranti Mark 1 utilizó con éxito un algoritmo para dominar a los verificadores. Posteriormente, Newell y Simon desarrollaron un algoritmo general de resolución de problemas para resolver problemas matemáticos. También en los años 50, John McCarthy, a menudo conocido como el padre de la IA, desarrolló el lenguaje de programación LISP, que se hizo importante en el aprendizaje automático

En la década de 1960, los investigadores enfatizaron el desarrollo de algoritmos para resolver problemas matemáticos y teoremas geométricos. A fines de la década de 1960, los científicos informáticos trabajaron en Machine Vision Learning y desarrollaron el aprendizaje automático en robots. WABOT-1, el primer robot humanoide "inteligente", fue construido en Japón en 1972.

Sin embargo, a pesar de este esfuerzo global bien financiado durante varias décadas, a los científicos informáticos les resultó increíblemente difícil crear inteligencia en las máquinas. Para tener éxito, las aplicaciones de inteligencia artificial (como el aprendizaje por visión) requerían el procesamiento de una enorme cantidad de datos. Las computadoras no estaban lo

suficientemente desarrolladas para procesar una gran cantidad de datos. Los gobiernos y las corporaciones estaban perdiendo la fe en la IA.

Por lo tanto, desde mediados de la década de 1970 hasta mediados de la década de 1990, los científicos informáticos se enfrentaron a una grave escasez de fondos para la investigación en IA. Estos años se conocieron como los 'AI Winters'.

A fines de la década de 1990, las corporaciones estadounidenses se interesaron una vez más por la IA. El gobierno japonés reveló planes para desarrollar una computadora de quinta generación para avanzar en el aprendizaje automático. Los entusiastas de la IA creían que pronto las computadoras serían capaces de mantener conversaciones, traducir idiomas, interpretar imágenes y razonar como personas. En 1997, la derrota de Deep Blue de IBM se convirtió en la primera computadora en vencer al campeón mundial de ajedrez, Garry Kasparov.

Algunos fondos de AI se secaron cuando estalló la burbuja de las dotcom a principios de la década de 2000. Sin embargo, el aprendizaje automático continuó su marcha, en gran parte gracias a las mejoras en el hardware de las computadoras. Las corporaciones y los gobiernos utilizaron con éxito métodos de aprendizaje automático en dominios estrechos. Las ganancias exponenciales en la capacidad de procesamiento de la computadora y la capacidad de almacenamiento permitieron a las empresas almacenar grandes cantidades de datos por primera vez. En los últimos 15 años, Amazon, Google, Baidu y otros aprovecharon el aprendizaje automático para su enorme ventaja comercial.

Además del procesamiento de datos de los usuarios para comprender el comportamiento del consumidor, estas compañías han continuado trabajando en la visión por computadora, el procesamiento en lenguaje natural y una gran cantidad de otras aplicaciones de inteligencia artificial. El aprendizaje automático ahora está integrado en muchos de los servicios en línea que utilizamos. Como resultado, hoy en día, el sector tecnológico impulsa el mercado de valores estadounidense.

MACHINE LEARNING

Machine Learning es un subconjunto de inteligencia artificial donde los algoritmos informáticos se utilizan para aprender de forma autónoma a partir de datos e información. En el aprendizaje automático, las computadoras no tienen que ser programadas explícitamente, pero pueden cambiar y mejorar sus algoritmos por sí mismas.

El aspecto iterativo del Machine Learning es importante porque a medida que los modelos son expuestos a nuevos datos, éstos pueden adaptarse de forma independiente. Aprenden de cálculos previos para producir decisiones y resultados confiables y repetibles. Es una ciencia que no es nueva pero que ha cobrado un nuevo impulso. El resurgimiento del interés en el aprendizaje basado en máquina se debe a los mismos factores que han hecho la minería de datos y el análisis Bayesiano más populares que nunca. Cosas como los volúmenes y variedades crecientes de datos disponibles, procesamiento computacional más económico y poderoso, y almacenaje de datos asequible.

Aunque muchos algoritmos de aprendizaje basado en máquina han estado entre nosotros por largo tiempo, la posibilidad de aplicar automáticamente cálculos matemáticos complejos al Big Data una y otra vez, cada vez más rápido es un logro reciente.

Hoy en día, los algoritmos de aprendizaje automático permiten a las computadoras comunicarse con humanos, conducir autos de manera autónoma, escribir y publicar informes de partidos deportivos y encontrar sospechosos de terrorismo. Creo firmemente que el aprendizaje automático tendrá un gran impacto en la mayoría de las industrias y en los puestos de trabajo dentro de ellos, por lo que cada gerente debe tener al menos algún conocimiento de qué es el aprendizaje automático y cómo está evolucionando.

Examinamos los orígenes del aprendizaje automático, así como los hitos más recientes, para poder comprender como Machine Learning se ha convertido en una de las tecnologías más importantes hoy en día.

- 1950: Alan Turing crea la "Prueba de Turing" para determinar si una computadora tiene inteligencia real. Para pasar la prueba, una computadora debe poder engañar a un humano para que crea que también es humano.
- 1952 - Arthur Samuel escribió el primer programa de aprendizaje por computadora. El programa era el juego de damas, y la computadora IBM mejoró en el juego cuanto más jugaba, estudiando qué movimientos inventan las estrategias ganadoras e incorporando esos movimientos a su programa.
- 1957: Frank Rosenblatt diseñó la primera red neuronal para computadoras (el perceptrón), que simula los procesos de pensamiento del cerebro humano.
- 1967: se escribió el algoritmo del "vecino más cercano", lo que permite que las computadoras comienzan a usar el reconocimiento de patrones muy básicos. Esto podría usarse para trazar una ruta para los vendedores ambulantes, comenzando en una ciudad aleatoria, pero asegurándose de que visiten todas las ciudades durante un breve recorrido.
- 1979 - Los estudiantes de la Universidad de Stanford inventan el "Carrito de Stanford" que puede sortear obstáculos en una habitación por sí solo.
- 1981: Gerald Dejong presenta el concepto de aprendizaje basado en explicaciones (EBL), en el que una computadora analiza los datos de capacitación y crea una regla general que puede seguir descartando los datos que no son importantes.
- 1985 - Terry Sejnowski inventa NetTalk, que aprende a pronunciar palabras de la misma manera que lo hace un bebé.
- Década de 1990: el trabajo en aprendizaje automático cambia de un enfoque basado en el conocimiento a un enfoque basado en los datos. Los científicos comienzan a crear programas para que las computadoras analizan grandes cantidades de datos y saquen conclusiones, o “aprendan”, a partir de los resultados.
- 1997 - Deep Blue de IBM vence al campeón mundial de ajedrez.
- 2006 - Geoffrey Hinton acuña el término "aprendizaje profundo" para explicar nuevos algoritmos que permiten a las computadoras "ver" y distinguir objetos y texto en imágenes y videos.

- 2010 - El Microsoft Kinect puede rastrear 20 características humanas a una velocidad de 30 veces por segundo, lo que permite a las personas interactuar con la computadora a través de movimientos y gestos.
- 2011 - Watson de IBM supera a sus competidores humanos en Jeopardy.
- 2011 - Google Brain se desarrolla y su red neuronal profunda puede aprender a descubrir y categorizar objetos de la misma manera que lo hace un gato.
- 2012 - X Lab de Google desarrolla un algoritmo de aprendizaje automático que permite navegar de forma autónoma los videos de YouTube para identificar los videos que contienen gatos.
- 2014 - Facebook desarrolla DeepFace, un algoritmo de software que es capaz de reconocer o verificar individuos en fotos al mismo nivel que los humanos.
- 2015 - Amazon lanza su propia plataforma de aprendizaje automático.
- 2015: Microsoft crea el Kit de herramientas de aprendizaje automático distribuido, que permite la distribución eficiente de los problemas de aprendizaje automático en varias computadoras.
- 2015: más de 3,000 investigadores de inteligencia artificial y robótica, respaldados por Stephen Hawking, Elon Musk y Steve Wozniak (entre muchos otros), firmaron una carta abierta de advertencia sobre el peligro de las armas autónomas que seleccionan y atacan objetivos sin intervención humana.
- 2016: el algoritmo de inteligencia artificial de Google supera a un jugador profesional en el juego de mesa chino Go, que se considera el juego de mesa más complejo del mundo y es muchas veces más difícil que el ajedrez. El algoritmo AlphaGo desarrollado por Google DeepMind logró ganar cinco juegos de cinco en la competencia Go.

Como se puede apreciar el uso de Machine Learning en las computadoras ha ido avanzando tanto a tal punto en que las computadoras han evolucionado de poder aprender, imitar y generar decisiones por ellas mismas, a un grado en la cual puede tomar conductas propias de un ser viviente y poder comportarse y comprender cómo al ser al cual “imita”.

PROCESAMIENTO DEL LENGUAJE NATURAL

El Procesamiento del Lenguaje Natural (PLN) es la rama de conocimiento de la Inteligencia Artificial que se encarga de investigar la manera de comunicar las computadoras con las personas mediante el uso de lenguas naturales como el español.

La historia del PLN empieza desde 1950, cuando Alan Turing publicó Computing Machinery and Intelligence en el cual proponía el test de Turing como criterio de inteligencia. El experimento de Georgetown en 1954 involucró traducción automática de más de sesenta oraciones de Ruso a Inglés. Se proponía que en tres o cinco años la traducción automática estaría resuelta. El progreso real en traducción automática fue más lento.

Virtualmente, cualquier lengua humana puede ser tratada por los ordenadores. Lógicamente, limitaciones de interés económico o práctico hace que solo las lenguas más habladas o utilizadas en el mundo digital tengan aplicaciones en uso.

Las lenguas humanas pueden expresarse por escrito (texto), oralmente (voz) y también mediante signos. Sin embargo, el PLN está más avanzado en el procesamiento de texto, ya que son más fáciles de obtener datos directos que de otras fuentes como el audio. El audio hay que procesarlo para transformarlo en texto y poder comprender el mismo.

Existe una serie de dificultades en el procesamiento del lenguaje natural, entre las cuales:

Ambigüedad.

- En el nivel léxico, una misma palabra puede tener varios significados, y la palabra correcta se debe deducir a partir del conocimiento básico. Este tipo de ambigüedades se tratan de resolver con diccionarios, gramáticas, bases de conocimiento y correlaciones estadísticas.
- La identificación de anáforas y catáforas determina la entidad lingüística que hacen referencia.

- En el nivel estructural, se requiere de la semántica para desambiguar la dependencia de los sintagmas preposicionales que conducen a la construcción de distintos árboles sintácticos.
- En el nivel pragmático, una oración no significa lo que realmente se está diciendo. Elementos como la ironía tienen un papel importante en la interpretación del mensaje.

Para resolver estos tipos de ambigüedades y otros, el problema central en el PLN es la traducción de entradas en lenguas naturales a una representación interna sin ambigüedad, como árboles de análisis.

Detección de separación entre las palabras

En la lengua hablada no se suelen hacer pausas entre palabras. El lugar en el que se debe separar las palabras a menudo depende de cuál es la posibilidad que mantenga un sentido lógico, tanto gramatical como contextual. En la lengua escrita, idiomas como el mandarín tampoco tienen separaciones entre palabras.

Recepción imperfecta de datos

Acentos extranjeros, regionalismos o dificultades en la producción del habla, errores de mecanografiado o expresiones no gramaticales, errores en la lectura de textos mediante OCR (Optical Character Recognition, o Reconocimiento Óptico de Caracteres en español).

Los componentes del procesamiento del lenguaje natural son:

- Análisis morfológico. El análisis de las palabras para extraer raíces, rasgos flexivos y unidades léxicas compuestas.
- Análisis sintáctico. El análisis de la estructura sintáctica de la frase mediante la gramática de la lengua en cuestión.

- Análisis semántico. El significado de la frase, y la resolución de ambigüedades léxicas y estructurales.
- Análisis pragmático. El análisis del texto sobre los límites de la frase para determinar los referentes de los pronombres.
- Planificación de la frase. Estructurar cada frase del texto con el fin de expresar el significado adecuado.
- Generación de la frase. La generación de la secuencia lineal de palabras a partir de la estructura general de la frase, con sus correspondientes flexiones, y concordancias.

Las principales tareas de trabajo en el procesado del lenguaje natural son:

- Síntesis del discurso
- Análisis del lenguaje
- Comprensión del lenguaje
- Reconocimiento del habla
- Síntesis de voz
- Generación de lenguajes naturales
- Traducción automática
- Respuesta a preguntas
- Recuperación de la información
- Extracción de la información

Los lingüistas computacionales se encargan de preparar el modelo lingüístico para que pueda ser implementado en código eficiente y funcional. Existen dos aproximaciones al problema de la modelización lingüística:

Modelos Lógicos

Los lingüistas escriben reglas de reconocimiento de patrones gramaticales estructurales. Estas reglas, en conjunto con la información almacenada en diccionarios computacionales, definen los patrones que hay que reconocer para resolver la tarea. Estos modelos lógicos pretenden reflejar la estructura lógica del lenguaje y surgen a partir de las teorías de Noam Chomsky en 1950.

Modelos probabilísticos del lenguaje natural.

Los lingüistas recogen colecciones de datos (corpus) y a partir de ellos se calculan las frecuencias de diferentes unidades lingüísticas (letras, palabras, oraciones) y su probabilidad de aparecer en un contexto determinado. Calculando esta probabilidad, se puede predecir cuál será la siguiente unidad en un contexto dado, sin necesidad de recurrir a reglas gramaticales explícitas.

MANIPULACIÓN DE GRÁFICOS 3D

En general se trata de software que se utiliza para la creación y manipulación de gráficos 3D por computadora. Actualmente existen muchas alternativas y cada una de ellas con diferentes especializaciones, pero en general cumplen con algunas o todas de las siguientes funciones:

- Representación: representación de exteriores y arquitecturas.
- Modelado: modelado, escultura, edición de curvas para el diseño de objetos 3d.
- Animación: animar objetos 3d.
- Renderización: generación de imágenes.
- Efectos visuales: agregar efectos visuales en uso de cámaras en escena.
- Dinámicas y Simulación: ofrece simulaciones de eventos de objetos reales.

- Edición de video: Bender ofrece una gama de herramientas en la edición de video
- Scripting: poder crear secuencias de comandos y personalización.

En cuanto a las herramientas más utilizadas para el desarrollo del proyecto, las características a utilizar son el modelado y animación. A continuación, se estará describiendo en manera global y general los fundamentos principales que se manejan dentro de estas dos áreas, las cuales serán detalladas y dirigidas en el desarrollo del proyecto en capítulos siguientes, para facilitar la explicación de los conceptos se utilizarán imágenes y ejemplos basados en software como Blender y Autodesk Maya.

Armadura o Esqueleto

La armadura en Blender sirve como el esqueleto del modelo 3D, y como un esqueleto real la armadura puede estar compuesta de varios huesos. Estos huesos pueden ser movidos, rotados, deformados y todo a lo que están unidos o asociados se moverá y deformará en una forma similar.

Las armaduras están diseñadas para realizar poses, ya sea para escenas estáticas o animadas y tienen un estado específico llamado “posición de descanso”. Dicho estado funciona como la “forma” por defecto de la armadura, igual que la posición, rotación y escala por defecto de los huesos.

Huesos. Son los elementos base de las armaduras y conformar los elementos móviles para poder generar una pose con la armadura. Al igual que los huesos reales en un esqueleto, los huesos en Blender están compuestos por una unión principal, el cuerpo del hueso y la unión final (tip, boy, top).

Los huesos pueden clasificarse de dos formas:

1. **Huesos Deformables:** son huesos que al transformarse resultara en los vértices asociados a ellos transformarse en una forma similar. Los huesos deformables están asociados directamente en la alteración de las posiciones de los vértices asociados con ellos.
2. **Huesos de Control:** Controlan como otros huesos u objetos reaccionan cuando son transformados. Los huesos de control no están asociados a posiciones de vértices; de hecho, los huesos de control no tienen vértices asociados con ellos mismos

Estructura de los huesos. Como se dijo previamente los huesos comprenden de tres partes: unión inicial, llamada raíz o cabeza, el cuerpo y la Unión final conocida como punta o cola. Las respectivas posiciones de la raíz y la punta definen el hueso. Las diferentes uniones entre los huesos y el manejo de sus vértices son las que se usan para poder crear una pose diferente a la que se posee en el estado por defecto de la armadura.

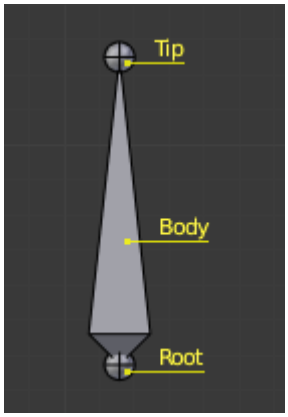

Figura 3. Hueso base

En cuanto a las uniones entre huesos pueden ser descritas en dos tipos de relaciones:

1. Emparentado: Son huesos que actúan como si no tuvieran relaciones, pueden ser arrastrados, rotados, escalados, etc. sin afectar a sus descendientes.

2. Conectados: Son huesos que tienen punta de un hueso padre conectado a su raíz, de esta forma al transformar un hueso afectará a todos huesos padre/hijo/hermano conectados.

Estructura y cadena de huesos. Las armaduras imitan esqueletos reales. Están hechas de huesos, que por defecto son elementos rígidos. Pero se tienen más posibilidades que con un esqueleto real: en adición a la rotación "natural" de los huesos, es posible trasladarlos e incluso escalarlos y los huesos no tienen que estar conectados los unos con otros, pueden estar completamente libres si se quiere. Sin embargo, así como en un esqueleto real donde varios huesos están conectados unos con otros. En la armadura el arreglo más natural implica que algunos huesos están relacionados entre sí, formando a lo que se conoce como cadena de huesos.

La cadena de huesos se forma cuando varios huesos se emparentan con otros huesos cercanos a ellos, como por ejemplo cuando se quiere crear una pierna, todos los huesos debajo del muslo deberían estar conectado para moverse de una forma uniforme. La cadena de huesos también puede presentar ramificaciones, como por ejemplo los huesos de los dedos pueden estar unidos a un mismo hueso de la mano.

La cadena se genera cuando la punta de un hueso padre se conecta con la raíz de un hueso hijo. El hueso al inicio de la cadena se conoce como el hueso raíz y el último hueso de la cadena como hueso de punta. Dentro de la cadena de huesos, un hueso puede ser el padre de varios huesos hijos.

Pose y animación

Cuando la armadura se considera que se encuentra bien estructurada, con los huesos definidos, conectados en una posición por defecto o de descanso, se puede proceder con la parte de poses. Una pose es una posición estática en la cual ciertos huesos (o todos) tienen su

propia posición, rotación y escala, diferente de las que poseía en la posición de descanso, generando así una nueva posición estática para la armadura.

Aunque las poses son para propósitos completamente estáticos, juegan un rol sumamente importante en las características y técnicas de animación.

Cuando empezamos a crear una pose nueva nos encontramos con tres factores principales que transforman a los huesos de su posición de descanso a una nueva pose, estos factores se describen como:

- **Ejes:** cuando se quiere realizar una nueva pose en Blender, los huesos deben tener una posición determinada dentro de un espacio de movimiento. Este espacio se conforma con los ejes X, Y, Z. Cada hueso al querer ser transformado se encuentra en un espacio propio el cual está conformado por estos ejes y rigen las direcciones en las que se puede mover. El espacio en que un hueso puede moverse en los huesos puede ser solamente en un eje o en la intersección de varios, lo que permite el crear poses 3d más detalladas y precisas.

- ***Traslación:*** Mover un hueso de una posición a otra ya se en el mismo eje, o de un eje a otro.

- ***Roll:*** Cuando un hueso roda sobre su propio eje Y.

- ***Rotación:*** Cuando un hueso rota en base a un vértice en un eje específico.

Animación de las poses. Cuando se tiene las poses necesarias modelada de acuerdo a como se necesiten para poder generar una animación, la posición de cada hueso en el espacio debe ser bloqueada, para que la información de posición de los huesos quede guardada.

A la información sobre la posición de los huesos en el espacio se le conoce como **Keyframe.** Para que las poses puedan pasar a hacer una animación, se requiere que las

keyframe que conforman una posición específica sean vinculadas a una línea de tiempo, la cual se le conoce como Dope Sheet.

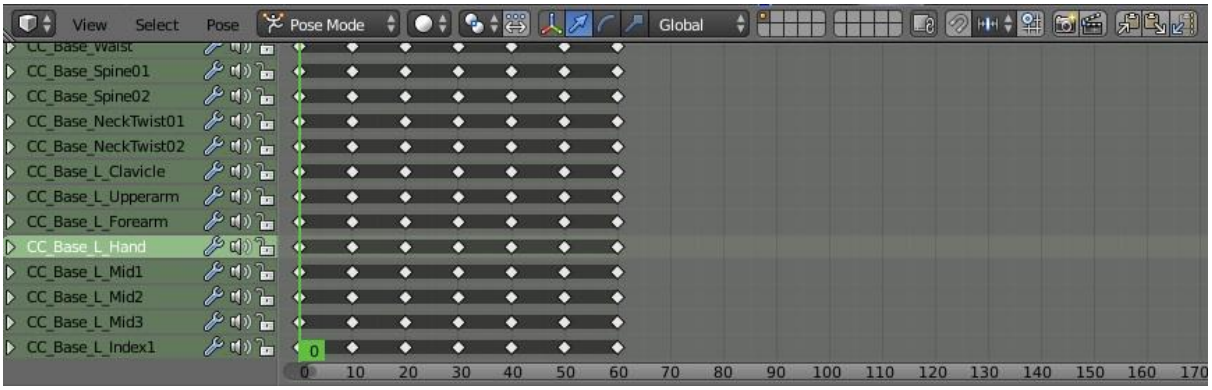

Figura 4. Ejemplo de Dope Sheet en Blender 3D

En figura 4 se muestra una Dope Sheet, los elementos que la conforman son:

1. El listado de huesos bloqueados que aparece a la izquierda.
2. Los keyframes representados por los rombos blancos
3. Línea de tiempo donde se ubican los keyframes.

En la línea de tiempo de la dopesheet es donde se ubican los keyframes de los huesos en un momento o cuadro determinado, así el cuadro 0 puede tener keyframes que conforman una pose predeterminada y en el frame 10 puede estar una pose totalmente distinta. Es en este cambio de posiciones de los huesos entre keyframes las que sirven como las bases para crear una animación.

La forma en que la animación se lleva a cabo es mediante "*interpolación de puntos"*, Blender interpola los cuadros necesarios para producir el traslado de movimiento de un keyframe a otro. Por ejemplo, en el keyframe en el cuadro 0 la posición del hueso del brazo está en una posición de descanso bajo, luego en el keyframe 10 el mismo hueso ahora está en una posición de descanso arriba. Al reproducir la animación Blender interpola calculando la distancia desde el punto A al punto B y genera los puntos necesarios para crear el moviendo, así la traslación se aprecia como cuando uno levanta el brazo normalmente.

MOTORES DE VIDEOJUEGOS

El término de Motor de videojuego es acuñado para aquel software que proporciona un conjunto de herramientas dentro de un entorno de desarrollo integrado que facilitan el diseño, creación, funcionamiento y mantenimiento de un videojuego. Uno de los aspectos más importantes sobre los motores de videojuegos actuales es la capa de abstracción de hardware, lo cual permite al desarrollador la realización de proyecto sin conocer la arquitectura de hardware de las plataformas objetivo.

Antes de la creación de los motores de videojuegos, estos eran desarrollados usualmente sin tomar en cuenta la separación de áreas (imagen, sonido, física, entre otras) y eran desarrollados desde cero. Por ejemplo, si alguien buscara crear un videojuego bajo estas técnicas tendría programar todo el manejo de gráficos y física desde cero y para un único juego exclusivamente. Fue en los años 80's cuando surgieron los primeros motores de videojuegos como Freescape (1987) para 3D y Adventure Game Interpreter (1984) para 2D, pero su utilización continuaba siendo muy exclusiva a un solo videojuego.

Fue en los años 90's cuando los motores de videojuegos se popularizaron realmente con juegos como Quake, ya que los motores utilizados no eran exclusivos para un único videojuego, fue así como Quake Engine permitió la creación de diversos juegos y mods. La principal característica es que los desarrolladores no tenían que comenzar de cero, ya que estos motores realizaban el manejo de renderizado, física, iluminación entre otros aspectos. Los motores de videojuegos son en la actualidad los programas de software complejos, por la cantidad de funciones que pueden llegar a manejar y el uso de recursos. Los componentes principales que se manejan por los distintos motores son los siguientes:

Motor de gráficos (Renderización)

En palabras simples un motor gráfico es un software utilizado para dibujar gráficos en una pantalla, renderización. La renderización es el procesamiento de ecuaciones y cálculos matemáticos por computador que dará como resultado una imagen, por lo general este término

es utilizado con gráficos en tres dimensiones, pero es totalmente valido para gráficos en dos dimensiones. El objetivo de la renderización en los videojuegos es realizar cálculos que permitan obtener una simulación realista de materiales, texturas, iluminación y comportamientos físicos, por ejemplo, podemos obtener representaciones sobre madera, agua, vidrio, etc.

La renderización es el proceso más demandante en recursos, por lo cual en general se utilizan diferentes técnicas como:

- Pre-rendering: consiste en mostrar el resultado sin ser procesado en tiempo real, es decir se utilizan imágenes previamente procesadas. La principal desventaja es que se puede perder el grado de interactividad con el usuario

- Lightmapping: es el proceso que consiste en calcular el brillo o luz sobre las superficies y guardando el resultado en un "mapa de luz" para su uso posterior. Esta técnica supone una gran mejora para el tiempo de renderización, pero puede perderse parte del realismo y por lo cual suele utilizarse sobre superficies estáticas en la escena.

- Post-processing: es el proceso de aplicar filtros y efectos sobre imágenes renderizadas, todo esto puede ocurrir en tiempo real, antes de ser mostradas al usuario logrando así efectos más realistas como profundidad y gradientes sin requerir una gran cantidad de recursos comparado a realizarlo directamente sobre la renderización.

Existen más técnicas como el nivel de detalle, la oclusión y otras que no serán discutidas en este proyecto, que pueden mejorar mucho el rendimiento de una aplicación.

Motor de física

El motor de físicas es el encargado de emular leyes de la física dentro de un videojuego o la aplicación, con el fin de generar sensaciones realistas en la interacción con los objetos y entorno dentro de la aplicación al momento de su ejecución. Se encarga de simular atributos

físicos tales como gravedad, peso, volumen, velocidad, aceleración, rotaciones, traslación, fricción, colisiones, entre otros. Se pueden clasificar los motores de física en dos categorías según la capacidad de cálculos que se requieran para el resultado:

- Motores de alta precisión: en general son utilizados para fines de investigación científica, donde la cantidad de cálculos a realizar, por la precisión requerida, es sumamente grande que un resultado en tiempo real se vuelve imposible.

- Motores de tiempo real: se emplean sobre todo para los videojuegos, donde se busca realizar un modelo del mundo real que busca tener una gran calidad y velocidad por encima de la precisión de los cálculos, ya que es lo más importante para los usuarios.

Motor de sonido

Si bien un videojuego no siempre se requiere del sonido para obtener la funcionalidad deseada, este puede brindar una experiencia sumamente más inmersiva y en ocasiones puede ser vital para proceder. El motor de sonidos es el encargado de manipular algoritmos relacionados con la carga, modificación y salida del audio. Existen diversas API que facilitan el manejo y renderización de efectos como eco, efecto Doppler, reverberación, sonido en espacio 3D, entre muchos otros. Entre las API más conocidas tenemos DirectSound por Microsoft, OpenAL y SDL OpenAudio.

Scripting y programa principal

Es la parte esencial de todo motor de videojuego y es la que permite por medio de lenguajes de programación el funcionamiento y la lógica de los elementos contenidos dentro del videojuego y aplicación. Mediante la programación se obtienen datos de entrada por parte del usuario u otro programa y se determina la respuesta correspondiente a cada una de ellas, asegurando que los eventos dentro del juego se ejecuten en el momento adecuado. Por ejemplo,

podemos brindar a un modelo humano en 3D la capacidad de reproducir rotaciones y translaciones es respuesta a comandos enviados desde otra aplicación por medio de la implementación de Websockets e incluso se podría implementar un sistema de inteligencia artificial.

Otra funcionalidad importante que se puede alcanzar es por medio de scripting, que permite a objetos ejecutar funcionalidad predefinidas sin requerir la intervención de un usuario, un ejemplo común de este es la reproducción de una escena renderizada en tiempo real, donde los distintos objetos realizan acciones en lugar de utilizar un video previamente renderizado.

Estos son los motores más utilizados actualmente:

- Unity3D
- Cocos2D
- Cryengine
- Unreal Engine
- Game Maker Studio

Bucle del Juego

El bucle del juego, o Game Loop por su nombre en inglés, es el componente principal para el funcionamiento de cualquier videojuego. Este permite que el juego permanezca en ejecución independientemente de las entradas de los usuarios o la ausencia de ellas. El siguiente pseudocódigo muestra una lógica básica aplicable a la mayoría de videojuegos.

```
1   Bucle: Mientras sea verdad ( Juego abierto ) hacer:
2     Comprobar interacción del usuario
3     Comprobar estado del personaje
4     Comprobar la gravedad
5     Buscar mensajes
6     Dibujar la escena
7     Reproducir Sonidos
8   Retorno al inicio del bucle
```

Figura 5. Pseudocódigo de un bucle

Generalmente las actividades dentro de un videojuego se desarrollan dentro de estos bucles, que nos permiten realizar una diversidad de acciones, como comprobaciones, sobre cada iteración y que puede ser terminada únicamente si se cumplen condiciones específicas como el presionar un botón o el mismo cierre del juego. Cada iteración realizada es conocida como cuadro, o frame por su nombre en inglés, y pueden verse como una actualización del estado actual del juego.

Tomemos como ejemplo el lanzamiento en caída libre de una manzana, a cada segundo la manzana es afectada por una gran cantidad de variables tales como la gravedad, distancia, desplazamiento, tiempo, viento y posibles obstáculos. Para poder simular su comportamiento de manera realista es necesario analizar continuamente todas las variables, aplicar sus consecuencias y finalmente mostrar el resultado. De esta manera cada cuadro representa el resultado de estas comprobaciones y cálculos, por ejemplo, el primer cuadro de un bucle puede representar el momento específico del lanzamiento de la manzana, mil cuadros después el efecto de un obstáculo sobre ella, y finalmente después de otra cantidad de cuadros el impacto sobre el suelo que podría significar el final del bucle o simplemente continuar.

Cada frame comprende una iteración completa del programa, y los videojuegos realizan estas iteraciones una gran cantidad de veces por segundo, es común escuchar velocidades de 30,60,120 cuadros por segundo, FPS por sus siglas en inglés, donde el número indica la cantidad de imágenes (resultado final) que se nos muestra en pantalla por el lapso de tiempo de un segundo. Entre mayor sea la cantidad de cuadros por segundo mayor será el consumo de recursos y la calidad final presentada.

Corrutinas

Normalmente cuando se ejecutan funciones sobre un programa estas deben ser ejecutadas en su totalidad, es decir evaluando todas las líneas de código, antes de poder retornar. En un motor de videojuego esto significa que todo el método debe ser ejecutado dentro de un mismo frame (cuadro) por lo que la siguiente actualización en pantalla mostrará el resultado de la total ejecución del método. Una corrutina es esencialmente una función, pero esta tiene la capacidad de pausar su ejecución y retornar el control al motor el cual posteriormente definido por algún tiempo devolverá el control a la función nuevamente para continuar con su ejecución.

Uno de los ejemplos más sencillos para mostrar su funcionalidad es cuando queremos hacer que un objeto se desvanezca, que para lograr ese efecto es necesarios cambiar la transparencia de este objeto gradualmente en diferentes momentos hasta que desaparezca por completo.

```
1  Bucle: Repetir 100 veces:
2      Porcentaje transparencia = número de iteración / 100
3   transparencia objeto = porcentaje transparencia
4   Retornar al inicio del bucle
```

Figura 6. Pseudocódigo de bucle imitando una corrutina

El pseudocódigo anterior muestra una posible manera de realizar este efecto mediante el uso de un bucle, el cual aumenta la transparencia hasta que esta sea totalmente transparente. Si ejecutamos este código en una función normal el objeto desaparece instantáneamente, ya que la función es ejecutada en su totalidad durante el lapso de un solo cuadro. En cambio, al utilizar cortinas podemos retornar el control al motor para que este actualice el resultado en pantalla en cada iteración, y este mismo encargará de regresar al bucle después que haya realizado la actualización, es decir un cuadro después o el valor establecido.

```
1   Bucle: Repetir 100 veces:
2   Porcentaje transparencia = número de iteración / 100
3       transparencia objeto = porcentaje transparencia
4       Pausar y regresar dentro de un cuadro
5   Retornar al inicio del bucle
```

Figura 7. Pseudocódigo de una corrutina

Una gran ventaja del uso de corrutinas es que se pueden ejecutar tareas que consumen mucho tiempo para su procesamiento, realizando pausas y retornos evitando así que la aplicación quede congelada tratando de procesar todo en un mismo frame. Permite también controlar la frecuencia con que se realizan tareas, por ejemplo, quizá no sea necesario revisar si hay un nuevo mensaje 60 veces en un mismo segundo, en su lugar puede realizarse esta comprobación una vez cada dos segundos, siendo así una gran optimización para el uso de esa función.

La corrutinas pueden ser ejecutadas parte por partes sobre el tiempo, pero es importante tener en cuenta que todo este procesamiento sigue realizando son el hilo principal de la aplicación, por lo que su uso no asegura evitar que la aplicación se congele.

Hilos

Un hilo, proceso ligero o subproceso son una secuencia de tareas encadenadas que puede ser ejecutada por un sistema operativo, siendo muy parecidas a un proceso normal, pero con la diferencia que estas pueden ser ejecutada al mismo tiempo que otras tareas.

Los hilos de ejecución que comparten los mismos recursos son conocidos como un proceso. El hecho de que los hilos de ejecución de un mismo proceso comparten los recursos hace que cualquiera de estos hilos pueda modificar estos recursos por lo cual cuando un hilo modifica un dato en la memoria, los otros hilos acceden a ese dato modificado inmediatamente. La ejecución de los hilos es manejada por el sistema operativo y depende de la implementación en los diferentes lenguajes de programación.

A diferencia de las corrutinas, los hilos si permiten realizar tareas sin la preocupación que el tiempo que tome en ejecutarse afecte de manera directa la aplicación principal por lo que una buena implementación de estos evita por completo las pausas o congelamientos no deseados sobre la aplicación, pero puede ralentizar un poco la velocidad general de la aplicación.

Singleton Pattern

El patrón de diseño de Instancia Única o Singleton Pattern, por su nombre en inglés, en la ingeniería de software es un patrón de diseño que permite que en cualquier momento dado de la aplicación exista únicamente una sola instancia del objeto y proporcionar además un punto de acceso global para todos. Para implementar este patrón se debe crear un método que permita realice una instancia del objeto, pero únicamente si no existe otra previamente.

En ocasiones como en algunos motores de videojuegos implementar este patrón también consiste es destruir cualquier otra instancia que haya sido creada. Este patrón es muy útil para compartir datos a través de la misma aplicación y poder mantener estados globales. Otro aspecto muy importante de este patrón de diseño es que permite reducir el uso de recursos generados por la instanciación de un mismo objeto múltiples ocasiones.

DIAGRAMAS DE CASOS DE USO

Los diagramas de casos de uso tienen como propósito demostrar las diferentes maneras en que una entidad, como un usuario, puede interactuar con un sistema. Son una herramienta que permite sumarizar las interacciones que pueden darse entre los distintos actores con el sistema. Una característica muy importante es que estos diagramas no entran en detalles específicos, ya que buscan dar una perspectiva o visión general de las relaciones entre casos de usos, actores y sistemas.

El lenguaje unificado de modelado (**UML**), es el lenguaje de modelado de sistemas de software utilizado para construir dichos diagramas. Los diagramas de caso de uso están compuestos por actores, casos de uso, sistemas y paquetes.

- **Actor.** Un actor es cualquier entidad externa al sistema que posee un rol en relación al sistema. Los actores pueden ser usuarios, grupos, organizaciones e incluso otros sistemas y usualmente son representados por la siguiente figura:

Figura 8. Representación de un actor

- **Caso de uso.** Los casos de uso representan y describen cualquier acción o actividad que realiza el sistema. Su representación se hace con un óvalo y el nombre en el interior.

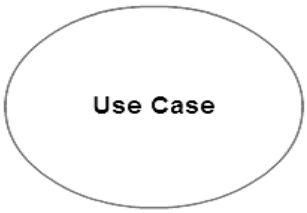

Figura 9. Representación de un caso de uso

- **Sistema.** Un sistema es utilizado para definir el ámbito o contexto de los casos de usos que este contiene, es representado con un rectángulo y un título. Es un elemento opcional, pero se vuelve más necesario conforme avanza la complejidad de un proyecto.

Figura 10. Representación de un sistema

Además de los elementos que componen los diagramas de casos de usos, existen las relaciones que pueden existir entre estos, hay cinco tipos de relaciones que pueden darse:

- **Relación entre un actor y un caso de uso.** Es la relación principal en cualquier diagrama de casos de uso, y representa la interacción entre los actores, que poseen roles, y los casos de usos que son las acciones que pueden realizar. Se representa con una línea recta.

- **Generalización de un actor.** La generalización se utiliza para indicar que un actor puede heredar el rol de otro actor, y con esto todos los casos de usos relacionados con el otro actor para a estar relacionados con él, es una especie de especialización. Se representa con una línea continua terminada en triángulo, que sale desde el actor especializado hacia el actor base.

- **Relación "incluye" entre dos casos de uso.** Esta relación indica que el caso de uso a ser incluido forma parte del caso de uso que lo incluye, y sin el cual estaría incompleto. Se representa con una línea discontinua terminada en triángulo, que sale desde el caso de uso incluido hacia el caso de uso base.

- **Generalización de un caso de uso.** Esta relación es utilizada para indicar cuando un caso de uso hereda las características y comportamientos de otro caso de uso. La existencia de esta relación indica una especialización, es decir que un caso uso es una

especialización de otro existente. Un ejemplo sencillo es al realizar un pago, la especialización puede darse en los distintos métodos de pagos que puedan existir. Se representa con una línea continua terminada en triángulo, que sale desde el caso de uso especializado hace el caso de uso base.

- **Relación "extiende" entre dos casos de uso.** Esta es una relación de adición, que indica que un caso de uso, que es extensión, puede brindar funcionalidades extra a otro caso de uso que sería su base. Se representa con una línea discontinua terminada en triángulo, que sale desde el caso de uso extensión hacia el caso de uso base.

