

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



**DESARROLLO DE UN SISTEMA INFORMÁTICO WEB
PARA LA GESTIÓN DE EXPEDIENTES DE LA
ADMINISTRACIÓN ACADÉMICA EN LA FACULTAD
MULTIDISCIPLINARIA ORIENTAL DE LA
UNIVERSIDAD DE EL SALVADOR**

PRESENTADO POR:

**ROMMEL AMADEUS ALBERTO MEJÍA
DIEGO ARMANDO HERRERA FLORES
RAFAEL ELÍAS MARIONA TURCIOS**

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

ASESORA:

ING. MILAGRO ALICIA GONZÁLEZ DE REYES

CIUDAD UNIVERSITARIA ORIENTAL
OCTUBRE 2019

UNIVERSIDAD DE EL SALVADOR

AUTORIDADES

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

VICERRECTOR ACADÉMICO:

DR. MANUEL DE JESÚS JOYA ÁBREGO

VICERRECTOR ADMINISTRATIVO:

ING. NELSON BERBANÉ GRANADOS

SECRETARIO GENERAL:

LIC. CRISTOBAL HERNÁN RÍOS BENÍTEZ

FACULTAD MULTIDISCIPLINARIA ORIENTAL

AUTORIDADES

DECANO:

ING. JOAQUÍN ORLANDO MACHUCA

VICEDECANO:

LIC. CARLOS ALEXANDER DÍAZ

SECRETARIO GENERAL:

LIC. JORGE ALBERTO ORTEZ HERNÁNDEZ

DIRECTOR GENERAL DE PROCESOS DE GRADUACIÓN:

LIC. JORGE PASTOR FUENTES CABRERA

**DEPARTAMENTO DE INGENIERÍA Y
ARQUITECTURA**

AUTORIDADES

**JEFE DEL DEPARTAMENTO:
ING. JUAN ANTONIO GRANILLO COREAS**

**COORDINADORA GENERAL DE PROCESOS DE
GRADUACIÓN:
ING. MILAGRO DE MARÍA ROMERO DE GARCÍA**

**COORDINADORA DE LA CARRERA DE INGENIERÍA DE
SISTEMAS INFORMÁTICOS:
ING. MILAGRO ALICIA GONZÁLEZ DE REYES**

TRIBUNAL EVALUADOR

JURADO ASESOR:

ING. MILAGRO ALICIA GONZÁLEZ DE REYES

DOCENTE JURADO CALIFICADOR:

ING. LIGIA ASTRID HERNÁNDEZ BONILLA

DOCENTE JURADO CALIFICADOR:

ING. JULIO CÉSAR ARIAS GUEVARA

AGRADECIMIENTOS

*Primeramente agradecer a **Dios** por la oportunidad que me brinda de culminar mi carrera. Gracias a Él he podido llegar hasta este momento, uno de los más esperados por cualquier estudiante. No ha sido fácil el camino, siempre lleno de innumerables obstáculos y situaciones poco favorables. Pero gracias a Dios por darle energías y fuerzas a mis padres para sacarme adelante a lo largo de todos estos años.*

*Agradezco a mis padres **Mauro Antonio Herrera y Morena Elizabeth Flores** por su enorme sacrificio y voluntad de sacarme adelante, no ha sido sencillo, fue especialmente difícil para ustedes, pero acá estoy ya, justo donde querían verme. Gracias infinitas. Agradezco también a mi novia **Diana Ivonne Quintanilla** y a su familia por todo su apoyo y cariño brindados a lo largo de todo este tiempo.*

*A mi asesora de tesis **Alicia Milagro de Reyes** por todas las recomendaciones y apoyo brindado a lo largo de estos años. Desde primer año hasta este momento, gracias por todo. Al ingeniero **Julio César Árias** por todas las enseñanzas y amistad brindada a lo largo de los años, fue mi primer docente y a su vez el último en evaluarme, gracias por todo. Al la ingeniero **Ligia Astrid Hernández** por su asesoría, tiempo y consejos brindados en todo este proceso, además de su colaboración en proyectos tanto académicos como laborales.*

*Agradecer a mis jefes en Cablesat **Roberto Milla, Manuel Mejía y Jennifer Zelaya** por el apoyo brindado durante todo este proceso, gracias por la oportunidad y todo el respaldo brindado.*

Diego Armando Herrera Flores

AGRADECIMIENTOS

Nihil novum sub sole

Rafael Elías Mariona Turcios

AGRADECIMIENTOS

*Agradezco a mi padre **Juan Carlos Alberto Rivera** y a mi madre **Ana Silvia Mejía Rosales** por toda la ayuda que me brindaron de principio a fin, moral, emocional y económicamente. Agradezco a mis hermana y hermanos **Eloisa, Oliver, Johan, Ludwig** y **Enzo** por el apoyo emocional y moral que me han brindado incluso antes de mi periodo universitario.*

Agradezco a todos mis docentes, desde las matemáticas y físicas, pasando por los de Contaduría y Administración de Empresas, hasta Diseño de Sistemas y Proyectos Informáticos, por sus enseñanzas y consejos.

Agradezco al personal de la facultad por ayudarme en este largo proceso y espero conseguir llegar lejos en mi vocación de Ingeniero de Sistemas Informáticos, y no defraudar a la universidad ni a mis padres en el camino de la rectitud que me han enseñado.

Agradezco a todos mis compañeros que me brindaron apoyo incluso en momentos de mayor necesidad, su apoyo es parte de este logro y jamás olvidare la ayuda de un amigo.

Gracias a todos.

Rommel Amadeus Alberto Mejía

Índice general

1. Planteamiento del problema	1
1.1. Planteamiento del problema	2
1.2. Justificación	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Alcances	5
1.5. Limitaciones	6
2. Marco Referencial	7
2.1. Marco Histórico	8
2.1.1. Universidad de El Salvador, Facultad Multidisciplinaria Oriental	8
2.1.2. Internet	8
2.1.3. La World Wide Web	11
2.1.4. Lenguajes de programación	15
2.2. Marco Teórico	21
2.2.1. Sistema	21
2.2.2. Metodologías de desarrollo	22
2.2.3. Ingeniería de software	24
2.2.4. Base de datos	26
2.2.5. Programación	27
2.2.6. Pruebas de validación	31
2.2.7. Estudio de factibilidad	32
2.2.8. Metodología de Investigación del problema	33
2.2.9. Hardware	36
2.3. Marco Legal	48
2.3.1. Ley de Acceso a la Información Pública	48
2.3.2. Ley de Archivo de la Nación	51
2.3.3. Ley de Procedimientos Administrativos.	53
2.3.4. Reglamento de la Gestión Académico Administrativa de la Universidad de El Salvador	55
2.4. Marco Conceptual	59
2.4.1. Definiciones	59
2.4.2. Supuestos teóricos	60
3. Metodología de la investigación	61
3.1. Tipo de investigación	62
3.1.1. investigación tecnológica	62
3.2. Universo y muestreo	63

3.2.1.	Universo	63
3.2.2.	Muestra	63
3.3.	Técnicas e instrumentos para la recolección de información	63
3.3.1.	Entrevista	63
3.3.2.	Cuestionario	65
3.3.3.	Observación directa	65
3.4.	Procedimiento para la validación de instrumentos	65
3.5.	Procedimiento para la recolección de información	66
3.6.	Procedimiento para procesar la información	66
3.7.	Procedimiento para presentar e interpretar la información	66
3.8.	Análisis e interpretación de resultados	67
3.8.1.	Análisis de requerimientos de la entrevista con el Administrador Académico.	67
3.8.2.	Cuestionario	69
4.	Desarrollo de software	70
4.1.	Definición del sistema	71
4.2.	Resumen de problemas del sistema actual	71
4.3.	Supuestos	71
4.4.	Restricciones	72
4.5.	Requerimientos del nuevo sistema	72
4.5.1.	Requerimientos funcionales	72
4.5.2.	Requerimientos no funcionales	73
4.6.	Estudio de factibilidad	74
4.6.1.	Factibilidad técnica	74
4.6.2.	Factibilidad económica	76
4.6.3.	Factibilidad operativa	77
4.6.4.	Factibilidad jurídica	77
4.7.	Herramientas de desarrollo	79
4.7.1.	Metodologías de desarrollo.	79
4.7.2.	Diseño	79
4.7.3.	Desarrollo	79
4.7.4.	Frameworks y librerías	79
4.7.5.	Editor de texto y Entorno de desarrollo integrado (IDE)	80
4.7.6.	Base de datos)	80
4.7.7.	Servidor)	80
4.7.8.	Software complementario)	80
4.7.9.	Cantidad de archivos promedio a almacenar	80
4.8.	Estandares de desarrollo	81
4.8.1.	Estandares para el desarrollo del sistema informático	81
4.9.	Metodología de programación	82
4.9.1.	Metodología de programación orientada a objetos	82
4.10.	Diseño del sistema informático	83
4.10.1.	Diagramas de casos de uso	83
4.10.2.	Diseño de la base de datos	92

5. Plan de implementación	93
5.1. Objetivos	93
5.1.1. Objetivo general	93
5.1.2. Objetivos específicos	94
5.2. Planeación	94
5.2.1. Actividad 1: Dividir el proceso en tareas y actividades	94
5.2.2. Actividad 2: Repartir actividades entre cada miembro del equipo de implementación	95
5.2.3. Actividad 3: Creación de un diagrama de Gant	95
5.3. Acondicionamiento	95
5.3.1. Actividad 1: Actualización de software del servidor	95
5.3.2. Actividad 2: Instalación de software y paquetes necesarios	95
5.3.3. Actividad 3: Preparación de aspectos de conexión y redes	95
5.4. Implementación	96
5.4.1. Actividad 1: Copiar archivos y paquetes del Nuevo Sistema al directorio /html.	96
5.4.2. Actividad 2: Escribir cambios.	96
5.5. Pruebas	96
5.5.1. Actividad 1: Verificar que el Sistema sea accesible para los usuarios y realizar pruebas.	96
6. Conclusiones y recomendaciones	98
6.1. Conclusiones	99
6.2. Recomendaciones	100
Bibliografía	101
Sitiografía	102
Anexos	103
Anexo 1. Entrevista dirigida al Administrador Académico	104
Anexo 2. Cuestionario dirigido al personal de Administración Académica	105
Anexo 3. Manual de usuario	106
Anexo 4. Manual de instalación	123
Anexo 5. Manual de programación	137
Anexo 6. Diccionario de datos	158

Índice de figuras

2.1. Características del ordenador	40
2.2. Especificaciones del ordenador	40
4.1. Diagrama general de casos de uso	83
4.2. Caso de uso de expedientes	83
4.3. Caso de uso de estudiantes	84
4.4. Caso de uso de configuración	84
4.5. Diagrama entidad relación	92
5.1. Diagrama de desglose analítico	94
5.2. Diagrama de Gantt para el plan de implementación.	97
6.1. Tabla carreras	158
6.2. Tabla departamentos	158
6.3. Tabla documentos f2	159
6.4. Tabla documentos de graduación	160
6.5. Tabla estudiantes	160
6.6. Tabla genero	161
6.7. Tabla observaciones	161
6.8. Tabla otros documentos	161
6.9. Tabla privilegios	161
6.10. Tabla registro de acciones	162
6.11. Tabla tipo de usuario	162
6.12. Tabla usuarios	162

Índice de cuadros

4.1. Características del ordenador cliente.	75
4.2. Características del servidor.	75
4.3. Costo de desarrollo del proyecto.	76
4.4. Caso de uso: Iniciar sesión.	85
4.5. Caso de uso: Crear nuevo expediente.	86
4.6. Caso de uso: Añadir archivos PDF.	86
4.7. Caso de uso: Modificar archivos del expediente.	87
4.8. Caso de uso: Buscar expediente.	87
4.9. Caso de uso: Eliminar un expediente.	88
4.10. Caso de uso: Añadir nuevo estudiante.	88
4.11. Caso de uso: Modificar datos de estudiantes.	89
4.12. Caso de uso: Buscar estudiante.	89
4.13. Caso de uso: Eliminar un estudiante.	90
4.14. Caso de uso: Añadir nuevos usuarios.	90
4.15. Caso de uso: Respaldo.	91

Introducción

Actualmente estamos enmarcados en la era de la información y el tratamiento informático de los datos. Existe una especial importancia y preocupación en cualquier institución ya sea de carácter público o privado acerca de la digitalización y el respaldo de su información y la de sus usuarios.

El mundo está yendo en general hacia la era de la digitalización, en la cual la información y los datos de las instituciones, su personal y sus usuarios en general están almacenados en bases de datos en las cuales la información está disponible desde cualquier lugar y en cualquier momento cada vez que sea requerida, gracias a los avances tecnológicos y al internet. Esto aumenta de manera exponencial la eficiencia y eficacia de las instituciones, ya que, no se requiere más tener la información almacenada y archivada de manera física en papel, y con esto a su vez, susceptible a daños y pérdidas de información por cualquier desastre natural o error humano.

Actualmente, la Administración Académica de la Universidad de El Salvador, Facultad Multidisciplinaria Oriental, cuenta con un sistema informático que almacena expedientes académicos de los estudiantes. Desafortunadamente, el sistema cuenta con muchas carencias y fallos a nivel de seguridad, sin mencionar que solo está implementado de manera local por una sola computadora, y esto por supuesto, genera un grave riesgo de pérdida de información delicada de los estudiantes de dicha facultad.

La presente tesis presenta y detalla una solución a todos estos problemas y carencias del actual sistema con el que cuenta la Administración Académica, y así mantener segura su información.

Capítulo 1

Planteamiento del problema

1.1. Planteamiento del problema

La Administración Académica de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador, cuenta con un sistema informático para la gestión de los expedientes digitalizados de los estudiantes, y su posterior almacenamiento. Sin embargo, dicho software cuenta con múltiples fallos y carencias que hacen ineficiente el proceso para el cuál fue desarrollado.