PRUEBAS DE USUARIO O USABILIDAD

Las pruebas de usuario o usabilidad se utilizan para la medición de la capacidad de un producto fabricado por el humano en cumplir con el propósito para el cual se diseñó. Estas pruebas miden la usabilidad o facilidad de uso de un producto específico o un conjunto de productos.

Usability.gov las define como: "La actividad de evaluar un producto o servicio probándolo con usuarios representativos. Normalmente, durante una prueba, los participantes tratarán de completar tareas típicas mientras los observadores observan, escuchan y toman notas. El objetivo es identificar cualquier problema de usabilidad, recopilar datos cualitativos y cuantitativos y determinar la satisfacción del participante con el producto."

Las pruebas de usabilidad buscan evaluar en general las siguientes características:

- **Exactitud:** Número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados.
- **Tiempo** requerido para concluir la actividad.
- **Recuerdo**: Que tanto recuerda el usuario después de un periodo sin usar la aplicación.

- **Respuesta emocional:** Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera).

Para la aplicación en cuestión se realizarán estos tipos de pruebas de usabilidad:

- **Sumativas:** son el tipo de pruebas que se realizan cuando el producto ya se encuentra terminado. El objetivo de este tipo de prueba es medir el desempeño del producto, con respecto al tiempo, precisión y errores, con valores cuantitativos, para un futuro análisis con los resultados de todas las pruebas en conjunto.
- **Presencial:** son el tipo de pruebas que se realizan en una habitación donde el investigador y los usuarios están presentes. En general, se realizan cuando se pretende obtener comentarios o ideas. También se realizan cuando se espera crear un ambiente de confianza con los usuarios, esto con el fin de debatir temas e interrogantes. Dependiendo de la duración de la prueba influye en el tipo de decisiones que se tomarán, entre más larga sea la prueba, es más conveniente hacerla presencial.
- **Guerrilla:** son conocidas como pruebas de campo que pueden realizarse en espacio públicos, como un café, restaurante, etc. De esta manera se puede registrar rápidamente mientras los usuarios utilizan un sitio web, una aplicación durante un tiempo corto. Los beneficios de este tipo de prueba es que se necesitan pocos usuarios y bajo costo para obtener la información que se necesita.

CAPITULO III

DETERMINACIÓN DE REQUERIMIENTO Y DISEÑO

DETERMINACIÓN DE REQUERIMIENTOS

En esta sección se describen los requerimientos, tanto funcionales como no funcionales, dando a conocer las características y funcionalidades que la solución debe poseer.

Tabla 1
Formato de Requerimientos

ID - Nombre	RF[N] / RNF[N] - XYZ
Descripción	Descripción del requerimiento.
Prioridad	Nivel de importancia, baja -media -alta.
Frecuencia de uso	Uso de la característica, baja- media -alta.
Otra información	Comentarios.

Requerimientos funcionales

Estos requerimientos definen las funciones que el sistema de software o sus componentes debe poseer, en otras palabras, nos describen lo que el sistema debe ser capaz de hacer. A continuación, se listan los requerimientos funcionales.

Tabla 2
RF1. Voz y texto

ID - Nombre	RF1 Voz y texto
Descripción	El sistema debe ser capaz de recibir como entrada texto y voz, para posteriormente ser interpretada al lenguaje de LESSA.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	Una misma frase, palabra, letra u oración ingresada debe generar el mismo resultado, ya sea que se ingrese mediante texto o voz.

Tabla 3
RF2. Identificación de entidades

ID - Nombre	RF2 - Identificación de entidades.
Descripción	El sistema debe ser capaz de reconocer entidades nombradas.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	Debe ser capaz de reconocer entidades como los siguientes ejemplos: buenosdias - buenasnoches - bienportado - jugodenaranja, etc.

Tabla 4
RF3. Cola de animación

ID - Nombre	RF3 - Cola de Animación
Descripción	Al recibir letras, palabras, frases u oraciones posteriormente a su interpretación, el programa debe ser capaz de mantener una estructura de datos para su posterior animación en el orden correcto.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	Si tres elementos son ingresados, estos deben ser devueltos según el orden en el que ingresan, es decir la primera en colocarse es la primera en salir.

Tabla 5
RF4. Animación

ID - Nombre	RF4 - Animación
Descripción	Al obtener una letra, palabra, frase u oración de la cola de animación, el modelo del personaje debe reproducir la animación correspondiente.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	Al no haber elementos dentro de la cola y luego de un lapso de tiempo, el modelo del personaje debe entrar a una animación de descanso, esperando que nuevos elementos ingresen a la cola para ser animados.

Tabla 6
RF5. Deletreo de palabras

ID - Nombre	RF5 - Deletreo de palabras
Descripción	Las palabras ingresadas y entidades que no posean una animación correspondiente, deben ser descompuestas en letras y posteriormente colocadas en la cola de animación para su posterior animación.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	Al ingresar “Darpa”, palabra sin animación correspondiente, la cola deberá contener los elementos “d”, “a”, “r”, “p”, “a” que serán animados de izquierda a derecha.

Tabla 7
RF6. Velocidad de animaciones

ID - Nombre	RF6 - Velocidad de animaciones
Descripción	Se debe proveer por medio de la interfaz, controles que permitan a los usuarios manipular la velocidad en que se reproducen las animaciones.
Prioridad	Media.
Frecuencia de uso	Baja.
Otra información	La velocidad para reproducir animaciones correspondientes a deletreo debe manejarse por separado a la velocidad para las otras animaciones.

Requerimientos no funcionales

Estos requerimientos especifican criterios que pueden usarse para evaluar las funcionalidades de un sistema o sus componentes, por lo tanto, no se refieren a los requisitos que describen funciones, sino a características de funcionamiento. A continuación, se listan los requerimientos no funcionales.

Tabla 8
RNF1. Interfaces sencillas

ID - Nombre	RNF1- Interfaces sencillas
Descripción	El programa debe poseer interfaces sencillas, que no afecten la atención del usuario. Además, las interfaces deben ser fáciles de comprender por los usuarios.
Prioridad	Alta.
Frecuencia de uso	Alta.

Otra información	Se debe proveer de la menor cantidad de opciones de configuración y contenido para no desviar la atención del usuario. Los menús y opciones del programa deben ser explícitas.

Tabla 9
RNF2. Tiempo de interpretación

ID - Nombre	RNF2- Tiempo interpretación
Descripción	El tiempo de respuesta entre el envío del mensaje y la interpretación debe ser menor a tres segundos.
Prioridad	Media.
Frecuencia de uso	Alta.
Otra información	El tiempo debe ser evaluado hasta el momento en que la petición ingresa a la cola de animación, sin considerar el tiempo de ejecución de la misma.

Tabla 10
RNF3. Múltiples usuarios

ID - Nombre	RNF3 - Múltiples usuarios
Descripción	El sistema de interpretación debe permitir el uso simultáneo entre distintos usuarios.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	Dos usuarios, ejecutando la aplicación en diferentes procesos, deben ser capaces de utilizar el sistema sin inconvenientes.

Tabla 11
RNF4. Procesamiento de caracteres

ID - Nombre	RNF4 - Procesamiento de caracteres
Descripción	El sistema de procesamiento del lenguaje natural debe ser capaz de procesar textos de hasta 4096 caracteres en menos de 1 segundo.
Prioridad	Alta.
Frecuencia de uso	Alta.
Otra información	4096 caracteres es el tamaño máximo de un mensaje en Telegram, por lo que se considera el tamaño máximo a procesar. 4096 caracteres equivalen a 1.36 páginas de un documento de MS Word con fuente de tamaño 11 puntos (0.38 cm).

Tabla 12
RNF5. Aspecto del modelado 1

ID - Nombre	RNF5 - Aspecto del modelo 1
Descripción	Los modelos de personajes deben tener el rostro y manos descubiertas.
Prioridad	Medio.
Frecuencia de uso	Alto.
Otra información	Los modelos con cabello deben evitar que este cubra el rostro. La vestimenta no debe incluir accesorios, como guantes o brazaletes, ni cubrir las manos o rostro del personaje.

Tabla 13
RNF6. Aspecto del modelado 2

ID - Nombre	RNF6 - Aspecto del modelo 2
Descripción	La vestimenta de los modelos de personajes debe utilizar tonalidades no brillantes y sin elementos como figuras, dibujos, patrones o estampados.
Prioridad	Medio.
Frecuencia de uso	Alto.
Otra información	Debe utilizarse tonalidades que no destaquen para el diseño de la vestimenta. La vestimenta debe ser monocromática.

CASOS DE USO

Para el análisis y diseño de la solución, se hará uso de las técnicas de casos de uso mediante diagramas UML, ya que permiten obtener elementos útiles para el análisis y diseño de sistemas, tales como:

- Funciones que el sistema debe cubrir.
- Requerimientos funcionales del sistema.
- Flujos de acciones en el sistema.
- Roles de los actores del sistema
- Identificar factores internos y externos del sistema, por ejemplo, un sistema puede ser factor externo de otro sistema.

El proyecto está dividido en dos subsistemas principales: Cliente NLP y Cliente Animado 3D; que combinados entre ellos y junto a otros servicios de terceros, brindarán la

funcionalidad completa de la solución. A continuación, se explica de manera más detallada los aspectos de ambos sistemas.

Cliente NLP

El cliente de Procesamiento de Lenguaje Natural cumple con la primera parte del objetivo general, interpretar el habla y texto en español. El cliente NLP será dividido en dos partes: una versión web básico con soporte para texto y voz, y un cliente implementado en un servicio de mensajería para recibir mensajes de texto y notas de voz para poder interpretarlos y transmitirlos al módulo de interpretación 3D para ser representados.

Descripción de los actores.

Tabla 14
Descripción De Los Actores

Nombre	**Descripción**
Usuario común Viñeta “a”, figuras 11 y 12	El usuario común del sistema es capaz de enviar mensajes al sistema cliente 3D por medio de la interfaz web o por medio del servicio de mensajería.
Sistema cliente 3D Viñeta “f” figura 11 Viñeta “h” figura 12	El sistema cliente 3D realiza la interpretación del mensaje recibido a través del modelo animado.
Sistema cliente NLP Viñeta “e” figura 11 Viñeta “g” figura 12	El sistema cliente NLP realiza la interpretación del mensaje en lenguaje natural, lo procesa y transmite al sistema cliente 3D.

Diagrama de Caso de Uso.

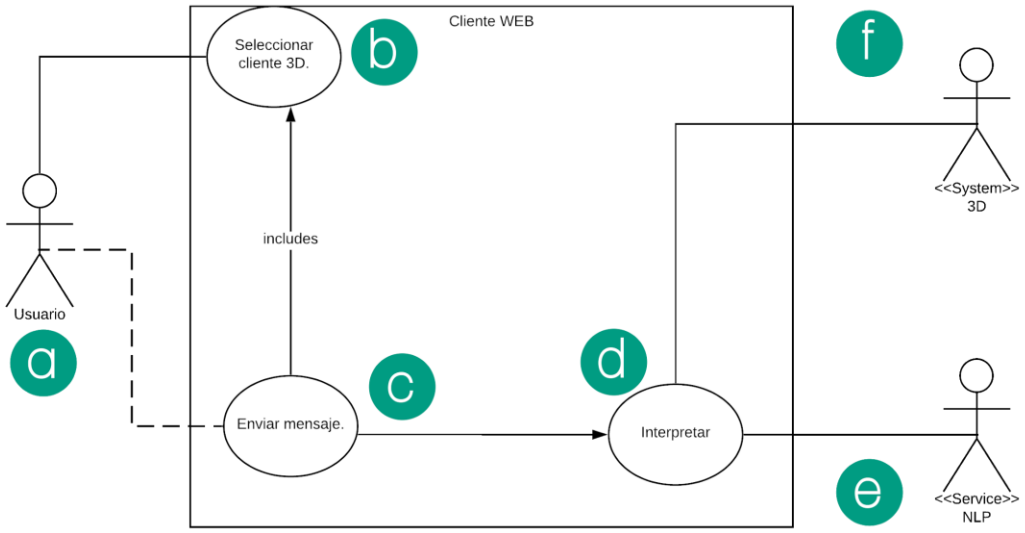

Figura 11. Diagrama caso de uso para Cliente Web

Especificación del Caso de Uso.

Tabla 15
Especificación Del Caso De Uso

Nombre	**Descripción**
Seleccionar cliente 3D. Viñeta “b” figura 11	El usuario debe seleccionar el cliente 3D al que enviará los mensajes a ser interpretados.
Enviar mensaje. Viñeta “c” figura 11	El usuario del sistema realiza el envío de mensajes de texto por medio de la interfaz web.
Interpretar. Viñeta “d” figura 11	El sistema NLP realiza la interpretación y transmite el mensaje al cliente 3D.

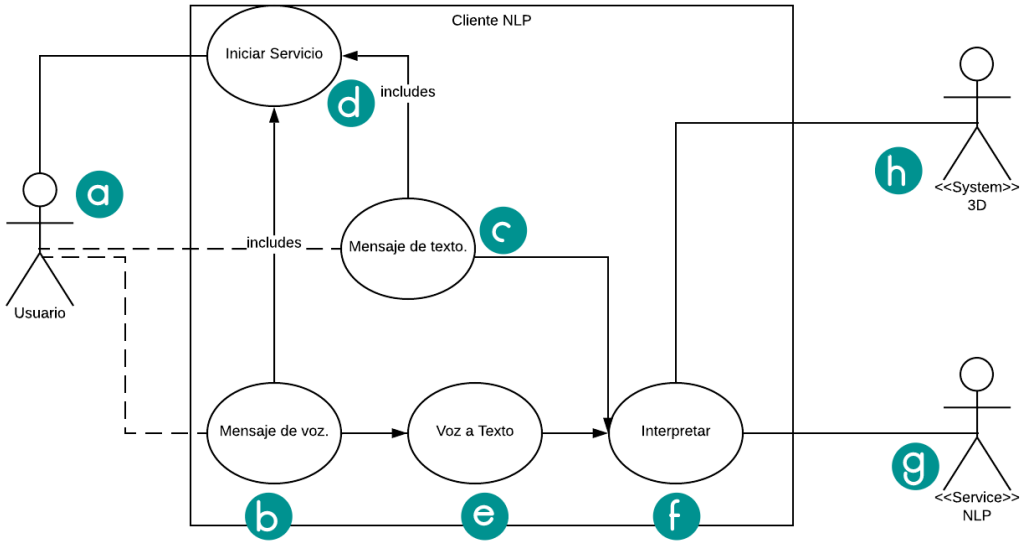

Figura 12. Diagrama caso de uso para Cliente NLP

Especificación de los Casos de Uso.

Tabla 16
Especificación Del Caso De Uso

Nombre	**Descripción**
Mensaje de voz. Viñeta “b” figura 12	El usuario puede utilizar el servicio de mensajería para enviar mensajes de voz (notas de voz en la aplicación) por medio de la interfaz por defecto que ofrece el servicio seleccionado.
Mensaje de texto. Viñeta “c” figura 12	El usuario puede utilizar el servicio de mensajería para enviar mensajes de texto al cliente 3D por medio de la interfaz por defecto del servicio seleccionado.

Iniciar servicio. Viñeta “d” figura 12	Para hacer uso de cualquiera de los servicios que ofrece el servicio de mensajería, debe iniciarse el mismo. Al iniciar el servicio se hará la configuración básica del mismo.
Voz a texto. Viñeta “e” figura 12	En caso que el usuario decida enviar una nota de voz por medio del servicio de mensajería, el sistema debe convertir la nota de voz en texto para poder ser procesado normalmente.
Interpretar. Viñeta “f” figura 12	La interpretación del texto, independiente del tipo de mensaje que se recibió, se realiza en el cliente NLP y luego se transmite el mensaje al cliente 3D.

Cliente animado 3D

Esta sección muestra la segunda parte del objetivo principal, que es mostrar la traducción en gestos a través de un modelo 3D. Lo que se busca es que al recibir la información o datos que se necesitan interpretar, previamente procesadas, este muestre el resultado mediante un modelo humano tridimensional.

Diagrama de casos de uso.

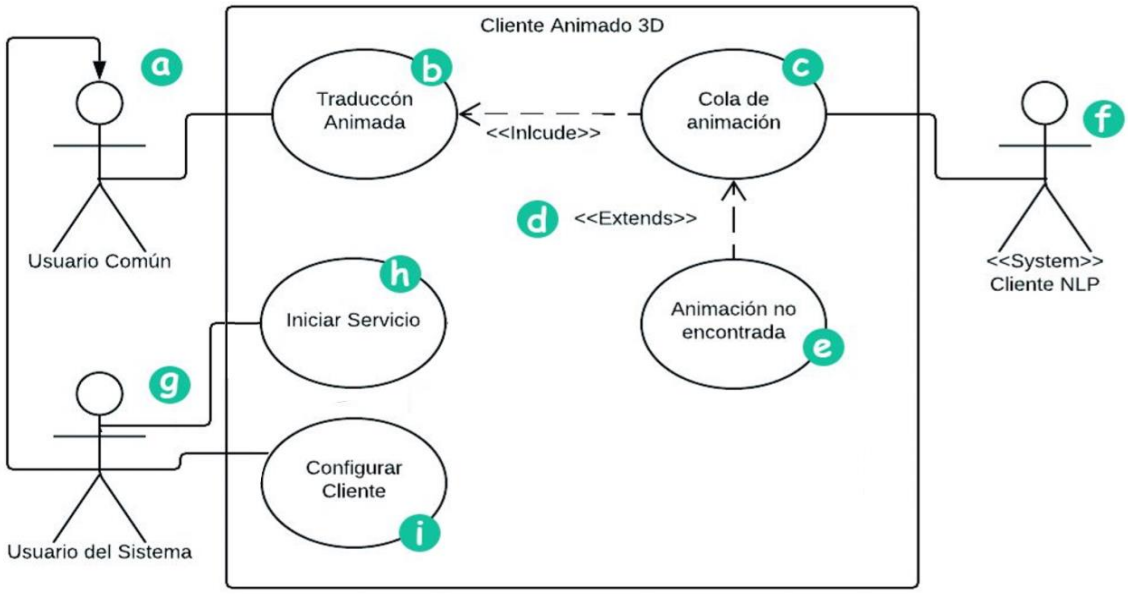

Figura 13. Diagrama caso de uso para Cliente animado 3D

Descripción de los actores

Tabla 17
Descripción De Los Actores, Cliente animado 3D

Nombre	**Descripción**
Usuario común Viñeta “a” figura 13	El usuario principal para este sistema. Este usuario es aquel que abarca a todo individuo que puede hacer uso de la aplicación sin privilegios de manejo del sistema.
Usuario del sistema Viñeta “b” figura 13	Además de heredar todas las características y comportamiento de un usuario común posee también otros roles administrativos. Un usuario del sistema es el encargado de administrar el Cliente Animado 3D, para realizar acciones como su ejecución y configuración.

Sistema cliente NLP Viñeta “f” figura 13	El sistema cliente NLP es el encargado de procesar el audio ingresado por los usuarios comunes a fin de ser transmitido al sistema cliente 3D. Se encarga de brindar interfaz de usuario, procesamiento y transmisión de la información.

Especificación de los Casos de Uso.

Tabla 18
Especificación Del Caso De Uso, Cliente animado 3D

Nombre	**Descripción**
Traducción animada Viñeta “b” figura 13	Recurso a consumirse por el usuario principal, es el resultado final del sistema que brinda la traducción del lenguaje oral o escrito, hacia el lenguaje de señas LESSA por medio de un modelo 3D. Este caso inicia una vez que un usuario del sistema inicie los servicios y que además existan elementos dentro de la cola de animación.
Cola de animación Viñeta “c” figura 13	Este se encarga de buscar la existencia o no de las animaciones solicitadas y crear una estructura de cola con todas las animaciones encontradas. Este proceso comienza cada vez que se recibe un mensaje desde el sistema externo “Cliente NLP”. En el caso en que este proceso no encuentre una animación correspondiente a la solicitada, pasará a un proceso alternativo llamado Animación No Encontrada
Animación no encontrada Viñeta “e” figura 13	Este es un proceso alternativo, que busca proveer animaciones sustitutas, como las animaciones necesarias para el deletreo de

	una palabra en lugar de una animación específica, y además proveer un mecanismo de registro para la concurrencia de estos casos.
Iniciar servicio Viñeta “h” figura 13	Al iniciar el cliente, este permite cambiar hacia la pantalla principal, iniciando los servicios necesarios para llevar a cabo la traducción animada, que será utilizada por el usuario común.
Configurar cliente Viñeta “i” figura 13	Permite realizar configuraciones para el cliente, desde configuraciones visuales como seleccionar colores para el fondo, hasta opciones funcionales como la velocidad de animaciones o cambio de personaje.

DISEÑO DE INTERFACES

MiaWeb

La interfaz de la versión web del sistema consiste en dos secciones principales. La primera está marcada con el rectángulo rojo y consiste en la interfaz de mensajería por texto para comunicarse con las instancias del servicio 3D (Figura 14). El listado de instancias o dispositivos disponibles para comunicarse se muestran en la sección de dispositivos resaltada en el rectángulo de color azul.

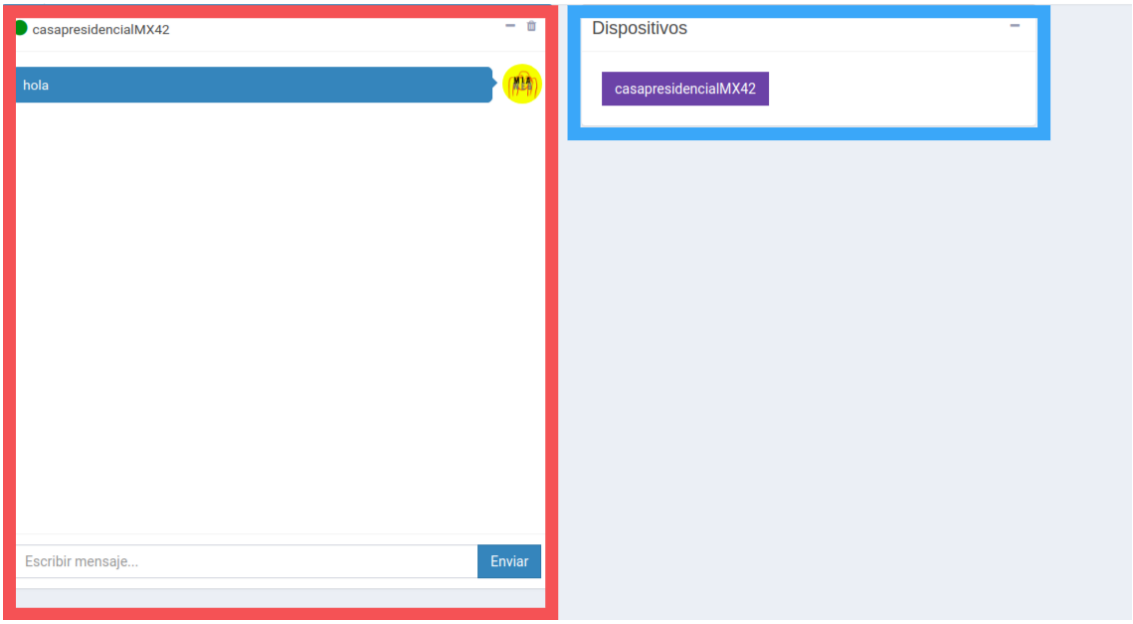

Figura 14. Prototipo interfaz 1, Mia web

La interfaz de la sección de mensajería está compuesta de 5 partes. En la sección resaltada en color rosa se muestra el nombre del servicio 3D al que se está conectado (Figura 15). En la sección resaltada en color púrpura se muestra una botonera para minimizar el panel de mensajes o para eliminar los mensajes que se encuentran en el momento en el diálogo.

En la sección resaltada en color azul se muestra la lista de mensajes que el usuario envía en orden cronológico, mientras que el cuadro de texto de la sección resaltada en color verde se utiliza para escribir el mensaje que se enviará al servicio 3D. Para enviar el mensaje se hace uso de botón “Enviar” resaltado en la sección marcada de color amarillo.

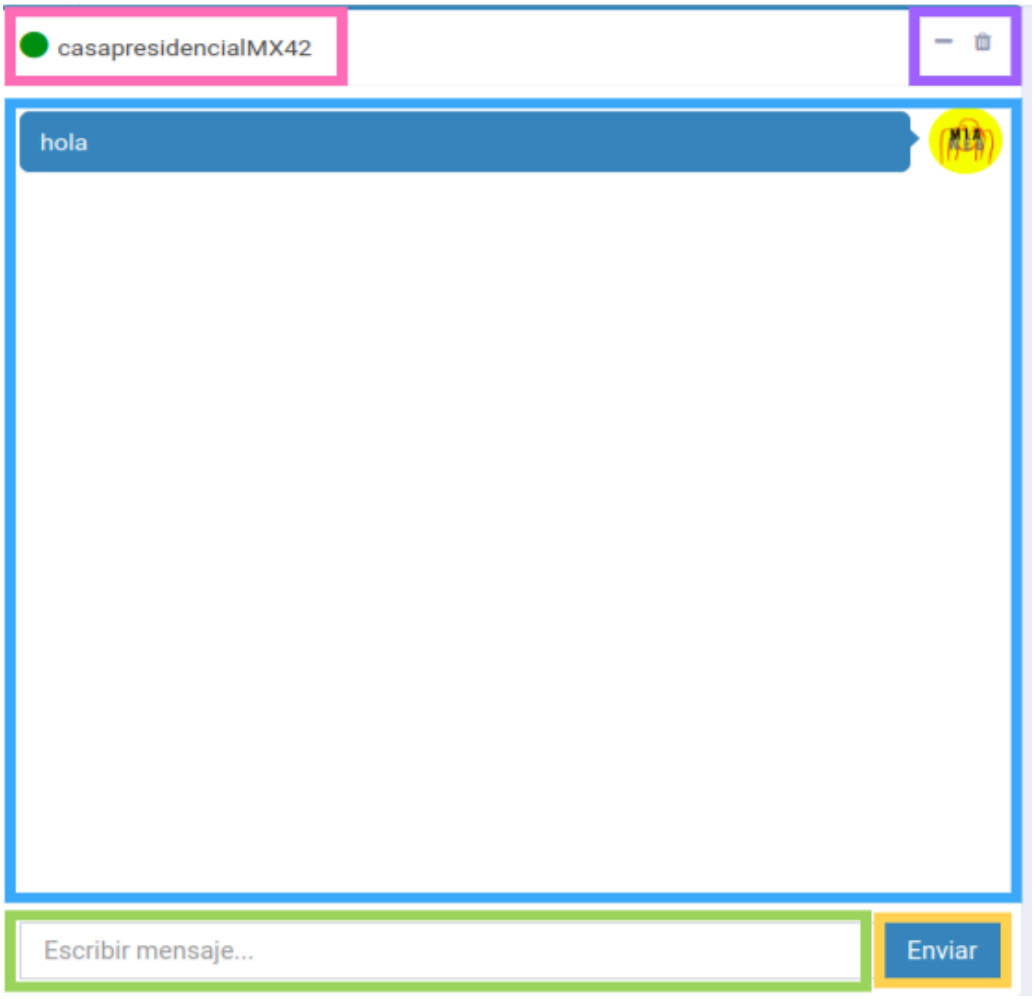

Figura 15. Prototipo interfaz 2, Mia web

En la sección de dispositivos existen dos secciones. La sección resaltada en color azul muestra el nombre de las instancias disponibles para mostrar mensajes, mientras que la sección resaltada en color rojo muestra un botón para minimizar u ocultar la lista de dispositivos (Figura 16).

Figura 16. Prototipo interfaz 3, Mia web

@lessamiabot

La interfaz de usuario del cliente de Procesamiento del Lenguaje Natural es la interfaz del servicio de mensajería Telegram. Para hacer uso del cliente NLP se utilizará el Telegram Bots API para construir un bot que permita interactuar con el resto del sistema.

El usuario de Telegram del bot es *@lessamiabot*. Al buscar el usuario en Telegram aparece un botón en la parte inferior que dice “Iniciar bot” o “Reiniciar bot” como se puede observar en la Figura 17 dentro del rectángulo rojo.

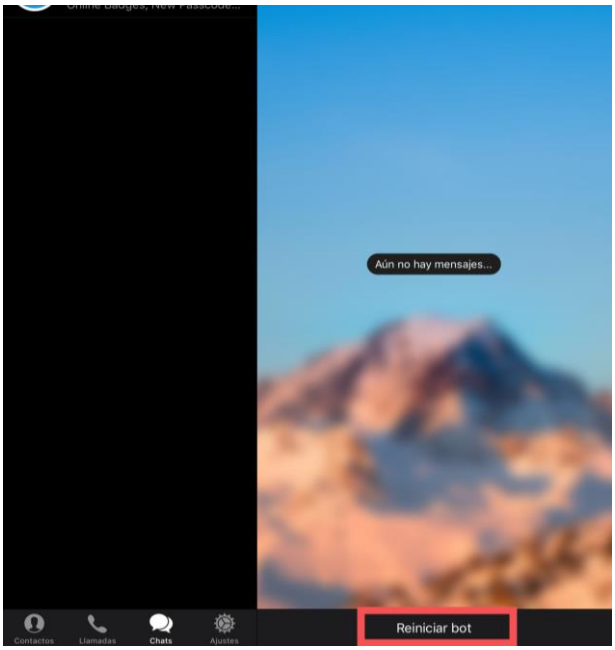

Figura 17. Ejemplo de interfaz 1, Telegram

Iniciar servicio. Después de presionar el botón “Reiniciar bot”, se muestra la información inicial del servicio. El rectángulo blanco de la Figura 18 muestra la información general sobre el servicio. En el rectángulo amarillo se encuentran el comando inicial “/start” y la opción para configurar el bot para mostrar o no acuse de recibo para cada mensaje de voz o texto que se envíe. En la parte inferior del servicio, se muestra una botonera (Figura 18,

rectángulo rojo) que muestra las palabras “SI” y “NO”, aceptando o rechazando la opción de acuse de recibo.

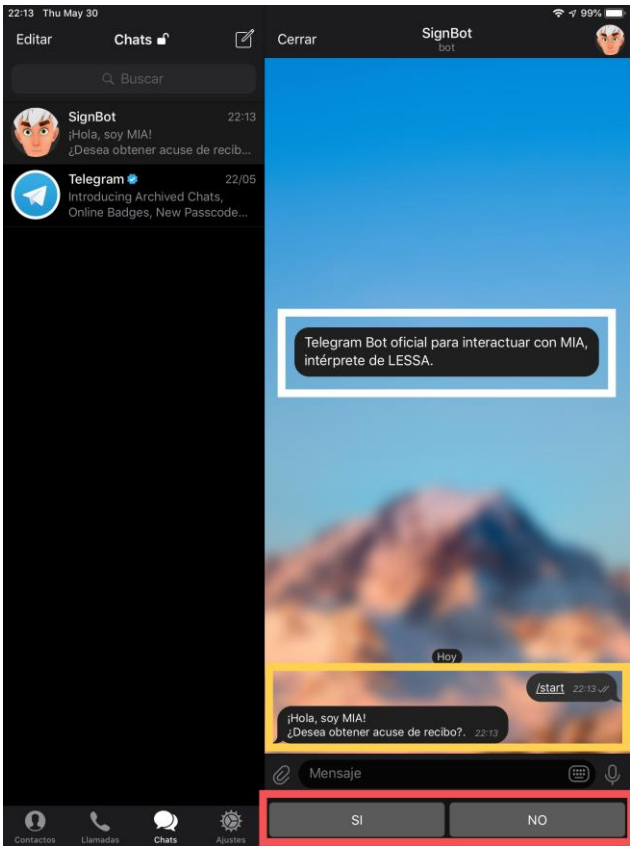

Figura 18. Ejemplo de interfaz 2, Telegram

Ayuda e interacción con el servicio. En el rectángulo amarillo se muestra la confirmación de la configuración de acuse de recibo y ofrece ayuda al usuario. Al solicitar ayuda (Figura 19, rectángulo rojo) con el comando “/help”, el bot muestra la información de ayuda que ofrece el bot y la lista de comandos y formas de interactuar con el mismo. En el rectángulo verde de la figura, se muestra la forma de interactuar con el servicio. Se puede enviar mensajes de texto o mensajes de voz, recibiendo en ambos casos la palabra “OK” como acuse de recibo. Finalmente (Figura 19, rectángulo rosa), con el comando “/cancel”, se termina el servicio y el bot se despide del usuario.

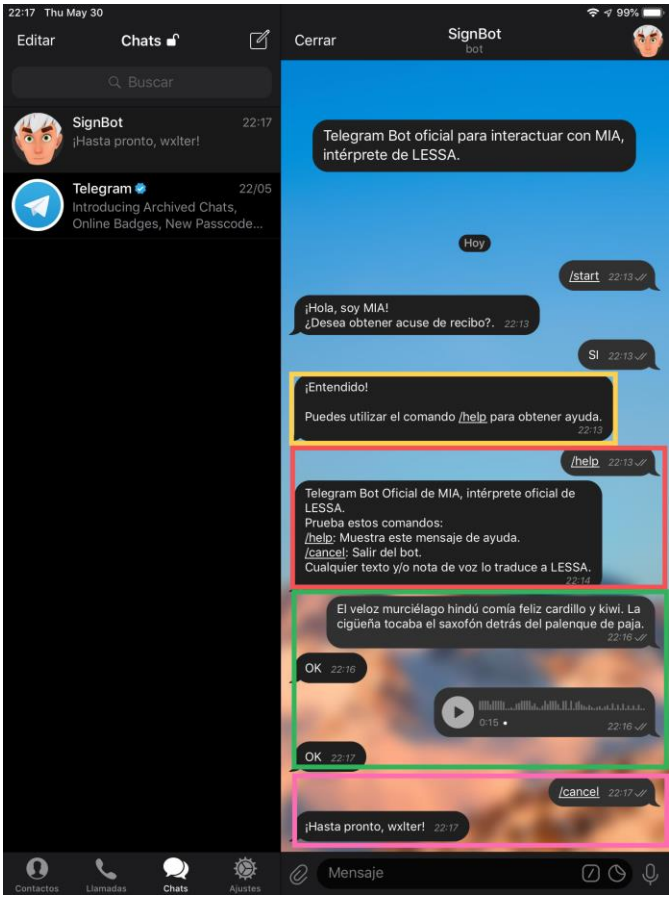

Figura 19. Ejemplo de interfaz 3, Telegram

Acuse de recibo desactivado. Al iniciar el servicio y configurar el bot, el mismo es quien pregunta al usuario si desea activar la opción de acuse de recibo. En caso de responder "NO", el servicio funcionará de la misma manera que se describió previamente, pero sin responder con la palabra "OK" para cada interacción como se puede observar en la Figura 20 en el rectángulo rojo.

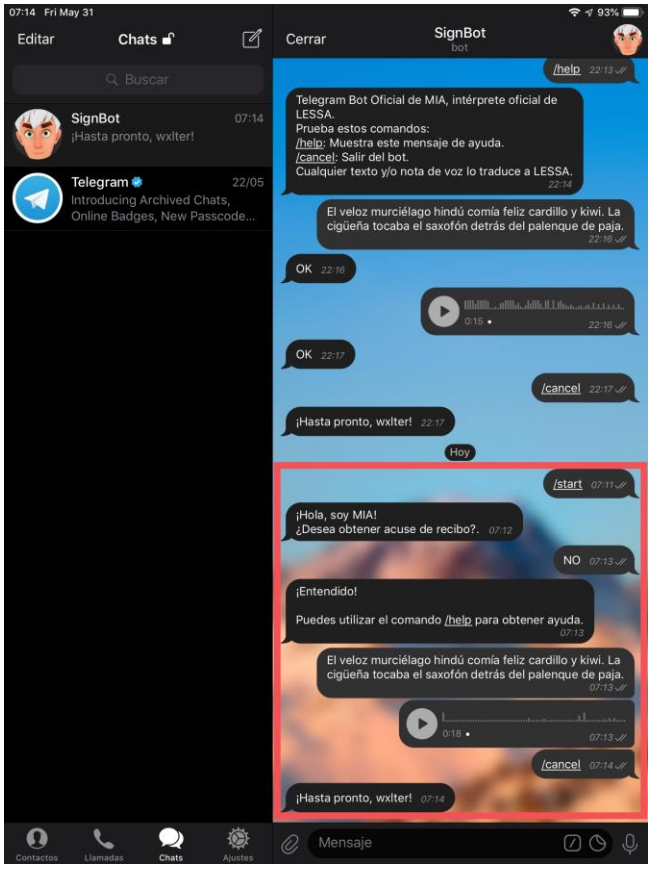

Figura 20. Ejemplo de interfaz 4, Telegram

Cliente animado 3D

El diseño de las interfaces de este cliente está orientado hacia sus principales usuarios, que son aquellas personas con discapacidad auditiva. Se han retomado las sugerencias y comentarios obtenidos mediante las entrevistas realizadas a personas con discapacidad, traductores y otras relacionadas a esta área.

En cuanto las interfaces, el principal punto a tomar en cuenta es la simpleza en general, tanto en colores, animaciones, figuras y transiciones, ya que las personas con discapacidad auditiva, especialmente si su uso es para niños o jóvenes, pueden generar distracción y por ende afectar en cómo el mensaje es captado por los usuarios.

Los colores principales del cliente se basan en el color blanco y tonos de colores neutros. Los colores pueden ser modificados por medio de configuraciones personales por parte de los usuarios.

Menú de configuración. Este menú estará disponible en todas las pantallas del cliente, ya sea en estado oculto o visible. En estado oculto, este menú únicamente muestra dos íconos, como se observa en la Figura 21. El primer ícono marcado con la viñeta “a” permite cambiar el estado del menú, pasando de esta oculto a visible y viceversa. El segundo ícono marcado con la viñeta “b” permite realizar un cierre ordenado de la aplicación.

Figura 21. Prototipo interfaz: menú de configuración en estado oculto, Mia 3D

En la Figura 22 se puede observar este menú en estado visible, este menú estará compuesto por un recuadro con transparencia señalado por el recuadro rojo, que listará una serie de opciones utilizando elementos como controles deslizantes, botones o paneles para realizar las configuraciones del cliente. En la parte superior señalada por el recuadro amarillo está el botón para el cierre de aplicación, los botones mostrar y ocultar que estarán siempre visibles para el usuario.

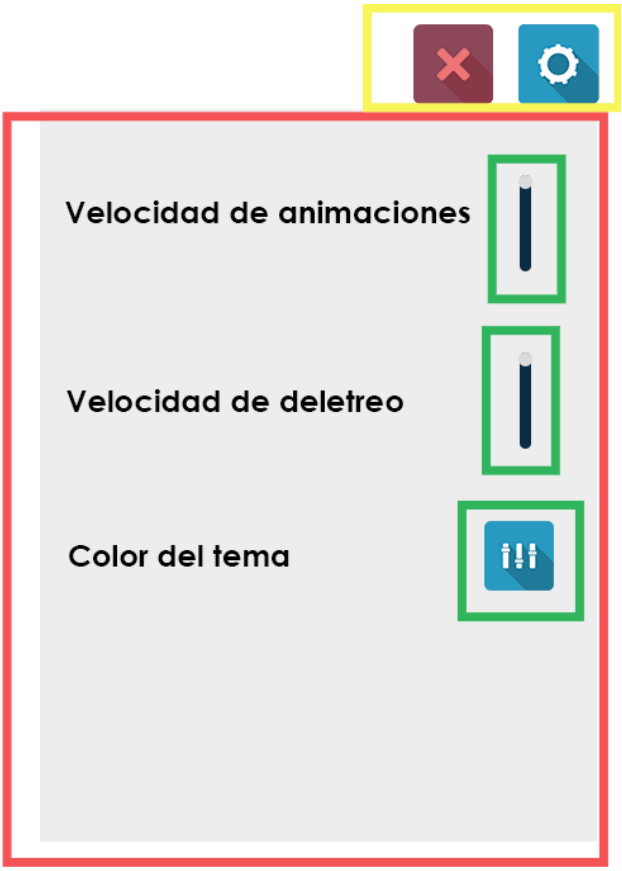

Figura 22. Prototipo interfaz: menú de configuración en estado visible, Mia 3D

Pantalla de inicio y reposo. Esta pantalla será presentada en la aplicación antes de la pantalla principal, y cuando la pantalla principal entre en un estado de inactividad después de un determinado tiempo. La Figura 23 muestra esta pantalla, que contendrá un fondo blanco con leve degradado, un logo del proyecto al centro, señalado por el recuadro rojo, para identificar la aplicación. Si el usuario desea continuar hacia la pantalla principal deberá dar click sobre un botón “Iniciar” ubicado en la parte central baja que consistirá en un icono tal como se muestra en el recuadro amarillo.

Figura 23. Prototipo interfaz: pantalla de inicio cliente animado, Mia 3D

En la parte superior derecha, marcado por el recuadro verde, se encuentran el menú de configuración con todas sus funcionalidades previamente expuestas.

Pantalla principal. Esta pantalla contendrá un fondo sólido sobre un escenario en tres dimensiones, el cual puede verse afectado visualmente por los componentes de iluminación en la escena, que pueden proyectar sombras y distintos efectos. Al centro de esta pantalla se encuentra el modelo humano tridimensional responsable de las animaciones, esta área se encuentra marcada por un recuadro amarillo en la Figura 24, figura que corresponde a la pantalla principal.

Por último, en la parte superior izquierda, marcado con un recuadro rojo, se encuentra un botón con un ícono en forma de flechas, que nos permitirá regresar a la pantalla de inicio o reposo, acción que también puede ser desencadenada por un período de inactividad en la pantalla principal.

Figura 24. Prototipo interfaz: pantalla principal cliente animado, Mia 3D

En la parte superior derecha, marcado por el recuadro verde, se encuentra el menú de configuración con todas sus funcionalidades previamente expuestas.

DATOS DE ANIMACIONES

Es necesario utilizar algún mecanismo de almacenamiento de datos para almacenar la información correspondiente a las distintas animaciones de LESSA, con el objetivo de posteriormente consultarlos, procesarlos y utilizar su resultado para reproducir la animación correspondiente al proceso de interpretación. Para llevar a cabo esta tarea se optó por utilizar una única tabla de datos ya que resulta suficiente para lograr cubrir las necesidades de la solución.

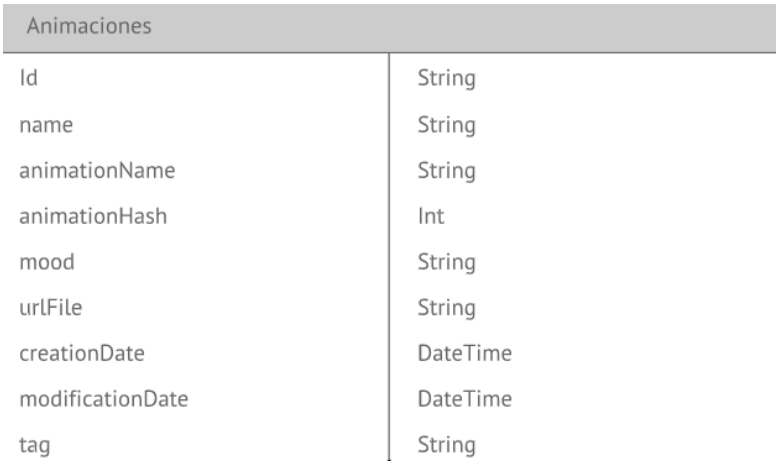

Figura 25. Modelo de datos de animaciones

Tabla 19
Modelo de datos de animaciones Descripciones

Id	**Es un identificador único para cada entrada de datos.**
Name	Este campo se utiliza para indicar el nombre de la letra, palabra o conjunto de estas, que posean una correspondencia en LESSA.
animationName	Indica el nombre de la animación que corresponde a la palabra.
animationHash	Id único generado a partir del nombre de la animación.
Modo	Indica la expresión facial correspondiente a cada animación.
urlFile	Url del archivo que contiene la animación.
creationDate	Fecha de creación de la entrada.
modificationDate	Fecha de la última modificación realizada.
Tag	Etiqueta utilizada para crear clasificaciones.
User	Último usuario en modificar la entrada.

Los últimos cinco campos de esta tabla no son requeridos para la funcionalidad de la aplicación, pero han sido agregados para llevar un control y organización de los datos.

TECNOLOGÍAS UTILIZADAS

Para desarrollar una solución que cumpla con los requerimientos, se requiere el uso de una variedad de software que formen parte de un flujo de desarrollo y se complementen entre ellos para brindar la funcionalidad. Con el fin de escoger las mejores alternativas se hace uso de una matriz de toma de decisiones personalizada, la cual brindará como resultado un puntaje que determinará el cumplimiento de una alternativa. La matriz listará diversas opciones y criterios a evaluarse para cada una de ellas, como se muestra en la siguiente tabla.

Tabla 20
Tabla De Importancia De Tecnologías

Criterios	Importancia	Opciones			
		Opción 1	Opción 2	Opción 3	Opción 4
Criterio 1	1	5	0	0	5
Criterio 2	2	5	0	5	0
Puntaje total:	15	15	0	10	5

Cada opción será puntuada con respecto a los criterios de cada comparativa, y estos valores indican el grado de cumplimiento. Los puntajes van de 0 a 5, un valor de 0 indica que no cumple con el criterio; y un valor de 5 indica que cumple completamente con el criterio.

Tabla 21
Escala De Puntuación

Escala de puntuación	
No Cumple	0
	1
	2
	3
	4
Cumple	5

Además, cada criterio tiene su propia importancia, entre mayor sea este valor, más relevante se vuelve dicho criterio. Este factor de importancia se multiplicará por la puntuación que cada opción obtiene en los distintos criterios. El puntaje final de cada opción será la sumatoria de cada uno de sus puntajes obtenidos multiplicados por la respectiva importancia de los criterios. Es importante destacar que las alternativas comparadas han sido previamente evaluadas y seleccionadas por lo que, otras alternativas quedaron fuera.

El primer parte del proyecto consiste en poder interpretar la voz y el texto y descomponerlo en oraciones, palabras y entidades nombradas (clasificación en categorías predefinidas) para poder transmitirlo al módulo de interpretación 3D. Además, se debe extraer palabras vacías de las oraciones (todas aquellas que no aportan significado a la oración) y proveer de información extra como palabras lematizadas, género y número. Para lograrlo, es necesario el uso de diferentes tecnologías: lenguaje de programación, intérprete de voz a texto y librería de procesamiento de lenguaje natural.

Para la segunda parte, se requiere representar una persona en un espacio tridimensional, con diferentes características; tales como tamaño, forma, texturas, expresiones y movimientos los cuales responden al resultado obtenido por un intérprete y se requiere que pueda mostrarse en tiempo real. Para lograr esto es necesario utilizar software que permita la manipulación de gráficos 3D y un motor de videojuegos para su ejecución en tiempo real.

A continuación, se describen brevemente las posibles alternativas para el desarrollo de la solución, las comparativas y decisiones realizadas entre dichas alternativas.

Lenguaje de Programación para NLP

Un paso muy importante para la selección de tecnologías consiste en elegir un Lenguaje de Programación que se ajuste a las necesidades de desarrollo del software. Existe una variedad de lenguajes de programación, pero en base a previa selección se tienen las siguientes opciones:

C++. Este es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C con mecanismos que

permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Java. Este es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "*write once, run anywhere*"), lo que quiere decir que el código es ejecutado en una plataforma y no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados.

Python. Este es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1.

Los criterios de selección son:

- **Simplicidad.** Facilidad para desarrollar en el lenguaje de programación seleccionado. Que el código no necesite de pasos innecesarios debido solamente al lenguaje de programación.
- **Velocidad de ejecución.** Velocidad de ejecución del software en los distintos lenguajes basados en evaluaciones y opiniones de terceros.
- **Multiplataforma.** Que el código pueda ejecutarse en distintos sistemas operativos.
- **Comunidad.** Que la comunidad de usuarios del lenguaje seleccionado sea extensa para solucionar problemas y dudas sobre el mismo durante el desarrollo.

- **Sintaxis.** Que el lenguaje de programación sea fácilmente entendible y sea lo suficientemente legible para otros desarrolladores.
- **Programación orientada a objetos.** Que el lenguaje de programación cumpla con los paradigmas de la Programación Orientada a Objetos.
- **Experiencia previa.** Experiencia previa por parte de los integrantes del grupo de trabajo en el lenguaje de programación para disminuir la curva de aprendizaje en el mismo y poder enfocarse en tareas específicas de NLP.
- **Librerías para NLP.** Que el lenguaje de programación tenga soporte de librerías de NLP.
- **Uso de memoria.** El uso de memoria por las aplicaciones escritas en el lenguaje seleccionado sea óptimo para poder ejecutarse en sistemas con poca memoria.
- **Uso de procesador.** El uso de procesador por las aplicaciones sea óptimo para poder ejecutarse en sistemas de bajo coste.

Tabla 22
Calificación De Importancia De Lenguajes De Programación

		Opciones		
Criterios	Importancia	Python	Java	C++
Simplicidad	3	5	4	4
Velocidad de ejecución	4	4	3	5
Velocidad de desarrollo	4	5	3	3
Multiplataforma	4	5	5	5
Comunidad	2	5	5	4
Sintaxis	1	5	4	4
Orientado a objetos	4	5	5	5
Experiencia previa	3	4	3	1
Librerías NLP	4	4	5	2
Uso de memoria	3	4	2	5
Uso de procesador	3	3	4	5
Puntaje total:	175	155	137	137

Luego de realizar la comparativa entre las diferentes opciones de lenguajes de programación, con un puntaje máximo de 190 puntos, se selecciona como lenguaje de programación para el módulo de procesamiento de lenguaje natural a Python, con un puntaje de 155 puntos. Se hace esta elección principalmente por la simplicidad, velocidad de desarrollo y por la experiencia previa que se cuenta en dicho lenguaje. Además, ofrece beneficios como el uso de memoria y procesador eficientes.

Reconocimiento de voz

Una vez conocido el lenguaje de programación a utilizar, el siguiente paso es seleccionar una librería de reconocimiento de voz que se adapte a las necesidades del software. Existen muchas librerías de reconocimiento de voz que pueden ser utilizadas con Python, pero la lista se puede resumir en dos de las principales. A continuación, se comparan y se selecciona una de ellas para ser utilizada en el proyecto.

SpeechRecognition. La librería SpeechRecognition actúa como un contenedor para varias API de voz populares y, por lo tanto, es extremadamente flexible. Uno de ellos, la API de Google Cloud Speech, es compatible con una clave de API predeterminada que está codificada en la biblioteca SpeechRecognition. Eso significa que puede desplegarse sin tener que registrarse para el servicio. SpeechRecognition requiere entrada de audio, y SpeechRecognition hace que la recuperación de esta entrada sea realmente fácil. En lugar de tener que crear scripts para acceder a los micrófonos y procesar archivos de audio desde cero, SpeechRecognition lo pone en funcionamiento en tan solo unos minutos. La flexibilidad y la facilidad de uso del paquete SpeechRecognition lo convierten en una excelente opción.

Google Cloud Speech. Google Cloud Speech es una API que permite a los desarrolladores convertir audio en texto fácilmente gracias a la aplicación de modelos de redes neuronales. Esta API reconoce 120 idiomas y variantes. Además, utiliza la tecnología del aprendizaje automático de Google para procesar audios grabados previamente o en tiempo real. Aplica los algoritmos más avanzados de aprendizaje profundo y redes neuronales a los audios para conseguir un reconocimiento de voz con precisión. Con Transcripción de voz de Cloud, es

posible identificar el idioma en que se pronuncia una frase (hasta un máximo de cuatro idiomas), lo cual resulta especialmente útil en búsquedas por voz.

Los criterios de selección son:

- **Volumen de consultas.** Volumen de consultas que pueden realizarse de manera gratuita.
- **Soporte para español.** Que la librería tenga soporte para idioma español.
- **Velocidad de respuesta.** La velocidad de respuesta de la librería a las peticiones.
- **Precio.** Relación precio/cantidad de peticiones. Menor es mejor.
- **Soporte para micrófono.** Soporte sin necesidad de código extra para entrada de audio directa de audio desde un micrófono.
- **Soporte para archivos de audio.** Soporte para transcripción directa desde archivos.

Tabla 23
Calificación De Importancia De Tecnologías De Reconocimiento De Voz

Criterios	Importancia	Opciones	
		SpeechRecognition	Google Cloud Speech
Volumen de consultas.	4	5	3
Soporte para Español.	4	5	5
Velocidad de respuesta.	2	5	5
Precio.	3	5	3
Soporte Micrófono.	3	5	3
Soporte Archivos de Audio.	3	5	5
Puntaje total:	95	95	75

SpeechRecognition hace uso de Google Cloud Speech, pero presenta ciertas ventajas como el volumen de consultas ilimitado de manera gratuita comparado con los sesenta minutos mensuales de Google Cloud Speech. Además, SpeechRecognition tiene soporte nativo para entrada de micrófono, lo que lo hace ideal para el proceso de desarrollo y depuración.

Procesamiento del Lenguaje Natural

Se debe seleccionarse la librería de Python que se utilizará para el procesado del lenguaje natural. Python es el lenguaje de referencia para software científico y relacionado a la Inteligencia Artificial, por lo que existe una amplia lista de librerías para procesado de lenguaje natural. Se realiza una comparación entre las principales librerías para determinar la que mejor se adapte a las necesidades del proyecto.

NLTK. El kit de herramientas de lenguaje natural, o NLTK, es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural en el lenguaje de programación Python. NLTK está destinado a apoyar la investigación y la enseñanza en PLN o áreas muy relacionadas, que incluyen la lingüística empírica, las ciencias cognitivas, la inteligencia artificial, la recuperación de información, y el Machine Learning. NLTK se ha utilizado con éxito como herramienta de enseñanza, como una herramienta de estudio individual, y como plataforma para los sistemas de investigación de prototipos y construcción.

Spacy. Esta es una librería de código abierto para el procesado del lenguaje natural, escrita en Python y Cython. La librería está publicada bajo la licencia MIT y actualmente ofrece modelos de redes neuronales estadísticos para inglés, alemán, español, portugués, francés, italiano, neerlandés y reconocimiento de entidades nombradas multi-idioma, así como tokenización para otros idiomas.

A diferencia de NLTK, que es ampliamente usado para enseñanza e investigación, SpaCy se centra en proveer software listo para producción. También soporta flujos de trabajo de aprendizaje profundo que permiten conectar modelos estadísticos entrenados por librerías populares de Machine Learning como TensorFlow, Keras, Scikit-Learn o PyTorch. La librería de Machine Learning de SpaCy, Thinc, también está disponible como una librería de código abierto separada. Además, soporta modelos de redes neuronales convolucionales para etiquetado de POS (Part Of Speech), conversión de dependencias y reconocimiento de entidades nombradas, así como mejoras al API sobre entrenamiento y actualización de modelos y construcción de flujos de procesado personalizados.

Stanford NLP. Este es un paquete de análisis de lenguaje natural de Python. Contiene herramientas, para convertir una cadena que contiene texto en lenguaje humano en listas de oraciones y palabras, para generar formas básicas de estas, sus partes del habla y características morfológicas, y para proporcionar una dependencia sintáctica de la estructura. Está diseñado para ser paralelo entre más de 70 idiomas, utilizando el formalismo de Dependencias Universales.

Además, es capaz de llamar al paquete CoreNLP de Java y hereda una funcionalidad adicional desde allí, como el análisis de circunscripción, la resolución de la referencia de núcleo y la comparación de patrones lingüísticos. Está construido con componentes de red neuronal de alta precisión que permiten capacitación y evaluación eficientes con sus propios datos. Los módulos están construidos sobre PyTorch. Se obtiene un rendimiento mucho más rápido si ejecuta este sistema en una máquina con GPU discreta.

Los criterios de selección son:

- **Velocidad.** Velocidad para la interpretación del texto. Es la capacidad de procesar mayor cantidad de texto en menor cantidad de tiempo.
- **Soporte para el español.** La librería seleccionada debe estar entrenada en un gran corpus del idioma español. Mejor reconocimiento de entidades y tokens significa mejor soporte.
- **Velocidad de desarrollo.** La velocidad y facilidad de desarrollo que ofrece la librería.
- **Funciones por defecto.** La librería debe contar con suficientes funciones por defecto para cubrir todos los requerimientos del software. Las funciones requeridas deben estar ya implementadas o que estas sean fáciles de implementar.
- **Licencia**. La licencia de uso del software, una licencia libre es preferible.
- **Documentación y soporte.** Documentación oficial y no oficial disponible sobre la librería.
- **Tokenización y oraciones.** Contar con soporte para separar documentos en oraciones y tokens.

Tabla 24
Calificación De Importancia De Modelos De PLN

Criterios	Importancia	Opciones			
		NLTK	Spacy	StanfordNLP	NLP Toolkit
Velocidad	4	4	5	3	4
Soporte para el español.	4	4	4	5	4
Velocidad de desarrollo	3	4	5	3	4
Funciones por defecto	3	4	5	4	4
Licencia	3	5	5	3	5
Documentación y soporte	4	5	4	3	4
Tokenización y oraciones	3	5	4	4	3
Puntaje total:	120	106	109	86	96

Aunque los resultados entre NLTK y spaCy son bastante cercanos, la velocidad de respuesta y desarrollo son determinantes para inclinarse al seleccionar spaCy. Además, spaCy tiene funciones por defecto listas para ser utilizadas, que al utilizar cualquiera de las otras librerías se debería implementar desde cero.

Motores de videojuego

Para mostrar un resultado final a los usuarios donde el modelo tridimensional muestre la interpretación, se necesita seleccionar un motor de videojuego para su renderización en tiempo real. Se realiza una comparación entre los principales motores, con soporte en la actualidad y alta documentación, para determinar el que mejor se adapte a las necesidades del proyecto.

Unity 3D. Es un motor que permite el desarrollo de juegos 2D y 3D, creado por Unity Technologies y se encuentra disponible sobre diversas plataformas como Windows, OS X y Linux, esta última en una fase beta. Esta herramienta provee soporte para una gran variedad de plataformas objetivo y que siguen ampliándose hasta la fecha. Para el año 2019 Unity Technologies pone a disposición una variedad de licencias para cada uso específico.

- Unity personal: todas las prestaciones del motor se encuentran disponibles de forma gratuita. Servicios con limitaciones, por ejemplo, Unity Collaborate. El principal requisito para el uso de esta licencia es que hay un límite de ingresos puesto en $100,000 sin importar la compañía, organización o incluso si se trata sobre trabajo individual/personal.
- Unity Plus: Esta licencia es sumamente similar a Unity Personal, a diferencia ésta posee un costo por suscripción mensual y el límite de ingresos está puesto sobre los $200,000.

Existen otra variedad de licencias especializadas, pero que no serán discutidas ya que no tiene mayor impacto o relevancia para este proyecto. En cuanto a los requerimientos de Unity3D para la versión 2018, Unity Technologies establece los siguientes requerimientos:

Tabla 25
Requerimientos Para Unity 3D

Requerimientos para Unity3D 2018 o versiones inferiores.	
Sistemas Operativos	Windows 7 SP1+, 8, 10, 64-bit; macOS 10.12+ Ubuntu
Procesador (CPU)	Soporte para el conjunto de instrucciones SSE2.
Gráficos (GPU)	Tarjeta de video con capacidad para DX10

Unreal Engine. Es un motor de videojuegos creado por la compañía Epic Games, permite tanto el juego de desarrollo 2D Y 3D. Se encuentra disponible para diversas plataformas como Windows, OS X y Linux para el desarrollo. Su código se encuentra escrito en C++ y es el mismo lenguaje utilizado para el desarrollo en esta plataforma.

Epic provee únicamente un tipo de licencia para el uso de Unreal Engine (EULA), que en pocas palabras da la posibilidad de utilizar todas las características de su herramienta sin un

costo de adquisición, pero Epic recibirá el 5% de los ingresos obtenidos por el producto, lo cual esta detallado en gran manera en su acuerdo de licencia. Los requerimientos recomendados de software y hardware especificados para UE4 para utilizar el editor son los detallados a continuación:

Tabla 26
Requerimientos Para Unreal Engine

Requerimientos para Unreal Engine 4	
Sistema Operativo	Windows 7 64-bit, Mac OS X 10.9.2 Ubuntu 16.04 LTS
Procesador (CPU)	Quad-core Intel o AMD a 2.5 GHz
Memoria (RAM)	8 GB RAM
Gráficos	NVIDIA GeForce 470 GTX o AMD Radeon 6870 HD series o superior

Los criterios de selección son para el motor de videojuego se listan a continuación:

- **Desarrollo multiplataforma:** Que el desarrollo pueda realizarse en distintos sistemas operativos.
- **Ejecución multiplataforma:** Que el producto final del desarrollo pueda ser compilado o exportado para su ejecución sobre distintos sistemas operativos.
- **Experiencias previas:** Experiencia previa por parte de los integrantes del grupo de trabajo en el uso de herramientas de desarrollo de videojuegos para disminuir el tiempo de aprendizaje y desarrollo.
- **Lenguaje de programación:** En cuanto al lenguaje de programación, se evaluará la facilidad que este brinde para el desarrollo y la legibilidad del código.
- **Curva de aprendizaje:** Se evaluará como positivo la facilidad que brinda la herramienta para su uso y aprendizaje del mismo.
- **Documentación:** Documentación oficial y no oficial disponible sobre la librería.

- **Requerimientos software y hardware:** Se evaluará los requerimientos necesarios para poder utilizar las herramientas por parte del equipo de desarrollo.
- **Licencia:** La licencia de uso del software.

Tabla 27
Calificación De Importancia Para Motor De Videojuego

Criterios	Importancia	Opciones	
		Unity3D	UnrealEngine
Ejecución Multiplataforma	4	5	5
Desarrollo Multiplataforma	3	5	5
Experiencias Previas	3	3	0
Lenguaje de programación	2	5	3
Curva de aprendizaje	3	4	3
Documentación	3	5	4
Requerimientos Software/Hardware	4	4	4
Licencia	4	5	5
Puntaje total:	110	97	78

Con los resultados obtenidos se puede observar que ambas alternativas son altamente elegibles, los factores principales que determinan la selección de Unity3D, son los requerimientos de hardware y software menores de estar herramienta, lo cual se adaptan de mejor manera a las posibilidades del equipo de desarrollo. Otro aspecto clave son la curva de aprendizaje y las experiencias previas con el uso de esta herramienta, lo cual permitirá realizar un mejor desarrollo en un lapso menor de tiempo.

Software de manipulación de gráficos 3D

Blender. Es un programa de Gráficos 3D, que tiene como principales usos el modelado, iluminación, renderizado y animación. Blender también cuenta con otra variedad de funciones como edición de videos, pintura digital, entre otros.

Es un programa multiplataforma compatible con Windows, Mac OS X, GNU/Linux y es distribuido como un proyecto impulsado por la comunidad bajo la Licencia Pública General de GNU (GPL). Se encuentra programado en C, C++ y Python. Una característica muy importante es que se pueden usar todas sus funciones por medio de scripts en Python, dando así la posibilidad de integrar sus funciones en un flujo de trabajo, realizar automatizaciones, pruebas e integraciones con otros productos.

Tabla 28
Requerimientos Para Blender

Requerimientos para Blender 2.7 / 2.8	
Sistema Operativo	Windows Vista, 7, 8, 10, macOS 10.12
Procesador (CPU)	32-bit dual core 2Ghz CPU con soporte SSE2
Memoria (RAM)	2 GB RAM
Gráficos	OpenGL 2.1 compatible con 512 MB RAM

Autodesk Maya. Es un programa para el desarrollo Gráficos 3D por ordenador, sus funcionalidades incluyen modelado, iluminación, renderizado y animación. Maya es sin duda uno de los más famosos en este ámbito gracias a los increíbles efectos visuales realizados para la industria cinematográfica. Es un programa multiplataforma compatible con Windows, Mac OS X, GNU/Linux y es distribuido bajo una licencia de software propietario.

Se puede hacer uso de este software sin costo por medio de licencias educativas y personales, pero limitan la distribución de los productos obtenidos con dicha herramienta.

Se encuentra escrito en los lenguajes de programación de C++, Python y Mel, este último un lenguaje especial para este software en específico. En Maya se tiene la posibilidad de utilizar scripting para crear herramientas propias, conjunto de comandos y plugins por medio de los lenguajes de Mel y Python.

Tabla 29
Requerimientos Para AutoDesk Maya

Autodesk® Maya® 2016.	
Sistema Operativo	Sistemas operativos Windows10, 8.1 y 7. Sistemas operativos Apple Mac OS X 10.9.5, 10.10.x y 10.11.3 Sistema operativo Red Hat Enterprise Linux 6.5 WS Sistema operativo CentOS 6.5 Linux
Procesador (CPU)	Procesador de varios núcleos de 64 bits Intel® o AMD®
Memoria (RAM)	4 GB de RAM (se recomiendan 8 GB)
Gráficos	Hardware certificado para Maya.

Cinema4D. Es un software de creación de gráficos y animación 3D, sus principales funciones son el modelado, texturización y animación. Se definen a sí mismos como un software rápido y fácil de usar, pudiendo realizar muchas cosas con poco conocimiento de la herramienta. Es un programa multiplataforma disponible para Windows, Mac OS X y Amiga OS.

Es distribuido para una licencia de software propietario. Las licencias de este producto, comparado con sus competidores, son las de valor más elevado y la única manera de obtenerlo sin costo es con licencias estudiantiles que están limitadas o con la prueba de 30 días.

Es muy importante mencionar que Cinema4D brinda la posibilidad de scripting por medio de Python, para que los usuarios puedan crear sus propias herramientas que faciliten el trabajo repetitivo o procedimental.

Tabla 30
Requerimientos Para Cinema 4D r15

Cinema 4D r15	
Sistema Operativo	Windows 7 SP1 64-bit o superior, MacOS 10.11.6 o 10.12.4+ 64-bit
Procesador (CPU)	Intel o AMD 64-bit CPU con soporte para SSE3, CPU funcionando en Apple Macintosh con Intel.
Memoria (Ram)	4 GB de RAM (8 GB o más recomendado)
Gráficos (GPU)	Tarjeta con soporte para OpenCL 1.2, se recomienda al menos 4 GB VRAM para el render GPU.

Los criterios de selección son:

- **Desarrollo multiplataforma:** Que el desarrollo pueda realizarse en distintos sistemas operativos.
- **Scripting:** Que la herramienta permite el uso de scripts para la automatización.
- **Modelado y animación:** Que el programa brinda las herramientas necesarias para llevar a cabo las tareas tanto de animación como el modelado.
- **Experiencias previas:** Experiencia previa por parte de los integrantes del grupo de trabajo en el uso de herramientas de desarrollo de manipulación de gráficos 3D para disminuir el tiempo de aprendizaje y desarrollo.
- **Curva de aprendizaje:** Se evaluará como positivo la facilidad que brinda la herramienta para su uso y aprendizaje del mismo.
- **Requerimientos software y hardware:** Se evaluará los requerimientos necesarios para poder utilizar las herramientas por parte del equipo de desarrollo.
- **Soporte:** Que el programa a la actualidad se encuentre con soporte por parte de los desarrolladores-.

- **Licencia:** La licencia de uso del software.

Tabla 31
Calificación De Importancia De Software De Manipulación De Gráficos 3D

Criterios	Importancia	Opciones		
		Blender	Maya	Cinema 4D
Desarrollo multiplataforma	3	5	4	4
Scripting	3	5	5	3
Modelado y animación	4	5	5	5
Experiencia Previa	3	3	0	4
Curva de aprendizaje	3	4	3	4
Compatibilidad con motores de juegos	4	5	5	5
Requerimientos Hardware/Software	4	5	4	4
Soporte	3	5	5	5
Licencia	4	5	0	0
Puntaje total:	155	146	107	116

Blender obtiene los puntajes más altos en todos los criterios evaluados, pero aspectos más importantes de esta selección son la libertad que esta herramienta provee, por ser Open Source, tanto para uso como para los productos resultantes de este. De igual manera es muy importante los requerimientos de software que resultan más aptos para el desarrollo y la experiencia previa que permitirá acelerar el proceso de diseño y animación del modelo 3D a utilizarse.

Herramienta especializada para el modelado del personaje

Para realizar el modelado del personaje es necesario utilizar herramientas extras que permitan realizar esta tarea sin la necesidad de conocimientos especializados en manipulación de gráficos en 3D. La principal razón es que para realizar un modelado desde cero con

herramientas como Blender 3D, se requiere tiempos de hasta meses para obtener un resultado final y sin garantizar una calidad adecuada.

Adobe Fuse CC. Es una herramienta de manipulación de gráficos por computadora en 3D desarrollado por Mixamo, que permite a los usuarios crear personajes en 3D por medio del uso assets y configuraciones. Su principal ventaja es la integración con otras herramientas como Adobe Photoshop que permitirán realizar retoques sobre textura y creación de poses sobre el personaje.

Fuse CC por sí sola no tiene la capacidad de crear los huesos sobre el modelo para su posterior animación, pero provee la opción de utilizar un sitio web de sus mismos desarrolladores en donde el modelo es subido, procesado y posteriormente descargado. Se encuentra disponible de manera gratuita y sin limitantes en cuanto a su uso, únicamente se requerirán licencias de otros productos de adobe como Photoshop si se buscara una integración entre ambas.

Tabla 32
Requerimientos Para Adobe Fuse CC

Adobe Fuse CC (Beta)	
Sistema Operativo	Microsoft Windows 7, Windows 8, Windows 8.1, o Windows 10.
Procesador (CPU)	Procesador dual core o superior.
Memoria (RAM)	4 GB de RAM.
Gráficos	1024 x 768 display con 16-bit color y al menos 512 MB de VRAM.

Character Creator 3D. Reallusion, la empresa creadora de esta herramienta, la describe como uno de los canales de creación de contenido más interoperables y fácil de usar. Esta herramienta permite la creación de personajes con la posibilidad de cambiar prácticamente

cualquier aspecto de este, desde aspectos como su cuerpo, piel, cabello, ropa hasta otros aspectos como el peso o influencia que cada hueso tiene en los movimientos.

La versión de prueba permite el acceso a todas las características individuales de esta herramienta con las únicas limitantes que pueden hacerse uso de estas durante un periodo máximo de 30 días.

Tabla 33
Requerimientos Para Character Creator 3

Character Creator 3	
Sistema Operativo	Windows 10, 8 y 7. Soporte para sistemas de 64-bit.
Procesador (CPU)	Procesador dual core o superior.
Memoria (RAM)	8 GB de RAM.
Gráficos	NVidia GeForce GTX 600 Series / AMD Radeon HD 7000 Series.

Los criterios de selección son:

- **Calidad de modelos:** Calidad del resultado final del personaje, tomando en cuenta aspectos como texturas y forma.
- **Listo para animaciones:** Los modelos resultantes deben poseer las estructuras de huesos necesarios para su animación, así como las deformaciones o blendshapes para las expresiones faciales.
- **Facilidad de uso:** Se evaluará como positivo la facilidad que brinda la herramienta para su uso y aprendizaje del mismo.
- **Requerimientos software y hardware:** Se evaluará los requerimientos necesarios para poder utilizar las herramientas por parte del equipo de desarrollo.
- **Soporte:** Que el programa a la actualidad se encuentre con soporte por parte de los desarrolladores.

- **Licencia:** La licencia de uso del software.

Tabla 34
Calificación De Importancia De Software De Modelado De Personaje

Criterios	Importancia	Opciones	
		Adobe Fuse	Character Creator 3
Licencia	3	5	3
Calidad de modelos	4	3	5
Listo para animaciones	4	3	5
Facilidad de uso	2	5	4
Soporte	3	2	5
Compatibilidad con motores de juegos	4	4	5
Requerimientos hardware/software	4	5	5
Puntaje total:	120	91	112

Las únicas ventajas notables que supone la herramienta de Adobe Fuse CC son el uso gratuito y los requisitos de hardware, aspectos que están ligados a los costes de utilizar una herramienta u otra. Por otro lado, Character Creator supone ventajas en cuanto a la calidad del resultado, variedad, compatibilidad, integración y la seguridad de que el producto será soportado los próximos años e incluso con mejoras adicionales. Además, este último, mediante la compra de licencias, abre la posibilidad de utilizar este programa con herramientas de captura de movimientos.

Datos de animaciones

Para la manipulación de los datos de las señas y animaciones durante el desarrollo y la primera de demostración se decidió buscar una manera de prescindir de cualquier herramienta, motor o servicio específico como lo son mysql, sqlite, mongodb, firebase entre otros a fin de que posteriormente esta pueda integrarse con la opción de mejor conveniencia.

ElasticSearch. Es un servidor de búsqueda basado en Lucene, que provee un motor de búsqueda de texto completo y distribuido. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache.Con esta herramienta podemos impulsar búsquedas extremadamente rápidas que respalden las aplicaciones de descubrimientos de datos.

La primera razón de utilizar esta herramienta es que posteriormente al migrar a otra herramienta se puede incluso seguir utilizando en conjunto esta para la realización de búsqueda de datos. Como segunda razón es que ElasticSearch es muy recomendable en proyecto de NLP, debido a sus capacidades de análisis, dejando así la posibilidad a nuevas funcionalidades que la solución pueda realizar a futuro, por ejemplo, consultas de palabras similares en base a definición o descripción o deducción de términos en base a descripciones brindadas por los usuarios. Por último, se busca explorar y demostrar la compatibilidad que existe entre las distintas tecnologías utilizadas y nuevas tecnologías como lo es ElasticSearch.

CAPÍTULO IV
DESARROLLO

MODELADO DE MIA

Esta sección consiste en el diseño del personaje Mia, uno de los assets o recursos principales para el cliente animado, donde se utilizará un modelo tridimensional para la representación de un personaje humano. Para la creación, diseño y modelado del personaje se hace uso de las herramientas Character Creator 3 y Blender 3D.

Character Creator 3 permite un modelado rápido de personajes realistas y listos para ser utilizados en animaciones. Se recurrió al uso de esta herramienta ya que el modelado requerido para el proyecto puede tomar tiempos prolongados, hasta meses, el cual puede ser reducido drásticamente con el uso de esta herramienta. Al inicio esta herramienta presenta un modelo humano base, a partir del cual se pueden modificar aspectos como rostro, cuerpo, cabello, maquillaje y vestimenta a fin de obtener un resultado lo más apegado a la necesidad del usuario.

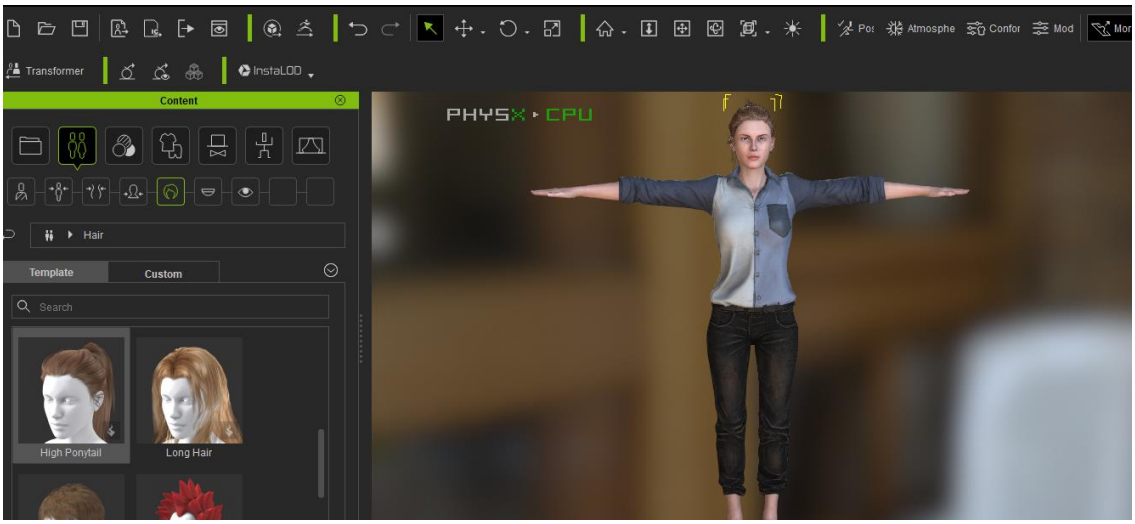

Figura 26. Pantalla inicial de Character Creator 3

Los modelos generados en esta herramienta son configurados automáticamente con un esqueleto para su animación y que es compatible con los distintos motores de videojuegos como Unity. Esta herramienta también provee al modelo con una serie de Blendshapes o deformaciones faciales que serán utilizadas para las animaciones de expresiones faciales en el modelo.

Al obtener el aspecto deseado sobre el modelo se procede a su exportación.