Posee diversos fallos de seguridad de considerable gravedad. Uno de los principales es que, a pesar que el sistema sí cuenta con un inicio de sesión mediante usuario y contraseña, carece de un adecuado manejo de sesiones, de tal manera que cualquier persona que no esté autorizada para ingresar, puede hacerlo mediante la barra de navegación, simplemente con conocer el nombre de la url de cualquiera de los módulos o secciones del sistema, y por medio del ingreso a alguna sección, tener acceso al sistema completo sin necesidad de tener que ingresar sus credenciales en el login, de tal manera que el login que posee dicho sistema es completamente un mero adorno. Otro de los graves problemas que presenta el actual sistema, es que este se encuentra alojado en un solo ordenador, de manera que al digitalizar los expedientes de los estudiantes, estos quedan almacenados en un solo disco duro de manera local, estando expuestos a cualquier accidente o desastre natural que pudiese existir, sin mencionar los daños que pudiera sufrir el disco duro en el que están almacenados; por otra parte, el sistema no genera ningún tipo de respaldo, por lo cual ante cualquier fallo del disco duro, daño físico o siniestro que pudiera ocurrir en la Administración Académica, perderíamos todos los expedientes digitalizados de miles de estudiantes.

Además de todos los problemas críticos mencionados anteriormente, tenemos también los inconvenientes de usabilidad del sistema, ya que, el personal administrativo de la Administración Académica se queja de lo poco intuitivo y amigable de cara al usuario que es dicho sistema. Los usuarios, que son a los que principalmente debe satisfacer el sistema, se les hace complicado navegar dentro de las opciones y menús que tiene el sistema actual, ya que no son del todo intuitivos, sin mencionar que algunos no funcionan como deberían. Por mencionar unas cuantas de las funciones que no funcionan bien, está la de modificar expedientes existentes en el sistema, cuando se requiere adjuntar un nuevo documento a un expediente ya almacenado en el sistema, este no hace nada, ya que carece de dicha funcionalidad, de manera que una vez registrado un expediente, este no puede modificarse.

Tampoco existe la posibilidad de ver un documento en concreto de manera individual. Los expedientes están conformados por muchos documentos diferentes, desde partidas de nacimiento, DUI, NIT o cualquier documento que se considere relevante, y en el sistema actual, no se pueden ver de manera individual, se muestra solamente el documento completo y hay que buscar manualmente en todo el PDF el documento deseado, lo cuál hace tedioso el proceso de búsqueda. También carece de la posibilidad de añadir información o documentos adicionales que se requieran en un momento determinado.

Todo lo anterior hace del actual sistema de información, un sistema totalmente inseguro, poco eficiente, sin escalabilidad y sin una experiencia de usuario satisfactoria.

1.2. Justificación

La administración académica realiza una serie de procedimientos administrativos, los cuales tienen que ver con la planificación de actividades académicas, inscripciones, expedientes académicos y trámites de alumnos, control de notas, reportes, estadísticos sobre la actividad académica, certificados, diplomas y control de becas. Especialmente se encargan de registrar a todos los estudiantes de nuevo ingreso mediante expedientes estudiantiles, estos conllevan una serie de documentos personales tales como el F1, partida de nacimiento, exámenes médicos, comprobante de pagos, entre otros. Actualmente la administración se encuentra en el proceso de convertir más de 40,000 expedientes a formato digital; sin embargo el sistema informático que poseen no es idóneo para realizar esa tarea de gestión de dichos expedientes digitalizados.

Se requiere el desarrollo de un nuevo sistema informático para la gestión y respaldo de los expedientes digitales de los estudiantes, ya que se ha hecho un análisis previo del sistema actual, y se ha llegado a la conclusión que es necesario realizar el desarrollo de un nuevo sistema ya que es muy poco factible y productivo aplicar mejoras o ajustes al sistema actual, ya que está mal diseñado, no se ha codificado bajo ningún estándar ni patrón de diseño definido. Tampoco está codificado de manera escalable, por lo cual, actualizar o mejorar el actual sistema se volvería una tarea mucho más difícil y generaría un mayor retraso que desarrollar uno desde el principio.

Debido a todos los motivos expuestos anteriormente se propone desarrollar un sistema informático web para la gestión de expedientes digitales de la Administración Académica de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

1.3. Objetivos

1.3.1. Objetivo general

- Desarrollar un sistema informático web para la gestión de expedientes digitales de la Administración Académica en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

1.3.2. Objetivos específicos

- Gestionar los expedientes digitalizados de los estudiantes de la Facultad Multidisciplinaria Oriental.
- Respaldar los expedientes digitalizados de manera remota.
- Agilizar los procesos académicos de los estudiantes de la Facultad Multidisciplinaria Oriental.
- Facilitar la búsqueda y edición de los expedientes de los estudiantes.

1.4. Alcances

- El sistema informático será manipulado solamente por el personal de la Administración Académica de la Facultad Multidisciplinaria Oriental.
- La gestión a realizar por el sistema comprenderá la completa administración de los expedientes de los estudiantes de la Universidad. Dicha administración contempla la creación de nuevos expedientes, edición de expedientes existentes, y eliminación de expedientes.
- Se contará con un mecanismo de búsqueda de expedientes para facilitar las tareas de búsqueda por parte del personal administrativo de la Administración Académica.
- El sistema informático contará con un mecanismo de registro de actividades para saber qué usuario realizó la última modificación de expedientes.
- No se dará mantenimiento al sistema por parte de los desarrolladores una vez implementado. En caso de ser necesario el mantenimiento, será responsabilidad de la Administración Académica.

1.5. Limitaciones

- El sistema no tendrá un mecanismo desde el cuál se pueda escanear expedientes para digitalizarlos en formato PDF, debe utilizarse software externo para dicho proceso.
- El sistema solo funcionará de manera óptima en navegadores web modernos. Internet explorer en sus versiones de la 9.0 hacia abajo no están soportadas de manera completa.

Capítulo 2

Marco Referencial

2.1. Marco Histórico

2.1.1. Universidad de El Salvador, Facultad Multidisciplinaria Oriental

2.1.1.1. Historia

El 17 de Junio de 1966, en Sesión No. 304. El Consejo Superior Universitario fundó el Centro Universitario de Oriente (CUO), en la Ciudad de San Miguel, como una extensión de los estudios universitarios de la Universidad Nacional de El Salvador hacia la Zona Oriental. Los Objetivos principales que motivaron su creación fue: Ampliar la capacidad de servicio docente de la Unidad.

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

Estos Objetivos aún siguen vigentes. En Abril de 1967, se adquirió un terreno de 108 manzanas de extensión en el Cantón el Jute a 6.5 kms. Al Sur Oriente de la Ciudad de San Miguel, donde se pretendía construir el Campus Universitario del CUO. Las actividades académicas se iniciaron el 17 de mayo de 1969, por lo que este año se cumplen 50 años de funcionamiento académico y el 17 de junio, se cumplieron 53 años de su creación. El primer Director fue el Dr. José Vinnatea; y las actividades académicas se iniciaron a través de tres Departamentos que impartían el servicio de áreas comunes a todas las carreras de la Universidad.

2.1.2. Internet

Internet ha supuesto una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Los inventos del telégrafo, teléfono, radio y ordenador sentaron las bases para esta integración de capacidades nunca antes vista. Internet es a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus ordenadores independientemente de su localización geográfica.

2.1.2.1. Historia

La historia de internet se remonta al temprano desarrollo de las redes de comunicación. La idea de una red de ordenadores creada para permitir la comunicación general entre usuarios de varias computadoras sea tanto desarrollos tecnológicos como la fusión de la infraestructura de la red ya existente y los sistemas de telecomunicaciones.

La primera descripción documentada acerca de las interacciones sociales que podrían ser propiciadas a través del networking (trabajo en red) está contenida en una serie de memorándums escritos por J.C.R. Licklider, del Massachusetts Institute of Technology, en Agosto de 1962, en los cuales Licklider discute sobre su concepto de Galactic Network (Red Galáctica).

El concibió una red interconectada globalmente a través de la que cada uno pudiera acceder desde cualquier lugar a datos y programas. En esencia, el concepto era muy parecido a la Internet actual. Licklider fue el principal responsable del programa de investigación en ordenadores de la DARPA desde Octubre de 1962. Mientras trabajó en DARPA convenció a sus sucesores Ivan Sutherland, Bob Taylor, y el investigador del MIT Lawrence G. Roberts de la importancia del concepto de trabajo en red.

En Julio de 1961 Leonard Kleinrock publicó desde el MIT el primer documento sobre la teoría de conmutación de paquetes. Kleinrock convenció a Roberts de la factibilidad teórica de las comunicaciones vía paquetes en lugar de circuitos, lo cual resultó ser un gran avance en el camino hacia el trabajo informático en red. El otro paso fundamental fue hacer dialogar a los ordenadores entre sí. Para explorar este terreno, en 1965, Roberts conectó un ordenador TX-2 en Massachusetts con un Q-32 en California a través de una línea telefónica conmutada de baja velocidad, creando así la primera (aunque reducida) red de ordenadores de área amplia jamás construida. El resultado del experimento fue la constatación de que los ordenadores de tiempo¹ compartido podían trabajar juntos correctamente, ejecutando programas y recuperando datos a discreción en la máquina remota, pero que el sistema telefónico de conmutación de circuitos era totalmente inadecuado para esta labor. La convicción de Kleinrock acerca de la necesidad de la conmutación de paquetes quedó pues confirmada.

A finales de 1966 Roberts se trasladó a la DARPA a desarrollar el concepto de red de ordenadores y rápidamente confeccionó su plan para ARPANET, publicándolo en 1967. En la conferencia en la que presentó el documento se exponía también un trabajo sobre el concepto de red de paquetes a cargo de Donald Davies y Roger Scantlebury del NPL. Scantlebury le habló a Roberts sobre su trabajo en el NPL así como sobre el de Paul Baran y otros en RAND. El grupo RAND había escrito un documento sobre redes de conmutación de paquetes para comunicación vocal segura en el ámbito militar, en 1964.

Ocurrió que los trabajos del MIT (1961-67), RAND (1962-65) y NPL (1964-67) habían discurrido en paralelo sin que los investigadores hubieran conocido el trabajo de los demás. La palabra packet (paquete) fue adoptada a partir del trabajo del NPL y la velocidad de la línea propuesta para ser usada en el diseño de ARPANET fue aumentada desde 2,4 Kbps hasta 50 Kbps. En Agosto de 1968, después de que Roberts y la comunidad de la DARPA hubieran refinado la estructura global y las especificaciones de ARPANET, DARPA lanzó un RFQ para el desarrollo de uno de sus componentes clave: los conmutadores de paquetes llamados interface message

¹ *Stephanie Falla Aroche (2014, 14 de febrero) La historia de internet. Maestros del web. Disponible en: <http://www.maestrosdelweb.com/internethis/> Consultado: 7 de marzo de 2019 07:46 pm.*

processors (IMPs, procesadores de mensajes de interfaz).

El RFQ fue ganado en Diciembre de 1968 por un grupo encabezado por Frank Heart, de Bolt Beranek y Newman (BBN). Así como el equipo de BBN trabajó en IMPs con Bob Kahn tomando un papel principal en el diseño de la arquitectura de la ARPANET global, la topología de red y el aspecto económico fueron diseñados y optimizados por Roberts trabajando con Howard Frank y su equipo en la Network Analysis Corporation, y el sistema de medida de la red fue preparado por el equipo de Kleinrock de la Universidad de California, en Los Angeles.

A causa del temprano desarrollo de la teoría de conmutación de paquetes de Kleinrock y su énfasis en el análisis, diseño y medición, su Network Measurement Center (Centro de Medidas de Red) en la UCLA fue seleccionado para ser el primer nodo de ARPANET. Todo ello ocurrió en Septiembre de 1969, cuando BBN instaló el primer IMP en la UCLA y quedó conectado el primer ordenador host. Así, a finales de 1969, cuatro ordenadores host fueron conectados conjuntamente a la ARPANET inicial y se hizo realidad una embrionaria Internet. Incluso en esta primitiva etapa, hay que reseñar que la investigación incorporó tanto el trabajo mediante la red ya existente como la mejora de la utilización de dicha red. Esta tradición continúa hasta el día de hoy.

Se siguieron conectando ordenadores rápidamente a la ARPANET durante los años siguientes y el trabajo continuó para completar un protocolo host a host funcionalmente completo, así como software adicional de red. En Diciembre de 1970, el Network Working Group (NWG) liderado por S.Crocker acabó el protocolo host a host inicial para ARPANET, llamado Network Control Protocol (NCP, protocolo de control de red). Cuando en los nodos de ARPANET se completó la implementación del NCP durante el periodo 1971-72, los usuarios de la red pudieron finalmente comenzar a desarrollar aplicaciones.²

2.1.2.2. Uso de TCP/IP

La primera conexión ARPANET fuera de los Estados Unidos se hizo con NOR-SAR en Noruega en 1973, justo antes de las conexiones con Gran Bretaña. Todas estas conexiones se convirtieron en TCP/IP en 1982, al mismo tiempo que el resto de las ARPANET.

DARPA formalizó tres contratos con Stanford (Cerf), BBN (Ray Tomlinson) y UCLA (Peter Kirstein) para implementar TCP/IP (en el documento original de Cerf y Kahn se llamaba simplemente TCP pero contenía ambos componentes). El equipo de Stanford, dirigido por Cerf, produjo las especificaciones detalladas y al cabo de un año hubo tres implementaciones independientes de TCP que podían interoperar. Este fue el principio de un largo periodo de experimentación y desarrollo para evolucionar y madurar el concepto y tecnología de Internet. Partiendo de las tres primeras redes ARPANET, radio y satélite y de sus comunidades de investigación iniciales, el entorno experimental creció hasta incorporar esencialmente cualquier forma de red y

² *Stephanie Falla Aroche (2014, 14 de febrero) La historia de internet. Maestros del web. Disponible en: <http://www.maestrosdelweb.com/internethis/> Consultado: 24 de febrero de 2019 07:46 pm.*

una amplia comunidad de investigación y desarrollo. Cada expansión afrontó nuevos desafíos.