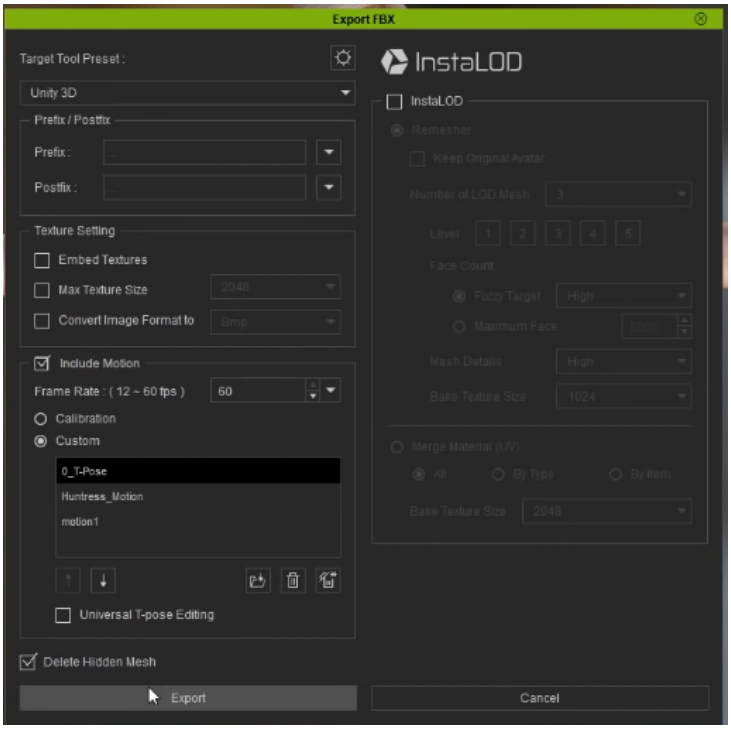

Figura 27. Menú de exportación de Character Creator 3

En las opciones se selecciona la plataforma a la que se desea exportar, en este caso Unity 3D, lo cual generará un archivo con extensión FBX. En la sección de animaciones o motion, es importante incluir la pose “T”, donde el modelo se encuentra viendo hacia el frente con los brazos extendidos a modo de representar una posición en T, ya que el motor de videojuegos hace uso de esta pose para poder realizar un mapeo de los huesos.

Las demás opciones son utilizadas para optimización del modelo y son opcionales, como la última opción seleccionada “Delete Hidden Mesh”, que realiza una fusión de todos los componentes tridimensionales, eliminando por ejemplo partes del cuerpo que están cubiertas por la vestimenta.

Una vez finalizada la exportación se obtiene un modelo como el siguiente:

Figura 28. Modelo exportado con Character Creator 3

Con el modelo exportado al formato FBX, desde la herramienta Character Creator, se procede a importarlo a un nuevo proyecto en Blender 3D, con el objetivo de poder realizar modificaciones extras al modelo a ser utilizado en Unity 3D y crear además un nuevo modelo optimizado que será utilizado para realizar las animaciones. Para importar un modelo en Blender basta con ingresar al Menú File > Import >FBX (.fbx) y posteriormente seleccionar el archivo a importar.

Los cambios a realizar sobre el modelo para utilizarse en Unity 3D, son inicialmente la eliminación de elementos en la escena como cámaras e iluminación. Con Blender queda abierta la posibilidad de realizar cambios de personalización sobre el modelo en cualquier momento del desarrollo, sin tener implicaciones mayores en el funcionamiento del proyecto y sin depender del tiempo de prueba que ofrece la herramienta de Character Creator 3.

El segundo modelo a utilizarse, se creará a partir del mismo, esta será una versión sin texturas, materiales, cabello entre otras características que no son de importancia al momento

de la realización de las animaciones, a fin de disminuir los requerimientos de hardware en el proceso de animación.

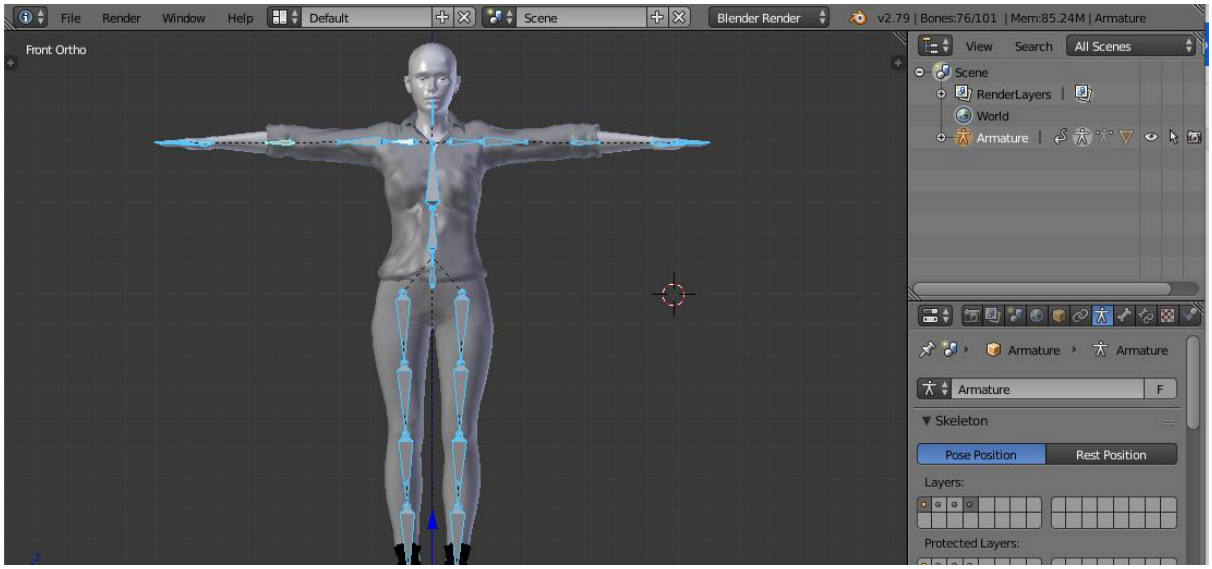

Figura 29. Modelo del personaje para la animación en Blender 3D

Convención de nombres

Todos los modelos a utilizarse, independientemente de la extensión del archivo (dae, fbx, blend) deben comenzar con la palabra clave "model" para indicar el tipo de contenido del archivo. Posteriormente se debe indicar si se trata de un personaje "char", un objeto dentro de una escena "obj", un escenario completo "esc" y en cualquier otro caso distinto "other". Por último, queda a discreción de cada individuo proveer nombres que brinden un descripción corta y precisa del archivo utilizando únicamente caracteres alfanuméricos. Cada sección del nombre debe ir separada por el carácter guion bajo "_", obteniendo nombres como los siguientes ejemplos:

- *model_char_mia01*
- *model_esc_habitacion*
- *model_obj_silla02*
- *model_otros_efecto*

ANIMACIÓN

Esta sección se abordará la animación del personaje Mia, el asset principal para el cliente animado, donde se utilizará un modelo tridimensional previamente creado en la sección modelado del capítulo de desarrollo. Para la creación de las distintas animaciones del personaje se hace uso de las herramientas Blender y Unity 3D.

Animaciones de señas

Esta etapa consiste en recrear las señas usadas en LESSA en acciones que el modelo realizará, manipulando sus propiedades que lo componen, como lo son los huesos, las conexiones entre huesos, traslaciones, rotaciones, etc.

Retomando descripciones previas expuestas en el Marco Teórico, la animación del modelo se logra a través de poses, las cuales son posiciones estáticas en la cual ciertos huesos (o todos) poseen su propia posición, rotación y escalas diferentes de las que poseían en la posición de descanso. Dichas poses son definidas en "*keyframes"* que contienen la información de la posición de los huesos en un punto definido en una línea de tiempo. Esta línea de tiempo en la cual se organizan los keyframes se le conoce como *"DopeSheet",* en ella se organizan el orden de las keyframes. La *DopeSheet* esta medida en cuadros de animación, las keyframes poseen un cuadro específico en el cual se encuentran ubicadas y la pose que el modelo adoptará en un cuadro específico.

Las animaciones creadas son guardadas dentro de "*Acciones".* Una acción puede ser descrita como la colección de keyframes organizadas dentro de la línea de tiempo y es identificada por un nombre específico (ya sea dado por defecto o por el usuario). Dentro de un archivo de Blender puede haber múltiples acciones para el modelo 3D.

La animación es generada durante la transición de un cuadro al otro en la línea de tiempo, la forma en la que se produce la animación es gracias a que Blender realiza la interpolación de

posiciones de los huesos de una pose a otra, es decir, Blender genera la animación de transición de un keyframe a otro.

No obstante, esto no significa que la animación de transición esté libre de problemas, un problema muy común con el cual se lidió en la animación es el cruce de las partes del cuerpo, es decir, ciertas partes del cuerpo se cruzaban entre sí como si fueran intangibles, por ejemplo, en la transición del keyframe A al keyframe B los brazos se cruzan y se atraviesan entre sí para poder llegar a su nueva posición.

Otro detalle con el que se trabajó fue en las animaciones de transición poco naturales, la transición de un keyframe a otro se realizaba sin problema, pero en la animación completa esta resultaba muy "robótica" y no como una acción que una persona realizaría con naturalidad. Estos problemas fueron disminuidos agregando keyframes nuevos entre el keyframe A y el Keyframe B para poder corregir la dirección en la que se mueven para evitar el cruce entre partes del cuerpo o agregar mayor fluidez o naturalizada a la animación.

Para ejemplificar de forma más detallada el cómo se realizaron las animaciones se continuará con la descripción de los pasos y procedimientos que se utilizaron en Blender para recrear las señas en el modelo 3D.

Al abrir en el archivo de Blender en el cual se trabaja, se encuentra con solamente el propio modelo sobre cual trabajaremos y debajo de él se encuentra la *DopeSheet* donde como hemos descrito con anterioridad comenzaremos con los diseños de las poses y sus keyframes.

Blender por defecto maneja dentro del *DopeSheet* que 30 cuadros dentro de la línea de tiempo son equivalente a un segundo, este parámetro se ha usado de referencia para medir la longitud de las señas en las animaciones basados en el tiempo real de duración que se considera conveniente para una seña específica, esto debido a que cada seña necesitará ir a su velocidad propia, debido a que algunas señas con respecto a su complejidad necesitarán una mayor cantidad de cuadros para lograr una animación comprensible.

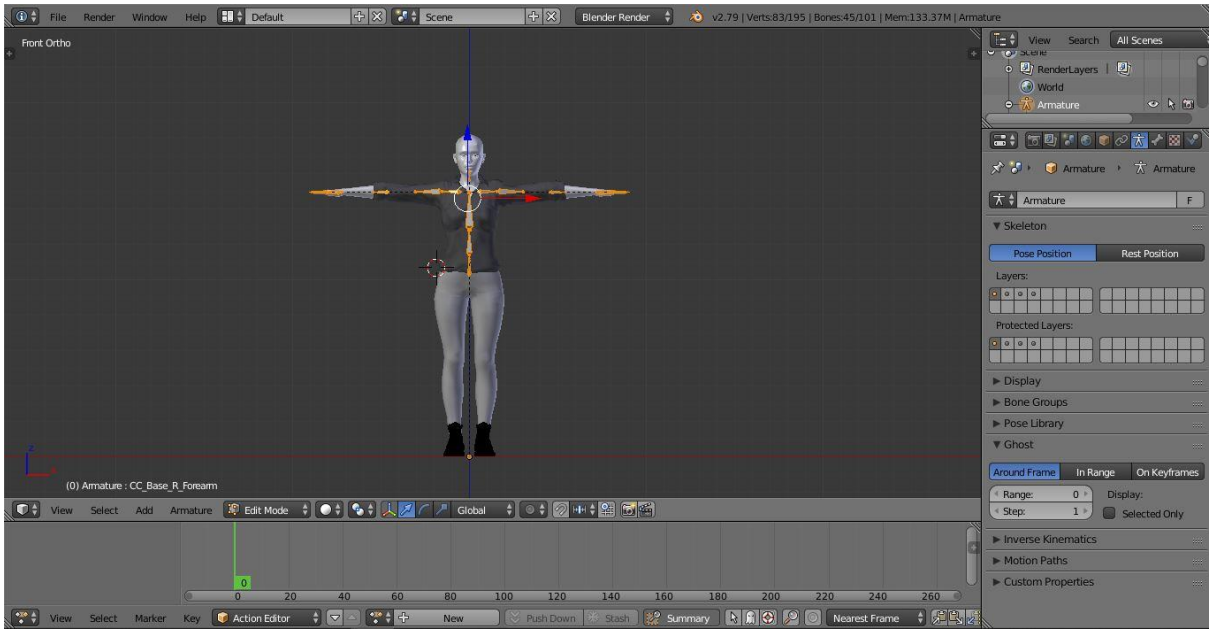

Figura 30. Vista Principal de Animación

Se crea una nueva acción donde quedarán guardadas todas las poses ancladas en los keyframes y ordenadas en la línea de tiempo, con lo cual se recrea varias señas por archivo en Blender. Para crear la acción se dirige a la barra debajo de la línea del tiempo y se escribe el nombre de la nueva acción donde está la palabra *New.*

Figura 31. Barra de Herramientas para crear una nueva animación

Cuando la nueva acción es creada, generará una nueva línea de tiempo vacía en la cual se definen los cuadros en los que se ubican los keyframes que contendrán una pose determinada. Para especificar en qué cuadro dentro de la línea de tiempo se ubicará el keyframe solamente es necesario especificar con el puntero y click izquierdo donde se ubica y quedará resaltado con una línea y cuadro verde indicando en qué frame se encuentra.

Una vez ubicados en el lugar correcto, realizar la pose que el modelo adoptará en el cuadro especificado, iniciando a transformar los huesos (moverlos y rotarlos) hasta poder

generar la pose que se necesita y dejarla anclada a un keyframe ubicado donde se especificó en la línea de tiempo. Este es el proceso que se sigue repetidamente para generar las distintas poses que se necesitan para poder recrear la seña en animación.

Cuando los huesos de modelo están posicionados de tal forma que el modelo esté en la pose que se necesita, se bloquea la posición actual de los huesos (*LocRot*) lo cual ancla las posiciones en el cuadro en el que se encuentra, en la línea de tiempo mediante un keyframe.

Figura 32. Keyframes ubicados en la Línea de Tiempo

¿Cómo sabe que pose hacer y en qué orden necesitan ir para generar una seña? Para poder recrear una seña, basado en material de referencia, mayormente los video oficiales del lenguaje LESSA en YouTube, así como con la ayuda de intérpretes del lenguaje para poder analizar cómo está conformada una seña específica.

Por ejemplo, la seña *"Buenos Días",* primero observar en el material de referencia para analizar y determinar las poses que son necesarias y que animación de transición son las indicadas para recrear la seña con el modelo 3D. Para esto se identifica que poses son de partida, intermedias, y de llegada. La pose de partida y de llegada son las poses ubicadas una después de otra de donde se requiere que las partes del cuerpo estén en un cuadro determinado para poder generar la animación de transición, estas se identifican ya que poseen un énfasis (ya sea una pausa, prolongación, un elemento que llama la atención de la persona).

Las poses intermedias son las que se ocupan entre una pose de partida y de llegada para agregar mayor naturalidad y fluidez a la animación o bien para corregir la dirección de ciertas partes del cuerpo y evitar que esta tome desvíos imprevistos o cruzan unas con otras.

En la seña de *Buenos Días*, con base en el material encontrado en YouTube para generar la seña, se necesita una pose de partida donde los puños con los pulgares hacia arriba estando unidos a la altura del pecho y una pose de llegada donde están completamente separados en una posición un poco por debajo del pecho. Tomando en cuenta el análisis se procede con la recreación de las poses y luego ver la animación para poder observar el resultado y si este es adecuado.

Un punto de aclaración en la recreación de las señas es que muchos tienen su forma de hacer las señas, nadie usa la misma manera, unas personas pueden elevar un poco más los brazos que otros, mover las manos con más rapidez, etc. Pero todos respetan la estructura propia se la seña sin hacerle cambios drásticos que alteren su significado. Por esa razón es que se decidió recrear las señas teniendo en mente los límites con los que se trabaja en el modelo reproducirá las señas en su propia forma de expresión, pero siempre respetando las normas que componen una seña.

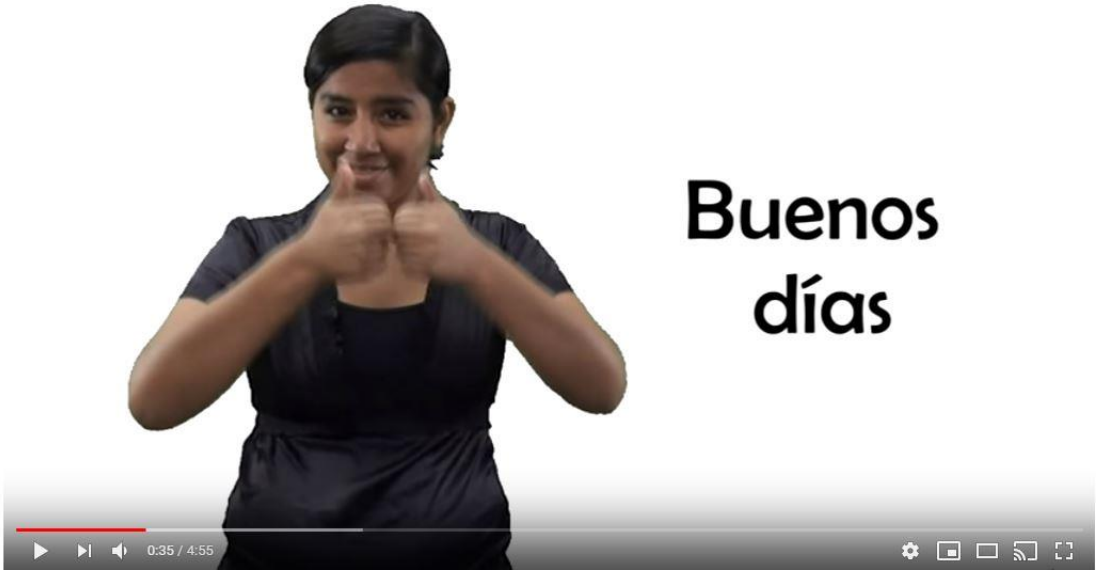

Figura 33. Seña hecha por interprete posición 1

Tomando desde el video se identifica que la seña comienza desde esta pose la cual se intenta recrear en el modelo de Blender.

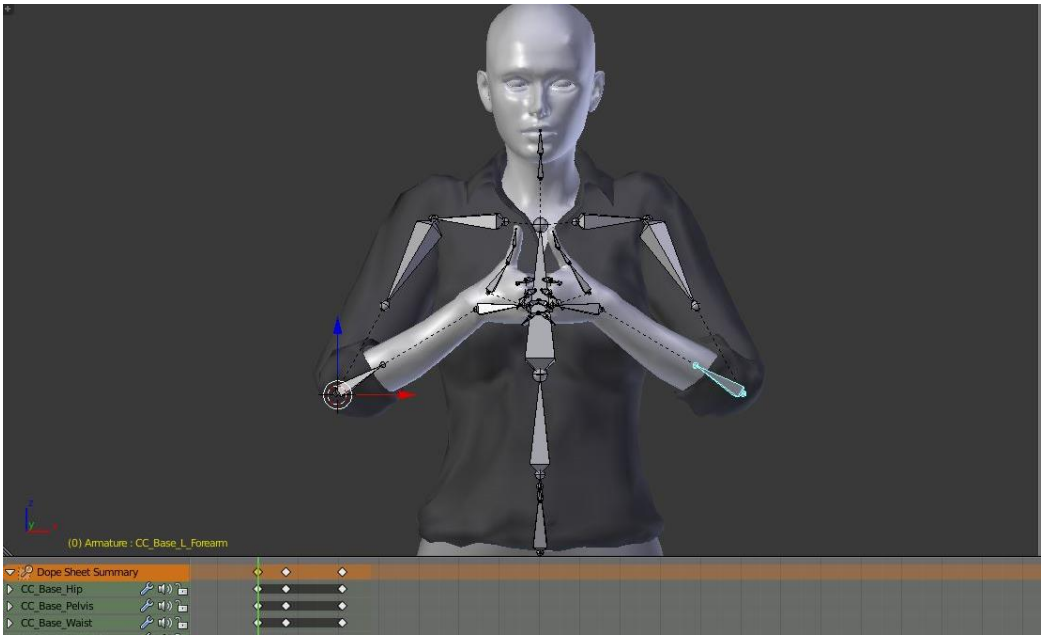

Figura 34. Seña recreada con ml modelo posición 1

Para crear esta pose nueva se ha cambiado los factores principales que transforman a los huesos de su posición de descanso a una nueva pose, siendo estos sus Ejes, Traslación, Roll y Rotación los cuales ya han sido definidos en el Marco Teórico. Ya que esta la primera pose que se encuentra en la seña se ubica dentro del frame 0. Luego recreamos la pose de llegada que también es la pose final de la seña.

Figura 35. Figura 35. Seña Hecha Por Interprete Posición 2

Ya que es la última pose que conforma la seña se ubica en un cuadro final en el cual se considera que la animación tendrá un tiempo de reproducción adecuado, en este caso el frame 20.

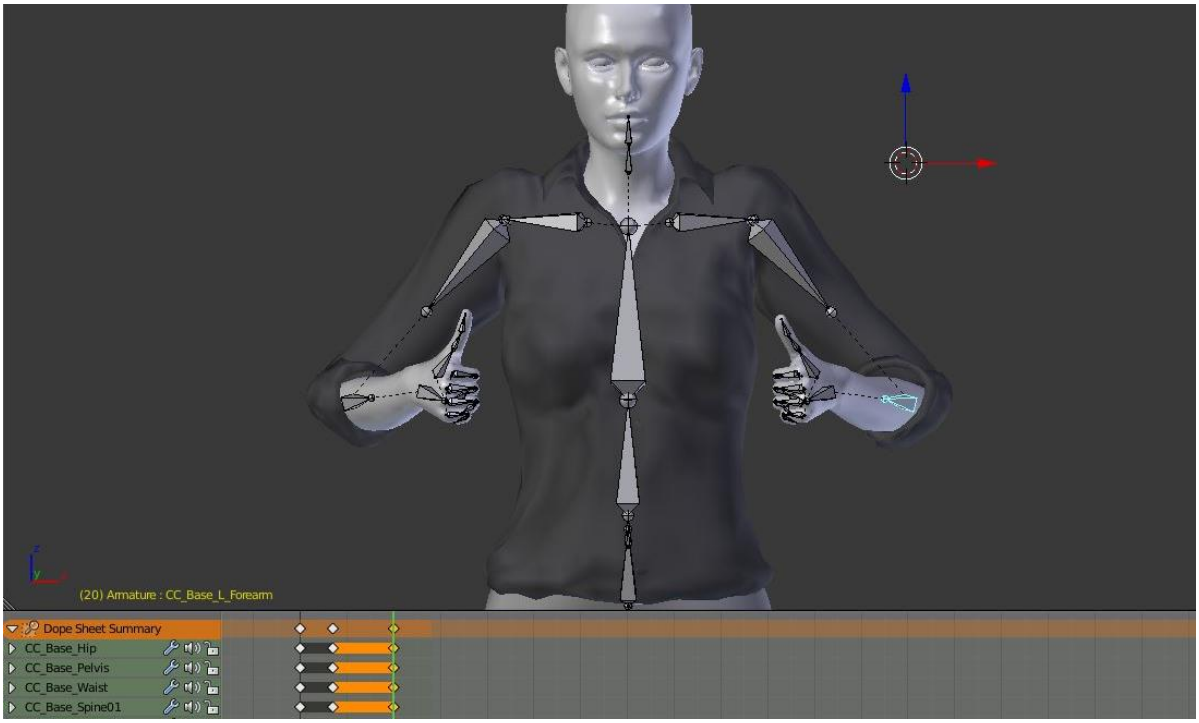

Figura 36. Seña recreada con el modelo posición 2

Para reproducir la animación completa y poder verificar si es aceptable la repetición de la seña, nos ubicamos en el Timeline donde podemos reproducir la animación. Gracias a esto, solo se deja la seña con las poses de partida y de llegada, está no se ve natural, se ve muy robótica y automática, es aquí donde entran las poses intermedias.

Al revisar la animación nos percatamos que la transacción se nota muy robótica por falta de movimientos en los hombros y brazos. Para poder corregir añadimos una animación intermedia donde elevamos la posición de los hombros y los movemos un poco hacia atrás y en los brazos los elevamos y los separamos levemente uno del otro.

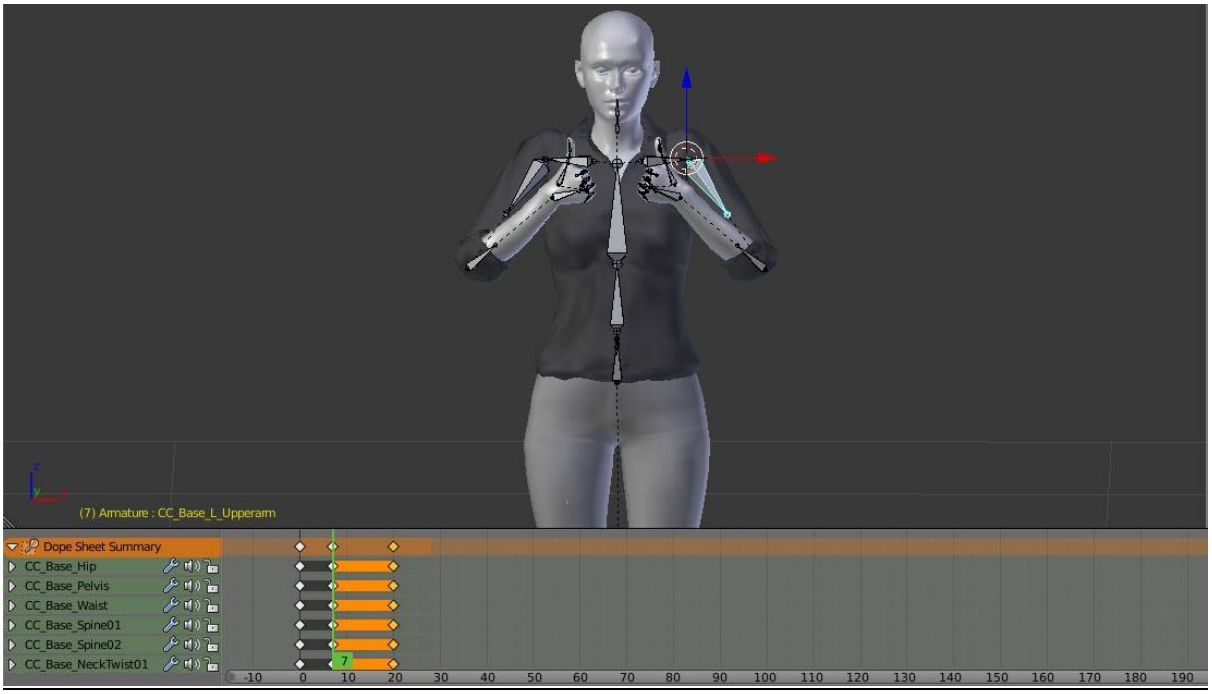

Figura 37. Seña Recreada Con el modelo posición de transición

La pose intermedia fue ubicada en el frame 7 y no en el 10 que se considera la mitad de la animación ya que el ver el material de referencia identificamos que los movimientos faltantes deberían de ir un poco después de la pose de llegada ya que se observa que la mayoría de la transición es la separación de los brazos y colocarlos a mitad de la animación haría que la seña perdiera entendimiento. De igual forma se observa que los movimientos faltantes que se agregaron ocurren casi inmediatamente luego de la seña de partida.

Animaciones faciales

Para la realización de estas animaciones se hace uso de las deformaciones de la geometría, conocidas comúnmente como Blendshapes. Los modelos previamente creados contienen una serie de blendshapes definidas para manejar las deformaciones del rostro del personaje, permitiendo crear animaciones de sonrisa, enojo, sueño, alegría, entre muchas otras. Las animaciones faciales son realizadas con el editor de Unity, desde la ventana Animation que puede abrirse desde el menú Window > Animation > Animation.

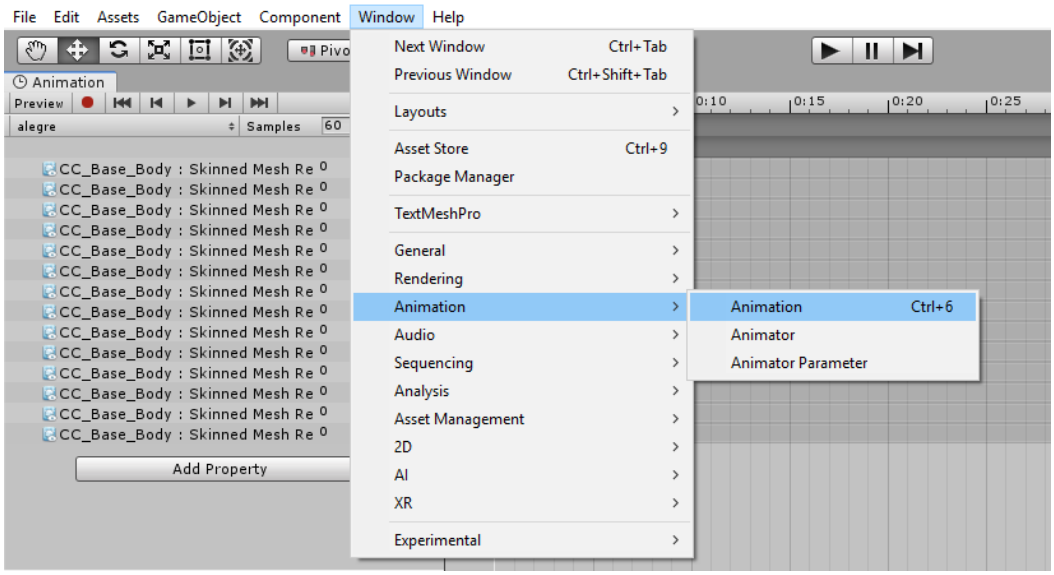

Figura 38. Menú para ingresar a la ventana Animation

Las animaciones mediante Blendshapes se realizan modificando valores que van desde 0 hasta 100. Un valor de cero indica que esta deformación tiene una influencia nula, mientras que un valor de 100 indica que la influencia es del 100%. Por ejemplo, si se tiene un Blendshape para la apertura de ojos, un valor de 0 indicará ojos cerrados y un valor de 100 indicará ojos completamente abiertos. Para ver los blendshapes disponibles del personaje es necesario acceder al modelo del personaje y luego al componente Skinned Mesh Renderer.

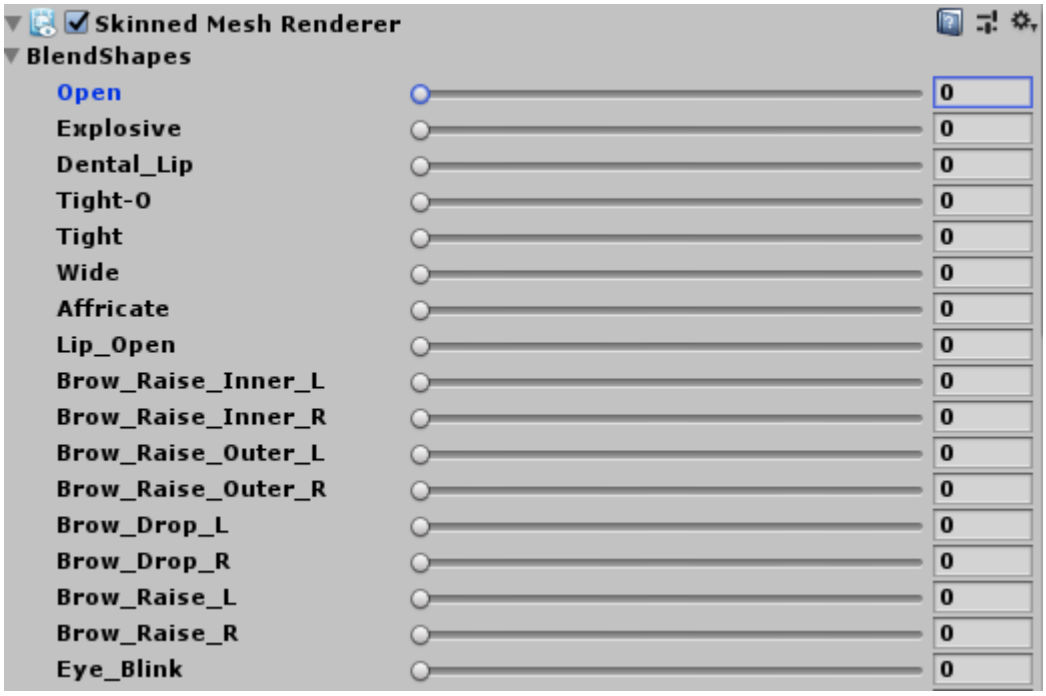

Figura 39. Blendshapes del personaje desde el editor de Unity 3D

Para realizar animaciones primero debe activarse el modo Recording, dando click en el icono rojo de la ventana Animation, lo cual permitirá guardar automáticamente la animación a medida que se realiza. Luego se modifican los valores de las distintas blendshapes hasta obtener el resultado deseado, el cual puede ser visualizado desde la ventana Scene. Al igual que en Blender las animaciones son realizadas sobre una línea de tiempo en la cual se insertan elementos llamados keyframe, que contendrán valores de propiedades como los blendshapes.

Esta línea de tiempo está dividida en fracciones de segundos, y la animación es generada durante la transición de una fracción de segundo a otro en la línea de tiempo, la forma en la que se produce la animación es gracias a que se realiza la interpolación de valores de una pose a otra, es decir que se generará automáticamente la transición de valores de un keyframe a otro.

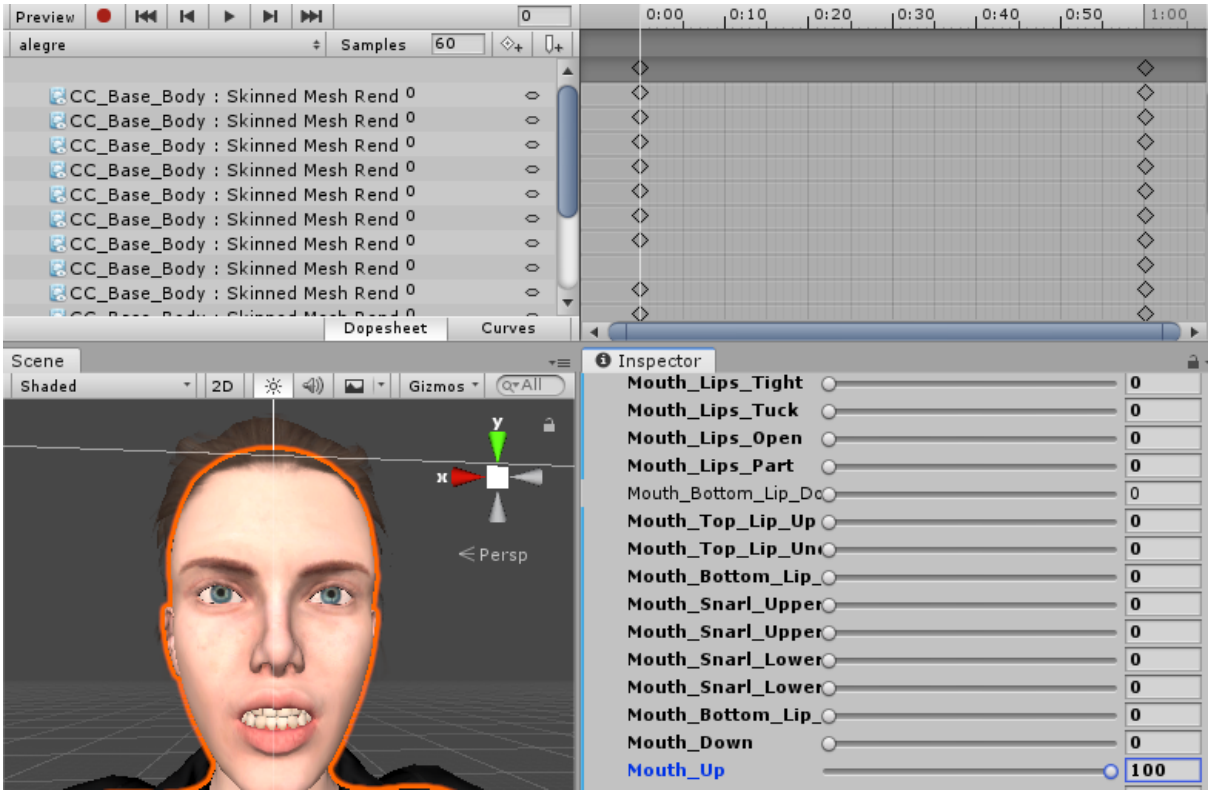

Figura 40. Animación facial del personaje desde Unity 3D

Convención de nombres

Las animaciones realizadas en Blender son contenidas en un archivo con extensión “. blend”, propia del programa. El nombre del archivo de Blender debe comenzar con la palabra clave “anim” para indicar el contenido del archivo. Luego debe agregarse un nombre que resuma el tipo o contexto de las animaciones contenidas por el archivo, se utilizará la palabra “lessa” para animaciones del lenguaje de señas, “faex” para expresiones faciales y “other” para cualquier otro caso. Por último, queda a discreción de cada integrante proveer nombres que brinden un descripción corta y precisa. Cada sección del nombre debe ir separada por el símbolo de guion bajo “_”, obteniendo nombres como los siguientes ejemplos:

- anim_lesaa_letras01
- anim_faex_alegre
- anim_other_sentarse

Los archivos de Blender para las animaciones no debe deben contener más de 30 animaciones por archivo. Cada animación dentro del archivo, así como las animaciones individuales creadas en el editor de Unity, debe ser nombrada iniciando un con una abreviación del grupo de animación y con el nombre propio de cada animación, ambas separadas por el símbolo de guion bajo "_", así como se muestra en los siguientes ejemplos:

- ver_correr, para señas del grupo verbos.
- fam_abuelo, para señas del grupo familia.
- exp_sonreir, para expresiones faciales.
- pos_iddle, para posturas.

ENTRENAMIENTO.

Para proyectos bastante complejos es recomendable entrenar un modelo estadístico de reconocimiento de entidades, sin embargo, los modelos estadísticos requieren de una gran cantidad de datos para el entrenamiento. Para la mayoría de situaciones, el reconocimiento de entidades basado en reglas son una solución más práctica.

Generalmente, entrenar un modelo es bastante útil en situaciones donde se tienen datos de entrenamiento y se desea generalizar basado en el contexto de los ejemplos. En cambio, en el caso del presente proyecto no se desea generalizar para las entidades agregadas, sino encontrar entidades nombradas bien establecidas. Por lo tanto, para identificar estas entidades basta con establecer los patrones de búsqueda necesarios.

La mejor manera de abordar el problema de reconocimiento de entidades nombradas es combinando ambos enfoques e identificar todas las entidades previamente entrenadas con el modelo estadístico y todas las entidades que no se encuentren entrenadas se deben identificar con el enfoque basado en reglas.

La determinación de las entidades nombradas que deben ser identificadas con el enfoque basado en reglas se hizo en base a la lista de palabras y frases que puedan ser representadas por

el presente proyecto en LESSA. Mediante observación en múltiples pruebas, toda entidad que el modelo pre-entrenado es incapaz de identificar de manera correcta es agregada a la lista que se presenta en el Anexo 5.

Dejando de lado previa justificación, se intentó entrenar un modelo estadístico para comparar resultados con el enfoque basado en reglas. Se formó un set de entrenamiento de 200 documentos por cada entidad aproximadamente con un total de 5130 documentos de un máximo de 4096 caracteres cada uno.

Los datos de entrenamiento están divididos en las siguientes categorías:

- 1517 documentos con expresiones de tiempo.
- 1214 documentos con expresiones con contexto especial en El Salvador.
- 2399 documentos que contienen saludos.

Una vez teniendo el set de entrenamiento se procedió a anotar el mismo. El proceso para anotar los documentos consiste en determinar la posición de la entidad en la oración. El resultado de anotar una oración es una tupla, y para entrenar el modelo se necesita una lista de tuplas con las anotaciones de todas las oraciones. Para agilizar el proceso de anotación se realizó de forma automática, etiquetando los ejemplos con el modelo pre-entrenado y etiquetando las nuevas entidades haciendo uso de expresiones regulares, las cuales consisten en patrones de búsqueda en los documentos. El resultado de la anotación es una lista de 5130 tuplas anotadas.

Para entrenar se usó el modelo pre-entrenado “es_core_news_md”, el cual es un modelo oficial de la librería spaCy del idioma español entrenado en noticias y artículos de Wikipedia.

Para la primera ronda se utilizó un método de entrenamiento por pequeñas porciones de datos y entrenando sobre el mismo modelo. La gráfica de pérdida (penalidades por error en predicción) es la siguiente (menor valor significa mejor entrenamiento, un modelo perfecto tiene pérdida de cero):

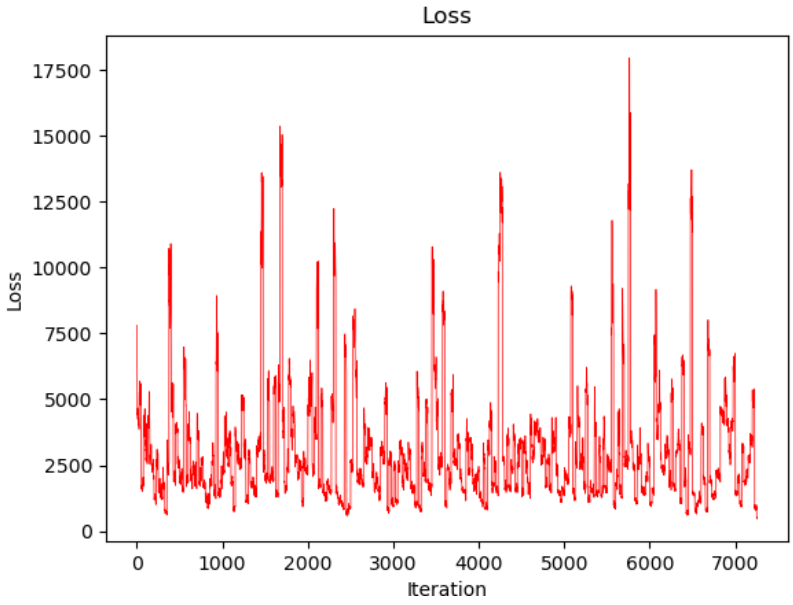

Figura 41. Gráfica de pérdidas del entrenamiento

Se puede observar en la gráfica que, tras el entrenamiento de los 7254 documentos, la pérdida se mantiene relativamente constante. La fluctuación en la gráfica es el comportamiento normal durante sesiones de entrenamiento no continuas, sin embargo, la tendencia debe ser reducir la pérdida durante el entrenamiento completo. Los valores pico de la primera ronda de entrenamiento se encuentran entre 221 y 17,508, sin embargo, no son significativos debido a la tendencia a mantenerse constante en valores entre 1500 y 5000.

No obstante, el principal problema del entrenamiento anterior fue al momento de realizar las pruebas de precisión del modelo. Se pudo observar que el modelo fue afectado por “olvido catastrófico”. El olvido catastrófico es la tendencia de una red neuronal de olvidar catastróficamente todo lo previamente aprendido debido al nuevo aprendizaje. Además, las nuevas entidades entrenadas presentaban inconsistencias en el reconocimiento, esto debido a la cantidad de datos de entrenamiento.

Para corregir el problema del olvido catastrófico, se procedió a agregar 2124 documentos aleatorios y se redujo la cantidad de documentos anotados con las entidades que se desean anotar, para cambiar la proporción de entidades nuevas y antiguas.

Los resultados de la segunda ronda de entrenamiento son los siguientes:

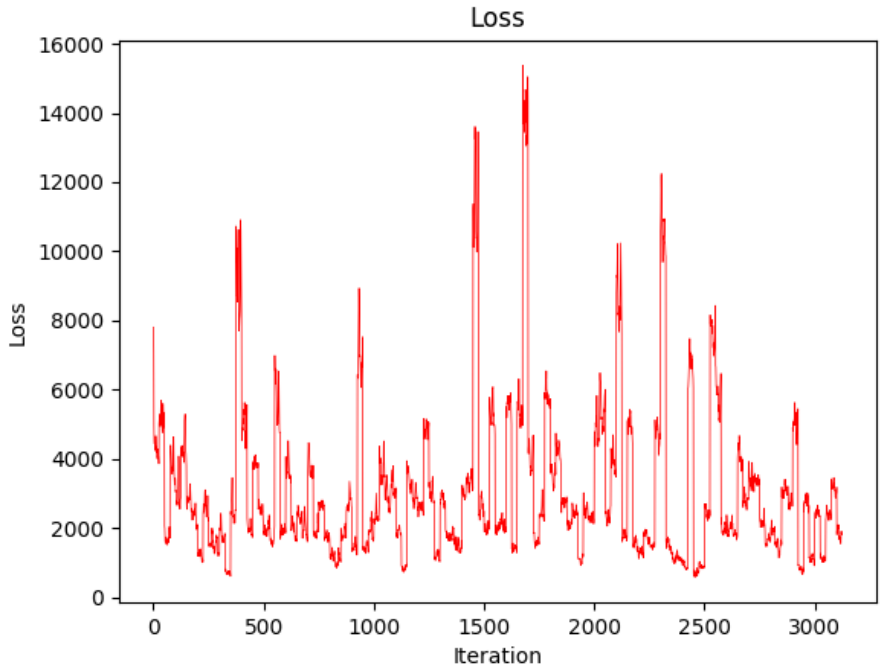

Figura 42. Gráfica de pérdidas actualizada

Se puede observar que se mantienen las fluctuaciones y la tendencia a mantenerse constantes los valores de la gráfica. Al final de la ronda de entrenamiento se realizaron pruebas de reconocimiento de entidades y se obtuvieron resultados favorables en entidades previamente entrenadas, por lo que se puede concluir que se corrigió el problema del olvido catastrófico. Sin embargo, al realizar las pruebas de reconocimiento de nuevas entidades los resultados fueron pésimos. Para un total de 500 documentos que contenían 722 nuevas entidades, solamente fue posible reconocer un total de 118, 16.34% del esperado.

Por lo tanto, comparando el enfoque estadístico de reconocimiento de entidades con el enfoque basado en reglas, se puede concluir que no es factible entrenar un modelo con todas las dificultades que presenta, incluso si se llegase a mejorar el método de entrenamiento, los resultados no estarían cerca de los resultados del enfoque basado en reglas (100% de los casos).

Procesamiento del lenguaje natural

La siguiente parte para el desarrollo del proyecto es el módulo de procesamiento del lenguaje natural, donde se hará uso del modelo actualizado con el enfoque basado en reglas de la sección anterior.

El modelo español de SpaCy contiene una gran cantidad de palabras vacías que pueden ser accedidas y actualizadas mediante la función “is_stop” de un token, sin embargo, no todas las palabras vacías del modelo deben ser reconocidas como tal por el módulo. La librería provee de métodos para agregar o eliminar palabras vacías de la lista, pero siendo las palabras que se desean eliminar demasiadas se prefiere utilizar una lista personalizada que se lee desde archivo de texto. El archivo contiene la lista de palabras o símbolos vacíos separados por saltos de línea. El archivo se lee línea a línea, agregando el contenido de cada una a una lista llamada “swspanish”. Se hace uso de la función strip para obtener solamente el texto, sin saltos de línea o espacios en blanco que pudieron ser agregados por accidente.

Para utilizar el enfoque basado en reglas, se agrega la lista de nuevas entidades en formato “.jsonl”. El archivo phrases.jsonl es un archivo que contiene todas las nuevas entidades que se desean reconocer. El archivo tiene el siguiente formato, con cada entidad separada por un salto de línea.

{ “label” : “GRTNGS”, “pattern” : [{ “lower” : “buenos” }, { “lower” : “dias” }] }
{ “label” : “TIME”, “pattern” : [{ “lower” : “pasado” }, { “lower” : “mañana” }] }

El objeto está dividido en dos partes principales, “label” y “pattern”. Label es la etiqueta que se agrega como nueva entidad y en este caso se utilizaron tres etiquetas: GRTNGS para saludos, SLV para expresiones con contexto especial en El Salvador y TIME para expresiones de tiempo. Pattern es el patrón de búsqueda de entidades y para generalizar se utiliza el modificador lower. Cada pattern contiene la frase que se desea buscar como nueva entidad o la misma separada en objetos por cada palabra. En el caso de “buenos días”, se hace la separación de la frase en dos tokens y a cada uno se le agrega el modificador lower, que denota que la frase

debe ser interpretada en minúsculas, pero que el modelo puede generalizar en cualquier forma que se presenten los mismos tokens en la misma estructura, sin importar la capitalización de cada uno.

Luego de cargar el modelo y agregar la lista de palabras vacías al arreglo de datos swspanish, se procede a agregar un nuevo EntityRuler, que denota las reglas de búsqueda personalizadas de entidades cargando el archivo phrases.jsonl. Se agrega el nuevo EntityRuler al modelo por medio del método add_pipe.

La función process es la función principal de la clase y es la que será invocada cuando quiera utilizarse el módulo. Además del identificador de la clase (self), se recibe como parámetro document, que es el texto que será procesado. Al inicio de la función se inicializa las variables que se utilizarán dentro de la misma. La variable doc contiene el documento procesado por el modelo actualizado; msg contiene el mensaje que será retornado por la función como resultado del procesamiento; id contiene el número de oración del documento que se está analizando; exp contiene el tipo de expresión que se utilizó en la oración.

Luego se itera sobre cada una de las oraciones del documento. Se suma uno al identificador de la oración y se determina la expresión de la misma. Para determinar la expresión se utilizan dos condicionales: en caso de que la oración contenga un símbolo de exclamación, se considera que la oración es exclamativa y en caso de contener un símbolo de interrogación, se considera que la oración es interrogativa. En caso de no cumplirse ninguna de las condiciones, se considera una oración normal.

Con los datos que se tienen, se procede a formar la primera parte del mensaje de retorno. La primera parte del mensaje contiene el instante del tiempo en formato UNIX en milisegundos; el tipo de expresión de la oración; el identificador de la oración. Además, se agrega el identificador words para iniciar la lista de las palabras descompuestas de la oración.

Luego, se itera sobre cada token de la oración. Cada token tiene un ent_iob que es el código IOB del token. I denota que está dentro de una entidad, O que está fuera de la entidad y

B que inicia una entidad. Si se utiliza la denotación por caracteres IOB el proceso es considerablemente más lento que si se utiliza el número que representa cada sigla, por lo tanto, se prefiere el método numérico. I es 1, O es 2 y B es 3. Además, si ent_iob es 0, significa que no es parte de ninguna entidad.

Por lo tanto, si se inicia una entidad con el código IOB 3, significa que es posible que el siguiente token sea parte de la misma. En caso de que el siguiente token tenga código IOB 1, el texto del token se agrega a la entidad anterior y en caso que el token tenga código IOB 2, denotando que está fuera de la entidad se comprueba que el token no sea el inicio de la oración y que el token anterior sea el inicio o parte de la entidad. En caso de cumplirse dichas condiciones, se agrega la entidad anterior a la lista de palabras; en caso de no cumplirse se agrega el token actual a la lista de palabras.

Se procede a formar de nuevo la oración con las entidades formadas. Por ejemplo, para la entidad “Unidad de Salud” el resultado es unidad de salud, esto sirve para identificar la entidad en el sistema intérprete de señas. Para cada token en la nueva oración se procede a verificar si es una de las palabras vacías definidas en el documento; en caso de no serlo, se extraen las características morfológicas y sintácticas del token. Con las características extraídas se forma un objeto que es agregado a la lista de palabras. Una palabra de la lista consiste en la palabra o entidad original, la palabra lematizada y las características extraídas.

Al terminar de recorrer todos los tokens en la oración, se procede a darle el formato final al mensaje para la transmisión. Y la función finalmente retorna el mensaje formado.

Se inicializa la variable recognizer y se lee el archivo de audio como fuente para el reconocimiento. Luego, la función trata de retornar el texto del resultado de la interpretación en el servicio de Google, fijando el idioma a español de El Salvador. En caso que el audio no pueda reconocerse o no contenga palabras, la función retorna "No se puede entender el audio" y en caso existan errores en la solicitud de la interpretación, el software informa sobre ello.

Telegram Bot

La primera parte para el desarrollo del bot en Telegram, es pedirle a BotFather que lo cree. BotFather es un bot que sirve para crear bots. Primero debe iniciarse el bot con el comando /start. Para crear un nuevo bot se debe usar el comando /newbot. BotFather pregunta por el nombre del nuevo bot que desea crearse, en este es "lessamiabot".

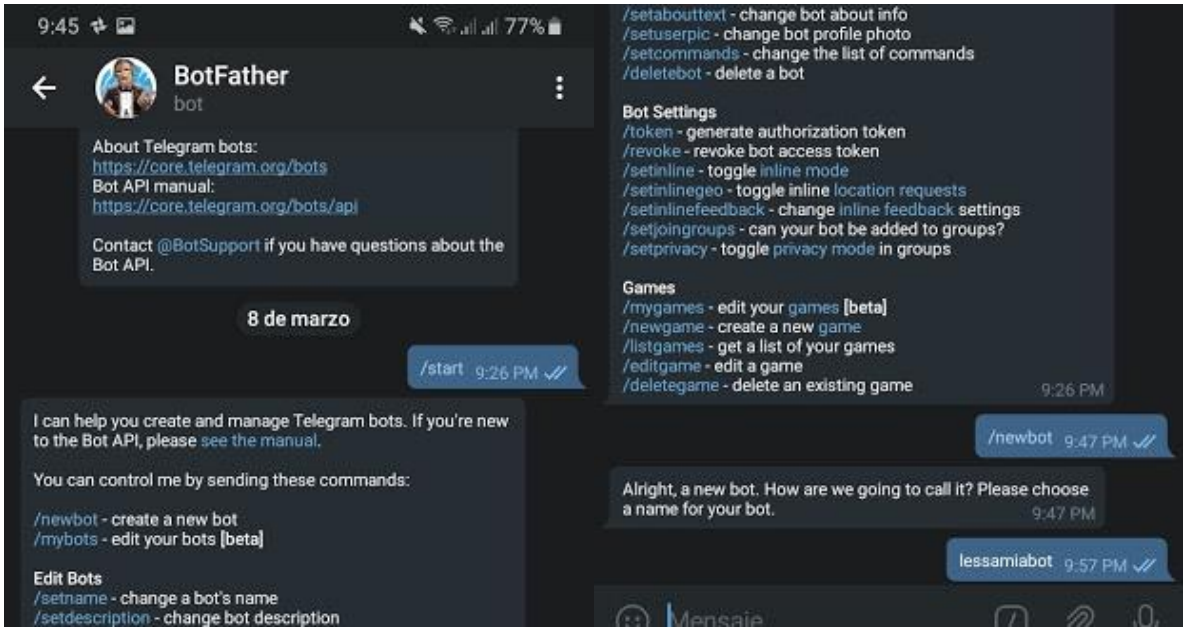

Figura 43. Creación del bot

Luego de asignarlo, se puede acceder a las opciones del nuevo bot, por lo que se procede a personalizarlo con la descripción y la sección “Acerca de” del bot. Además, puede asignarse una imagen que identifique al bot. Luego de esto, el bot queda creado y personalizado y se procede al desarrollo de las funciones que inicien el mismo.

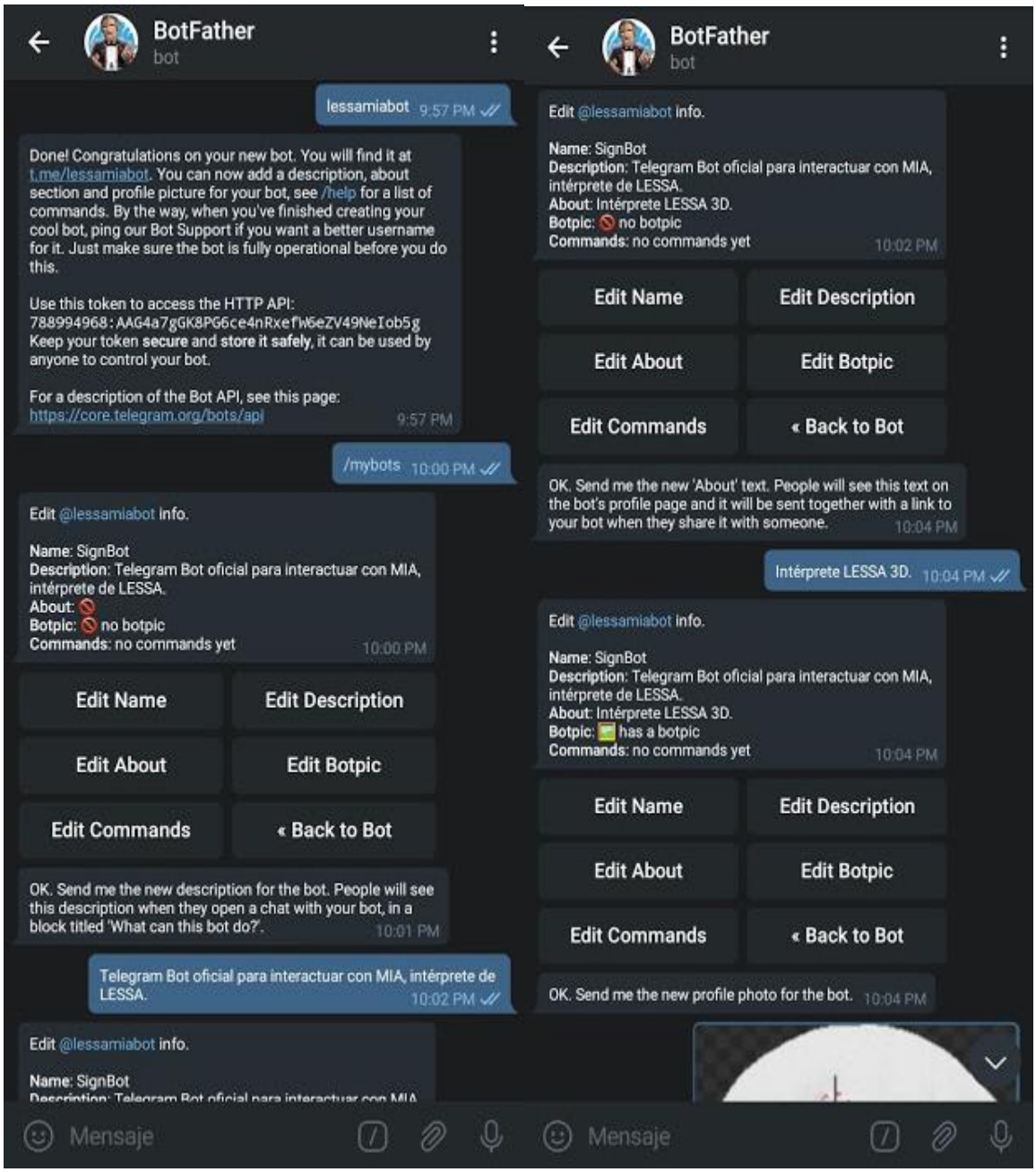

Figura 44. Personalización del bot

Debe definirse la función start, que es la que se ejecuta cuando el usuario inicia el chat en Telegram. Se saluda al usuario y se le pregunta si desea obtener acuse de recibo. El chat muestra un teclado de respuesta con las frases “SI” y “NO” para responder.

La función set_ack se ejecuta para establecer si se desea recibir acuse de recibo durante la sesión. Se notifica al usuario que la selección fue establecida y se le muestra información sobre la ayuda del bot.

Translate es la función principal del bot. Obtiene el mensaje que el usuario envía. El mensaje del usuario puede ser mensaje de texto o mensaje de voz, por lo que se realiza la comprobación del tipo de mensaje. En cualquiera de los casos, si el usuario estableció recibir acuse de recibo, el bot responde con “OK”. Si el mensaje es un mensaje de texto se pasa directamente el mensaje al procesamiento de la clase Processing, el cual retorna el mensaje a transmitir. Con ws.send se envía el mensaje al servicio 3D previamente establecido. Si el mensaje es un mensaje de voz, se descarga el archivo y se hace uso de ffmpeg, una librería de software libre para convertir el tipo de archivo de audio.

Se realiza este paso extra, debido a que Telegram almacena las notas de voz en formato OGG, pero SpeechRecognition solo tiene soporte para archivos de formato FLAC. Luego se hace uso de la función de reconocimiento de voz que se desarrolló previamente y la transcripción es almacenada en la variable transcription, que luego es analizada por la clase Processing para ser transmitida al servicio 3D por medio de WebSockets.

La función help envía un mensaje al usuario con el texto de ayuda del bot.
La función cancel termina la sesión del usuario con el bot y se despide del mismo, usando el nombre de usuario desde donde se había iniciado.

La función main es la que se ejecuta al ejecutar el script del bot, define el updater con el token del bot que fue otorgado por BotFather. Además, se crea el dispatcher y se define el ConversationHandler, que es el que define el flujo de uso del bot. Define como punto de entrada la función start y agrega la funcionalidad de los comandos “/cancel” y “/help”.

Al dispatcher se le agrega el ConversationHandler y se define que para los mensajes de texto y voz que no sea un comando definido previamente se envíe a la función translate. Se inicia el servicio y queda a la espera de conexiones de parte de los usuarios.

CLIENTE MIAWEB

El cliente MiaWEB se ha realizado en el lenguaje de programación JavaScript, siendo este el estándar de desarrollo web. El cliente consiste en una sección con un área de texto para los mensajes que se envían al cliente 3D y una sección para listar los dispositivos a los que se puede enviar mensajes.

El envío de mensajes se hace de la misma manera que en el cliente de Telegram. Se procesan por medio de la clase Processing, pero el NLP se hace en Python y para poder acceder a él desde JavaScript se expuso el mismo en un API escrita en Python, específicamente en el framework Flask, el cual es un framework minimalista que permite crear aplicaciones web rápidamente y con un mínimo de líneas de código.

Para el desarrollo del API en Flask, simplemente se importa la clase Processing que se describió previamente, además de las utilidades de Flask. Se debe definir una ruta donde se atenderán las peticiones y se debe permitir el método POST del protocolo HTTP. En el formulario de datos de la petición se debe incluir un campo que contiene el mensaje llamado “msg”. Por lo tanto, en caso de ser una petición POST se retorna el mensaje procesado en formato JSON. Además, se define una ruta para renderizar la plantilla que contiene la página web principal en la ruta “/”. Flask se ejecuta por defecto en el puerto 5000 en el modo de desarrollo y pruebas.

Al tener el código que atenderá las peticiones para procesarlos en el módulo NLP, se debe proceder a crear la plantilla que se cargará en el cliente web. La plantilla es un archivo HTML sencillo, y por motivos de simplicidad, se incluye el código en
JavaScript en la etiqueta <script> del mismo archivo. Se hace uso de la biblioteca jQuery para simplificar la interacción con HTML.

Los manejadores de evento onClick de los botones s1 y s2, contienen la actualización a la URL del WebSocket y la llamada al método para conectarse al mismo, es decir que en el momento en que el usuario hace click en cualquiera de los dos botones se hace el cambio de conexión al nuevo seleccionado. Las URL que se presentan son exclusivamente para fines ilustrativos.

Luego se define el manejador del evento submit del formulario, es decir cuando el usuario presione el botón Enviar. Luego de hacer las comprobaciones necesarias sobre la existencia de la conexión al WebSocket y que el campo para mensajes contenga un mensaje, se definen las opciones para la petición al módulo NLP. Las opciones consisten en la URL del API que se está ejecutando en Flask, el método del protocolo HTTP que se utilizará (POST) y los datos de la petición. Los datos de la petición se agregan a un diccionario con la llave "msg" y el valor del campo de texto del formulario.

La función para conectar al WebSocket que se mencionó previamente consiste en inicialmente comprobar si existe una conexión previamente establecida, en caso de ser cierto se procede a cerrar la conexión. Luego se eliminan todos los mensajes que hayan sido enviados y se establece la nueva conexión y definen los métodos al recibir mensajes (agregarlos a la lista de la plantilla) y al cerrar la conexión (alertar al usuario).

CLIENTE ANIMADO

El desarrollo del cliente animado se ha realizado utilizando el editor oficial de Unity 3D y en base a la amplia documentación y tutoriales propios de la API de scripting en su versión 2018.4.6f1. Todo el código ha sido desarrollado bajo el lenguaje de programación de c#, utilizando librerías como websocket-sharp 1.0.3-rc11, NEST 7.3.0 y Elasticsearch 7.3.0.

Jerarquía del proyecto y nombres de archivos.

Para el cliente animado la organización se basará principalmente en el tipo de GameObject o Asset, todo archivo debe ir agrupado según su tipo. Por ejemplo, los scripts deben contenerse bajo una carpeta de scripts y todas las imágenes bajo una de imágenes.

A continuación, se presentan las principales carpetas del proyecto:

- **Assets:** Esta carpeta se encuentra presente en todo proyecto Unity y es la carpeta principal, todo archivo, objeto y carpeta es contenido dentro de esta.
- **Animations:** Contiene todas la animaciones u objetos con animaciones, tanto para elementos 3D como personajes u objetos como también para elementos de interfaz.
- **Characters:** Contiene todos los modelos de personajes.
- **Editor**: Carpeta especial, contiene archivos .cs que son utilizados en el editor.
- **Scenes:** Contiene las distintas escenas del programa o pantallas.
- **Prefabs:** Contiene todos los prefabs del proyecto.
- **Scripts:** Contiene todos los scripts utilizados para el juego.
- **Plugins:** Carpeta especial, contiene archivos de librerías.
- **Images:** Contiene todas las imágenes a utilizarse en el proyecto.
- **Settings:** Contiene todos los archivos de configuraciones del proyecto, algunos de estos creados automáticamente gracias a la utilización de una plantilla en la creación.

Cada una de estas carpetas puede contener otra serie de carpetas, que a su vez pueden contener más carpetas y así sucesivamente, a fin de crear una organización que no se base exclusivamente en el tipo, sino también en características o funcionalidades.

El nombramiento de las carpetas y archivos se rigen bajo las siguientes reglas:

- Estar compuesto de una sola cadena sin espacios en blanco, que inicie con mayúscula.
- Si contiene más de una palabra se la debe concatenar igualmente escribiendo el inicio de cada palabra en mayúscula.

- Los nombres de los archivos y por ende de las clases no pueden contener espacios ni caracteres especiales.

Assets

Un asset en teoría es cualquier objeto que puede utilizarse dentro de un proyecto en Unity, ya sean creados desde el editor o importados desde archivos externos. Para el cliente animado se ha realizado el uso de distintos tipos de assets, tales como imágenes, sprites, modelos 3d, animaciones, entre otros. En esta sección se aborda las configuraciones necesarias para los assets.

Archivos de animación

Las animaciones son el asset más importante de este proyecto y deben ser configuradas como se explica a continuación.

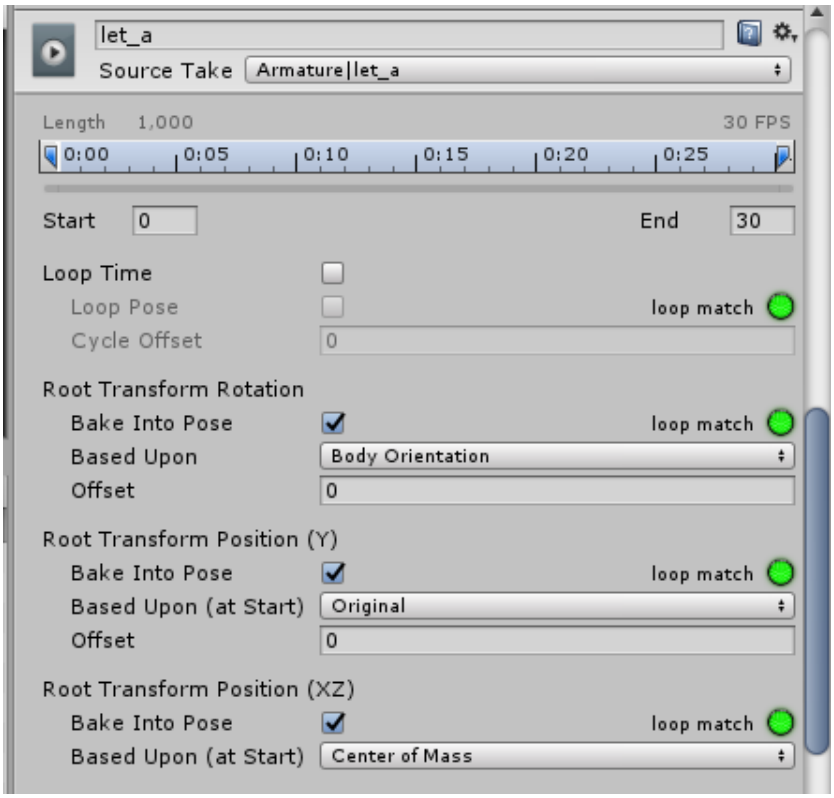

Figura 45. Valores de los archivos de animación

- Renombramiento de animaciones: Las animaciones exportadas por Blender agregan caracteres a los nombres de las animaciones, lo cual generará una falta de correspondencia con la base de datos. Por ejemplo, en la figura 45, la animación fue renombrada de “Armature|let_a” hacia “let_a” eliminando la palabra “Armature” generada por la exportación en Blender.

- LockRootHeighyY, LockRootRotation, LockRootPositionZ:Ya que los modelos para el cliente animado permanecen en una posición fija, no se necesitan generar desplazamientos en ninguna dirección ni rotaciones sobre el modelo en general. Estas tres opciones permiten forzar este comportamiento, por ejemplo, se podría crear una animación de alguien corriendo sin moverse de lugar o saltar sin modificar la posición en Y. Esto es importante ya que en ocasiones las animaciones pueden generar este tipo de desplazamientos o rotaciones no deseadas, y generar efectos como un modelo del personaje dando la espalda, fuera de cámara o inclinaciones en cualquier dirección no deseada.

- Asignar un mismo Avatar: Los avatares son definiciones de cómo las animaciones afectan las transformaciones de un modelo, para nuestros modelos de personajes se utilizarán avatars tipo humanoide, ya que permitirá aplicar el mismo conjunto de animaciones a varios modelos de personajes, característica conocida como retargeting.

La cantidad de animaciones puede crecer casi de manera indeterminada, por lo es necesario procesar estas importaciones y configuraciones de manera automática, a fin de disminuir los errores humanos y reducir el tiempo invertido en esta tarea repetitiva. En Unity es posible crear scripts que son utilizados únicamente dentro del editor, para ello debe hacerse uso de la clase especial AssetPostprocessor, que permite acceder al proceso de importación de assets y ejecutar script antes o después de cada importación. El script permite detectar de manera automática, gracias a la utilización de convención de nombres de archivos, cuando el asset a ser importado se trata de una animación y realiza todas las configuraciones previamente mencionadas de forma automática.

Creación y configuraciones del Proyecto

Para crear un nuevo proyecto al ejecutar Unity se hace click en New o Nuevo, lo que desplegará la vista Crear Proyecto. Desde esta vista es posible nombrar el proyecto, establecer la localización de los archivos y realizar configuraciones previas tales como el uso de plantillas y paquetes. Para el cliente animado se utiliza la plantilla Lightweight RP, que permite crear un proyecto con paquetes y configuraciones pre-establecidos para proyectos con gráficos ajustables a dispositivos con rendimiento bajo. Para finalizar la creación del nuevo proyecto basta con hacer click en el botón Create project. La figura 46 muestra la configuración desde la vista Crear Proyecto de la Pantalla de Inicio.

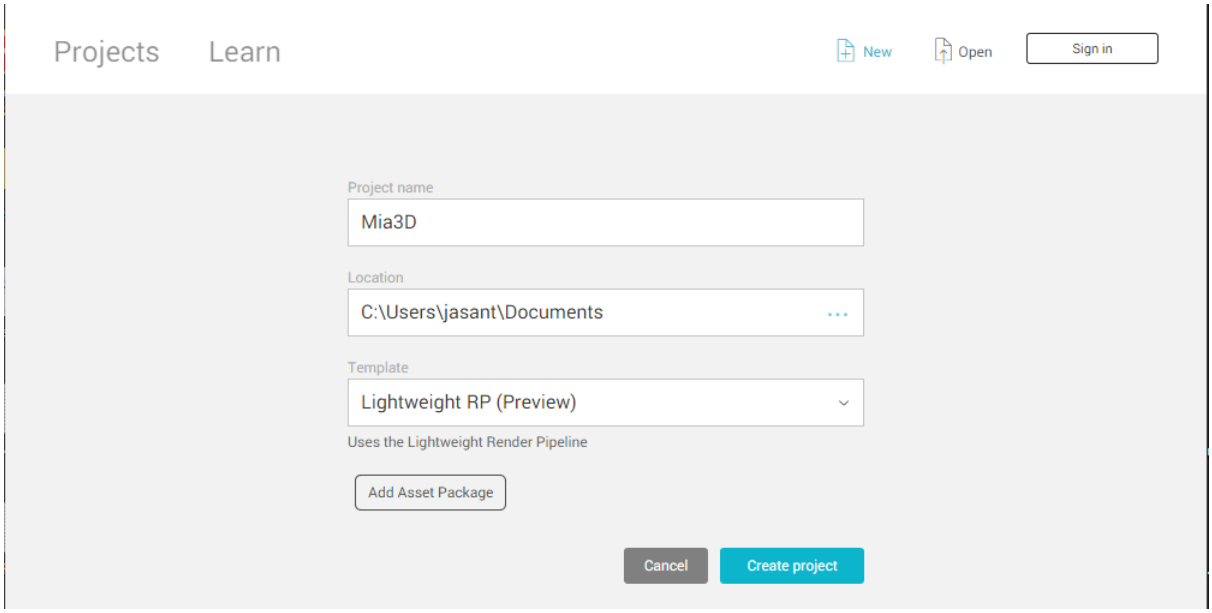

Figura 46. Vista Crear Proyecto

Una vez terminada la creación del proyecto, se abrirá el editor de Unity. La plantilla utilizada provee las configuraciones necesarias para el proyecto, que pueden ser modificadas en cualquier momento desde el menú Edit > Project Settings.

Configuraciones de interfaces de usuario, UI. Las interfaces de usuario o simplemente UI, por sus siglas en inglés, siempre deben ser contenidas por un objeto del tipo Canvas. Estos componentes son los que permiten que todos los elementos de UI puedan ser dibujados en

pantalla. Todas las interfaces del proyecto son elaboradas con una resolución de referencia de 3840 x 2160 pixeles especificada en un componente Canvas Scaler, que permite escalar la interfaz con las distintas resoluciones para evitar así errores como interfaces ilegibles, mal posicionadas, imágenes pixeladas, entre muchos otros. La figura 47 muestra la configuración desde el editor de Unity para el Canvas a utilizar con todos los elementos UI.

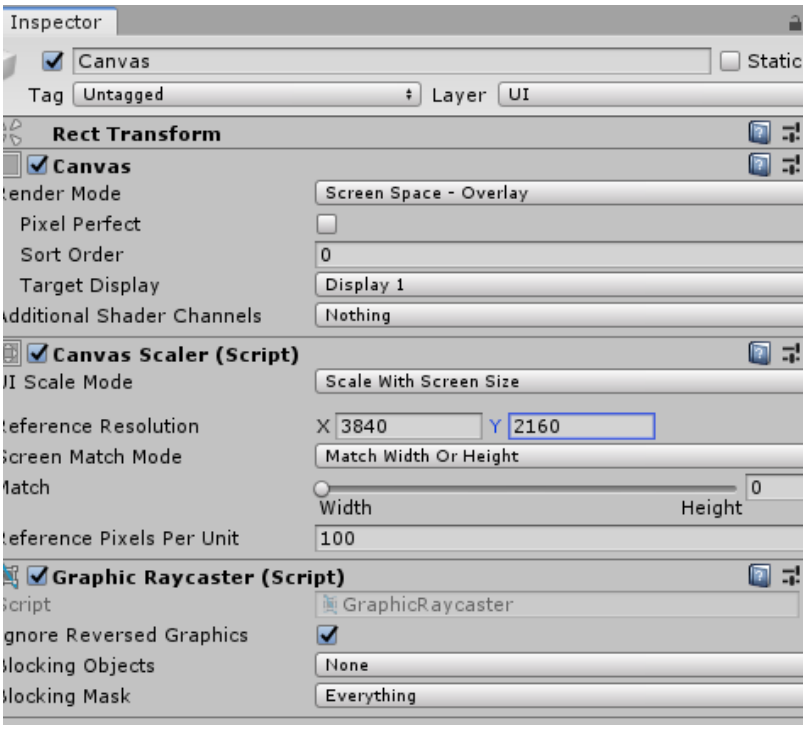

Figura 47**.** Configuración desde el editor de Unity para el Canvas

Managers

Los manager o administradores son elementos que en general permiten realizar el seguimiento del estado en que se encuentra la aplicación, administrar el sistema de pausado, lectura y escritura de información, cambio de pantallas, instanciar objetos, manejo de eventos entre muchas otras funciones según la necesidad.