Las primeras implementaciones de TCP se hicieron para grandes sistemas en tiempo compartido. Cuando aparecieron los ordenadores de sobremesa (desktop), TCP era demasiado grande y complejo como para funcionar en ordenadores personales. David Clark y su equipo de investigación del MIT empezaron a buscar la implementación de TCP más sencilla y compacta posible.

A medida fue avanzando el tiempo y desarrollándose las nuevas tecnologías, se llegó hasta lo que conocemos en nuestros días, un mundo interconectado y globalizado a través de internet.³

2.1.3. La World Wide Web

La World Wide Web (WWW o simplemente la Web) es un medio global de información cuyos usuarios pueden leer y escribir a través de computadoras conectadas a Internet. El término es a menudo usado erróneamente como un sinónimo para la Internet misma, pero la Web es un servicio que opera sobre la Internet, como también lo hace el correo electrónico. La historia de Internet data de un tiempo significativamente anterior a la historia de la World Wide Web.

2.1.3.1. Historia

En 1980, Tim Berners-Lee, un contratista independiente en la Organización Europea para la Investigación Nuclear (CERN por sus siglas en inglés), Suiza, desarrolló ENQUIRE, como una base de datos personal de gente y modelos de software, pero también como una forma de interactuar con el hipertexto; cada nueva página de información en ENQUIRE debía estar enlazada a una página existente.

En 1984 Berners-Lee regresa al CERN, y consideró los problemas de la presentación de información: los físicos alrededor del mundo necesitaban compartir datos y sus computadoras y software de presentación eran diferentes. En marzo de 1989 él escribió una propuesta para “una gran base de datos de hipertexto con enlaces tipados”, pero esto generó poco interés. Su jefe Mike Sendall, alentó a Berners-Lee a comenzar a implementar su sistema sobre la recientemente adquirida estación de trabajo NeXT.

En la navidad de 1990, Berners-Lee había desarrollado todas las herramientas necesarias para trabajar la Web: el Protocolo de transferencia de hipertexto (HTTP por sus siglas en inglés), el Lenguaje de Marcado de Hipertexto (HTML por sus siglas en inglés), el primer navegador web (llamado WorldWideWeb, que fue también un editor de páginas web), el primer servidor de aplicaciones HTTP (luego conocido como CERN httpd), el primer servidor web (<http://info.cern.ch>) y las primeras páginas web que describían el proyecto mismo.

³ *es.wikipedia.org (2019, 19 de febrero) Historia de internet. Wikipedia. Disponible en: <https://es.wikipedia.org/wiki/HistoriadeInternet> Consultado: 25 de febrero de 2019 09:01 pm.*

El buscador podía acceder los grupos de noticias de Usenet así como archivos en un FTP. Sin embargo no podía correr sobre la NeXT. Nicola Pellow luego creó un simple buscador de texto que podía correr sobre al menos alguna computadora y fue llamado Line Mode Browser.

De acuerdo con Tim Berners-Lee, la Web fue principalmente inventada en el Edificio 31 del CERN, pero también en su hogar, en las dos casas que vivió durante ese tiempo (una en Francia y la otra en Suiza).⁴

2.1.3.2. Crecimiento de la WWW

Período de 1992-1995: De acuerdo con su nacimiento en el CERN, los primeros en adoptar la World Wide Web fueron principalmente departamentos científicos de universidades o laboratorios de física como Fermilab y SLAC.

Los primeros sitios web mezclaban vínculos para el protocolo web HTTP y el entonces popular protocolo Gopher, el cual proveía acceso al contenido a través de menús de hipertexto presentados como un sistema de archivos en vez de como archivos HTML. Los primeros usuarios web podían navegar por populares páginas de directorios, como el primer sitio web de Berners-Lee en <http://info.cern.ch/>, o mediante la consulta a listas actualizadas como la página "What's New" de NCSA.

2.1.3.3. Navegadores web

El primer navegador fue desarrollado por Tim Berners-Lee, en la CERN, en 1990; el navegador web llamado WorldWideWeb era bastante sofisticado y gráfico, pero solo funcionaba en estaciones NeXT.

El punto de giro para la World Wide Web fue la introducción del navegador web Mosaic, un programa gráfico desarrollado por un equipo de la Universidad de Illinois en el NCSA, dirigido por Marc Andreessen en 1993. Los fondos para Mosaic vinieron del High-Performance Computing and Communications Initiative, un programa de fondos iniciado por el entonces senador Al Gore.

El navegador Mosaic, que funcionaba inicialmente en entornos Unix sobre XFree86 (X11), fue el primero que se extendió debido a que pronto el National Center for Supercomputing Applications (NCSA) preparó versiones para Windows y Macintosh.

Los orígenes de Mosaic comenzaron en 1992. En el mes de noviembre de 1992, el NCSA en la Universidad de Illinois (UIUC) estableció un sitio web. En diciembre de 1992, Andreessen y Eric Bina, estudiantes de UIUC que trabajaban en NCSA, comenzaron a trabajar en Mosaic. Ellos lanzaron un buscador basado en X Window en febrero de 1993. Dicho navegador ganó popularidad debido a su fuerte soporte de multimedia integrada y los autores rápidamente respondieron a los reportes de problemas y recomendaciones para nuevas características realizados por los usuarios.

⁴ *es.wikipedia.org* (2019, 07 de marzo) *Historia de la world wide web*. *Wikipedia*. Disponible en: https://es.wikipedia.org/wiki/Historia_de_la_World_Wide_Web Consultado: 7 de marzo de 2019 09:55 pm.

Después de su graduación de UIUC(Universidad de Illinois), Andreessen y James H. Clark, antiguo CEO de Silicon Graphics, se conocieron y fundaron la antigua Mosaic Communications Corporation para desarrollar el navegador Mosaic comercialmente. La compañía cambió su nombre a Netscape en abril de 1994 y el navegador fue desarrollado luego como Netscape Navigator.

Sin embargo, Netscape Navigator al poco tiempo entró en el mercado y rápidamente superó en capacidades y velocidad al Mosaic. Este navegador tuvo la ventaja de funcionar en casi todos los sistemas Unix, y también en entornos Windows.

Internet Explorer (anteriormente Spyglass Mosaic) fue la apuesta tardía de Microsoft para entrar en el mercado y consiguió desbancar al Netscape Navigator entre los usuarios de Windows, debido a la integración del navegador con el sistema operativo y al hecho de que era gratuito, mientras que Netscape tenía costo, llegando a poseer cerca del 95 % de la cuota de mercado. Netscape Communications Corporation liberó el código fuente de su navegador, naciendo así el proyecto Mozilla.

Finalmente Mozilla (Mozilla Application Suite) fue reescrito desde cero tras decidirse a desarrollar y usar como base un nuevo conjunto de widgets multiplataforma basado en Extensible Markup Language (XML) llamado XUL y esto hizo que tardara bastante más en aparecer de lo previsto inicialmente, apareciendo una versión 1.0 de gran calidad y para muchísimas plataformas a la vez el 5 de junio de 2002.

El 7 de enero de 2003, Apple lanzó al mercado el navegador web Safari. Este navegador se hace con casi la totalidad del mercado de las microcomputadoras Mac, debido a su velocidad y gran cantidad de actualizaciones. A finales de 2004 aparece en el mercado Mozilla Firefox, una rama de desarrollo de Mozilla que pretende hacerse con parte del mercado de Internet Explorer. Se trata de un navegador más ligero que su hermano mayor.

El 2 de septiembre de 2008, Google Chrome aparece en el mercado. Es el navegador web desarrollado por Google y compilado con base en componentes de código abierto como el motor de renderizado de WebKit y su estructura de desarrollo de aplicaciones (framework). Está disponible gratuitamente bajo condiciones de servicio específicas. El nombre del navegador deriva del término usado para el marco de la interfaz gráfica de usuario (Chrome”). En diciembre de 2011, Google Chrome superó a Internet Explorer 8.0 como el navegador web más utilizado a nivel mundial.⁵

2.1.3.4. Web 2.0

Iniciando el 2002, nuevas ideas para compartir e intercambiar contenido en la red, como los blogs y RSS, rápidamente ganaron aceptación en la Web. Este nuevo modelo para el intercambio de información, primeramente caracterizando DIY(Hágalo usted mismo, por sus siglas en inglés.) editado por el usuario y sitios web generados, acuñaron la Web 2.0.

⁵ *es.wikipedia.org (2019, 07 de marzo) Navegador web. Wikipedia. Disponible en: https://es.wikipedia.org/wiki/Navegador_web Consultado: 9 de marzo de 2019 3:27 pm.*

El auge de la Web 2.0 vio muchas nuevas compañías (startup) orientadas a servicios alojadas en una nueva y democrática Web. Algunos creen que esto sería seguido por una completa realización de la Web Semántica.

El término Web 2.0 o Web social comprenden aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Web 2.0 permite a los usuarios interactuar y colaborar entre sí, como creadores de contenido. La web en este caso se convierte en una plataforma de trabajo colaborativo. Ejemplos de la Web 2.0 son las comunidades web, los servicios web, las aplicaciones Web, entre otros. La evolución de las aplicaciones, que pasan de ser estáticas a dinámicas, las cuales implican la colaboración del usuario.

Origen

El término Web 2.0 tiene un origen claro: fue utilizado por primera vez por O'Reilly Media (empresa conocida por su editorial de libros de tecnología) en una conferencia en octubre de 2004. Meses después fue Tim O'Reilly, fundador de la empresa, quien definió el concepto y lo hizo conocido en la comunidad.

Incluso antes de que el término se hiciera de uso común, O'Reilly lo había registrado como marca, lo que luego dio algunos dolores de cabeza a otras empresas que intentaron usarlo.

La Web 2.0 es una segunda generación de servicios basados en la Web, que enfatiza en la colaboración online, la conectividad y la posibilidad de compartir contenidos entre los usuarios.

La Web 2.0 implica la evolución de las aplicaciones digitales hacia aplicaciones dirigidas al usuario final, que incluyen servicios como redes sociales, blogs wikis y las folcsonomías.

2.1.3.5. Web 3.0

La Web 3.0 es considerada también como una extensión de Web Semántica definida por el premio Nobel Sir Timothy "Tim" John Berners Lee. Y trata de poder enriquecer la comunicación mediante metadatos semánticos (ontologías) que aportan un valor añadido a la información, la diferencian y la hacen más inteligente. Por lo que los contenidos ahora ya no son tratados por su sintáctica sino por su semántica.

2.1.3.6. Web 4.0

La Web 4.0 es un término que se va acuñando recientemente y trata de movernos hacia una Web Ubicua donde el objetivo primordial será el de unir las inteligencias, para que tanto las personas como las cosas se comuniquen entre sí para generar la toma de decisiones.

La evolución puede resumirse en:

- Web 1.0: Personas conectándose a la Web y la Web como punto de información estática.
- Web 2.0: Personas conectándose a personas, la inteligencia colectiva como centro de información y la Web es sintáctica.
- Web 3.0: Aplicaciones Web conectándose a aplicaciones Web, las personas siguen siendo el centro de la información y la Web es semántica.
- Web 4.0: Personas conectándose con Personas y aplicaciones Web de forma ubicua, se añaden tecnologías como la Inteligencia Artificial, la Voz como vehículo de intercomunicación para formar una web total.⁶

2.1.4. Lenguajes de programación

Los primeros lenguajes de programación preceden a la computadora moderna. En un inicio los lenguajes eran códigos.

La máquina del telar de Jacquard, creada en 1801, utilizaba los orificios en tarjetas perforadas para representar los movimientos de un brazo de la máquina de tejer, con el objetivo de generar patrones decorativos automáticamente.

Durante un período de nueve meses entre 1842 y 1843, Ada Lovelace tradujo las memorias del matemático italiano Luigi Menabrea acerca de la nueva máquina propuesta por Charles Babbage, la Máquina Analítica. Con estos escritos, ella añadió unas notas en las cuales especificaba en detalle un método para calcular los números de Bernoulli con esta máquina.

Herman Hollerith se percató de que podía codificar la información en tarjetas perforadas cuando observó a los conductores de trenes que identificaban a los pasajeros según el orificio que hacían en su respectivo ticket. En 1890 Hollerith codificó los datos del censo en tarjetas perforadas.

Los primeros códigos de computadora estaban especializados según sus aplicaciones. En las primeras décadas del siglo 20, los cálculos numéricos estaban basados en los números decimales. Eventualmente se fueron dando cuenta que la lógica podía ser representada con números, no sólo con palabras. Por ejemplo, Alonzo Church fue capaz de expresar el cálculo lambda a través de fórmulas. La máquina de Turing estableció las bases para almacenar programas como datos en la arquitectura de von Neuman de una computadora. Sin embargo, a diferencia del cálculo lambda, el código de Turing no serviría satisfactoriamente como base para lenguajes de más alto nivel- su principal uso es en el análisis riguroso en la complejidad algorítmica.

Para algunas personas, lo que sería el primer lenguaje de programación moderno depende de cuánto poder y legibilidad humana se requería antes de que se concediera el estado de "lenguaje de programación". Tanto el telar de Jacquard como la Máquina Diferencial de Babbage, tenían lenguajes muy simples y extremadamente limitados

⁶ *Comunicación multimedia (2015, 08 de mayo) El avance de la web. Comunicación multimedia. Disponible en: <http://comunicacionmultimedia2015.blogspot.com/2015/05/el-avance-de-la-web-10-la-web-40.html> Consultado: 9 de marzo de 2019 8:50 pm.*

para describir las acciones que estas máquinas realizaran. Se puede incluso considerar los agujeros perforados en los rollos de pianola como un limitado lenguaje de dominio específico, a pesar de no estar diseñado para el consumo humano.⁷

2.1.4.1. Historia

En la década de 1940 fueron creadas las primeras computadoras modernas, con alimentación eléctrica. La velocidad y capacidad de memoria limitadas forzaron a los programadores a escribir programas, en lenguaje ensamblador muy afinados. Finalmente se dieron cuenta de que la programación en lenguaje ensamblador requería de un gran esfuerzo intelectual y era muy propensa a errores.

En 1948, Konrad Zuse publicó un artículo acerca de su lenguaje de programación Plankalkül. Sin embargo, no fue implementado en su vida y sus contribuciones fueron aisladas de otros desarrollos.

Entre algunos lenguajes importantes que fueron desarrollados en este período se encuentran:

1. 1943 - Plankalkül, diseñado, pero sin implementar durante medio siglo.
2. 1943 - se crea el sistema de codificación ENIAC.
3. 1949 - 1954 – una serie de conjuntos de instrucciones nemotécnicas, como la de ENIAC, comenzando en 1949 con C-10 para BINAC (que luego evolucionaría en UNIVAC). Cada conjunto de instrucciones estaba destinado a un fabricante específico.

2.1.4.2. La década de 1980: consolidación, módulos, rendimiento

La década de 1980 fueron años de consolidación relativa en los lenguajes imperativos. En vez de inventar nuevos paradigmas, se comenzó a trabajar a partir de las ideas inventadas en la década anterior. C++ combinaba la programación orientada a objetos y la programación de sistemas. El gobierno de Estados Unidos estandarizó Ada, un lenguaje de programación de sistemas destinado a ser utilizado por contratistas de defensa. En Japón y en otras partes, se gastaron enormes sumas investigando los llamados lenguajes de programación de quinta generación que incorporaban construcciones de la programación lógica. La comunidad de los lenguajes funcionales llevaron a cabo la estandarización de ML y Lisp. La investigación en Miranda, un lenguaje funcional, con evaluación perezosa, comenzó a tomar fuerza en esta década.

Una nueva tendencia importante en el diseño de lenguajes era un mayor enfoque en la programación de sistemas a gran escala a través del uso de módulos, una organización a gran escala de unidades de código. Modula, Ada, y ML desarrollaron sistemas de módulos notables en la década de 1980. Los sistemas de módulos eran relacionados con frecuencia con construcciones de programación genéricas, que serían, en esencia,

⁷ *es.wikipedia.org* (2019, 01 de febrero) *Historia de los lenguajes de programación*. *Wikipedia*. Disponible en: https://es.wikipedia.org/wiki/Historia_de_los_lenguajes_de_programacion Consultado: 9 de marzo de 2019 9:29 pm.

módulos parametrizados (véase también polimorfismo en programación orientada a objetos). Aunque no aparecían nuevos paradigmas de lenguajes de programación imperativos, muchos investigadores expandieron las ideas de los lenguajes anteriores y las adaptaron a nuevos contextos. Por ejemplo, los lenguajes de los sistemas Argus y Emerald adaptaron la programación orientada a objetos para sistemas distribuidos.

La década de 1980 también trajo avances en la implementación de lenguajes de programación. El movimiento de RISC en la arquitectura de computadoras postulaba que el hardware debía estar diseñado para los compiladores más que para los programadores de ensamblador humanos. Con la ayuda de las mejoras en la velocidad del procesador, permitiendo técnicas de compilación cada vez más agresivas, el movimiento RISC despertó un mayor interés en la tecnología de compilación de los lenguajes de alto nivel.

Las tecnologías de los lenguajes continuaron sobre estas líneas entrando en la década de 1990.

Algunos de los lenguajes más importantes que se desarrollaron en este período son:

1. 1980 - C++ (C con clases, el nombre cambió en julio de 1983)
2. 1983 - Ada
3. 1984 - Common Lisp
4. 1984 - MATLAB
5. 1986 - Objective-C
6. 1987 - Perl

2.1.4.3. La década de 1990: la era del Internet y los lenguajes de programación orientados a la web.

El rápido crecimiento de Internet en la década de 1990 fue el siguiente gran acontecimiento histórico para los lenguajes de programación. Con la apertura de una plataforma totalmente nueva para los sistemas informáticos, Internet creó una oportunidad adoptar nuevos lenguajes. En particular, el lenguaje de programación JavaScript se hizo popular debido a su pronta integración con el navegador web Netscape Navigator, y varios lenguajes de scripting alcanzaron un amplio uso en el desarrollo de aplicaciones personalizadas para servidores web. La década de 1990 no vio ninguna novedad fundamental en los lenguajes imperativos, pero sí mucha recombinación y la maduración de viejas ideas. Esta era comenzó la difusión de los lenguajes funcionales. Una filosofía de conducción grande era la productividad del programador. Surgieron muchos lenguajes de “aplicaciones de desarrollo rápido” (RAD), los cuales usualmente venían con un IDE, recolector de basura, y eran descendientes de lenguajes anteriores. Todos estos lenguajes eran orientados a objeto. Estos incluían Object Pascal, Visual Basic y Java. Java, en particular, recibió mucha atención. Pero más radicales e innovadores que los lenguajes de RAD eran los nuevos lenguajes de script. Estos no descendían directamente de otros lenguajes y ofrecieron

nuevas sintaxis e incorporación más liberal de otras características. Muchos consideran estos lenguajes de script más productivos que los lenguajes de RAD, aunque esto se debe a menudo a que es más difícil escribir y mantener largos programas que pequeños programas simples. Sin embargo, no es menos cierto que los programas de script llegaron para convertirse en los más prominentes en la conexión con la Web.

Algunos de los lenguajes más importantes que se desarrollaron en este período son:

1. 1990 - Haskell.
2. 1991 - Python.
3. 1991 - Visual Basic.
4. 1991 - HTML (lenguaje de marcado de hipertexto).
5. 1993 - Ruby.
6. 1995 - Java.
7. 1995 - JavaScript.
8. 1995 - PHP.⁸

2.1.4.4. Historia de PHP

PHP tal y como se conoce hoy en día es en realidad el sucesor de un producto llamado PHP/FI. Creado en 1994 por Rasmus Lerdorf, la primera encarnación de PHP era un conjunto simple de ficheros binarios Common Gateway Interface (CGI) escritos en el lenguaje de programación C. Originalmente utilizado para rastrear visitas de su currículum online, llamó al conjunto de scripts "Personal Home Page Tools", más frecuentemente referenciado como "PHP Tools". Con el paso del tiempo se quiso más funcionalidad, y Rasmus reescribió PHP Tools, produciendo una implementación más grande y rica. Este nuevo modelo fue capaz de interactuar con bases de datos, y mucho más, proporcionando un entorno de trabajo sobre cuyos usuarios podían desarrollar aplicaciones web dinámicas sencillas tales como libros de visitas. En junio de 1995, Rasmus » publicó el código fuente de PHP Tools, lo que permitió a los desarrolladores usarlo como considerasen apropiado. Esto también permitió -y animó- a los usuarios a proporcionar soluciones a los errores del código, y generalmente a mejorarlo.

En septiembre de ese mismo año, Rasmus amplió PHP y, por un corto periodo de tiempo, abandonó el nombre de PHP. Ahora, refiriéndose a las herramientas como FI (abreviatura de "Forms Interpreter"), la nueva implementación incluía algunas de las funciones básicas de PHP tal y como la conocemos hoy. Tenía variables como las de Perl, interpretación automática de variables de formulario y sintaxis incrustada HTML. La sintaxis por sí misma era similar a la de Perl, aunque mucho más limitada, simple y algo inconsistente. De hecho, para embeber el código en un fichero HTML, los desarrolladores tenían que usar comentarios de HTML. Aunque este

⁸ *es.wikipedia.org (2019, 27 de febrero) Historia de los lenguajes de programación. Wikipedia. Disponible en: https://es.wikipedia.org/wiki/Historia_de_los_lenguajes_de_programacion Consultado: 9 de marzo de 2019 9:29 pm.*

método no era completamente bien recibido, FI continuó gozando de expansión y aceptación como una herramienta CGI, pero todavía no completamente como lenguaje. Sin embargo, esto comenzó a cambiar al mes siguiente; en octubre de 1995 Rasmus publicó una versión nueva del código. Recordando el nombre PHP, ahora era llamado (resumidamente) "Personal Home Page Construction Kit,^z fue la primera versión que presumía de ser, en aquel momento, considerada como una interfaz de scripts avanzada. El lenguaje fue deliberadamente diseñado para asemejarse a C en estructura, haciéndolo una adopción sencilla para desarrolladores familiarizados con C, Perl, y lenguajes similares. Habiendo sido así bastante limitado a sistemas UNIX y compatibles con POSIX, el potencial para una implementación de Windows NT estaba siendo explorada.

El código fue completamente rehecho de nuevo, y en abril de 1996, combinando los nombres de versiones anteriores, Rasmus introdujo PHP/FI. Esta implementación de segunda generación comenzó realmente a desarrollar PHP desde un conjunto de herramientas dentro de un lenguaje de programación de derecho propio. Incluía soporte interno para DBM, mSQL, y bases de datos Postgres95, cookies, soporte para funciones definidas por el usuario, y mucho más. Ese mes de junio, PHP/FI brindó una versión 2.0. Sin embargo, un interesante hecho sobre esto, es que sólo había una única versión completa de PHP 2.0. Cuando finalmente pasó de la versión beta en noviembre de 1997, el motor de análisis subyacente ya estaba siendo reescrito por completo.

Aunque vivió una corta vida de desarrollo, continuó gozando de un crecimiento de popularidad en el aún joven mundo del desarrollo. En 1997 y 1998, PHP/FI tenía un culto de varios miles de usuarios en todo el mundo. Una encuesta de Netcraft en mayo de 1998 indicó que cerca de 60,000 dominios reportaron que tenían cabeceras que contenían "PHP", indicando en efecto que el servidor host lo tenía instalado. Este número se correspondía con aproximadamente el 1% de todos los dominios de Internet del momento. A pesar de estas impresionantes cifras, la maduración de PHP/FI estaba condenada por limitaciones; mientras habían varios contribuidores menores, aún era desarrollado principalmente por un individuo.

A partir de acá, es cuando comienza el gran crecimiento y auge de PHP a través de todo internet. A partir de 1997 hasta la actualidad, el lenguaje PHP fue ganando gran popularidad y extensibilidad, adoptando diversas mejoras tanto en rendimiento como en características que facilitan aun más la interacción y el desarrollo con éste. PHP continuó evolucionando a lo largo del nuevo milenio, pasando por PHP 3.0, PHP 4.0 (1998), llegando a PHP 5.0 allá por el año 2004, que fue una de las versiones de PHP que mayores mejoras y cambios introdujo. Actualmente en 2019, PHP se encuentra en su versión 7.3.2 (Marzo de 2019).

Hemos analizado la historia del lenguaje de programación libre para la web por excelencia, sin el cual, la web no sería lo grande y extendida que es en la actualidad.⁹

⁹ *php.net (2001, 8 de enero) Historia de PHP. PHP Group. Disponible en: <https://www.php.net/manual/es/history.php.php> Consultado: 9 de marzo de 2019 11:14 pm.*

2.1.4.5. Historia de Javascript

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios complejos.

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes (como ScriptEase) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje LiveScript. Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época.

La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1. Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3. JScript era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales.

Para evitar una guerra de tecnologías, Netscape decidió que lo mejor sería estandarizar el lenguaje JavaScript. De esta forma, en 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association).

ECMA creó el comité TC39 con el objetivo de "estandarizar de un lenguaje de script multiplataforma e independiente de cualquier empresa". El primer estándar que creó el comité TC39 se denominó ECMA-262, en el que se definió por primera vez el lenguaje ECMAScript.

Por este motivo, algunos programadores prefieren la denominación ECMAScript para referirse al lenguaje JavaScript. De hecho, JavaScript no es más que la implementación que realizó la empresa Netscape del estándar ECMAScript. Actualmente (Marzo de 2019) ECMAScript se encuentra en su versión 7.¹⁰

¹⁰ *uniwebsidad.com (2018, 20 de noviembre) Introducción a Javascript. uniwebsidad. Disponible en: <https://uniwebsidad.com/libros/javascript/capitulo-1/breve-historia> Consultado: 9 de marzo de 2019 11:46 pm.*

2.2. Marco Teórico

2.2.1. Sistema

Un sistema ¹¹ es un conjunto de elementos relacionados entre sí que funciona como un todo. La palabra sistema procede del latín *systema*, y este del griego *systema*, identificado en español como “unión de cosas de manera organizada”.

2.2.1.1. Sistema informático

Un sistema informático (SI) ¹² es un sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático. El hardware incluye computadoras o cualquier tipo de dispositivo electrónico, que consisten en procesadores, memoria, sistemas de almacenamiento externo, etc. El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos. Por último, el soporte humano incluye al personal técnico que apoyan y mantienen el sistema (analistas, programadores, operarios, etc.) y a los usuarios que lo utilizan.

Los sistemas informáticos pasan por diferentes fases en su ciclo de vida, desde la captura de requisitos hasta el mantenimiento. En la actualidad se emplean numerosos sistemas informáticos en la administración pública, por ejemplo, las operadoras de la policía, el servicio al cliente, entre otras.

2.2.1.2. Clasificación de los sistemas informáticos

Los sistemas informáticos pueden clasificarse con base a numerosos criterios. Las clasificaciones no son estancas y es común encontrar sistemas híbridos que no encajen en una única categoría.

- Por su uso
 - Sistemas de uso específico. En sistemas complejos es frecuente tener subsistemas que se encargan de tareas específicas como por ejemplo el sistema de detección de intrusos o el sistema de monitorización.
 - Sistemas de uso general.
- Por el paralelismo de los procesadores
 - MIMD, Multiple Instruction Multiple Data.
 - SIMD, Single Instruction Multiple Data.
 - SISD, Single Instruction Single Data.
- Por el tipo de computadora utilizado en el sistema

¹¹ "Sistema" (s. f.). En: *Significados.com*. Disponible en: <https://www.significados.com/sistema/> Consultado: 8 de mayo de 2019, 04:11 pm.