Clase GameManager. Esta clase permite el manejo de las distintas escenas dentro del proyecto, su carga asíncrona y la transición entre. Permite también el manejo de las preferencias de usuario, su carga al inicio del programa, su modificación durante ejecución y el

almacenamiento en memoria. En un cierre ordenado de la aplicación, esta clase se encargará de ejecutar operaciones de guardado automático de sus archivos de configuración.

Clase EventSystemSystemManager. Esta clase es la responsable crear, registrar, eliminar, escuchar, detener y lanzar los eventos durante la ejecución del programa. Su función principal es permitir la comunicación entre diferentes elementos como componentes y objetos sin la necesidad de existir una referencia directa entre ellos, si no que cada uno es capaz de crear y ejecutar eventos por medio de su implementación en la clase. La utilización de eventos es una característica recomendada, pero es importante tener en cuenta que, si un evento es llamado de manera continua en lapsos cortos, es importante evaluar el uso de referencias directas en su lugar.

Clase MainDispatcher. Todos los objetos relacionados con Unity no son seguros para subprocesos (safe thread) y solo se puede acceder desde el hilo principal o dará como resultado un comportamiento inesperado. Por esta razón se utiliza la implementación de un dispatcher o despachador, que nos permitirá ejecutar acciones generadas desde un hilo secundario (subproceso) sobre el hilo principal. Esta clase permite crear una cola de acciones, que mientras exista un elemento dentro de ella, intentara ejecutar la acción correspondiente sobre el hilo principal del programa.

Para la implementación de los distintos administradores se utiliza el patrón de diseño de instancia única, usualmente conocido por su nombre en inglés como Singleton Pattern. Es necesario el uso de esta ya que asegura que durante toda la ejecución del programa existirá una única instancia sin la posibilidad de duplicados. Además, se logra que la información y las distintas funciones puedan ser accesadas en cualquier momento y por cualquier objeto o componente en el programa.

Un singleton como cualquier otra clase normal dentro de Unity, se deriva de MonoBehaviour, una de las clases que forman el núcleo principal de las API de Unity. Ya que se necesita que la instancia pueda ser accedida por cualquiera otra clase, es necesario crear una instancia del tipo estática para esta clase. Luego para asegurar que la instancia sea única, es

necesario que en la clase se realice una comprobación cada vez que se intente crear u obtener una nueva instancia. Si ya existe una instancia, la comprobación resultara verdadera y la nueva instancia es destruida, caso contrario la nueva instancia se convierte en la instancia única.Es muy importante realizar comprobación de existencia dentro de la función Awake(), derivada de la clase MonoBehaviour, ya que esta función es ejecutada una vez que todos los objetos estén inicializados y antes que cualquier otra función dentro de la misma clase.

La comprobación de instancias también puede realizarse en el método publico get de la instancia, así nos permitirá retornar siempre la misma instancia y realizar su creación si esta no existe.

Escena principal

Esta muestra el resultado final del proceso de interpretación hacia el lenguaje de señas LESSA. En la figura 48 el recuadro rojo muestra los distintos objetos que la componen y el recuadro azul el resultado final para el usuario.

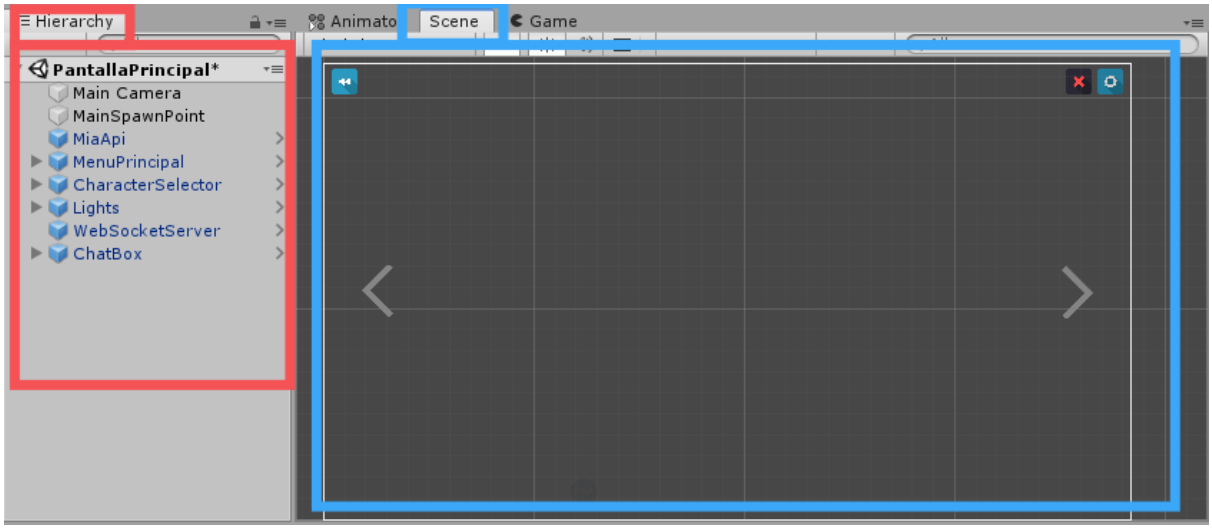

Figura 48. Pantalla principal del cliente animado

A continuación, se describen cada uno de los objetos, su función dentro de la escena y los distintos componentes que los conforman.

Personaje. Es un objeto que contiene el personaje 3D para la interpretación, a través del cual se mostrarán las animaciones correspondientes al lenguaje de señas LESSA. Este no es visible directamente desde la estructura principal de la escena, pero es instanciado en ejecución posteriormente. Además del modelo tridimensional, tiene adjunto dos componentes claves para su funcionamiento: CharacterAnimator y AnimatorQueued.

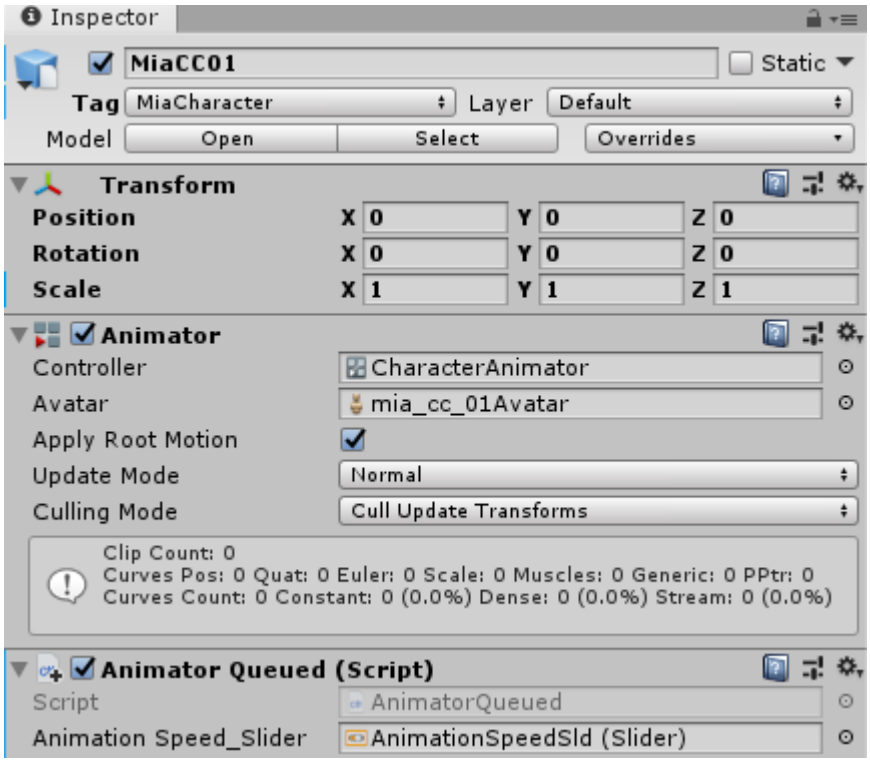

Figura 49. Inspector Para CharacterAnimator y AnimatorQueued

Character Animator. Este componente es del tipo Animator Controller, que es definido por Unity y permite animar al modelo del personaje, indicando que animación debe reproducir y la manera en que lo hace. Además, permite acceder a información para conocer el estado actual, por ejemplo, determinar si una animación se está reproduciendo, el tiempo restante de la animación, su velocidad, si se trata de transición entre otros aspectos.

Desde la ventana Animator pueden manejarse los componentes del tipo Animator Controller, en la cual se pueden definir aspectos como capas, parámetros, animaciones, transiciones, curvas, eventos y configuraciones.

Figura 50. Vista del componente AnimatorController desde la ventana Animator

En la imagen 50 se puede observar que se definen cuatro capass: BaseLayer, SignLayer, FaceLayer y BlinkingLayer. El objetivo de crear múltiples capas es para reproducir diferentes animaciones al mismo tiempo, por ejemplo, para reproducir una animación del personaje saludando y al mismo tiempo otra animación donde esté sonriendo.

A continuación, se detallan cada una de las capas utilizadas:

- BaseLayer: Se crea una capa vacía colocándola en primera posición, el único objetivo de esta capa es el de establecer la posición del hueso base o RootBone del personaje, para mantener su posición fija. Para la primera capa es necesario que esta sea tipo override y sin máscaras.

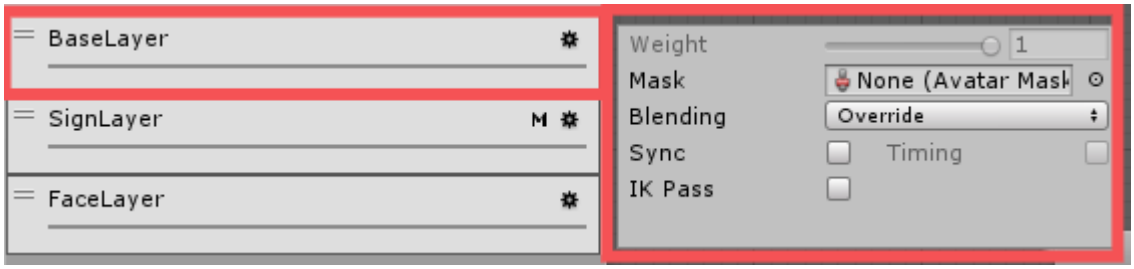

Figura 51. Capa BaseLayer del componente Animator Controlle

- SignLayer: Esta capa constituye la capa principal, puesto que almacenará las referencias hacia a todas las animaciones disponibles, estas referencias se denominan estados y es a los cuales el modelo puede transicionar o ejecutar directamente. En su configuración es muy importante colocar un peso o weigth de 1 y que sea del tipo Override, ya que de esta manera estas animaciones tendrán prioridad sobre cualquiera otra animación.

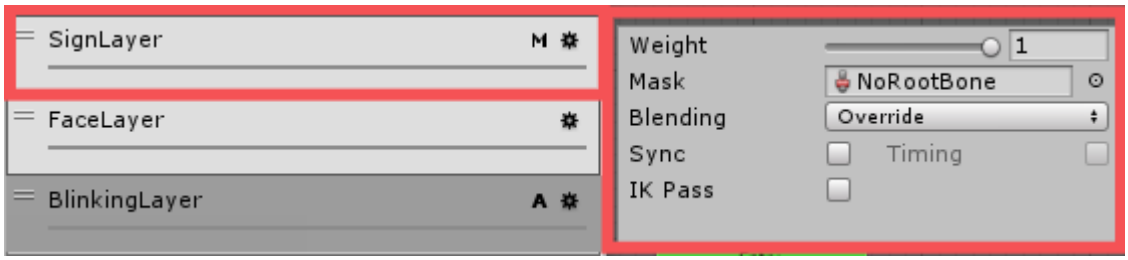

Figura 52. Capa SingLayer del componente Animator Controller

Como se puede observar en la figura 52, esta capa posee una máscara llamada NoRootBone, que se utiliza para excluir todos los huesos que no forman parte de las animaciones para las señas de lessa, como lo es el hueso principal y las extremidades inferiores.

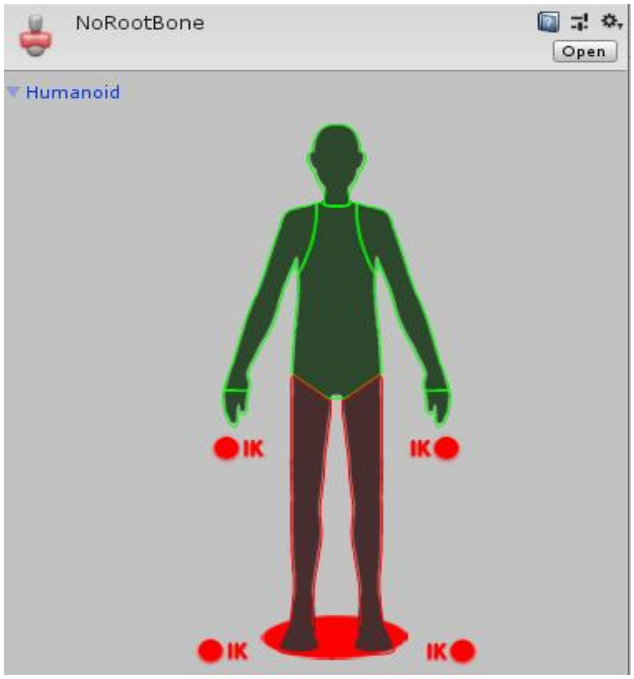

Figura 53. Máscara “NoRootBone”

- FaceLayer: Esta capa contiene únicamente las animaciones para las expresiones faciales del personaje. Estas animaciones como detalló previamente en la sección de animación de este capítulo, son realizadas por medio de blendshapes por lo cual el uso de máscaras en esta capa no tendría efecto.

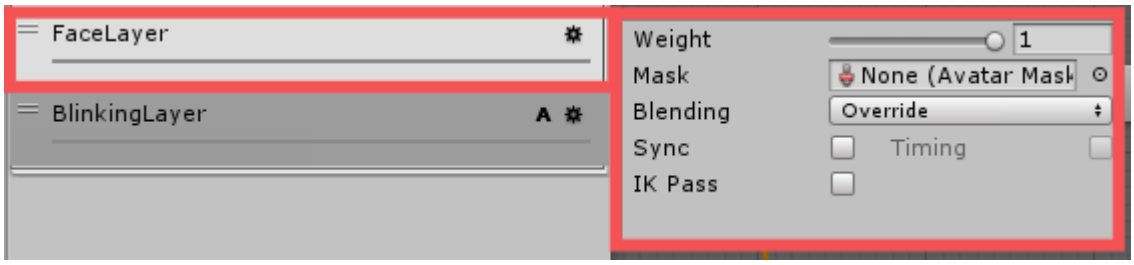

Figura 54. Capa FaceLayer del componente Animator Controller

- BlinkingLayer: Esta capa únicamente contiene una animación para simular el parpadeo del personaje. Se realizó sobre una capa aparte a las expresiones faciales ya que se busca que el parpadeo esté presente incluso durante las expresiones faciales. Esta capa es la única que se encuentra en modo Additive, por lo cual la animación de parpadeo puede

sumarse a las otras animaciones y no sustituirlas. Además, si dos animaciones afectan los ojos del personaje, esta capa cederá el control.

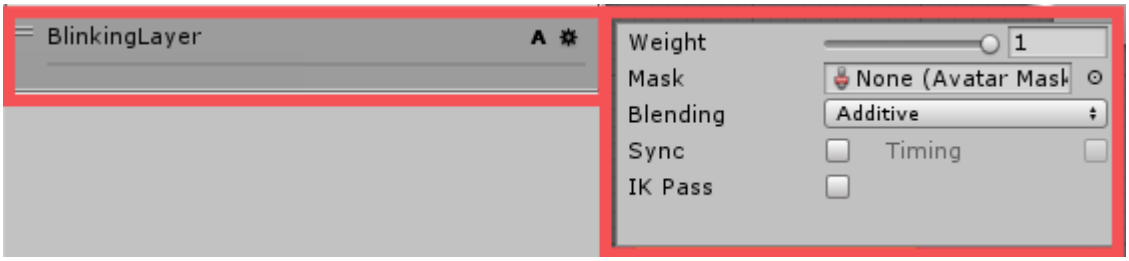

Figura 55. Capa BlinkingLayer del componente Animator Controller

Clase AnimatorQueued. Esta clase permite controlar la reproducción de las animaciones del personaje. Por medio de ella se verifica la disponibilidad de animaciones en la cola de la instancia de la clase MiaApi, para su posterior reproducción sobre el personaje tomando en cuenta tanto la seña LESSA como su respectiva expresión facial.

Debido a que el número de animaciones a utilizar es alto y cada animación puede transicionar hacia cualquiera, se generan miles combinaciones y vuelve imposible su manejo por medio de la interfaz que Unity provee. Por lo anterior, es esta clase la encargada de manejar la transiciones y velocidad entre animaciones. De igual manera es mediante esta clase que se determina el estado de inactividad de la aplicación utilizando una especie de temporizador o contador que es lanzado al no existir animaciones en ejecución y luego de un tiempo determinado la aplicación retorna a la pantalla de inicio o reposo.

Para el manejo de las animaciones faciales se utilizan BlendTrees o árboles de mezcla, utilizados para que múltiples animaciones sean mezcladas. Por ejemplo, un rostro en un momento dado puede estar mezclado entre una animación de expresión simple con un grado de 60% y una animación de expresión triste con grado de 40%, que podría significar ya sea que el personaje está pasando de un estado simple a uno triste o viceversa. Para realizar transiciones entre las animaciones de expresiones faciales se modifican valores de 0 - 1, tanto de la expresión actual y la expresión siguiente. En una transición al comienzo una animación tendría un valor

de 1 y la otra de 0, en una etapa intermedia ambas tendrían valor de 0,5 y finalmente los valores iniciales se verían invertidos.

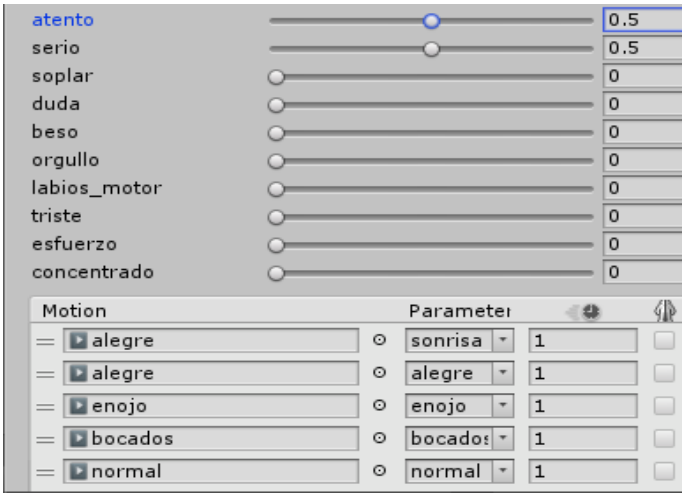

Figura 56. BlendTree para el manejo de expresiones faciales

MainCamera. Cámara principal y única de la escena, es creada automáticamente por el editor de Unity. Para lograr la vista deseada se ha realizado una configuración tal como se muestra en la figura 57.

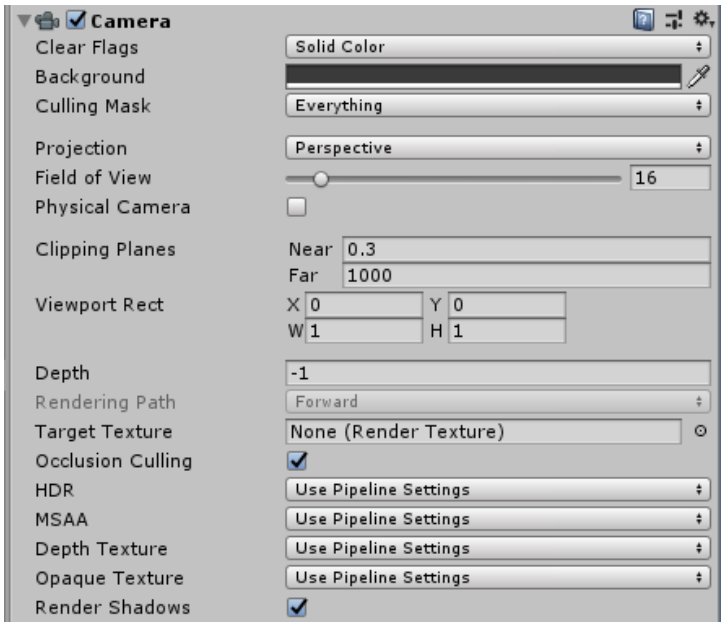

Figura 57. Configuración de cámara en escena principal

Es necesario que la cámara se encuentre en modo perspectiva, ya que de esta manera se obtiene una mejor percepción de la profundidad de los objetos en escena, lo que es conveniente para la claridad en movimientos o animaciones que se realicen de adentro hacia afuera, es decir en el eje Z.

ChatBox. Es un objeto que contiene una ventana estilo chat, por medio del cual se puede enviar mensajes para ser animados sobre el personaje, siendo esta una alternativa a los mensajes recibidos por websocket.

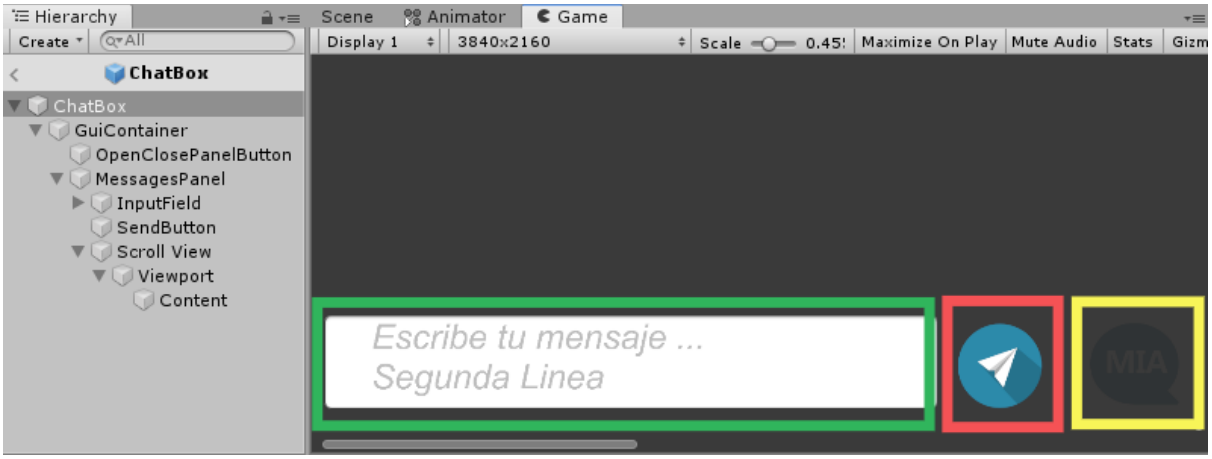

Figura 58. Visualización del objeto ChatBox

En cuanto a interfaz la ventana de chat es unos de los elementos más complejos de lograr, ya que se necesita un panel creciente que muestre una fracción y que permita la navegación para visualizar más contenido. Además, cada mensaje que es mostrado y colocado en la ventana es un objeto previamente preelaborado donde únicamente se le agrega texto al momento de su instanciación.

OpenCloseButton. Señalado por un recuadro amarillo en la figura 4.5.14, es un objeto del tipo Button que mediante el uso del evento OnClick permite mostrar y ocultar el cuadro de chat. El evento hace referencia directa hacia el objeto MessagePanel y se ejecuta el método ShowHidePanel que se encarga de mostrar y ocultar dicho panel.

SendButton. Señalado por un recuadro rojo en la figura 58, es un objeto del tipo Button que mediante el uso del evento OnClick permite enviar el texto capturado mediante InputField y generar la animación de su correspondiente sobre el modelo del personaje.

MessagesPanel. Objeto que funciona como contenedor de la interfaz principal del ChatBox, que puede ser mostrada y ocultada. Únicamente posee un componente del tipo Image que se utiliza para dar un fondo al panel.

InputField. Señalados por un recuadro azul en la figura 58, muestra un objeto con componente del tipo InputField que permite al usuario ingresar texto mediante el teclado para su envío y posterior animación.

ScrollView y Viewport. Estos dos objetos en su conjunto permiten restringir el área en donde se muestran los mensajes. El manejo del área consiste en el tamaño máximo que el panel puede contener y la fracción que será mostrada en pantalla.

Content. Como su nombre lo dice este objeto será el lugar donde todo el contenido, es decir cada mensaje a mostrar será colocado. Para su correcto funcionamiento se utilizan los componentes Content Size Fitter y Vertical Layout Group que permiten manejar los tamaños de cada mensaje, su posición y distancia entre cada uno de ellos.

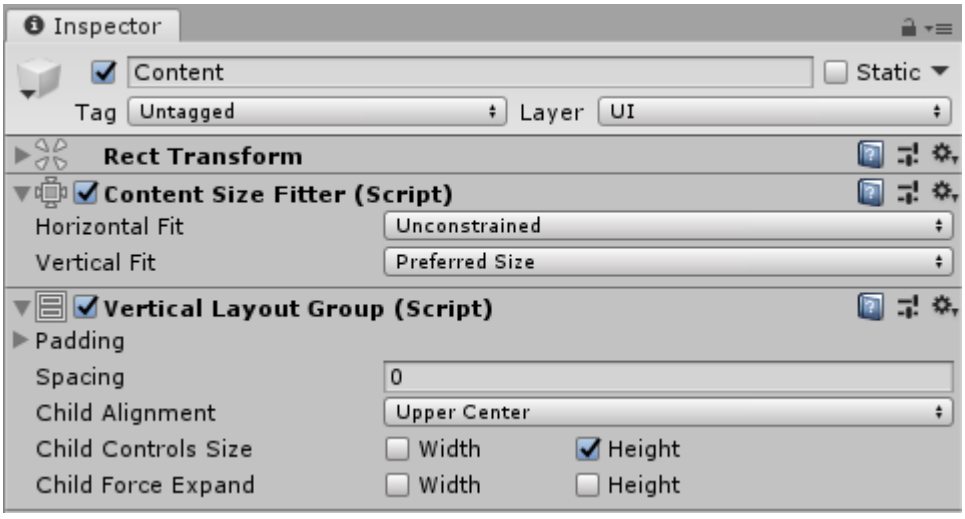

Figura 59. Configuración del objeto Content de ChatBox

CharacterSelector. Este objeto es el responsable de instanciar al personaje en la escena, además permite cambiar entre los distintos personajes disponibles. Este objeto tiene adjunto, además de los componentes básico para elementos UI, un componente del tipo script que hace referencia hacia el archivo CharacterSelector, tal como se muestra en la figura 60.

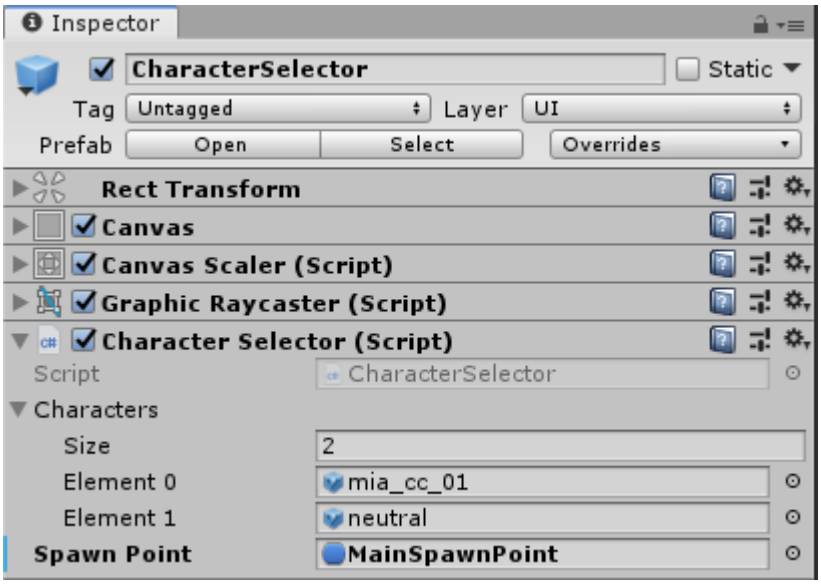

Figura 60. Configuración del objeto CharacterSelector

Mediante el uso de un arreglo de personajes que es prácticamente una lista de los personajes disponibles, se permite la selección del personaje deseado. Para la selección del personaje es necesario destruir el personaje en pantalla mediante el método Destroy y posteriormente instanciar el nuevo personaje mediante el método Instantiate, ambos perteneciente al api de Unity. La variable SpawnPoint es únicamente un objeto con la posición en que se desea colocar al personaje. Es necesario asignar las referencias hacia los personajes a utilizar y la variable SpawnPoint.

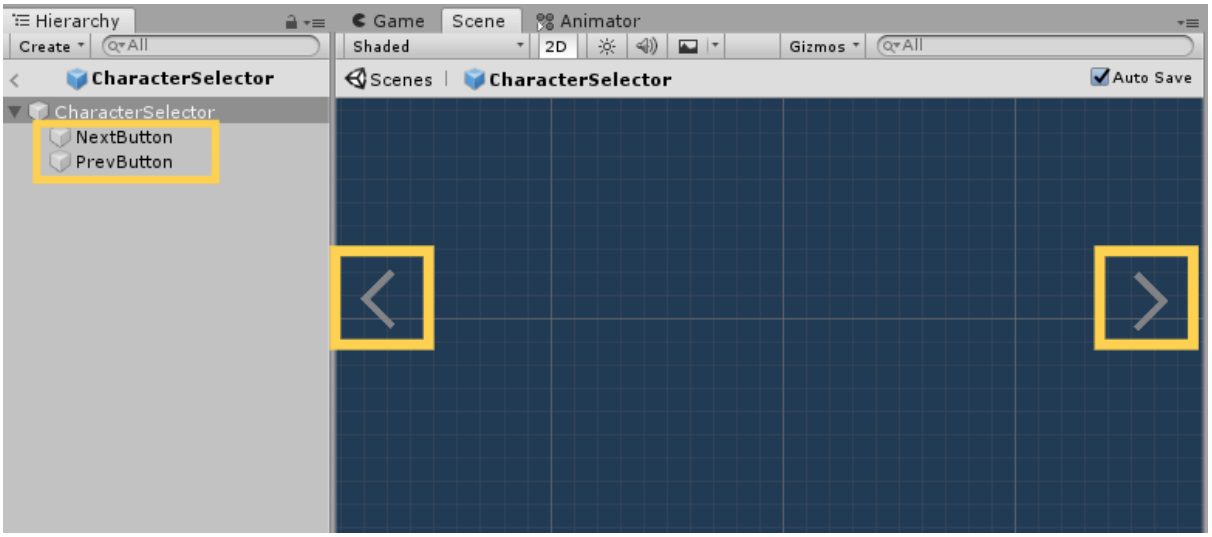

Figura 61. Vista en escena de CharacterSelector

NextButton y PrevButton. objetos con componente del tipo Button, que nos permiten navegar en el arreglo de personajes para poder realizar su selección. Estos mediante el evento OnClick hacen referencia a los métodos NextCharacter y PrevCharacter respectivamente. Los recuadros amarillos en la figura 61 señalan estos elementos.

WebSocketInstance, ElasticCliente y MiaApi. Estos tres objetos son creados a partir de objetos vacíos y que únicamente poseen adjunto un componente del tipo script, los archivos adjuntos respectivamente son WsIntance, ElasticSearchClient y MiaApi.

Clase WsInstance. Esta clase permite el manejo conexión con websocket, manejando su creación, conexión, desconexión, errores, recepción y envío de mensajes. Posterior a la recepción de un mensaje esta clase de encarga de enviarlo hacia la instancia de la clase MiaApi para su procesamiento. Se utiliza la librería websocket-sharp para su implementación, para lo cual es necesario agrega el archivo websocket-sharp.dll en la carpeta plugins del proyecto.

Clase ElasticSearchClient. A través de esta clase se realiza la conexión con el servicio de ElasticSearch, donde se realiza la búsqueda de la información correspondiente a las distintas señas de LESSA recibidas por medio de websocket. Esta clase permite realizar operaciones de lectura y escritura de la información.

Clase MiaApi. Cada mensaje enviado desde la instancia en escena de la clase WebSocketInstace, es transformado en un elemento según el modelo de la clase Animation y posteriormente es agregado a una estructura de cola, siendo así que el primer elemento en ingresar sea el primero en salir. Previamente a su ingreso a la cola, cada elemento es buscado por medio de la instancia en escena de la clase MiaApi y luego dentro de las animaciones disponibles. Si una animación no existe, esta clase se encarga de descomponer la palabra en letras y cada letra es agregada a la cola.

Ilumination. Está compuesto por dos objetos con componentes del tipo Ligth, utilizados para brindar la iluminación necesaria en la escena, a fin de crear una vista clara del personaje y sus animaciones.

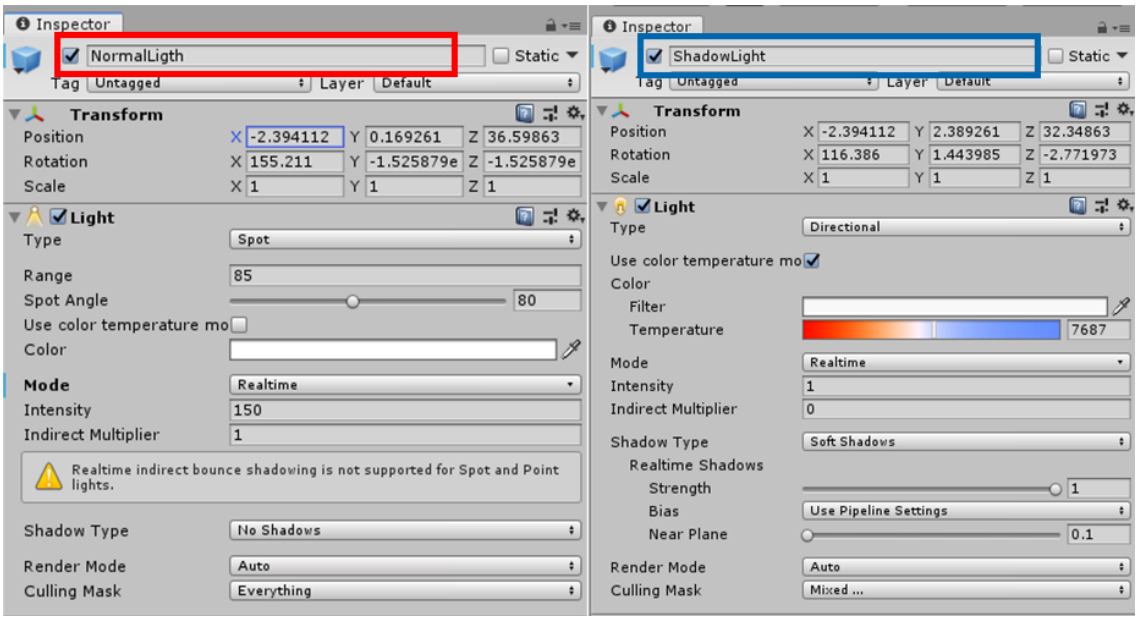

Figura 62. Vista en escena de CharacterSelector

La primera luz NormalLight como se puede observar en la figura 62 señalado por recuadro rojo, se ha utilizado únicamente para aclarar la escena en manera general y no genera ningún tipo de sombras. La segunda luz señalada en misma figura por el recuadro azul, se ha utilizado para mejorar el efecto de profundidad en la escena gracias a la proyección de sombras. La iluminación es muy importante, ya que mejora no solo la calidad del personaje si no también la claridad de las distintas animaciones.

Figura 63. Proceso de iluminación

Pantalla de inicio

En la figura 64 se muestra esta escena en el editor de Unity, en la ventana Hierarchy señalada por un recuadro rojo se pueden observar todos los elementos que componen la escena y en la ventana Scene señalada por un recuadro azul el resultado.

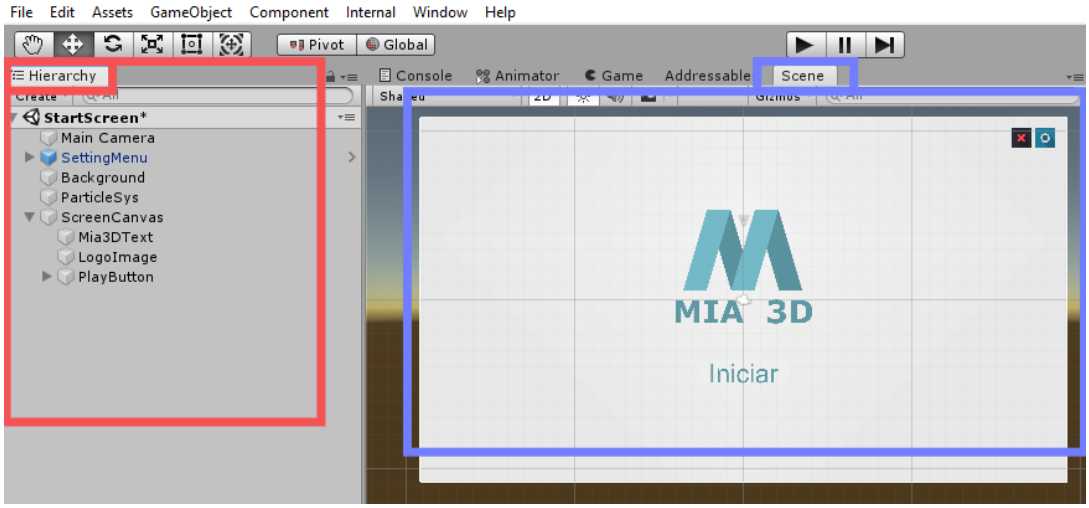

Figura 64. Pantalla de inicio del cliente animado

A continuación, se describen cada uno de los objetos, su función dentro de la escena y los distintos componentes que los conforman.

MainCamera. Cámara principal y única de la escena, es creada automáticamente por el editor de Unity. Para lograr la vista deseada se ha realizado una configuración mediante el editor tal como se muestra en la figura 65.