¹² "Sistema informático" (s. f.). En: *Wikipedia.com*. Disponible en: https://es.wikipedia.org/wiki/Sistema_informático Consultado: 8 de mayo de 2019, 04:15 pm.

- Estaciones de trabajo (workstations).
 - Minicomputadoras (por ejemplo, computadoras personales).
 - Microcomputadoras (servidores pequeños).
 - Supercomputadoras.
 - Terminales ligeros (thin clients).
- Por la arquitectura
 - Arquitectura cliente-servidor.
 - Arquitectura de 3 capas.
 - Arquitectura de 4 capas.
 - Arquitectura de n capas.
 - Monitor de teleproceso o servidor de transacciones.
 - Servidor de aplicaciones.
 - Sistema aislado.

2.2.2. Metodologías de desarrollo

2.2.2.1. Cascada

El modelo en cascada las actividades el desarrollo fundamental de este modelos de derivan en etapas las cuales son: ¹³

Análisis y definición de requerimientos: Definición en detalle las especificaciones del sistemas como son los servicios, restricciones y metas del sistema.

Diseño del sistema hardware: Establecimiento de una arquitectura completa del sistema describiendo las abstracciones fundamentales del sistema y sus relaciones.

Implementación y prueba de unidades: Durante esta etapa el diseño se lleva a cabo como unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.

Integración y prueba del sistema: Los programas individuales se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software, después el sistema se entrega al cliente.

Funcionamiento y mantenimiento: El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en la etapa anteriores del ciclo de vida, mejorar la implementación de los unidades del sistema y resaltar los servicios del sistema una vez que se descubre nuevos requerimientos.

¹³ "Metodologías clásicas" (s. f.). En: MarcoTeórico.com. Disponible en: <https://www.marcoteorico.com/curso/45/ingenieria-de-software/253/metodologias-clasicas>
Consultado: 8 de mayo de 2019, 03:32 pm.

Las ventajas del modelo en cascada son que la documentación se produce en cada fase y este cuadra con otros modelos del proceso de ingeniería.

La principal desventaja es su inflexibilidad en dividir el proyecto en distintas etapas. Se debe hacer compromisos en las etapas iniciales, lo que hace difícil responder a los cambios en los requerimientos del cliente.

2.2.2.2. Incremental

El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software combinada con los elementos del modelo lineal secuencial con la filosofía interactiva de construcción de prototipos.

El modelo de proceso incremental, como la construcción de prototipos es iterativo por naturaleza. Pero a diferencia de la construcción de prototipos, el modelo incremental se centra en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación. El desarrollo incremental es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en la fecha límite que se haya establecido para el proyecto. Los primeros incrementos se pueden implementar con menos personas.

2.2.2.3. Evolutivo

Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más completas del software.

2.2.2.4. Espiral

El modelo en espiral es un proceso evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

Modelo en espiral contiene seis regiones de tareas las cuales son:

- Comunicación con el cliente: Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- Planificación: Las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.
- Análisis de riesgos: Las tareas requeridas para evaluar riesgos técnicos y de gestión.

- Ingeniería: Las tareas requeridas para construir una o más representaciones de la aplicación.
- Construcción y acción: Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- Evaluación del cliente: Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

2.2.2.5. Prototipos

El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

2.2.2.6. Desarrollo basado en componentes

El desarrollo basado en componentes incorpora muchas de las características del modelo en espiral. Es evolutivo por naturaleza y exige un enfoque iterativo para la creación del software. Sin embargo, el modelo de desarrollo basado en componentes configura aplicaciones desde componentes preparados de software. Conduce a la reutilización del software, y la reutilización proporciona beneficios a los ingenieros de software.

2.2.3. Ingeniería de software

Tomando en cuenta las siguientes definiciones de Ingeniería de Software:¹⁴

- La ingeniería de software es la rama de la ingeniería que aplica principios de las ciencias de la computación y matemáticas para lograr soluciones rentables a los problemas de software.
- Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software.
- La Ingeniería de Software es la disciplina de la ingeniería que estudia la naturaleza, los enfoques y metodologías de desarrollo a gran escala del software.
- Además de las teorías y leyes detrás de las prácticas ingenieriles y el comportamiento del software.

¹⁴ Matla Cruz, E, (2014). *DESARROLLO DE SOFTWARE GUIADO POR LA NORMA ISO/IEC 29110 y SCRUM: SIDEV V.2.0 (Tesis de pregrado)*. UNAM, México.

Podemos construir nuestra propia definición:

- Llamaremos Ingeniería de Software a aquel proceso que involucra métodos, herramientas y técnicas que nos guiarán en el proceso de desarrollo de software y que tiene como fin la construcción de software correcto.

De manera genérica, todo proceso de Ingeniería de Software involucra las siguientes etapas de desarrollo:

Análisis de requerimientos. La captura, análisis y especificación de requerimientos es la primer etapa para la creación de software, se conocen los requerimientos que debe cumplir el sistema a través de la retroalimentación de parte del cliente y el conocimiento de las reglas de negocio.

Diseño. Tras pasar el análisis, en esta etapa se proponen los módulos a desarrollar y se deberán detallar cada uno de ellos. Es en esta fase cuando se definen aspectos como la arquitectura del sistema, la forma en que se desarrollarán las aplicaciones, el diseño de la base de datos y aquellas herramientas que se utilizarán para el desarrollo, pruebas y liberación.

Construcción. Se resume esta etapa en una palabra: codificación, se procede a codificar cada uno de los módulos planteados en la etapa de diseño, respetando estándares de codificación y de nombrado. Dentro de esta etapa, el equipo de desarrollo deberá probar cada uno de los componentes que conforman el sistema, realizando pruebas de caja negra y caja blanca.

Integración. En esta fase se unen los módulos creados en la etapa de construcción de acuerdo a las especificaciones obtenidas en la etapa de diseño, en cada integración de módulos se comienza en parte con la etapa de pruebas, ya que cada integración conlleva la necesidad de comprobar que en conjunto los módulos funcionan correctamente.

Pruebas. Las pruebas pueden dividirse en dos partes, primeramente el equipo de desarrollo deberá verificar que las funcionalidades implementadas trabajan de manera correcta. Posteriormente, al haberse asegurado del buen funcionamiento del sistema, se deberán realizar pruebas con usuarios finales del sistema para validar que éste cumple con los requerimientos que inicialmente se establecieron. Durante esta fase se recabarán sus comentarios sobre lo que se debe mejorar y las fallas que se detectaron en el uso del sistema.

Liberación. Al concluir la fase de pruebas, el sistema se encuentra listo para su uso. Es por ello que se entrega junto con la documentación al cliente, misma que servirá de referencia para los usuarios y el equipo encargado de su administración, operación y mantenimiento.

2.2.4. Base de datos

El día de hoy las bases de datos ¹⁵ se han vuelto una herramienta muy importante, no sólo por el hecho de permitir el almacenamiento de grandes cantidades de datos, sino también por que facilitan la obtención de información para la toma de decisiones. Diremos que una base de datos es un conjunto de datos los cuales mantienen una relación entre si al pertenecer a un mismo contexto, estos datos son administrados a través de un Sistema Manejador de Bases de Datos (DBMS por sus siglas en inglés). Al ser ésta el “almacén” de los datos del sistema, se convierte en el cimiento del mismo. Es por ello que se debe contar con un buen diseño de la misma para que sobre ésta se construyan los módulos del software.

2.2.4.1. Tipos de bases de datos

Existen varios tipos de bases de datos, entre los que podemos destacar:

- **Estáticas:** Son bases de datos que se utilizan sólo para lectura, conocidas como “Data warehouse” o cubos de datos, su función principal es ser la fuente de datos para la generación de reportes, toma de decisiones, análisis empresariales y minería de datos (extracción de conocimiento).
- **Dinámicas.** Son la mayoría de las bases que existen en la actualidad, permiten la modificación de datos mediante inserciones, actualizaciones y borrados además de consulta de datos.

2.2.4.2. Modelos de bases de datos

Entre los modelos de bases de datos podemos destacar:

- **Bases de datos jerárquicas.** La estructura de estas bases de datos consiste de un nodo padre el cual puede tener varios hijos. Optimizadas por índices, este modelo es eficiente para el manejo de grandes cantidades de datos, en su contra está la limitación para eficientar la redundancia de datos
- **Bases de datos transaccionales.** Permiten la consulta y modificación de datos a gran velocidad, al ser transaccionales sus operaciones son atómicas; para comprender más este concepto el ejemplo común es realizar un traspaso de la cuenta A a la cuenta B, se realizan dos operaciones una que decrementará el saldo en A y otra que incrementará el saldo en B, de esta manera o bien se llevan a cabo ambas operaciones o no se lleva a cabo ninguna.
- **Bases de datos relacionales.** Los datos son almacenados en entidades las cuales se encuentran relacionadas, las entidades se encuentran compuestas de atributos y almacenan tuplas las cuales contienen los datos que estamos almacenando. Para manipular la información se cuenta con un lenguaje formal llamado álgebra relacional que sienta las bases para construir las consultas las cuales se realizan en lenguaje estructurado de consultas (SQL por sus siglas en inglés). Durante el diseño de la base en este modelo se pasa por varias fases, como son: creación de diagramas entidad-relación, normalización de relaciones y creación de diagramas de clases UML

¹⁵ Silberschatz, A., Korth, H. F., Sudarshan, S., Pérez, F. S., Santiago, A. I., & Sánchez, A. V. (2002). *Fundamentos de bases de datos*.

Para este proyecto se ha elegido utilizar una base de datos relacional debido a la necesidad de representar entidades con múltiples características. Estas entidades se encuentran relacionadas entre sí, además de la facilidad para modelar los supuestos que se obtienen tras el análisis de la lógica que deberá llevar el sistema.

2.2.5. Programación

Un lenguaje de programación ¹⁶ es un lenguaje formal que especifica una serie de instrucciones para que una computadora produzca diversas clases de datos. Los lenguajes de programación se usan para crear programas de computadora. Un lenguaje de programación permite poner algoritmos en práctica, los cuales controlan el comportamiento físico y lógico de una computadora.

Está formado por un conjunto de símbolos (llamado alfabeto), reglas gramaticales (léxico/morfológicas y sintácticas) y reglas semánticas, que en conjunto definen las estructuras válidas en el lenguaje y su significado. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.

A continuación se detallan los lenguajes de programación y otras herramientas que se usarán para crear el sistema informático de gestión de expedientes.

2.2.5.1. PHP

PHP ¹⁷, acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3_0, es una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, esta licencia no tiene restricciones de copyleft.^a asociadas con GPL.

¹⁶ Olarte Gervacio, L. (2018, 06 de febrero) *Lenguaje de Programación. Conogasi, Conocimiento para la vida. Consultado: 8 de mayo de 2019, 04:15 pm.*

¹⁷ Pérez Valdeés, D. (2017, 02 de noviembre) *Los diferentes lenguajes de programación para la web. Maestros del web. Disponible en: <http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/> Consultado: 10 de mayo de 2019 08:32 pm.*

2.2.5.2. JavaScript

JavaScript ¹⁸ (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5., una versión de javascript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

2.2.5.3. HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web

¹⁸ *Ibidem*

y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.) No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores todavía mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, todavía existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

2.2.5.4. CSS

CSS (siglas en inglés de Cascading Style Sheets), en español "Hojas de estilo en cascada", es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS).

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del

documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en voz (mediante un navegador de voz o un lector de pantalla, y dispositivos táctiles basados en el sistema Braille). También se puede mostrar una página web de manera diferente dependiendo del tamaño de la pantalla o tipo de dispositivo. Los lectores pueden especificar una hoja de estilos diferente, como una hoja de estilos CSS guardado en su computadora, para sobrescribir la hoja de estilos del diseñador.

La especificación CSS describe un esquema prioritario para determinar qué reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas son aplicadas con un sistema llamado de cascada, de modo que las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C). El MIME type `text/css` está registrado para su uso por CSS descrito en el RFC 2318. El W3C proporciona una herramienta de validación de CSS gratuita para los documentos CSS.

2.2.5.5. Bootstrap

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Bootstrap es el segundo proyecto más destacado en GitHub y es usado por la NASA y la MSNBC entre otras organizaciones.

2.2.5.6. MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius —fundador de MySQL—, la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria —que reemplaza a MyISAM— y otro llamado XtraDB —en sustitución de InnoDB—. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Este SGBD surge a raíz de la compra de Sun Microsystems —compañía que había comprado previamente MySQL AB— por parte de Oracle. MariaDB es una bifurcación directa de MySQL que asegura la existencia de una versión de este producto con

licencia GPL. Widenius decidió crear esta variante porque estaba convencido de que el único interés de Oracle en MySQL era reducir la competencia que MySQL suponía para el mayor proveedor de bases de datos relacionales del mundo, que es Oracle.

2.2.5.7. jQuery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.¹⁹

2.2.6. Pruebas de validación

Las pruebas de validación en la ingeniería de software son el proceso de revisión que verifica que el sistema de software producido que cumple con las especificaciones y que logra su cometido. Es normalmente una parte del proceso de pruebas de software de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones y tutoriales. La validación es el proceso de comprobar que lo que se ha especificado es lo que el usuario realmente quería.²⁰

2.2.6.1. Alfa

Es la primera versión del programa, la cual es enviada a los verificadores para probarla. Algunos equipos de desarrollo utilizan el término alfa informalmente para referirse a una fase donde un producto todavía es inestable, aguarda todavía a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos.

2.2.6.2. Beta

Representa generalmente la primera versión completa del producto, que es posible que sea inestable pero útil para que las demostraciones internas y las inspecciones previas seleccionen a clientes. Esta etapa comienza a menudo cuando los desarrolladores anuncian una congelación de las características del producto, indicando que no

¹⁹ Pérez Valdeś, D. (2017, 02 de noviembre) *Los diferentes lenguajes de programación para la web. Maestros del web. Disponible en: <http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/> Consultado: 10 de mayo de 2019 08:32 pm.*

²⁰ Blanco-Llano, Javier, Rodríguez-Hernández, Aida, *REVISIÓN, VERIFICACIÓN Y VALIDACIÓN EN UN PROCESO DE DESARROLLO DE SOFTWARE. Ingeniería Industrial [en línea] 2011, XXXII (Enero-Abril) : [Fecha de consulta: 26 de mayo de 2019] Disponible en: <http://www.redalyc.org/articulo.oa?id=360433575005>ISSN 0258-5960*

serán agregadas más características a esta versión y que solamente se harán pequeñas ediciones o se corregirán errores. Muchos de estos programas beta, son de uso privado solo permitiendo a un número determinado de usuarios probarlo, y de esta manera mantener un control más eficiente de la calidad y las opiniones de los usuarios que lo están probando. Este tipo de programas casi siempre incluye instrucciones específicas para reportar estos bugs y recibir ayuda en caso de ser necesario.

Por otro lado, tenemos los betas públicos, que son programas como los que se han comentado al principio, software dirigido a cualquier usuario con un ordenador para que lo prueben y analicen ellos mismos.

2.2.7. Estudio de factibilidad

Factibilidad ²¹ se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados. Generalmente la factibilidad se determina sobre un proyecto.

El estudio de factibilidad, es una de las primeras etapas del desarrollo de un sistema informático. El estudio incluye los objetivos, alcances y restricciones sobre el sistema, además de un modelo lógico de alto nivel del sistema actual (si existe). A partir de esto, se crean soluciones alternativas para el nuevo sistema, analizando para cada una de éstas, diferentes tipos de factibilidades.

2.2.7.1. Factibilidad técnica

Factibilidad de sistemas Técnica es una evaluación que debe demostrar la facultad del sistema para ponerse en marcha y mantenerse durante el tiempo, además debe demostrar que la planeación del sistema ha sido desarrollada cuidadosamente contemplando todas las restricciones y objetivos, aprovechando los recursos que entrega la organización.

Los conceptos que hay que considerar en la planeación de la Factibilidad de sistemas técnica es:

- El sistema funciona como corresponde (números de pruebas)
- El sistema está desarrollado para mantenerse cerca de los consumidores.
- Escalas de producción (Ampliación o reducción de producción).
- Complementos que ayuden el desarrollo del proyecto: ¿Existe la tecnología necesaria?, ¿De dónde se obtendrá la tecnología, ¿Se puede capacitar al personal con la nueva tecnología? ¿Hay proveedores alternativos para el sistema?

2.2.7.2. Factibilidad económica

En esta etapa, hay que comprobar que el proyecto es sustentable económicamente. Justificar que la inversión genera una ganancia, demostrar que si el sistema no

²¹ "Definición de Factibilidad" (s. f.). En: Alegsa. Disponible en: <http://www.alegsa.com.ar/Dic/factibilidad.php>. Consultado: 12 de mayo de 2019, 11:23 pm.

cumple con su objetivo no habrán pérdidas económicas o serán las mínimas.

Los Costos: Considerar costos Fijos y variables

Las Ventas: demostrar como se ha definido el costo del producto y cuales son los estimados de ventas por el periodo de al menos un año, justificando cada calculo, investigación de mercado y estadísticas.

2.2.7.3. Factibilidad operativa

La Factibilidad de sistemas Operativa, tiene como objetivo comprobar que a empresa u organización sera capaz de darle uso al sistema, que cuenta con el personal capacitado para hacerlo o tiene los recursos humanos necesarios para mantener el sistema. para esto, el sistema debe contemplar cuatro puntos importante al momento de desarrollarse.

El sistema no debe ser complejo para los usuarios de la organización o los que operan el sistema, hay que evitar que el usuario ocupe el sistema de manera que pueda ocasionar errores o darle un uso indebido, simplificar las funciones y dar todo por servido.

Evitar que a los usuarios les incomode el nuevo sistema, ya sea por que se sientan desplazados de sus obligaciones o por la costumbre a un sistema antiguo, mantenerlo amigable y comprensible para los operadores.

Un cambio repentino, puede ocasionar un lento aprendizaje, capacitar y permitir al personal adaptarse a el con la tranquilidad y apoyo necesario, manuales, charlas, capacitaciones.

Como ultimo punto a considerar es la posibilidad de la obsolescencia subsecuente. La tecnología existe, pero aun no esta disponible en ese caso, es mejor constar con tecnología que este disponible en el momento y sea fácil de obtener o este mas al alcance de la mano (por si se requieren repuestos o correcciones sea fácil de conseguir). También tener en consideración las políticas habidas y por haber, de manera que si hay un cambio administrativo el sistema no quede obsoleto muy pronto.

2.2.8. Metodología de Investigación del problema

La metodología de solución que se usará para la problemática que posee la Administración Académica de la Facultad Multidisciplinaria Oriental, consiste en una investigación exhaustiva del problema y sus causas.

A continuación se detallan las técnicas que se utilizarán, para la recopilación de datos:

- **Entrevistas:** Se realizarán entrevistas al jefe de la administración académica, así como a los encargados de la gestión de los expedientes. Esta metodología fue la base para la determinación del problema de la institución de forma general.
- **Cuestionarios:** Se solicitará información necesaria para la investigación, mediante cuestionarios dirigidos a cada uno de los usuarios finales del sistema.

- **Observación directa:** Se realizarán una serie de visitas a las oficinas de la Administración Académica, con el fin de observar el ambiente de trabajo actual, el funcionamiento de el sistema existente, así como también las condiciones del equipo informático disponible.
- **Documentos:** Se cumplirá con los lineamientos de los manuales existentes para la administración académica que proveerá el jefe del departamento.

2.2.8.1. Ciclo de vida de desarrollo de software

Luego de realizar la etapa de investigación del problema, se opta por un ciclo de vida estructura de cinco etapas, según Pressman.²² El ciclo de vida estructurado sugiere un enfoque sistemático y desarrollo de sistemas. Cada etapa del proyecto es una macro actividad que se puede descomponer en actividades y tareas. A continuación se definirán las mismas.

Etapa I: Análisis de los requisitos del software:

El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Dentro del proceso de análisis, es fundamental que a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión. [PRR98]. Es de suma importancia que antes de empezar a codificar los programas, se tenga una completa y plena comprensión de los requisitos del software.

Pressman establece que la tarea del análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El análisis de requisitos permite al desarrollador o desarrolladores especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.

El análisis de requisitos del software puede dividirse en cinco áreas de esfuerzo, que son:

1. Reconocimiento del problema. Reconocer los elementos básicos del problema tal y como los perciben los usuarios finales.
2. Evaluación y síntesis. Definir todos los objetos de datos observables externamente, evaluar el flujo y contenido de la información, definir y elaborar todas las funciones del software, entender el comportamiento del software en el contexto de acontecimientos que afectan al sistema.
3. Modelado. Crear modelos del sistema con el fin de entender mejor el flujo de datos y control, el tratamiento funcional y el comportamiento operativo y el contenido de la información.

²² Roger S. Pressman, "INGENIERÍA DEL SOFTWARE UN ENFOQUE PRACTICO". McGrawhill, España, 1998.

4. Especificación. Realizar la especificación formal del software
5. Revisión. Un último chequeo general de todo el proceso.

Etapas II: Diseño:

Según Pressman, el diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software [PRR98].

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. De acuerdo con Pressman, el objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente.

El diseño, es la primera de las tres actividades técnicas que implica un proceso de ingeniería de software; estas etapas son diseño, codificación y pruebas. Generalmente la fase de diseño produce un diseño de datos, un diseño arquitectónico, un diseño de interfaz, y un diseño procedimental.

El diseño de datos esencialmente se encarga de transformar el modelo de dominio de la información creado durante el análisis. En el diseño arquitectónico se definen las relaciones entre los principales elementos estructurales del programa. Para una herramienta de software basada en el desarrollo e implementación de ambientes virtuales éste es un aspecto fundamental dado que en esta representación del diseño se establece la estructura modular del software que se desarrolla.

El diseño de interfaz describe cómo se comunica el software consigo mismo, con los sistemas que operan con él, y con los operadores que lo emplean [PRR98].

Etapas III: Generación de Código:

Esta actividad consiste en traducir el diseño, en una forma legible por la máquina. La generación de código se refiere tanto a la parte de generación de los ambientes virtuales, como a la parte en la cuál se añadirá comportamiento a estos ambientes. Por ejemplo, el lenguaje de programación VRML 2.0 es un lenguaje de modelado en 3D en el cuál se dibuja por medio de generar código de programación de formato y marcado para especificar las características del objeto u objetos que se van agregando a un mundo o entorno virtual. El comportamiento de las escenas virtuales es decir, su funcionalidad, se puede construir a través de algún otro lenguaje de programación, como clases Java o scripts especificados en JavaScript. Todas estas actividades implican generar código.

Etapas IV: Pruebas:

Una vez que se ha generado código, comienzan las pruebas del software o sistema que se ha desarrollado. De acuerdo con Pressman, el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores [PRR98]. En el caso de una herramienta de software, es necesario tener etapas de pruebas tanto para la parte funcional del

software, como para la parte aplicativa del mismo.

Se requiere poder probar el software con aplicaciones reales que puedan evaluar el comportamiento del software, con el fin de proporcionar retroalimentación a los desarrolladores. Es sumamente importante que durante el proceso de desarrollo no se pierda el contacto con los interesados o solicitantes del desarrollo de software, de esta manera los objetivos de proyecto se mantendrán vigentes y se tendrá una idea clara de los aspectos que tienen que probarse durante el periodo de pruebas.

Etapa V: Mantenimiento.

El software indudablemente sufrirá cambios, y habrá que hacer algunas modificaciones a su funcionalidad. Es de suma importancia que el software de calidad pueda adaptarse con fines de acoplarse a los cambios de su entorno externo. Por medio de la documentación apropiada y atinada del software se pueden presentar las vías para el mantenimiento y modificaciones al mismo.

2.2.9. Hardware

2.2.9.1. Servidor

Un servidor²³ es un ordenador de gran potencia en comparación con los ordenadores de escritorio o personales, destinado a alojar software y demás datos como base de datos, entre otros, para ofrecer sus servicios en una red, ya sea internet, una intranet, una red de área local, extendida, etc.

Estos servicios se ofrecen a otros ordenadores o dispositivos llamados "clientes", que, mediante la red, solicitan servicios, que son atendidos por el servidor; ejemplo: un ordenador cliente visita un sitio web, solicitando una determinada página, como "index.php", que es atendido y respondido por el servidor web ejecutando la página en cuestión, y enviando el resultado al cliente, de manera que el cliente puede ver la página index en el navegador.

Existen diferentes tipos de servidores²⁴, dependiendo de los servicios que ofrece, ergo, el software que aloja:

- **Servidor de correo:** Estos servidores tienen programas capaces de almacenar correos para los usuarios locales y con un conjunto de reglas definidas por el usuario que determinan cómo el servidor de correo debe reaccionar ante el destino de un mensaje específico.

Normalmente estos servidores se dividen en otros 2 diferentes, una para el correo entrante (llamados POP3) y otro para el correo saliente (llamados SMTP).

- **Servidor FTP:** Se trata de uno de los más antiguos en Internet, "file transfer protocol." en Español Protocolo Para la Transferencia de Archivos. Se utilizan para realizar una transferencia segura de archivos entre ordenadores (envío de

²³ "Servidor", Disponible: <https://www.areatecnologia.com/informatica/servidor-y-tipos.html>. Consultado el: 5 de Marzo de 2019.

²⁴ "Tipos de servidores". Disponible en: <https://www.areatecnologia.com/informatica/servidor-y-tipos.html>, <http://www.onyxsystems.es/que-es-un-servidor.html>

archivos de un sitio a otro). Los FTP garantiza la seguridad de los archivos y control de su transferencia.

- Servidor Web: Todas las páginas web que puedes ver por internet están almacenadas en servidores, llamados servidores web.

Un servidor web almacena los archivos de una web y los proporciona a los clientes que los solicitan haciendo la transferencia de los archivos a través de la red mediante los navegadores. El cliente lo pide a través de su navegador y el servidor web lo envía al mismo navegador del cliente para que este lo pueda visualizar.

- Servidor de base de datos: Son ordenadores preparados para alojar bases de datos para ser utilizadas por uno o más clientes. Además estos servidores realizan tareas como el análisis de los datos, el almacenamiento, la manipulación de datos, y otras tareas específicas.

- Servidor de nombre de dominio: son los que se encargan de gestionar los nombres de los dominios de las páginas web (las direcciones de las webs).

- Servidor Chat: Es un equipo dedicado a manejar y mantener un chat y sus usuarios. Los más famosos son los IRC. Ahora también se les conoce como servidores en tiempo real, porque permiten intercambiar información de forma instantánea.

- Servidor Groupware: Son servidores que facilitan el trabajo en grupo de varios ordenadores, con un objetivo común (por ejemplo un proyecto).

Estos servidores disponen de software que permite colaborar a los usuarios del servidor independientemente de donde están ubicados, permitiéndoles así hacer un trabajo colaborativo.

Los archivos y datos almacenados en un servidor groupware pueden ser alterados, acceder y recuperados por los miembros del grupo de trabajo. Groupware también se conoce como software de colaboración.

- Servidor Telnet: Son servidores que nos permiten iniciar sesión en cualquier ordenador y realizar tareas en otro ordenador. Podemos trabajar con nuestro ordenador de forma remota, es decir desde otro ordenador.

- Servidores SIP: Se encargan de gestionar el establecimiento de las llamadas telefónicas por internet. Los SIP almacenan la dirección IP donde deben acceder para realizar la comunicación con un usuario. No transmite ni audio ni video, solo establece la comunicación.

- Servidores de lista: Permiten gestionar listas de correos.