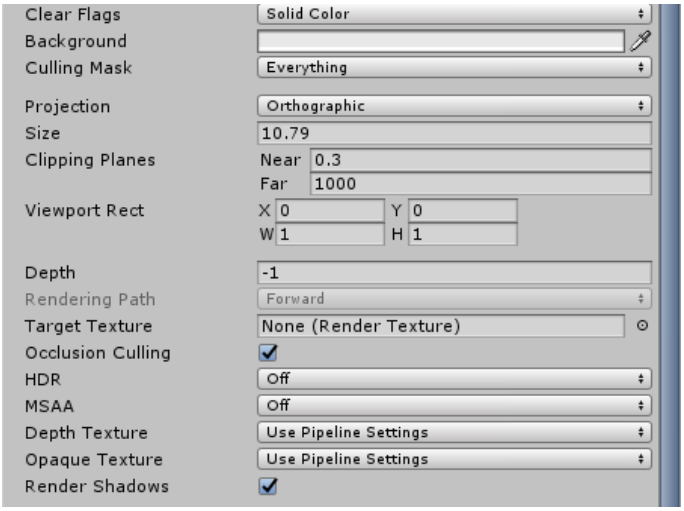

Figura 65. Configuración de cámara en la pantalla de inicio

ScreenCanvas. Es el canvas principal de la escena, contiene tres elementos UI: imagen para el logo, texto para el título, y un botón. Además, a esto, posee un componente script adjunto que hace referencia hacia el archivo StartScreen. Al presionar el botón Start de este objeto, por medio de su evento OnClick se ejecuta el método StartApp del archivo StartScreen, que permite avanzar hacia la pantalla principal y a su vez ejecutar un guardado de los archivos de configuración.

Menú Principal

Es un objeto que contiene la interfaz de usuario para el menú del cliente animado, el cual permitirá realizar operaciones tales como configuración del cliente, retorno hacia la pantalla de inicio y el cierre ordenado de la aplicación. Este menú será utilizado en las distintas pantallas del cliente con pequeñas variaciones en cada una. En la figura 66 se puede observar este el menú dentro del editor de Unity, en la ventana Hierarchy señalada por un recuadro rojo podemos observar todos los objetos que lo componen y en la ventana Scene señalada por un recuadro azul el resultado en escena.

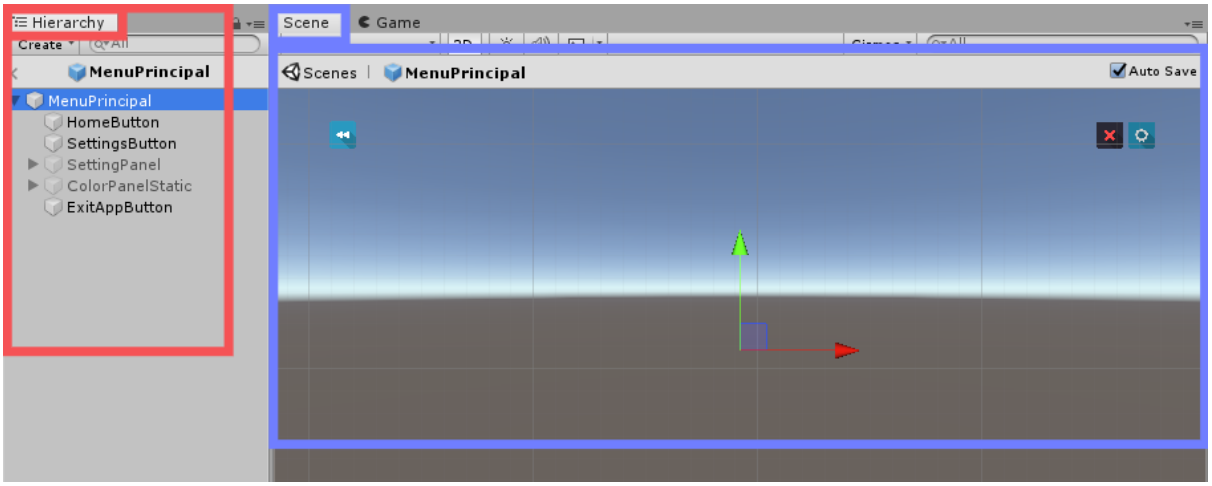

Figura 66. Visualización de Menú Principal desde la vista de escena

El menú principal tiene adjunto un componente del tipo script, que hace referencia hacia el archivo SettingsMenu, mediante el cual se establecen referencias directas hacia distintos objetos como SettingsPanel, AnimatorSpeedSlider y LetterSpeedSlider que deben ser especificadas desde el editor para su correcta inicialización de valores en ejecución.

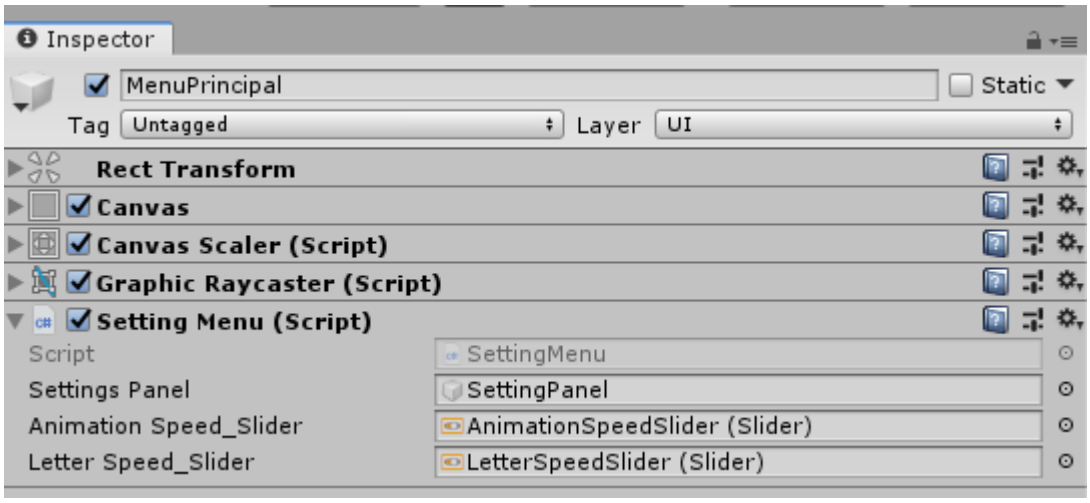

Figura 67. Configuración del objeto MenuPrincipal

A continuación, se describen cada uno de los objetos, su función y los distintos componentes que conforman el menú principal.

HomeButton, ExitAppButton y SettingsButton. Como su nombre lo denota son objetos con un componente del tipo Button adjunto, que mediante el uso del evento OnClick permite realizar la ejecución de métodos según el caso específico de cada botón, tal como se detalla a continuación.

- **HomeButton:** El evento de este botón hace referencia hacia el método ReturnHome del script SettingsMenu, permitiendo así el cambio de escena hacia la pantalla principal.

- **ExitAppButton:** El evento de este botón hace referencia hacia el método ExitApplication del script ExitHelper, brindando así la funcionalidad de terminar la aplicación de manera ordenada mediante el GameManager.

- **SettingsButton:** El evento de este botón hace referencia hacia dos métodos distintos, el primero es el método OpenCloseMenu del script SettingsMenu que nos permitirá mostrar y ocultar el panel de configuraciones SettingsPanel; el segundo es hacia el método SetActive del objeto ColorPanel y se ejecuta pasando un valor falso al parámetro solicitado. Ambos eventos son necesarios para mostrar y ocultar correctamente los paneles mediante la interacción del usuario.

SettingsPanel. Es un objeto que contiene un panel para las configuraciones del cliente como el color de preferencia, velocidad de deletreo y velocidad de señas. En la figura 68 se puede observar los distintos objetos que lo componen y el resultado.

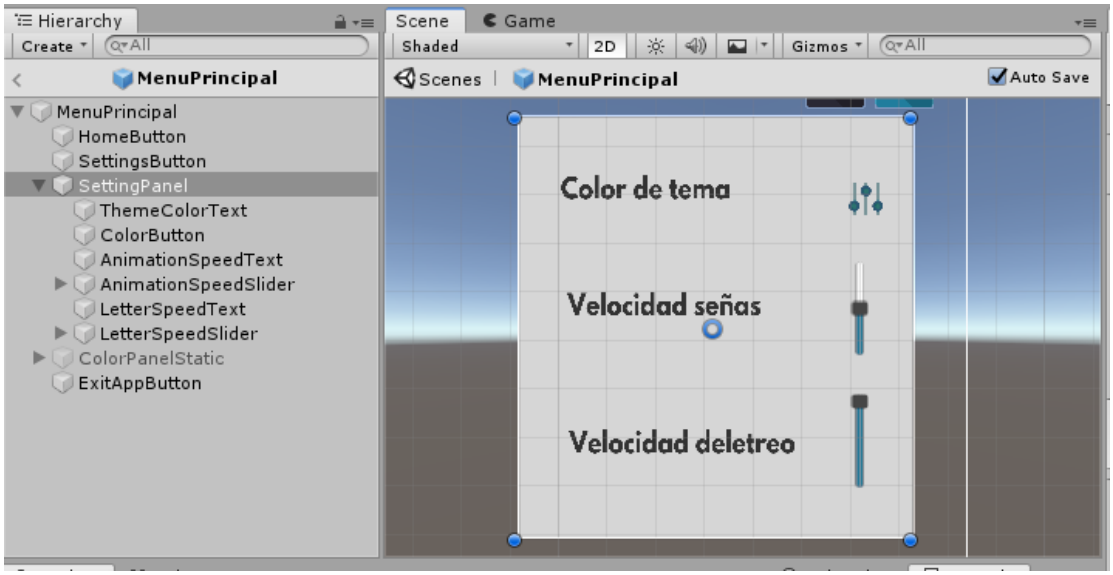

Figura 68. Visualización de SettingPanel del menú principal

LetterSpeedSlider y AnimationSpeedSlider. Por medio de estos objetos se pueden modificar los valores de la velocidad para las animaciones de deletreo y animaciones de señas respectivamente. Se utilizan objetos con componentes del tipo slider, que brindan una barra deslizante para ajustar valores y que al ser modificados se ejecuta el evento OnChangeValue permitiendo ejecutar métodos que se utilizan para actualizar los cambios en las configuraciones sobre el GameManager.

ColorButton. Objeto con un componente del tipo Button, que mediante el uso de dos eventos OnClick permite mostrar y ocultar el panel ColorPanel. El primer evento hace referencia directa hacia el objeto ColorPanel y se ejecuta el método SetActive pasando un valor verdadero al parámetro solicitado. El segundo evento hace referencia directa hacia el objeto SettingsPanel y se ejecuta el método SetActive pasando un valor falso al parámetro solicitado. Ambos eventos son necesarios para mostrar y ocultar correctamente los paneles mediante la interacción del usuario.

ColorPanel. Es un objeto que contiene un panel para la selección de los distintos colores del tema de la aplicación. En la figura 69 se puede observar los distintos objetos que lo componen y el resultado.

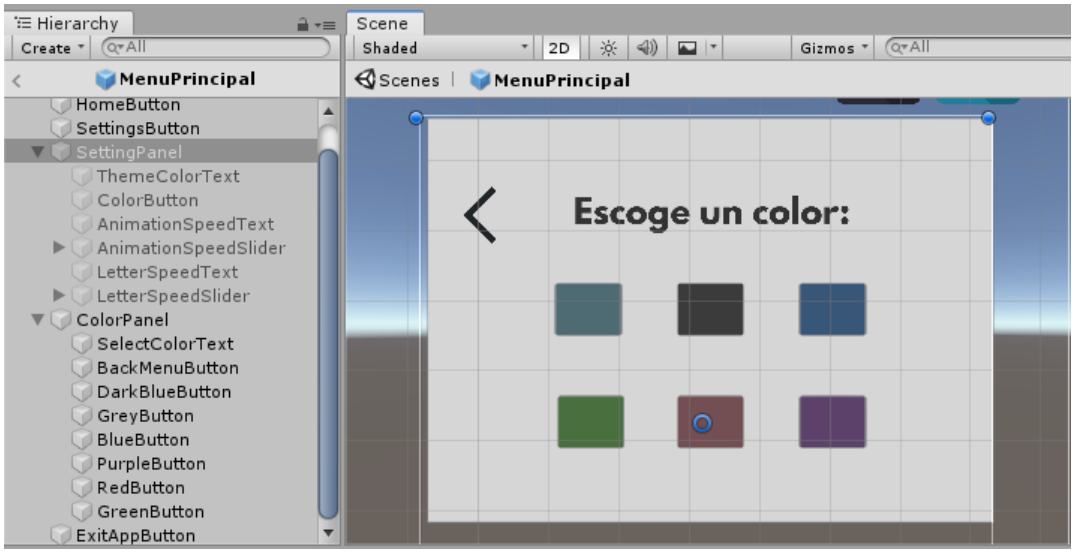

Figura 69. Visualización de ColorPanel del menú principal

ColorButton. Para la selección del color se hace uso de 6 objetos, cada uno de ellos con un componente del tipo Button, que mediante el uso del evento OnClick permite realizar el cambio del color, dicho evento hace referencia hacia el método ChangePrefColor del script SettingMenu, pasando un parámetro del tipo string con el valor del color correspondiente en código hexadecimal.

BackToSettings. Objeto con un componente del tipo Button, que mediante el uso de eventos OnClick permite ocultar el panel ColorPanel. El primer evento hace referencia directa hacia el objeto ColorPanel y se ejecuta el método SetActive pasando un valor falso al parámetro solicitado. El segundo evento hace referencia directa hacia el objeto SettingsPanel y se ejecuta el método SetActive pasando un valor verdadero al parámetro solicitado.

Los demás objetos como SelectColorText, ThemeColorText, AnimationSpeedText y LetterSpeedText contienen únicamente componentes del tipo Text. La figura 70 muestra la configuración que se utiliza en cada uno de ellos.

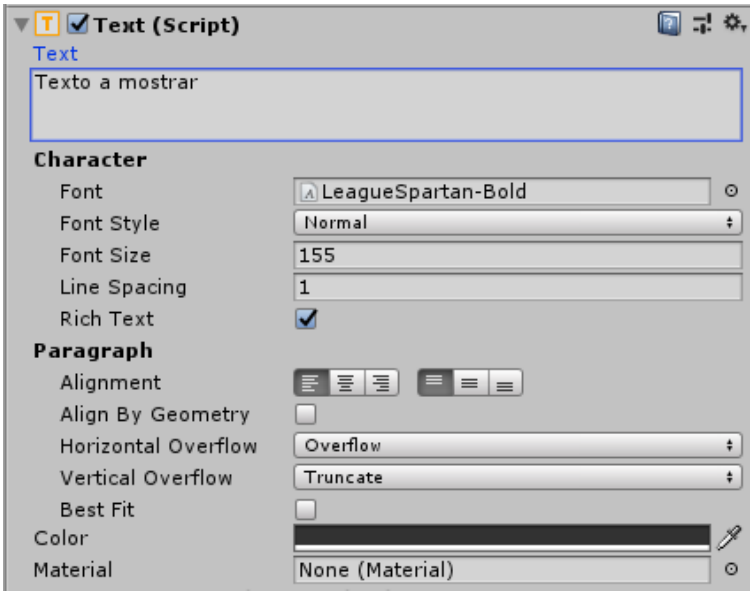

Figura 70. Configuración para los distintos objetos Text de ColorPanel

Se utilizan tamaños de fuente 155 y una escala reducida con el objetivo de asegurar la calidad del texto en las distintas resoluciones disponibles, evitando errores como textos ilegibles o borrosos para el usuario.

OTROS DESARROLLOS

Anteriormente en este capítulo se detallaron todos los clientes, software y herramientas resultantes del desarrollo, que forman parte directa del resultado o demo final del proyecto. Este apartado busca mostrar aquellos elementos que sirvieron para su desarrollo, pero que no forman parte de la funcionalidad del producto final.

Aplicación web para datos de animaciones

Como se detalló en el capítulo de diseño, se ha utilizado el motor de ElasticSearch para realizar la búsqueda de las señas disponibles. Para facilitar el manejo de estos datos se desarrolló una aplicación web mediante net core 2.2 utilizando el patrón MVC. El objetivo es proveer una interfaz sencilla y rápida para la visualización, creación y actualización de los datos correspondiente a las señas LESSA. La aplicación es desplegada utilizando contenedores Docker, asegurando su funcionamiento en los distintos sistemas operativos.

Figura 71. Aplicación web para datos de animaciones, inicio

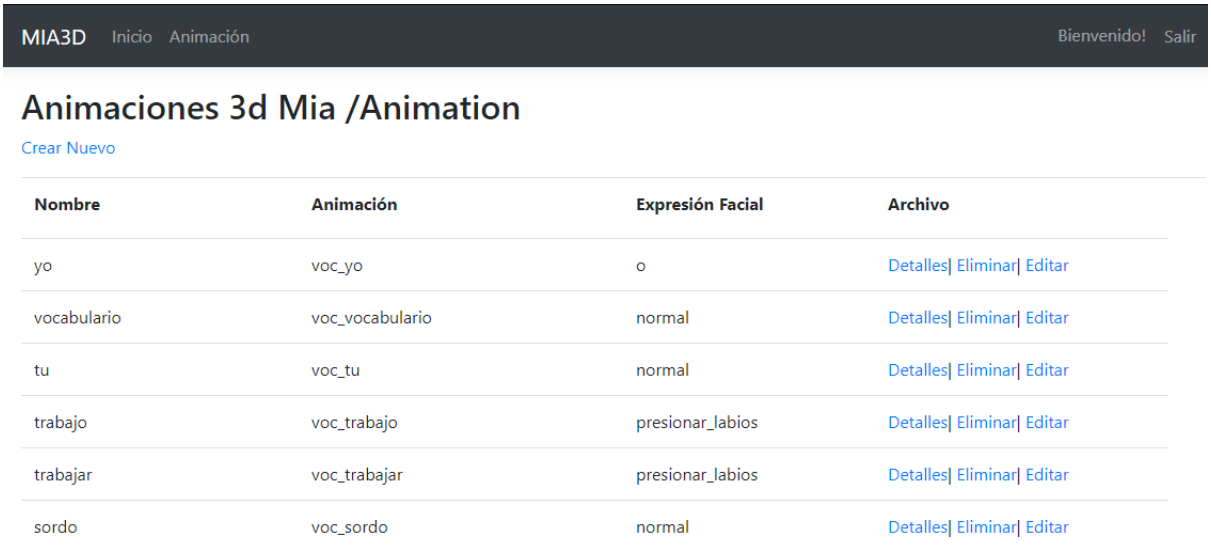

Figura 72. Aplicación web para datos de animaciones, listado

Se utilizaron servicios de terceros en sus versiones gratuitas como Bonsai y Heroku para el despliegue de elasticsearch y la aplicación web respectivamente, facilitando así la disponibilidad y el trabajo en equipo para el proyecto.

CAPÍTULO V
DISEÑO Y RESULTADO DE PRUEBAS

DISEÑO DE PRUEBAS

Pruebas de Usuario

El tipo de prueba de usuario seleccionado para el testeo de calidad de señas de MIA, son las pruebas Sumativas y Presenciales de Guerrilla, especificadas en el Marco Teórico del mismo documento. Se considera importantes las pruebas de usuario para el desarrollo del proyecto, ya que este está enfocado en la ayuda para un grupo de personas específico, para su beneficio, en consecuencia, es importante conocer lo que las personas con discapacidad auditiva necesitan en ese escenario específico.

Desarrollo de la Prueba. El desarrollo de esta prueba tiene como objetivo comprobar el grado de comprensión de las señas animadas e interpretadas por MIA, para ser entendidas con exactitud por personas conocedoras de LESSA, siendo calificadas por el usuario: personas con discapacidad auditiva, a las cuales se beneficiaría con el desarrollo del proyecto.

La prueba consiste en cuatro partes diferentes en la que al usuario se le mostrarán diferentes señas de LESSA a través de MIA, que el usuario deberá observar y luego escribir, de poder hacerlo, el significado correspondiente a lo que se muestra según la parte de la prueba.

La prueba está destinada a realizarse a 11 personas de diferentes edades y géneros, con el objetivo de cubrir rangos distintos de edades. La prueba se realizará con personas con discapacidad auditiva que interpreten LESSA y puedan escribir. La prueba espera tener una duración de 30 minutos por persona.

La prueba está enfocada en los siguientes ámbitos:

- Comprensión de la seña: si el usuario comprende la seña inmediatamente, luego de varias repeticiones o no la comprende. También está involucrada la posibilidad que sea comprendido, pero que necesite alguna mejora.

- Diferenciación de señas similares: en LESSA algo que puede distinguirse de otra seña puede ser solo la expresión facial o la rotación de una mano. Las señas que se le presentan al usuario, no deberían confundirse con otra.
- Respuesta del usuario: si el usuario se siente cómodo con el intérprete digital o no hay distracciones en el diseño.

Se evalúan estas áreas con el objetivo de examinar los aspectos de mayor interés para el usuario y para los desarrolladores de la aplicación.

Uno de los principales objetivos de esta prueba es cumplir con el alcance establecido que por lo menos el 85% de las señas sean comprendidas por las personas que hablan LESSA. También se busca que las personas no confundan señas con otras, y que las señas puedan ser comprendidas cuando se muestran individualmente y cuando se muestran en oraciones o párrafos.

La planeación de la ejecución de la prueba se espera realizar en dos grupos, días distintos a 5 y 6 personas en cada ocasión, por factores del tiempo y la disponibilidad de los usuarios con discapacidad auditiva. Se pretende realizar la explicación de la prueba a las 6 personas al mismo tiempo, pero esta característica puede variar dependiendo de lo que en el momento se crea para la mejor realización de la prueba.

La prueba se planifica de esta manera basada en los exámenes del curso básico de LESSA al que uno de los individuos del proyecto asistió para el aprendizaje de LESSA.

Partes de la Prueba. En los siguientes apartados se muestran las cuatro partes en las que consiste la prueba, para lo que se hará una explicación:

- **Parte 1:** Esta parte consiste en que por medio de MIA se le mostrarán al usuario 5 palabras distintas en forma de deletreo con LESSA, con la mano derecha de la animación, el usuario deberá escribir una letra por recuadro conforme a lo que interprete por cada seña de cada letra. Aunque el usuario conozca la palabra, pero observa o

considera que una letra no está bien, podrá dejar el espacio en blanco e informar al encargado de la prueba.

- **Parte 2:** Consiste en que por medio de MIA a cada usuario se le mostrarán 15 señas diferentes, cada una referenciando una palabra en LESSA, el usuario deberá escribir la palabra que interprete según el orden corresponda. De igual manera deberá notificar al encargado de la prueba cualquier duda.

- **Parte 3:** En este apartado se mostrará al usuario, por medio de MIA, 5 oraciones cortas con sentido gramatical. El usuario deberá interpretar la oración y escribir lo que perciba en los espacios asignados para cada oración. El encargado de la prueba deberá seguir el protocolo de la prueba, preguntar al usuario la opinión de la calidad de la seña animada.

- **Parte 4:** En diferencia a las partes anteriores, en este apartado se mostrará al usuario un párrafo lógico y consentido gramatical, para que el usuario interprete y capte el mensaje que transmita el párrafo previamente estipulado por los encargados de la prueba. Al contrario que en partes anteriores, el usuario no escribirá nada, sino, deberá expresar el encargado de la prueba el mensaje del párrafo y el nivel que el usuario considere de expresión y calidad de las señas.

Peculiaridades de la prueba. Las peculiaridades de estas pruebas son debido al caso en particular que se pretende emplear, explicando, para el desarrollo de esta, es necesario la presencia de un intérprete de LESSA que pueda explicar al usuario de la manera más clara cómo se realizará la prueba y las partes que la componen:

- Los usuarios a realizar la prueba son personas con discapacidad auditiva con capacidad de escribir en español.
- Serán 11 personas realizando la prueba.
- La prueba se realizará de forma individual a cada usuario.

- Habrá la ayuda de un intérprete de LESSA para facilitar la comunicación entre los usuarios y los encargados de prueba, consecuentemente beneficiará los resultados de la prueba.
- La documentación de la prueba consiste en un documento principal: el primer parte del encabezado será llena por el encargado de la puebla y el resto por el usuario, que llenará con sus datos y con las respuestas de la prueba
- Las partes 3 y 4 de la prueba, consisten en la composición de oraciones simples o compuestas para formar un texto que pueda ser entendido en español. Sin embargo, la composición gramatical en LESSA difiere del español, por esa razón en la sección de respuestas se mostrará cómo el intérprete MIA signará la oración escrita en español, ignorando determinantes como *la, el,* etc, cambiando el orden de las palabras en la oración, verbos sin conjugación o buscando sinónimos de algunas palabras.

Hoja de prueba. La Hoja de Prueba está incluida en ANEXOS 03: Hoja de prueba. La prueba deberá ser llevar el dato del nombre del encargado de la prueba y fecha en la que se realiza para tener el orden debido y mejor análisis de los datos. Se muestra también indicaciones dirigidas al encargado de la prueba.

Los datos del usuario deberán ser llenados por el mismo, para el que también se muestra una lista de instrucciones que le serán explicadas por el usuario. Esta prueba cuenta con 10 versiones distintas por lo cual se enlistan por partes, mostrando la respuesta esperada.

Parte 4. Como anteriormente mencionado la parte 4 de la prueba será la misma en las diez versiones de la prueba, es decir los usuarios verán el mismo párrafo signado para interpretar.

Aplicación de la prueba. La prueba, como mencionado, se llevó a cabo en dos fechas distintas: sábado 21 y lunes 23 de septiembre de 2019. La primera fecha se llevó a cabo 5 pruebas con estudiantes de bachillerato a distancia en Centro Escolar Tomás Medina y la segunda fecha en Centro Escolar de Sordos de Santa Ana, con 6 estudiantes de la institución.

Tabla 35
Listado de los estudiantes en orden de la realización de la prueba.

VERSIÓN	NOMBRE	EDAD	SEXO	PROFESIÓN
1	Magaly Abigail	18	Femenino	Estudiante
2	Jaenine Nineth	17	Femenino	Estudiante
3	Gerardo Ignacio Torres	20	Masculino	Estudiante
4	Jorge Alberto Castro	30	Masculino	Estudiante
5	Luis Enrique	20	Masculino	Estudiante
6	Susana Luna	22	Femenino	Estudiante
7	Evelyn Sermeño	14	Femenino	Estudiante
8	Fátima González	22	Femenino	Estudiante
9	Katherinne Alvarenga	14	Femenino	Estudiante
10	Gerson Martínez	16	Masculino	Estudiante
11	Moisés Hernández	16	Masculino	Estudiante

Cambios. En el momento de la implementación de la prueba fue informado al grupo de trabajo que, para la facilitación de los usuarios, estos solo escribirían en la Parte01 de la prueba, que consta de deletreo. La Parte02 y Parte03, fueron escritas por el encargado de la prueba conforme a la interpretación del usuario, comunicándose por medio del intérprete presente, que en ambos casos resultó ser docente de la institución a cargo del aula que fue asignada al grupo de trabajo.

Análisis de pruebas de usuario. Para la mejor comprensión de la evaluación de la prueba se deben tomar estas características a consideración:

- La Parte01 de la prueba consiste en el deletreo por medio de señas de 5 palabras, siendo 10 versiones de la prueba, son 50 palabras distintas, esto quiere decir que en cada prueba varía el número distinto de letras utilizado es por eso que para el mejor análisis se ha contado las diferentes señas distintas de letras que hay en cada versión.
- Así también en la Parte03 que consiste en oraciones, en cada versión son oraciones distintas con el número distinto de señas en cada una, también se contó en número de señas distintas a utilizar para la mejor evaluación de la prueba.
- En la Parte 02, todas las versiones de prueba cuentan con 15 señas, así como también en la Parte04, todas las versiones cuentan con 24 señas.
- Se tomaron en cuenta 4 respuestas del usuario al momento de la evaluación de las pruebas:

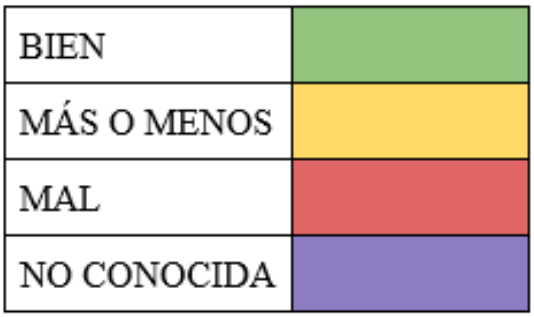

Figura 73. Código De Colores Para La Calificación De Los Resultados

- **Bien:** la seña se comprendió correctamente y no tiene correcciones.
- **Más o menos:** la seña se comprende con ciertas dificultades, luego de varias repeticiones y se hacen comentarios de mejora.
- **Mal:** La seña no se comprende o ha sido obsoleta.
- **No conocida:** el usuario no conocía la seña mostrada.

Tomando en cuenta los aspectos anteriores cada prueba contaba con un número de señas distinto.

Evaluación de las pruebas. Siendo así la metodología de análisis de la prueba la evaluación de fue de la siguiente manera:

- Considerando que cada prueba por consiguiente a las Parte01 y Parte03, tiene un número de señas distinto, para cada una se hizo una sumatoria total de las señas que contenía y el resultado de señas clasificadas como BIEN.

Ejemplo 01. Para el conteo de señas por versión se realizó el siguiente proceso:

Tabla 36
Ejemplo 01, versión 01 de prueba

1 BUEY
2 CABALLOS
3 QUERER
4 WALTER
5 ZANAHORIA

1. Se descompone la palabra BUEY en cuatro señas distintas B, U. E, Y. Total: 4.
2. Luego, la palabra CABALLOS, en siete señas: C, A, B, A, LL, O, S. Ahora, estas señas son comparadas con la palabra anterior para ver si hay repeticiones y se observa que la seña B se repite, por lo tanto, se ignora y se concluye que la segunda palabra cuenta con seis señas: C. A, A, LL, O, S. Ahora podemos ver que la seña A, se repite en dos ocasiones, por lo cual se ignora de la misma manera y la palabra queda con un total de cinco señas: C, A, LL, O, S.
3. Se repite el mismo proceso con las siguientes tres palabras, para concluir que las señas únicas utilizadas por palabra fueron:
    a. B U E Y
    b. C A LL O S
    c. Q R
    d. L T
    e. Z N H I

Haciendo el conteo en total son **17** señas distintas.

Ejemplo 02.

Tabla 37
Ejemplo 02, versión 01 de prueba

1 TU CUMPLEAÑOS CUANDO
2 ESCRITORIO CAFE NEGRO
3 BIENVENIDO ALCALDIA SANTA ANA
4 NOMBRE MI PAPA CARLOS
5 MI CUADERNO AZUL

1. La oración TU CUMPLEAÑOS CUÁNDO, indica 3 señas distintas.
2. La oración 2 contiene 3 señas distintas.
3. La oración 3 contiene 3 señas distintas: BIENVENIDO, ALCALDIA, SANTA ANA (dos palabras compuestas en una seña)
4. Oración 4 contiene 3 señas NOMBRE, MI, PAPA. La palabra Carlos siendo nombre propio se desglosa en: C, A, R, L, O, S. En este caso de haber un nombre propio que MIA muestre en forma de deletreo, se compara con la lista de señas individuales en la Parte01, en este caso como las señas de la palabra Carlos ya fueron utilizadas en la Parte01, ya no se toman en cuenta.
5. En la última oración MI CUADERNO AZUL, que consta de tres señas, la seña MI, como ya fue tomada en la oración anterior NOMBRE MI PAPA CARLOS, no se toma en cuenta, entonces esta oración consta de dos señas nuevas. Para obtener un total de 14 en esta parte.

- En la evaluación de señas por versión de prueba, se toman en cuenta las señas que aparecen una vez, las que aparecen más de una vez, se comprende como si fue entendida la cantidad de veces que se mostró, ya sea aparezca dos veces o 10, se toma como una vez, sacando así el total de señas por versión de prueba.

- Para el porcentaje total de señas que se esperaba fueran 85% comprendidas, no se toman en cuenta las señas que aparecen en dos o más versiones de pruebas, como por ejemplo las 27 letras del abecedario aparecen distribuidas en todas las versiones de prueba, siendo así cuando se cuenta el total de señas mostradas en la parte 1, se toma el total de 27 letras, no se cuentan las repeticiones, siento así también con señas que se repiten en varias oraciones como YO, MI y TÚ, en las oraciones de la Parte03.

En conclusión, para la evaluación completa de la prueba se toman las 17 señas de la parte01, más las 15 señas de la Parte02, más las 14 de la Parte03 y las 24 señas que abarca el párrafo de la parte04. Son un total de 70 señas en la versión 01 de la prueba.

1	DOCE	
2	BUSCAR	
3	NIT	
4	DICIEMBRE	
5	EL SALVADOR	
6	CARRO	
7	BAJAR	
8	ANARANJADO	
9	PAPEL BOND	
10	TE	
11	MAS O MENOS	
12	ADIOS	
13	ADENTRO	
14	UNIDAD DE SALUD	
15	SUPERMERCADO	

Figura 74. Muestra De Resultados De Las Pruebas

En la versión01 de la prueba todas las señas de las partes 01, 03 y 04 fueron comprendidas, solo hubo 4 señas de la parte 02 que no se comprendieron como muestran los resultados:

Para resultados más concretos y precisos de la prueba solo se toman como correctas las señas en verde calificadas como BIEN. Eso quiere decir que de esta parte hay 11/15 señas correctas, que trae un total para la versión01 de 66/70 señas correctas.

Tabla 38
Resumen De Resultados Obtenidos

SEÑAS APROBADAS	66
SEÑAS TOTALES	70
DIVISIÓN	0.9428571429
PORCENTAJE	94.29%
NOTA	9.33

- **Señas aprobadas:** número total de señas que fueron marcadas como BIEN o verde en el proceso de revisión de las pruebas.
- **Señas totales:** número de señas individuales utilizadas en cada versión de prueba.
- **División:** señas aprobadas / señas totales.
- **Porcentaje:** porcentaje total por versión de prueba, para obtener el resultado esperado, este debería ser mayor a 85%.
- **Nota:** esta cantidad es obtenida, brindándole a cada parte un porcentaje de valor de 25%, conforme a los literales de cada parte, se obtiene una nota o calificación por versión.

Resultados generales. Los resultados serán listados por versión de prueba en el orden que fueron presentados.

Tabla 39
Resultados Prueba Versión 1

SEÑAS APROBADAS	66
SEÑAS TOTALES	70
DIVISIÓN	0.9428571429
PORCENTAJE	94.29%
NOTA	9.33

Tabla 40
Resultados Prueba Versión 2

SEÑAS APROBADAS	67
SEÑAS TOTALES	71
DIVISIÓN	0.9436619718
PORCENTAJE	94.37%
NOTA	8.33

Tabla 41
Resultados Prueba Versión 3

SEÑAS APROBADAS	68
SEÑAS TOTALES	71
DIVISIÓN	0.9577464789
PORCENTAJE	95.77%
NOTA	9.16

Tabla 42
Resultados Prueba Versión 4

SEÑAS APROBADAS	65
SEÑAS TOTALES	72
DIVISIÓN	0.9027777778
PORCENTAJE	90.28%
NOTA	7.83

Tabla 43
Resultados Prueba Versión 5

SEÑAS APROBADAS	67
SEÑAS TOTALES	72
DIVISIÓN	0.9305555556
PORCENTAJE	93.06%
NOTA	8.16

Tabla 44
Resultados Prueba Versión 6

SEÑAS APROBADAS	66
SEÑAS TOTALES	70
DIVISIÓN	0.9428571429
PORCENTAJE	94.29%
NOTA	8.66

Tabla 45
Resultados Prueba Versión 7

SEÑAS APROBADAS	68
SEÑAS TOTALES	72
DIVISIÓN	0.9444444444
PORCENTAJE	94.44%
NOTA	9.33

Tabla 46
Resultados Prueba Versión 8

SEÑAS APROBADAS	65
SEÑAS TOTALES	72
DIVISIÓN	0.9027777778
PORCENTAJE	90.28%
NOTA	8.33

Tabla 47
Resultados Prueba Versión 9

SEÑAS APROBADAS	63
SEÑAS TOTALES	70
DIVISIÓN	0.9
PORCENTAJE	90.00%
NOTA	8.33

Tabla 48
Resultados Prueba Versión 10

SEÑAS APROBADAS	62
SEÑAS TOTALES	69
DIVISIÓN	0.8985507246
PORCENTAJE	89.86%
NOTA	8

Tabla 49
Resultados Prueba Versión 11

SEÑAS APROBADAS	78
SEÑAS TOTALES	81
DIVISIÓN	0.962962963
PORCENTAJE	96.30%
NOTA	9.16

Promedio de valores de PORCENTAJE de aprobación en las 11 versiones de la prueba:

Tabla 50
Porcentaje De Aprobación

PROMEDIO	0.9286090205
PORCENTAJE	92.86%

En total el número de señas probadas individuales fueron 253 señas, y con repetición fueron 370 en total. Uno de los alcances más importantes del proyecto es que el 85% de las señas realizadas y probadas fueran comprendidas por el usuario, personas con discapacidad auditiva. Siendo el resultado de porcentaje obtenido 92.86% se concluye que el resultado es mejor de lo esperado.

Señas clasificadas como MAL. Se listan a continuación:

1. **NIT**: Versión 01, Parte 02.
   Observación: no se comprende la segunda parte de la seña, cuando deletrea la palabra NIT.

3	NIT	

2. ANARANJADO: Versión 01, Parte 02.

   Observación; no se comprende qué es, comentario apunta a colocar la seña de color antes de decir un color.

8	ANARANJADO	

3. **GRANDIOSO**: Versión 02, Parte01.

   Observación: no se comprende la transición de la letra G a la R, la letra G tiene una mala ubicación, debe estar donde se deletrean las demás letras.

4	GRANDIOSO	

4. **PASAPORTE**: Versión 03, Parte 02.

   Observación: seña incorrecta, debe ser: abrir el pasaporte y sellarlo, no debe llevar la seña del color azul.

3	PASAPORTE	

5. **CHOFER**: Versión 04, Parte01.

   Observación: no se comprende la transición de la letra CH a la O, debido a que la CH tiene una posición incorrecta, igual que la letra G, debería estar en la misma posición que las otras letras.

5	CHOFER	

6. **HALLAR:** Versión 04, Parte 02.

   Observación: seña mal hecha.

9	HALLAR	

7. **¿QUÉ?:** Versión 05, Parte 02.

   Observación: falta expresión facial.

2	¿QUÉ?	

8. **PERDER:** Versión 06, Parte 02.

   Observación: las manos se mueven hacia adelante, se deberían mover hacia abajo para comprender la seña.

5	PERDER	

9. **CAMPO:** Versión 07, Parte 02.

   Observación: debe llevar la seña del color verde antes de hacer la propia seña de campo, mejorar para que no se vea tan robótica.

11	CAMPO	

10. **RESPIRAR:** Versión 07, Parte 02.

    Observación: falta expresión y más movimiento de manos, puede confundirse con la seña de Feliz o Joven.

15	RESPIRAR	

11. **SEGUIR:** Versión 08, Parte 02.

    Observación: mejorar movimiento para parecer más humano y velocidad debe ser más lenta.