- Servidores de la nube o cloud server: Realmente estos servidores lo único que hacen es dejarte o alquilarte un espacio del servidor. La mayoría se utilizan para almacenar grandes cantidades de información en el servidor y tenerla protegida fuera de nuestro ordenador. Muchas empresas alquilan servidores cloud (en la nube) para tener en ellos toda la valiosa información de la empresa, utilizándola cuando quieran y realizando el propio servidor copias de seguridad.

- Servidor de archivos: es aquel que almacena y sirve ficheros a equipos de una red.
- Servidor de directorio activo/dominio: es el que mantiene la información sobre los usuarios, equipos y grupos de una red.
- Servidor de impresión: se encarga de servir impresoras a los equipos cliente y poner en la cola los trabajos de impresión que estos generan.
- Servidor de fax: gestiona el envío, recepción y almacenamiento de los faxes.
- Servidor de proxy. Se utilizan para administrar una red de ordenadores, permitiendo el acceso o no a la red de los clientes. Suelen incluir protección de la red como por ejemplo un firewall (cortafuegos).
- Servidor DHCP: este dispone de un rango de direcciones con el cual, asigna automáticamente los parámetros de configuración de red IP a las máquinas cliente cuando estas realizan una solicitud.
- Servidor de telefonía: realiza funciones relacionadas con la telefonía, como es la de contestador automático, realizando las funciones de un sistema interactivo para la respuesta de la voz, almacenando los mensajes de voz, encaminando las llamadas y controlando también la red o el Internet, p. ej., la entrada excesiva de la voz sobre IP (VoIP), etc.
- Servidor de uso: realiza la parte lógica de la informática o del negocio de un uso del cliente, aceptando las instrucciones para que se realicen las operaciones de un sitio de trabajo y sirviendo los resultados a su vez al sitio de trabajo, mientras que el sitio de trabajo realiza la interfaz operadora o la porción del GUI del proceso (es decir, la lógica de la presentación) que se requiere para trabajar correctamente.
- Servidor de seguridad: Tiene software especializado para detener intrusiones maliciosas, normalmente tienen antivirus, antispyware, antimalware, además de contar con cortafuegos redundantes de diversos niveles y/o capas para evitar ataques, los servidores de seguridad varían dependiendo de su utilización e importancia.
- Servidor de reserva: tiene el software de reserva de la red instalado y tiene cantidades grandes de almacenamiento de la red en discos duros u otras formas del almacenamiento (cinta, etc.) disponibles para que se utilice con el fin de asegurarse de que la pérdida de un servidor principal no afecte a la red. Esta técnica también es denominada clustering.
- etc.

Los servidores deben estar siempre encendidos y disponibles para ofrecer sus servicios, con lo cual la instalación de estos debe ser segura ante apagones eléctricos y asumiendo un alto consumo eléctrico al tratarse de aparatos de gran potencia.

Dell PowerEdge T410

Para el servidor dedicado a nuestro proyecto, se utilizara el Dell PowerEdge T410²⁵, de la casa Dell, es un servidor de dos núcleos, según el sitio web oficial de Dell,

El Dell™ PowerEdge™ T410 es el servidor en torre de 2 sockets flexible y de alto valor de Dell, diseñado para abordar las amplias necesidades de las empresas en crecimiento y las oficinas remotas corporativas.

Rendimiento de procesamiento, memoria y almacenamiento avanzados
Diseño silencioso y compacto inspirado en los clientes Administración de sistemas simplificada

El sitio también añade en el apartado "Características":

Rendimiento de procesamiento, memoria y almacenamiento avanzados Compacto, flexible, silencioso y fácil de usar, el Dell PowerEdge T410 aprovecha las tecnologías de última generación de Dell para brindar un rendimiento confiable.

El rendimiento avanzado de los procesadores de la serie Intel® Xeon®, la memoria DDR3 y hasta seis discos duros de 3,5" (o de 2,5" de manera opcional) satisfacen las necesidades de las diversas aplicaciones de una empresa en crecimiento.

El Dell PowerEdge T410 cuenta con una acústica silenciosa, fuentes de alimentación redundantes disponibles y un medio de diagnóstico excelente a través de una LCD interactiva disponible.

Diseño compacto inspirado en los clientes La profundidad del chasis de 24 pulgadas da una flexibilidad óptima para la implementación en empresas con restricciones de espacio.

El Dell PowerEdge T410 cuenta con características comunes de sistema e imagen de primera clase y una facilidad de uso inspirada en los clientes. Tiene un diseño lógico e impecable y la fuente de alimentación está ubicada de manera que la instalación se pueda realizar rápidamente y la reimplementación sea más simple.

El Dell PowerEdge T410 también incluye un chasis de vanguardia, además de portadoras y ventiladores mejorados para los discos duros, que brindan una confiabilidad excepcional, operaciones silenciosas y fluidas y facilidad de uso.

Administración simplificada de sistemas La nueva generación del conjunto de herramientas Dell OpenManage™ permite realizar operaciones eficientes y ejecuta comandos basados en estándares, por lo que se integra con los sistemas existentes para lograr un control eficaz.

OpenManage reduce las tareas de varios pasos a un único paso y define el marco para las operaciones automatizadas. Las funciones incluyen una

²⁵ "Dell PowerEdge T410", Disponible: <http://www1.la.dell.com/content/products/productdetails.aspx/server-poweredge-t410?c=mx&l=es&s=corp&cs=mxcorp1> Consultado el: 5 de Marzo de 2019.

interfaz de consola única que da la posibilidad de conectar tecnologías de otros socios para llevar a cabo una administración eficaz de software y sistemas.

Este ordenador tiene las siguientes especificaciones:

Característica	Especificaciones técnicas
Factor de forma	Torre
Procesadores	Procesadores Intel® Xeon® serie 5500 y 5600 de cuatro o seis núcleos
Sockets del procesador	2
Bus frontal lateral o HyperTransport	Intel® QuickPath Interconnect (QPI)
Caché	4 MB y 8 MB
Chipset	Intel® 5500
Memoria ¹	Hasta 128 GB (8 ranuras DIMM): 1GB/2GB/4GB/8GB/16GB DDR3 hasta 1333MHz
Ranuras de E/S	5 ranuras PCIe G2: Cuatro ranuras x8 Una ranura x16
Controladora RAID	Interna: PERC H200 (6 Gb/s) PERC H700 (6 Gb/s) con 512 MB de memoria caché respaldada por la batería; 512 MB, 1 G de memoria caché no volátil respaldada por la batería SAS 6/iR PERC 6/i con 256 MB de memoria caché respaldada por la batería PERC S100 (basada en software) PERC S300 (basada en software) Externa: PERC H800 (6 Gb/s) con 512 MB de memoria caché respaldada por la batería; 512 MB, 1 G de memoria caché no volátil respaldada por la batería PERC 6/E con 256 MB o 512 MB de memoria caché respaldada por batería HBA externos (no RAID): HBA SAS de 6 Gbps HBA SAS 5/E HBA SCSI con PCIe LSI2032
Compartimientos de unidades	6 discos duros cableados de 3.5" O 6 discos duros intercambiables en caliente de 3.5" O 6 discos duros intercambiables en caliente de 2.5" Y 2 compartimientos de unidades de 5,25" para DVD-ROM, DVD+/-RW o TBU
Capacidad máxima de almacenamiento interno	Hasta 12 TB
Discos duros ¹	Opciones de disco duro de conexión en marcha: Unidades de estado sólido SATA de 2.5", SAS (10.000) SAS (15.000, 10.000), SAS nearline (7.200), SATA (7.200) de 3.5" Opciones de disco duro con cable: SAS de 3.5" (15.000, 10.000), SATA (7.200)

Figura 2.1: Características del ordenador

Comunicaciones	Gigabit Ethernet de doble puerto Broadcom® NetXtreme® II 5716 integrada Adaptador Intel® Adaptador Intel® Gigabit ET de dos puertos para servidor y adaptador Intel® Gigabit ET de cuatro puertos para servidor NIC Intel de 1 GbE y un puerto NIC Intel de 1 GbE y dos puertos NIC Intel de 1 GbE y cuatro puertos NIC Broadcom de 1 GbE y dos puertos
Fuente de alimentación	PSU no redundante, 525 W (80+) PSU redundante opcional, 580 W (80+ GOLD)
Disponibilidad	Paquete de cuatro LED de diagnóstico o LCD de diagnóstico con chasis para disco duro intercambiable en caliente; TPM, discos duros intercambiables en caliente opcionales; fuente de alimentación redundante intercambiable en caliente opcional; controladora PERC 6/i RAID opcional con memoria caché respaldada por batería; chasis que no requiere herramientas
Vídeo	Matrox® G200 integrado
Administración remota	iDRAC6 Enterprise opcional, iDRAC6 Express
Administración de sistemas	BMC, cumple con IPMI 2.0 Dell™ OpenManage™ Microsoft® System Center Essential (SCE) 2010 v2
Sistemas operativos	Microsoft® Windows® Small Business Server 2011 Microsoft® Windows® Small Business Server 2008 Microsoft® Windows Server® 2008 SP2, x86/x64 (x64 incluye Hyper-V™) Microsoft® Windows Server® 2008 R2, x64 (incluye Hyper-V™ v2) Novell® SUSE® Linux® Enterprise Server Red Hat® Enterprise Linux®
Aplicaciones de bases de datos destacadas	Para obtener más información sobre las versiones y agregados específicos, visite www.dell.com/OSsupport . Soluciones de Microsoft® SQL Server® (consulte Dell.com/SQL)

¹ GB equivale a mil millones de bytes, y 1 TB equivale a 1 billón de bytes. La capacidad real depende del material que se haya cargado previamente y del entorno operativo, lo cual puede determinar que dicha capacidad sea menor.

Figura 2.2: Especificaciones del ordenador

Este ordenador ya está en uso en Administración Académica, y ya ha estado en uso, de manera que se tiene plena confianza de su funcionamiento correcto, así como su eficacia y eficiencia.

2.2.9.2. Escáner

El escáner²⁶ es un dispositivo que captura la imagen de documentos de distintos tipos, para la creación posterior de un archivo digital con la misma información, su funcionamiento fundamental es básicamente como el de una cámara fotográfica, simplemente captura la imagen cruda, con la diferencia de que el scanner lo hace con carencia de ambiente al rededor de la imagen, y evitando luces externas y otro tipo

²⁶ "Escaner"; Disponible: <https://definicion.de/escaner/>, <https://tecnologia-facil.com/que-es-que-es-un-scanner/>; Consultado el: 5 de Marzo de 2019.

de influencias, que perturben la calidad original de la imagen, obteniendo una copia casi idéntica del documento en digital.

Tipos de escáner

Existen diferentes tipos de scanner²⁷, se citan a continuación.

- **Escáneres planos (de mesa)** Los escáneres planos son el tipo de escáner más conocido y vendido, y por buenas razones. Son versátiles, fáciles de manejar, y con una amplia disponibilidad. Su popularidad para la publicación en la Web abrió un gran mercado, forzando los precios de las unidades a nivel de entrada por debajo de los \$100. En el otro extremo, las unidades profesionales para el mercado de gráficos ahora compiten con los escáneres de tambor en cuanto a calidad. Todos utilizan la misma tecnología básica, en la cual un sensor de luz (por lo general un CCD) y una fuente de luz, ambos montados sobre un brazo móvil, pasan sobre el documento, que está fijo sobre una placa de vidrio. Algunos modelos poseen manipuladores de documento automáticos (ADH), que pueden aumentar el rendimiento y disminuir la fatiga del operador en el caso de grupos de documentos uniformes que se encuentran en condiciones razonablemente buenas. Una variante especializada del escáner plano es el escáner de libros de trayectoria aérea, en el cual la fuente de luz, la selección de sensores y la óptica son trasladados a un ensamble de brazo de trayectoria aérea bajo el cual puede colocarse un volumen encuadernado con las hojas hacia arriba, para ser escaneado.
- **Escáneres con alimentador de hojas** Los escáneres con alimentador de hojas utilizan la misma tecnología básica que los escáneres planos, pero maximizan el rendimiento, por lo general a expensas de la calidad. Diseñados generalmente para entornos de negocios de grandes volúmenes, típicamente escanean en blanco y negro o escala de grises con resoluciones relativamente bajas. Se espera que los documentos sean de un tamaño uniforme y con una solidez suficiente para soportar una manipulación bastante brusca, aunque los mecanismos de transporte de algunos modelos más nuevos reducen la tensión. Ya sea que utilicen un transporte de rodillos, cinta, tambor o de vacío, el sensor de luz y la fuente de luz permanecen fijos mientras que se mueve el documento. Una subclase de escáneres con alimentador de hojas son los modelos de pie específicamente diseñados para los documentos de gran formato, como los mapas y los planos arquitectónicos.
- **Escáneres de tambor** Los escáneres de tambor producen escaneados con la mayor resolución y calidad que cualquier otro tipo de escáner, pero esto tiene su precio. Además de su costo, los escáneres de tambor son lentos, no son indicados para documentos de papel quebradizo y requieren un alto nivel de habilidad por parte del operador. Por eso típicamente se los encuentra en agencias de servicios que satisfacen las necesidades del mercado de pre-impresión a color.
- **Escáneres para microfilm** Los escáneres para microfilm son dispositivos altamente especializados para digitalizar películas en rollo, microfichas y tarjetas de apertura. Puede ser difícil obtener una calidad buena y consistente en un

²⁷ "Tipos de Escaner"; Disponible: <https://definicion.de/escaner/>, <https://tecnologia-facil.com/que-es/que-es-un-scanner/>; Consultado el: 5 de Marzo de 2019.

escáner para microfilm debido a que los mismos pueden tener un funcionamiento complejo, la calidad y condición de la película puede variar, y debido a que ofrecen capacidad de mejora mínima. Sólo unas pocas empresas fabrican escáneres para microfilm, y la falta de competencia contribuye al alto costo de estos dispositivos.