6	SEGUIR	

12. **CAMBIAR:** Versión 08, Parte 02.

    Observación: hacer movimiento más notable de los dedos.

7	CAMBIAR	

13. **ORACIÓN 3**: Versión 08, Parte03.

    Observación: no se comprendió la transición de las señas Traer, Lápiz y no hubo comprendió del color Gris.

3	TU TRAER LÁPIZ GRIS PASADO MAÑANA	

14. **DICCIONARIO:** Versión 09, Parte 02.

    Observación: Seña incorrecta.

4	DICCIONARIO	

15. **CRECER:** Versión 09, Parte 02.

    Observación: la palabra crecer depende del contexto, seña se confundió con la palabra Flor.

6	CRECER	

16. **ORACIÓN 2:** Versión 09, Parte 03.

    Observación: gramática de la oración incorrecta.

2	LIBRERÍA CERRAR 8 NOCHE	

17. **CATORCE:** Versión 10, Parte 02.

    Observación: no se comprendió porque cuando se muestra el número 4 debe bajarse la otra mano para no indicar un 0, sino se comprende como 10 y 40.

6	CATORCE	

18. **AMBULANCIA:** Versión 11, Parte 02.

    Observación: movimiento de dedos muy lento, falta expresión facial.

2	AMBULANCIA	

Señas comprendidas con calidad MÁS O MENOS. Se listan a continuación:

1. **BAJAR:** Versión 01, Parte 02.

   Observación: debe bajar más y más inclinada.

7	BAJAR	

2. **TE:** Versión 01, Parte 02.

   Observación: la mano derecha no debe bajar tanto.

10	TE	

3. **KIOSKO:** Versión 02, Parte 01.

   Observación: no es correcta la seña de la letra K, puede confundirse con la V.

5	KIOSKO	

4. **QUINCE:** Versión 02, Parte 02.

   Observación: no se comprendió porque cuando se muestra el número 5 debe bajarse la otra mano para no indicar un 0, sino se comprende como 10 y 50.

5	QUINCE	

5. **ORACIÓN 5:** Versión 02, Parte 03.

   Observación: no se comprendió la seña Olvidar.

5	TIA OLVIDAR MI NOMBRE	

6. **SONREIR:** Versión 03. Parte 02.

   Observar: mejorar expresión facial.

2	SONREIR	

7. **ORACIÓN 3:** Versión 03, Parte 03.

3	YO ESTUDIAR 5 AÑOS UNIVERSIDAD	

8. **AGUJA:** Versión 04, Parte 02.

   Observación: no se comprende la letra G.

4	AGUJA	

9. **GUARDAR:** Versión 04, Parte 02.

   Observación: debe ser más clara la seña, puede confundirse con la seña de Banco.

2	GUARDAR	

10. **SUEGRA:** Versión 04, Parte 02.

    Observación: el movimiento de manos es muy abajo de la posición que debería ser.

8	SUEGRA	

11. **DORADO:** Versión 04, Parte 02.

    Observación: más expresión facial.

12	DORADO	

12. **ORACION 3:** Versión 04, Parte 03.

    Observación: mala gramática en la oración.

3	YO BICICLETA ESCUELA AYER	

13. **WEB:** Versión 05, Parte 01.

    Observación: confusión en la transición de letras.

1	WEB	

14. **¿CÓMO?:** Versión 05, Parte 02.

    Observación: el movimiento debe ser más fluido.

7	¿CÓMO?	

15. **ORACIÓN 3:** Versión 05, Parte 03.

    Observación: transición confusa de señas.

3	MI PRIMA TENER LICENCIA DE CONDUCIR	

16. **ORACIÓN 4:** Versión 05, Parte 03.

    Observación: transición confusa de señas.

4	YO ESCRIBIR NOCHE	

17. **BOTE:** Versión 06, Parte 02.

    Observación: primera parte debe llevar más duración.

8	BOTE	

18. **ORACIÓN 2:** Versión 06, Parte 03.

    Observación: seña de ayudar es incorrecta.

2	YO AYUDAR ESCRIBIR LIBRO	

19. **INSTAGRAM:** Versión 07, Parte 02.

    Observación: la mano está muy arriba en el rostro, debe ser más abajo.

10	INSTAGRAM	

20. **MÉXICO:** Versión 08, Parte 02.

    Observación: se utilizan 3 dedos, no 2.

2	MÉXICO	

21. **VOCABULARIO:** Versión 08, Parte 02.

    Observación: debe ser más preciso el movimiento.

10	VOCABULARIO	

22. **ORACIÓN 4:** Versión 08, Parte 03.

    Observación: seña Molestar, no está muy precisa.

4	ABUELO MOLESTAR ABUELA	

23. **NOVIEMBRE:** Versión09, Parte02.

    Observación: movimiento muy abajo del torso.

1	NOVIEMBRE	

24. **ALMOHADA:** Versión 10, Parte 01.

    Observación: extraña transición entre letras.

3	ALMOHADA	

25. **DIVORCIADO:** Versión 10, Parte02.

    Observación: segunda parte de la seña no es necesaria.

2	DIVORCIADO	

26. **PERDÓN:** Versión 10, Parte 02.

    Observación: mejorar expresión facial.

9	PERDÓN	

Todas las señas anteriormente presentadas como MAL o MÁS O MENOS, fueron corregidas y/o cambiadas según se indicó, también fueron probadas de nuevo para corroborar su compresión, las cuales fueron aprobadas al ser mostradas.

CAPÍTULO VI
CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

La Lengua de Señas Salvadoreña, es un medio de comunicación de personas con discapacidad auditiva, empleado como medio de comunicación principal para ellos, no un medio secundario. Sin embargo, en El Salvador no hay medios o sistemas que ayuden o beneficien a estas personas a comunicarse de una manera fácil con el resto de la sociedad.

No es posible obtener la aprobación total de todas las señas hechas en el diccionario digital, ya que no existe una estandarización de cómo deben realizarse las señas en LESSA. Esto debido a la existencia de múltiples organizaciones de personas con discapacidad auditiva y cada una de ellas tienen su perspectiva de cómo se debe usar alguna seña específica.

Es necesaria la ayuda de una persona con discapacidad auditiva que conozca LESSA, porque son ellas las que en verdad luchan con las barreras de comunicación día a día, y son las que pueden expresar lo que es necesario en un intérprete y conocen mejor las señas en LESSA. Esta información fue obtenida de las entrevistas realizadas a personas con discapacidad auditiva y en las entrevistas presentes en Anexos.

El procesamiento del Lenguaje Natural sirve como herramienta importante para el desarrollo de aplicaciones con impacto en la vida cotidiana, especialmente el caso del presente proyecto donde su principal objetivo es la mejora de la comunicación.
Con base en el listado de palabras y expresiones que componen LESSA, se hace necesario el uso de reconocimiento de entidades nombradas adaptadas al proyecto.

El entrenamiento de modelos de procesamiento de Lenguaje Natural supone un gran desafío para el desarrollo de cualquier tipo de aplicación que necesite de su uso. En el caso del presente proyecto, se puede concluir que, por el tipo de aplicación y las necesidades de la misma, el uso de modelos personalizados no supone una mejora sobre el uso de las herramientas preestablecidas por las librerías, con reconocimiento de entidades nombradas basado en reglas.

La utilización de motores de videojuegos, para el desarrollo de este tipo de proyectos representa una las mejores opciones, ya que permiten crear programas que mezclan el desarrollo de código y gráficos sin la necesidad de preocuparse por el manejo, sincronización, cálculos de gráficos y otros aspectos relacionados. Además, en la actualidad estos motores proveen una gran compatibilidad con distintas plataformas, permitiendo así que exista un mayor alcance en cuanto a la cantidad de usuarios finales, aspecto especialmente importante para proyectos que buscan beneficiar de alguna forma a la inclusión de personas.

RECOMENDACIONES

Se recomienda al lector, si desea formar parte del aprendizaje de LESSA, buscar un curso básico que sea impartido por una persona con discapacidad auditiva, ya que éstas son las personas más conocedoras del lenguaje, y contar con un intérprete para iniciar y tener mejor comprensión de todo.

Para adquirir el mejor conocimiento de LESSA es recomendable estudiar el curso completo, socializar con personas con discapacidad auditiva y además informarse de ASL (American Sign Lenguage) propio de Estados Unidos, que es muy usado en El Salvador también, sin embargo, no es LESSA.

Es recomendable que al trabajar en las animaciones se cuente con la ayuda de un intérprete de lenguaje de señas, quien puede brindar su aprobación u observaciones a corregir.

Para el modelado y animación se recomienda la obtención de licencias completas de los programas mencionados, ya que permiten obtener resultados con mejores acabados y aún más adaptables a otras necesidades que puedan surgir a futuro. Se recomienda además la utilización de equipos de captación de movimientos, ya que proveerán una vía más rápida para la animación y con resultados de mayor calidad. Además, el uso de estos equipos permitirá el trabajo en conjunto con personas que conozcan el lenguaje de LESSA, quienes podrían realizar las animaciones sin necesidad de conocimiento en programas de manipulación de gráficos 3D.

En caso de futuros cambios en el lenguaje de señas o en la adaptación del presente proyecto en distintas adaptaciones del mismo de otras regiones, en donde sea necesaria la generalización del reconocimiento de entidades nombradas basadas en el contexto de las oraciones y no en entidades nombradas preestablecidas, se recomienda el entrenamiento de un modelo personalizado de reconocimiento de entidades.

En cuanto al motor de videojuego se recomienda migrar este proyecto a versiones más actuales de la herramienta, las cuales poseen optimizaciones y mejoras en cuanto al uso de recursos y mejor compatibilidad con plataformas, especialmente con las plataformas web y a futuro con herramientas de streaming de videojuegos.

Para una implementación específica de este proyecto, se recomienda utilizar la base de datos o servicio preferible por el usuario, ya sea manteniendo el uso en conjunto con ElasticSearch o realizando una migración completa de los datos de animaciones.

BIBLIOGRAFÍA

Comunicación

- Autor: Adrián, Yirda. (11 de octubre del 2019). *Definición de Comunicación.* Recuperado de: https://conceptodefinicion.de/comunicacion

- Autor: Isbel Delgado. (16 de septiembre de 2019. *Comunicación.* Recuperado de: https://www.significados.com/comunicacion

Procesamiento del Lenguaje Natural

- Autor: Antonio Moreno. (octubre 17, 2017) *Procesamiento del lenguaje natural ¿qué es?* Recuperado de: http://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural

- Autor: varios. (Julio 2019) *Procesamiento de lenguajes naturales*. Recuperado de: https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales

Lenguaje de Señas

- Autor: varios. (noviembre 2019) *Lengua de señas.* Recuperado de: https://es.wikipedia.org/wiki/Lengua_de_se%C3%B1as

- Autor: varios. (2017) *Lenguaje de señas.* Recuperado de: https://www.ecured.cu/Lenguaje_de_se%C3%B1as

- Autor: Alejandro Oviedo (2016) *¿"Lengua de señas", "lenguaje de signos", "lenguaje gestual", ¿"lengua manual"? Argumentos para una denominación.* Recuperado de: https://cultura-sorda.org/lengua-de-senas-lenguaje-de-signos-lenguaje-gestual-lengua-manual

- Autor: Myriam Patricia Carguancundo Nuela (2017). "*DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA TRADUCTOR DE TEXTO Y VOZ A LENGUAJE DE SEÑAS A TRAVÉS DE UNA INTERFAZ GRÁFICA CON UN AVATAR MEDIANTE DISPOSITIVOS ANDROID".* Recuperado de: http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/13830/T-ESPEL-MEC-0119.pdf?sequence=1

- Autor: Asociación salvadoreña de sordos (1996). *Diccionario de señas básicas salvadoreñas.* Recuperado de: https://issuu.com/fundasordo/docs/diccionario_de_lessa_digital

Discapacidad

- Autor: Organización Mundial de la salud. (11 de octubre de 2018). *Ceguera y discapacidad visual.* Recuperado de: https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment

- Autor: Plena inclusión (2017) *Qué es discapacidad visual.* Recuperado de: https://www.plenainclusion.org/discapacidad-intelectual/que-es-discapacidad-intelectual

Diseño y Desarrollo

- Autor: Cook, A. M., Polgar, J. M, *Assistive Technologies: Principles & Practices*. (2015)

- Autor: Keimel Marina Gisele (junio 2016). *Integración de dispositivos en una plataforma de vida cotidiana asistida por computadora.*

- Autor: Cultural Heritage Experiences (2016). *The CHESS Project. Recuperado de:* http://www.chessexperience.eu/

- Autor: Monica Burns. (30 de octubre de 2018). *Shapes 3D Augmented Reality Geometry Drawing App*. Recuperado de: https://classtechtips.com/2018/10/30/augmented-reality-geometry/

- Autor: varios. (noviembre 2019). *Python.* Recuperado de: https://es.wikipedia.org/wiki/Python

- Autor: varios. (noviembre 2019). *C++. Recuperado de:* https://es.wikipedia.org/wiki/C%2B%2B

- Autor: varios. (noviembre 2019). *spaCy. Recuperado de:* https://en.m.wikipedia.org/wiki/SpaCy

- Autor: varios. (noviembre 2019) *Java (lenguaje de programación).* Recuperado de: https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)

- Autor: Blender org. Recuperado de: https://www.blender.org/download/requirements

- Autor: Maxon company. *Requisitos del sistema.* Recuperado de: https://www.maxon.net/es/productos/infosites/system-requirements/

- Autor: Autodesk.support (11 de abril de 2015). *Requisitos del sistema de Autodesk Maya 2016.* Recuperado de: https://knowledge.autodesk.com/es/support/maya/learn-explore/caas/sfdcarticles/sfdcarticles/ESP/System-requirements-for-Autodesk-Maya-2016.html

- Autor: Autodesk.support (2016). *Maya 2016 Extension 2.* Recuperado de: https://knowledge.autodesk.com/sites/default/files/Hardware_Certification_Results_Maya2016_EXT2.pdf

- Autor: Unity User Manual Team (Abril 2018). *Unity User Manual 2018.* Recuperado de: https://docs.unity3d.com/2018.4/Documentation/Manual/

Casos de Uso

- Autor: Universidad Nacional abierta y a distancia. *Diagramas de casos de uso.* Recuperado de: http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_casos_de_uso.html

Pruebas de Usabilidad

- Autor: Andrea Cantú. (septiembre 2017) *Qué son: Pruebas de Usabilidad.* Recuperado de: https://blog.acantu.com/que-son-pruebas-usabilidad

- Autor: Hassan Montero, Yusef y Martín Fernández, Francisco J. (9 de diciembre de 2003) *Método de Test con Usuarios.* Recuperado de: http://www.nosolousabilidad.com/articulos/test_usuarios.html

- Autor: Gamma UX. (10 de julio de 2018) *UX Test, ¿presencial o remoto?* Recuperado de: http://gammaux.com/blog/2018/07/10/test-ux-presencial-o-remoto/

- Autor: Eduardo (13 de marzo de 2014). *Test de usuarios de guerrilla: si ni testeas es porque no quieres.* Recuperado de: https://dispersium.es/test-de-usuarios-de-guerrilla

Named Entity Recognition

- Autor: John Foley, Sheikh Muhammad Sarwar y James Allan. (13 de junio de 2018). *Named Entity Recognition with Extremely Limited Data*. Recuperado de: https://arxiv.org/pdf/1806.04411.pdf

- Autor: Jason Brownlee (3 de diciembre de 2018). *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks.* Recuperado de: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks

- Autor: SpaCy. *Training spaCy's Statistical Models.* Recuperado de: https://spacy.io/usage/training

- Autor: Jason Brownlee (21 de julio de 2017). *A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size.* Recuperado de: https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size

- Autor: Manivannan Murugavel (29 de marzo de 2019). *How to Train NER with Custom training data using spaCy.* Recuperado de: https://medium.com/@manivannan_data/how-to-train-ner-with-custom-training-data-using-spacy-188e0e508c6

Inteligencia Artificial

- Autor: Iberdrola (2019). *¿Somos conscientes de los retos y principales aplicaciones de la Inteligencia Artificial?* Recuperado de: https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial

ANEXOS

Anexo 1: Entrevista a personas con discapacidad auditiva

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA
CRITERIO SOBRE LAS CONDICIONES SOCIALES A LAS QUE SE ENFRENTAN LAS PERSONAS CON DISCAPACIDAD AUDITIVA EN SANTA ANA, EL SALVADOR.
Nombre del entrevistado:

Edad:

---

1. ¿Cómo considera la situación de la discriminación hacia las personas con discapacidad auditiva en la ciudad de Santa Ana?
2. ¿Alguna vez ha estado en una situación de discriminación en contra de alguien con discapacidad auditiva? Explique su experiencia.
3. ¿Es conocedor de medidas que se han implementado en instituciones públicas que beneficien a personas con discapacidad auditiva?
4. ¿Cuáles cree que son los obstáculos más grandes que enfrentan las personas con discapacidad auditiva?
5. ¿Considera que las instituciones públicas hacen o han hecho un buen trabajo intentando incluir a personas con discapacidad en sus instituciones?
6. ¿Qué recomendaciones daría a las instituciones públicas en general para evitar casos de discriminación o mejorar la inclusión en dichas instituciones?

Anexo 2: Resumen de entrevista de Anexo 1

Este resumen hace referencia a la entrevista plateada al anexo 1 que fue realizada en dos fechas distintas: 25 de mayo de 2019 y 1 de junio de 2019, a dos mujeres con discapacidad auditiva que practican y enseñan LESSA con ayuda de una intérprete en ambas ocasiones.
Como se puede observar en el anexo 1, la entrevista consta de 6 preguntas, todas con énfasis en la discriminación hacia personas con discapacidad auditiva en Santa Ana, El Salvador. Así también como las circunstancias o situaciones que viven las personas con discapacidad auditiva en instituciones públicas con la barrera de la comunicación.

Ambas participantes de la entrevista declararon haber sufrido de algún tipo de discriminación en el curso de su vida y haber conocido casos de discriminación en Santa Ana, así como también casos de ignorancia de parte de instituciones públicas como la PNC que llevó a la muerte de un joven con discapacidad auditiva en el municipio, como también en instituciones educativas públicas donde levantan falso testimonio de ellas.

Comentan en sus respuestas que las personas que conocen con discapacidad auditiva, la mayoría trabaja en la empresa privada, ya que las instituciones públicas del estado no brindan ayuda a ellos. Señalan que el problema más grande es la barrera de la comunicación, que han tenido experiencias que van a hospitales o a la policía y no pueden comunicarse con nadie ahí porque la institución no cuenta con un intérprete, y siempre les piden llevar ellos un intérprete para poder comunicarse, cuando ellas, comentan, son personas independientes.

La barrera de la comunicación considera que es el problema más grande dentro de la sociedad, piensan ellas que no hay apoyo en las instituciones, siempre les dicen que esperen por algo que nunca llega, también sienten poco apoyo de sus mismas familias, incluso en algunos institutos escolares públicos no ofrecen cursos o talleres a los padres de los alumnos o simplemente los padres no tiene el interés de aprenderlo. Recomienda Cristina, que se necesita borrar la barrera de la comunicación, incluir a personas sordas en las instituciones públicas, y conocer que son personas capaces de realizar actividades y son independientes.

Anexo 3: Hoja de prueba

PRUEBA DE USABILIDAD

Encargado de la prueba:	

Fecha:	

Nota al encargado de la prueba:

- Explicar las instrucciones de la manera más sencilla posible al usuario, con la ayuda del intérprete.
- Mantener actitud de paciencia con el usuario.
- Realizar anotaciones u observaciones en caso sea necesario.
- Hacer la correcta grabación de la prueba, en pantalla y al usuario.
- Por cada ítem de cada parte deberá preguntar al usuario si la seña no se entiende, más o menos, o si sí se entiende, con lo que el usuario responderá con “pulgar hacia abajo” , “más o menos”, “pulgar hacia arriba”, correspondientemente.

Datos del usuario:

Nombre:	

Edad:	

Profesión:	

Parte 1.

Deletreo

Se le presentarán al usuario 5 palabras distintas en forma de deletreo desde MIA, se le pide que escriba exactamente las letras que vea para formar la palabra. Si conoce la palabra, pero el deletreo posee un error, escriba exactamente la letra que vio.

1.
2.
3.
4.
5.

Parte 2.

Palabras.

Se le presentarán al usuario 15 palabras en señas, una a la vez. Deberá escribir la palabra correspondiente a cada seña.

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.

13.
14.
15.

Parte 3.

Oraciones.

Ahora se le presentará el usuario una serie de oraciones en señas que deberá escribir conforme a las entienda.

1.
2.
3.
4.
5.

Parte 4.

Párrafo.

A continuación, se le presentará un párrafo en señas al usuario. El encargado de la prueba deberá preguntar al usuario sobre lo que comprendió del párrafo y tomar las anotaciones sobre las respuestas del usuario, además deberá preguntar sobre la calidad de las señas. El usuario no escribirá nada.

ANEXO 4: Respuestas Esperadas a las diferentes versiones de la Prueba 02.

Respuestas Parte 1.

Versión 1.

1. BUEY
2. CABALLOS
3. QUERER
4. WALTER
5. ZANAHORIA

Versión 2.

1. CANGREJO
2. CHOCOLATE
3. ELEFANTE
4. GRANDIOSO
5. KIOSKO

Versión 3.

1. ZAPATO
2. EXITO
3. UNIDAD
4. INTELIGENTE
5. BURRO

Versión 4.

1. TAXIS
2. CEBOLLA
3. ZORROS
4. AGUJA
5. CHOFER

Versión 5.

1. WEB
2. PROYECTAR
3. PROVECHO
4. AFLOJAR

5. TORNILLO

Versión 6.

1. ZAPATO
2. EXITO
3. UNIDAD
4. INTELIGENTE
5. BURRO

Versión 7.

1. WEB
2. PROYECTAR
3. PROVECHO
4. AFLOJAR
5. TORNILLO

Versión 8.

1. FOCA
2. AGACHAR
3. LLUVIA
4. TELA
5. KILOS

Versión 9.

1. LLAVE
2. ELEFANTE
3. EXAMEN
4. EXCLUYENTE
5. HORA

Versión 10.

1. QUERIDO
2. ESPAÑOL
3. ALMOHADA
4. VAYA
5. PIZZA

Versión 11.

1. PIÑA
2. KIWI
3. FECHA
4. GRANJA
5. EXCLUSIVO

Respuestas Parte 2.

Versión 1.

1. DOCE
2. BUSCAR
3. NIT
4. DICIEMBRE
5. EL SALVADOR
6. CARRO
7. BAJAR
8. ANARANJADO
9. PAPEL BOND
10. TE
11. MAS O MENOS
12. ADIOS
13. ADENTRO
14. UNIDAD DE SALUD
15. SUPERMERCADO

Versión 2.

1. LECHE
2. CUIDESE MUCHO
3. CRUZROJA
4. AMARILLO
5. QUINCE
6. ¿CUÁL?

7. VIERNES
8. ESTADOS UNIDOS
9. AGOSTO
10. ¿QUIÉN?
11. PADRASTRO
12. MADRASTRA
13. CAFE(COLOR)
14. NEGRO
15. CELESTE

Versión 3.

1. BUS
2. SONREIR
3. PASAPORTE
4. MIRAR
5. ENEMIGOS
6. APRENDER
7. JULIO
8. CENTROAMERICA
9. DIECISIETE
10. ROSADO
11. CABAÑAS
12. LLEGAR
13. CUSCATLÁN
14. LA UNION
15. BIENVENIDO

Versión 4.

1. ENERO
2. GUARDAR
3. TREINTA

4. ¿POR QUÉ?
5. MORADO
6. SEMANA
7. HOSPITAL
8. SUEGRA
9. HALLAR
10. PARTIDA DE NACIMIENTO
11. CINE
12. DORADO
13. SEMANA PASADA
14. MORAZAN
15. SAN VICENTE

Versión 5.

1. HOLA
2. ¿QUÉ?
3. MAYO
4. ¿CÓMO ESTÁS?
5. BUENAS TARDES
6. GRACIAS
7. ¿CÓMO?
8. ABRIL
9. JUNIO
10. SEPTIEMBRE
11. AHUACHAPÁN
12. SOLTERO
13. GUATEMALA
14. LESSA
15. CENTRO

Versión 6.

1. CAFETERIA
2. ANTEAYER
3. BORRAR
4. ABRIR
5. PERDER
6. CARRO
7. PRIMO
8. BOTE
9. PORTARSE BIEN
10. SUBIR
11. 19
12. 14
13. BICICLETA
14. BORRADOR
15. PERSONAS

Versión 7.

1. ¿CÓMO ESTÁS?
2. BUENAS TARDES
3. GRACIAS
4. ¿CÓMO?
5. ABRIL
6. JUNIO
7. SEPTIEMBRE
8. CUMPLEAÑOS
9. CORREO ELECTRONICO
10. INSTAGRAM
11. CAMPO
12. YOUTUBE

13. TRABAJAR
14. VIVIR
15. RESPIRAR

Versión 8.

1. CASAR
2. MÉXICO
3. FAMILIA
4. LA LIBERTAR
5. TAXI
6. SEGUIR
7. CAMBIAR
8. SORDO
9. MÁS
10. VOCABULARIO
11. 21
12. LENGUAJE
13. MOTO
14. DAR
15. GANAR

Versión 9.

1. NOVIEMBRE
2. SONSONATE
3. IGLESIA
4. DICCIONARIO
5. FACEBOOK
6. CRECER
7. COMUNICAR

8. TAXI
9. YERNO
10. LLENO
11. MAMÁ
12. VERDE
13. WHATSAPP
14. UNO
15. ARRIBA

Versión 10.

1. FIN DE SEMANA
2. DIVORCIADO
3. MARZO
4. COMER
5. CALCULADORA
6. CATORCE
7. BLANCO
8. DIRECTOR
9. PERDÓN
10. NIÑO
11. ABAJO
12. AFUERA
13. BANCO
14. LO SIENTO
15. DISCULPA

Versión 11.

1. RESTAURANTE
2. AMBULANCIA
3. COMPUTADORA
4. SABER
5. PERMISO
6. TELÉFONO
7. PODER
8. CELULAR
9. OBEDECER
10. MANEJAR
11. EXPLICAR
12. CURRICULUM VITAE
13. TARDE
14. MERCADO
15. ¿PARA QUÉ?

Respuestas Parte 03.

Versión 1.

1. TU CUMPLEAÑOS CUANDO
2. ESCRITORIO CAFE NEGRO
3. BIENVENIDO ALCALDIA SANTA ANA
4. NOMBRE MI PAPÁ CARLOS
5. MI CUADERNO AZUL

Versión 2.

1. HOY O MAÑANA
2. NIETO TRABAJAR SÁBADO
3. YO MAÑANA IR LA PAZ
4. SEMANA DOCE DIAS
5. TIA OLVIDAR MI NOMBRE

Versión 3.

1. CERRAR PUERTA
2. NOS VEMOS OTRO DOMINGO
3. YO ESTUDIAR 5 AÑOS UNIVERSIDAD
4. LUNES MARTES MIERCOLES
5. MI CUMPLEAÑOS FEBRERO

Versión 4.

1. TU HIJOS CUÁNTO
2. NOMBRE ESPOSO RAÚL
3. YO BICICLETA ESCUELA AYER
4. BUENAS NOCHES CON PERMISO
5. JUEVES 4 PM

Versión 5.

1. YO 4 HERMANOS 5 HERMANAS
2. YO AYER VOLAR ESTADOS UNIDOS
3. MI PRIMA TENER LICENCIA DE CONDUCIR
4. YO ESCRIBIR NOCHE
5. YO BEBER AGUA AYER

Versión 6.

1. YO ESCRIBIR MI CUADERNO
2. YO AYUDAR ESCRIBIR LIBRO
3. PROFESOR ESCRIBIR LIBRETA
4. NO COMER CINE
5. YO CAMINAR CENTRO AHUACHAPAN

Versión 7.

1. HOY MIÉRCOLES
2. MI MOCHILA CUATRO LIBROS
3. 1 AÑO CUÁNTOS MESES
4. YO IR AVION PUERTO RICO
5. TU ASUSTAR YO

Versión 8.

1. MI HERMANO ESTUDIA UNIVERSIDAD
2. YO NECESITAR VISA PARA VOLAR
3. TU TRAER LÁPIZ GRIS PASADO MAÑANA
4. ABUELO MOLESTAR ABUELA
5. CUÑADO AYUDAR LEER MI HIJO

Versión 9.

1. JOVER IR ESCUELA
2. LIBRERÍA CERRAR 8 NOCHE
3. YO LLAMAR TIA
4. NOVIO NECESITAR HABLAR
5. AMIGO CAMINAR COLEGIO

Versión 10.

1. BUENOS DÍAS MI NOMBRE JAVIER
2. YO ESCRIBIR MI AMIGO
3. BIENVENIDO ENTRAR PUERTA SENTARSE
4. MI FOLDER ROJO
5. YO IR ESTADOS UNIDOS

Versión 11.

1. YO BEBER CHOCOLATE
2. MI NOMBRE FACEBOOK ANDREA
3. MI CELULAR ES 78423112
4. MI SACAPUNTA GRIS
5. MI PAPA VIVIR FELIZ

Respuesta Parte 04.

Texto normal.

"*Hola buenos dias como estas. Mi nombre es mia mi seña es "seña" mucho gusto. Yo estoy feliz de estar aquí. Necesito aprender más vocabulario para comunicarme con otras personas sordas. Gracias buen dia."*

Texto LESSA.

"Hola buenosdias comoestas mi nombre mia lessa seña muchogusto yo feliz aqui yo necesitar aprender más vocabulario comunicar otro personas sordo gracias buenosdias"

Anexo 5.

Lista de entidades nombradas no reconocidas por el modelo pre-entrenado.

- Saludos.
    - Buenos días.
    - Buenas tardes.
    - Buenas noches.
    - Con permiso
    - Cómo estás.
    - Cuídese mucho.
    - Buen provecho.
    - Nos vemos
    - Lo siento.
    - Por favor.
    - Por qué.
    - Más o menos.
    - Te veo luego.
    - Te veo más tarde.
    - Hacer caso.

- Frases con contexto especial en El Salvador.
    - Café con leche.
    - Té con limón.
    - Jugo de naranja.

- Expresiones de tiempo.
    - Fin de semana.
    - Día entre semana.

- Pasado mañana.
- Esta semana.
- Próxima semana.
- Otra semana.
- Semana pasada.
- Esta tarde.
- Esta noche.
- Próximo día.

Anexo 6. Diccionario de palabras

El diccionario de palabras está clasificado por temas y en orden ortográfico. Las letras del abecedario y los números no serán incluidas en esta lista.

Saludos y Normas de Cortesía.

1. ¿Cómo estás?
2. Adiós
3. Bien
4. Bienvenido
5. Buen Provecho
6. Buenas noches.
7. Buenas tardes.
8. Buenos días
9. Con Permiso
10. Cuídese Mucho
11. Disculpe
12. Es Propio
13. Feliz día
14. Gracias
15. Hola
16. Lo siento
17. Mal
18. Más o menos
19. Mucho Gusto
20. Nos vemos
21. Perdón
22. Siéntese

Lugares.

1. Alcaldía
2. Banco
3. Cafetería
4. Campo
5. Centro de Ciudad
6. Cine
7. Colegio
8. Cruz Roja
9. Escuela

10. Hospital
11. Iglesia
12. Librería
13. Mercado
14. Restaurante
15. Supermercado
16. Tienda
17. Unidad de Salud
18. Universidad

Expresiones de tiempo.

1. Año
2. Anteayer
3. Aquí
4. Ayer
5. Día
6. Día Anterior
7. Día entre Semana
8. Día Siguiente
9. Fin de Semana
10. Hoy
11. Mañana
12. Mes
13. Meses
14. Noche
15. Pasado Mañana
16. Próxima Semana
17. Semana
18. Semana Pasada
19. Tarde

Días de la Semana.

1. Lunes
2. Martes
3. Miércoles
4. Jueves
5. Viernes
6. Sábado
7. Domingo

Meses del Año.

1. Enero
2. Febrero
3. Marzo
4. Abril
5. Mayo
6. Junio
7. Julio
8. Agosto
9. Septiembre
10. Octubre
11. Noviembre
12. Diciembre

Medios de Transporte.

1. Ambulancia
2. Avión
3. Bicicleta
4. Bote
5. Bus
6. Carro
7. Helicóptero

8. Microbús
9. Motocicleta
10. Taxi

Útiles Escolares.

1. Borrador
2. Calculadora
3. Corrector
4. Cuaderno
5. Diccionario
6. Director
7. Engrapadora
8. Folder
9. Lápiz
10. Libreta
11. Libro
12. Mochila
13. Mochila de Rodos
14. Morral
15. Papel Bond
16. Profesor
17. Sacabocado
18. Supervisor

Verbos.

1. Abrir
2. Alcanzar
3. Almacenar
4. Andar
5. Aprender
6. Apresurar
7. Apurar
8. Arreglar
9. Asustarse
10. Ayudar
11. Bajar
12. Beber
13. Borrar
14. Buscar
15. Caer
16. Cambiar
17. Caminar
18. Casar
19. Cerrar
20. Comenzar
21. Comer
22. Comprar
23. Conducir
24. Conocer
25. Crecer
26. Cumplir
27. Dejar
28. Encontrar
29. Enseñar
30. Escribir
31. Escuchar
32. Estudiar
33. Explicar
34. Ganar
35. Guardar
36. Hacer Caso
37. Hallar

38. Hostigar
39. Ir
40. Leer
41. Llamar
42. Llenar
43. Llevar
44. Manejar
45. Mirar
46. Molestar
47. Necesitar
48. No Poder
49. Obedecer
50. Obligar
51. Observar
52. Oír
53. Olvidar
54. Ordenar
55. Pedir
56. Perder
57. Permitir
58. Poder
59. Quebrar
60. Recibir
61. Respirar
62. Saber
63. Salir
64. Sangrar
65. Seguir
66. Sonreír
67. Subir
68. Suponer
69. Tener
70. Terminar
71. Tocar
72. Tomar
73. Traer
74. Vivir
75. Volar

Países.

1. Alaska
2. Argentina
3. Barbados
4. Bolivia
5. Brasil
6. Canadá
7. Chile
8. Colombia
9. Costa Rica
10. Cuba
11. Dominicana
12. Ecuador
13. El Salvador
14. Estados Unidos
15. Granada
16. Guatemala
17. Haití
18. Honduras
19. Jamaica
20. México
21. Nicaragua
22. Paraguay

23. Perú
24. Puerto Rico
25. República Dominicana
26. Santa Lucia
27. Trinidad y Tobago
28. Uruguay
29. Venezuela

Divisiones de América

1. Centro América
2. América del Norte
3. América del Sur
4. Caribe

Redes Sociales

1. Facebook
2. Instagram
3. Whatsapp
4. Youtube

Tecnología

1. Celular
2. Computadora
3. Correo Electrónico
4. Internet
5. Laptop
6. PC
7. Teléfono

Documentos

1. Carne
2. Curriculum
3. Curriculum Vitae
4. Diploma
5. Diploma de Bachiller
6. Diploma de Bachillerato
7. Diploma de Universidad
8. DUI
9. Licencia de Conducir
10. NIT
11. Partida de Nacimiento
12. Pasaporte
13. Título
14. Título de Bachiller
15. Título de Bachillerato
16. Título de Universidad
17. Título Universitario
18. VISA

Departamentos de El Salvador

1. Ahuachapán
2. Cabañas
3. Chalatenango
4. Cuscatlán
5. La Libertad
6. La Paz
7. La Unión
8. Morazán
9. San Miguel
10. Santa Ana
11. San Vicente
12. Sonsonate

13. Usulután

Expresiones de Tiempo

1. Año
2. Anteayer
3. Ayer
4. Día
5. Día Anterior
6. Día Entre Semana
7. Día Siguiente
8. Próximo Día
9. Fin de Semana
10. Aquí
11. Hoy
12. Mañana
13. Mes
14. Meses
15. Noche
16. Pasado Mañana
17. Otra Semana
18. Próxima Semana
19. Siguiente Semana
20. Semana
21. Semana Pasada
22. Tarde

Colores

1. Amarillo
2. Anaranjado
3. Azul
4. Blanco
5. Café
6. Celeste
7. Dorado
8. Gris
9. Morado
10. Negro
11. Oro
12. Plateado
13. Rojo
14. Rosado
15. Verde

Relaciones Personales

1. Divorciado
2. Enemigos
3. Joven
4. Muchacho
5. Muchacha
6. Mujer
7. Novia
8. Novio
9. Soltero

Números

1. Uno
2. Dos
3. Tres
4. Cuatro
5. Cinco
6. Seis

7. Siete
8. Ocho
9. Nueve
10. Diez
11. Once
12. Doce
13. Trece
14. Catorce
15. Quince
16. Dieciséis
17. Diecisiete
18. Dieciocho
19. Diecinueve

17. O
18. P
19. Q
20. R
21. RR
22. S
23. T
24. U
25. V
26. W
27. X
28. Y
29. Z

Letras

1. A
2. B
3. C
4. CH
5. D
6. E
7. F
8. G
9. H
10. I
11. J
12. K
13. L
14. LL
15. M
16. N

Familia

1. Abuela
2. Abuelo
3. Cuñada
4. Cuñado
5. Esposa
6. Esposo
7. Familia
8. Hermana
9. Hermano
10. Hija
11. Hijo
12. Madrastra
13. Madrina
14. Mamá
15. Nieta
16. Nieto

17. Nuera
18. Padrastro
19. Padrino
20. Papá
21. Prima
22. Primo
23. Sobrina
24. Sobrino
25. Suegra
26. Suegro
27. Tía
28. Tío

Bebidas

1. Agua
2. Café
3. Cerveza
4. Chocolate
5. Coca-Cola
6. Leche
7. Té
8. Té Con Leche
9. Té Con Limón
10. Vino

Vocabulario

1. Abajo
2. Adentro
3. Afuera
4. Arriba
5. Comunicar
6. Cumpleaños
7. Escritorio
8. Feliz
9. Gente
10. Lenguaje
11. Lessa
12. Mejorar
13. Mas
14. Mi
15. Nombre
16. O
17. Otro
18. Personas
19. Puerta
20. Seña
21. Sordo
22. Trabajar
23. Tu
24. Vocabulario
25. Yo
26.

Preguntas

1. ¿Cómo?
2. ¿Cuál?
3. ¿Cuándo?
4. ¿Cuántos?
5. ¿Dónde?
6. ¿Para Qué?
7. ¿Por Qué?
8. ¿Qué?
9. ¿Quién?

10. ¿Quiénes?

Continentes

1. África
2. América
3. Asia
4. Europa
5. Oceanía