- **Escáneres para diapositivas** Los escáneres para diapositivas se utilizan para digitalizar colecciones de diapositivas ya existentes como así también materiales fotográficos intermedios de objetos tridimensionales y documentos que no son adecuados para el escaneado directo. El uso de medios transparentes por lo general entrega una imagen con un buen rango dinámico, pero, dependiendo del tamaño del original, la resolución puede ser insuficiente para algunas necesidades. El rendimiento puede ser lento.
- **Cámaras digitales** Las cámaras digitales combinan un escáner con óptica de cámara para formar una herramienta versátil que puede producir imágenes de calidad superior. A pesar de ser más lentas y más difíciles de utilizar que los escáneres planos, las cámaras digitales se adaptan a una amplia variedad de documentos y objetos. Se pueden capturar en forma segura los materiales más frágiles, aunque la necesidad de proporcionar iluminación externa significa que el daño causado por la luz puede ser una preocupación. La tecnología de las cámaras digitales continúa mejorando, ayudada por el creciente mercado de consumidores.
- **De mano:** Estos dispositivos son muy económicos y prácticos. Normalmente poseen su propia tarjeta, en caso que no sea así pueden ser conectados a las impresoras de la computadora. Su resolución es muy baja, como máximo cuatrocientos PPP. Para obtener buenos resultados es necesario mucha práctica.
- **Digitalizadores de video:** Estas son usadas con el fin de crear documentos que posean aplicaciones informáticas. Es por ello que fueron producidas las tarjetas que poseen la capacidad de digitalizar imágenes. Además dichas tarjetas tienen la competencia de convertir imágenes, sin importar cual sea su fuente, en imágenes electrónicas analógicas.
- **De transparencias:** Esta función también puede ser realizada por escáneres de tipo plano, sin embargo existen estos dispositivos especializados. Son empleadas para captar películas transparentes, siendo estos en blanco y negro o a color, positivo o negativo.

Scanner HP Scanjet Enterprise Flow N9120

Este es la marca y modelo de escáner usado en Administración Académica, para el escaneo de expedientes físicos de estudiantes.

Según el sitio web:

El escáner departamental A3 más rápido y seguro acelera el avance de sus negocios. Las funciones de alto rendimiento permiten aumentar la productividad de su oficina.^{1,3}

Recursos

Envío digital de alto rendimiento

Utilice el OCR integrado para escanear de manera rápida y sencilla en texto editable, archivos PDF cifrados y otros tipos de archivos.

Logre más. Más rápido que nunca. 1

Administre grandes volúmenes a velocidades impresionantes de hasta 120 ppm/240 ipm.1

Capture cada página con facilidad, incluso pilas de medios mixtos, con HP EveryPage y un sensor ultrasónico.4

Confíe siempre en este escáner: recomendado para un máximo de 20000 páginas al día.

Administre directamente los trabajos con el panel de control táctil de 8 pulgadas (20,3 cm) y teclado extraíble completo.

El dispositivo de escaneo más seguro del mundo3

La arquitectura de firmware HP FutureSmart puede escalarse en todos los dispositivos y mejorar con el tiempo mediante nuevas funciones.

La plataforma OXP de HP ofrece soluciones previamente configuradas o personalizadas para procesos como la seguridad, la contabilidad, etc.

Centralice el control de su entorno de escaneo con HP Web Jetadmin y mejore la eficacia de su negocio.2

El disco duro seguro y de alto rendimiento de HP lo ayuda a proteger los datos confidenciales. Elimine archivos de forma segura.

Integre fácilmente soluciones de flujo de trabajo

Haga más con los datos mediante las diferentes opciones de destino de escaneos: carpetas, sitios FTP, correo electrónico, unidades USB, Share-Point, etc.

Defina perfiles de escaneo, escanee en la nube y en otros destinos y mejore las imágenes de forma automática.

El OCR integrado crea archivos con función de búsqueda que se pueden escanear en varios destinos mediante la creación automática y avanzada de imágenes.

Escanee imágenes directamente a aplicaciones, con las completas opciones incluidas: TWAIN e ISIS®.

Sus especificaciones técnicas según el sitio web son las siguientes:

Tipo de escáner

Cama plana, alimentador automático de documentos (ADF)

Recursos de envío digital

Escaneo para correo electrónico; Guardar en carpeta de red; Guardar en unidad USB; Enviar para FTP; Enviar para fax de LAN; Enviar para fax

de Internet; Libreta de direcciones locales; SMTP sobre SSL

Resolución de escaneo, óptica Hasta 600 ppi

Resolución de escaneado mejorada Hasta 600 ppi

Ciclo de trabajo (diario) Ciclo de trabajo diario recomendado: 20.000 páginas

Profundidad en bits 24 bits externa, 48 bits interna

Niveles de escala de grises 256

Tamaño de escaneo, máximo 297 x 432 mm

Tipos de medios admitidos AAD: Papel (normal, inyección de tinta, inyección de tinta para folletos/brillante), papel fotográfico; Escáner de superficie plana: Todos los medios del AAD, sobres, etiquetas, tarjetas, libros

Pesos de medios, alimentador de documentos automático (ADF) admitido 45 a 199 g/m²

Formato del archivo de digitalización PDF, JPEG, TIFF, MTIFF, XPS, PDF/A, TEXTO (OCR), TEXTO Unicode (OCR), RTF (OCR), PDF con búsqueda (OCR), PDF/A con búsqueda (OCR), HTML (OCR), CSV (OCR); Escaneo a USB de fácil acceso: PDF, JPEG, TIFF, MTIFF, XPS, PDF/A, TEXTO (OCR), TEXTO Unicode (OCR), RTF (OCR), PDF con búsqueda (OCR), PDF/A con búsqueda (OCR), HTML (OCR), CSV (OCR); Para HP Scan: PDF, JPEG, PNG, BMP, TIF, Texto (.txt), texto enriquecido (.rtf), PDF con función de búsqueda (.pdf), PDF/A (.pdf); Para HP Easy Scan: TIFF, PNG, JPEG, JPEG-2000, PDF, PDF con función de búsqueda, RTF, TXT; Para Linux: JPEG, PDF, PNG, PNM, PostScript, TEXT, TIFF

Modos de entrada de digitalización Aplicaciones del panel frontal: Correo electrónico; Guardar en carpeta de red; Guardar en USB; Guardar en memoria de dispositivo; Aplicaciones de la plataforma Open Extensibility Platform (OXF); escaneo con software mediante HP Scan; aplicación de usuario a través de los controladores profesionales TWAIN, ISIS, WIA o Kofax VirtualRescan (VRS)

Funciones avanzadas del escáner Optimizar texto/imagen; Ajustes de imagen; Trabajos combinados; Configuración de calidad de salida; Resolución de escaneo seleccionable de 75 a 600 dpi; Detección automática

de color; Borrado de bordes; Notificación de trabajo; Supresión de páginas en blanco; HP Quick Sets; HP EveryPage; Orientación automática; Reconocimiento óptico de caracteres (OCR), recorte automático para página; Tonalidad automática

Disco duro

Disco duro seguro y de alto rendimiento HP estándar de 500 GB

Panel de control

Pantalla LCD XGA 1024 x 768 Tecnología táctil ligera que admite interacciones basadas en gestos Control de ajuste de brillo de pantalla Ícono de inicio para volver rápidamente a la interfaz de usuario de la pantalla de inicio Teclado virtual y teclado físico

2.2.9.3. UPS (Uninterrupted Power Supply) o SAI (Sistema de Alimentación ininterrumpida)

Los UPS²⁸, son dispositivos que se encargan de proveer una alimentación eléctrica constante e ininterrumpida a dispositivos, principalmente computadoras, para evitar accidentes y errores producidos por factores externos, como apagones, cortos circuitos, etc.

Se altamente recomendado usar esta clase de dispositivos para usar computadoras de escritorio, principalmente si se va a usar en sesiones de muchas horas, además altamente recomendado su uso en servidores de red, que deben estar disponibles las 24 horas al día para ofrecer sus servicios al público.

A continuación se listan diferentes tipos de UPS²⁹:

- Los UPS Standby

Estos no-breaks se caracterizan por presentar en modo red la tensión y frecuencia en la salida igual a la de la red eléctrica, o sea, la potencia es transferida directamente de la red a la salida. A falta de red eléctrica o de variación de la tensión o frecuencia fuera de los niveles seguros especificados, el inversor entra en operación alimentando a través de las baterías internas los equipos conectados en la salida.

- Los UPS interactivos

También se caracterizan por presentar en modo red la frecuencia en la salida igual a la de la red eléctrica, pero poseen un estabilizador interno, donde la tensión de salida es corregida a un valor seguro si la tensión de entrada varía para valores que puedan perjudicar a los equipos conectados. En otras palabras, entre la entrada y la salida del producto, se tiene un estabilizador basado en un

²⁸ "Uninterrupted Power Supply"; Disponible: <https://administracioninformatica.wordpress.com/2012/08/31/definicion-de-UPS-y-su-funcion/>; Consultado el: 5 de Marzo de 2019.

²⁹ "Tipos de UPS"; Disponible: <http://www.informaticamoderna.com/UPS.htm>; Consultado el: 5 de Marzo de 2019.

transformador responsable por la corrección de la tensión de salida. A falta de la red eléctrica, tensión o frecuencia fuera de los niveles seguros especificados, el inversor entra en operación alimentando los equipos.

- Los UPS en línea de doble conversión
Proporcionan el mayor nivel de protección aislando los equipos de la red eléctrica a través de convertidores de potencia. El término doble conversión significa que el producto realiza en modo red la conversión de la corriente alterna de la red eléctrica a corriente continua (a través del rectificador PFC) y luego de corriente continua a corriente alterna (a través del inversor). Así, la tensión y frecuencia de salida no son totalmente dependientes de la entrada y como el inversor está alimentando la carga en todo momento, el tiempo de transferencia es nulo en la ocurrencia de un blackout. A falta de la red eléctrica, la energía entonces se transfiere de las baterías para alimentar las cargas.
- Standby on-line hybrid
La UPS Standby on-line hybrid es la topología utilizada para muchas UPS de alrededor de 10kVA que se etiquetan "on line?". La conversión DC a DC de la batería se enciende cuando se detecta un fallo de alimentación de CA, al igual que en un SAI de Standby. Debido a los condensadores en el combinator DC, el SAI Standby on-line hybrid no presenta ningún tiempo de transferencia durante una falla de energía AC. Este diseño es a veces equipado con un interruptor de transferencia adicional para la derivación durante una avería o la sobrecarga. La Figura 3 ilustra esta topología.
- Standby-Ferro
La UPS standby-Ferro una vez fue la forma dominante de UPS en el rango 3-15kVA. Este diseño depende de un transformador de saturación especial que tiene tres bobinados. El camino de alimentación primaria es de entrada de CA, a través de un interruptor de transferencia, luego través del transformador, y a la salida. En el caso de un corte de corriente, se abre el interruptor de transferencia, y el inversor recoge la carga de salida.

2.2.9.4. Ordenador Cliente

El ordenador cliente³⁰ es todo aquel que una red, envía solicitudes de uno o varios recursos al servidor, es decir, solicita a este último sus servicios, que son respondidos por el servidor mediante retransmitiendo la información solicitada a través de la red, usando la dirección IP.

Especificaciones mínimas de el ordenador cliente

- Procesador IntelTM i3 de cuatro núcleos.
- 4 GB de RAM.
- Disco duro de 500 GB, preferiblemente de 1 TB.
- Windows 7, 8, 8.1 y 10, MAC OS X o cualquier distribución de Linux.

³⁰ "Ordenador Cliente"; Disponible: <https://blog.infranetworking.com/modelo-cliente-servidor/>, <https://definicion.de/cliente-servidor/>; Consultado el: 5 de Marzo de 2019.

- Conexión a internet.

Estas especificaciones se cumplen y superan por la mayoría de los ordenadores instalados en Administración Académica, principalmente el ordenador previamente usado para escanear los expedientes físicos.

2.3. Marco Legal

La Universidad de El Salvador es un centro de educación superior, aparte es el único campus estatal de la nación. Por ende, todas las sub-entidades de esta manejan muy importante información, que no debe tratarse a la ligera, y que además debe ser legal y legítima para ser válida al estado y forjar profesionales bajo la ley. Existen una serie de leyes nacionales y universitarias que rigen como se debe tratar la documentación y administración, y todo lo relacionado a estos.

A continuación, se presentan las leyes que rigen lo mencionado a Administración Académica en la Facultad y a la universidad en general.

2.3.1. Ley de Acceso a la Información Pública

Según el sitio www.gobiernoabierto.gov.sv:

La Ley de Acceso a la Información Pública (LAIP)³¹ es la ley nacional que establece el derecho de acceso a la información pública y dice que toda persona tiene derecho a solicitar y recibir información generada, administrada o en poder de las instituciones públicas y demás entes obligados de manera oportuna y veraz, sin que te pregunten para qué quieres la información.

Esta ley promueve el acceso a la información mediante procedimientos sencillos y rápidos, propicia la transparencia mediante la difusión de la información, impulsa la rendición de cuentas y promueve la participación ciudadana y el control en la gestión de los recursos que realizan las instituciones del Estado.

Esta ley se encarga de permitir al público en general investigar información pertinente de instituciones estatales debido a su calidad pública, a su vez rige los procedimientos estrictamente establecidos para facilitar al público su búsqueda.

La ley se fundamenta en el derecho fundamental del público de saber todo sobre las instituciones que administran información pertinente al estado.

El sitio web antes citado también expresa:

¿Quiénes están obligados a dar información?

Todas las entidades del Estado: Órgano Ejecutivo, Legislativo y Judicial y sus dependencias, instituciones autónomas (como el FOVIAL, FONAVIPO, FOSALUD, INDES, ISSS, CEL, etc.) y las 262 municipalidades. También se incluyen sociedades de economía mixta, personas naturales

³¹ *LEY DE ACCESO A LA INFORMACIÓN PÚBLICA. DECRETO LEGISLATIVO No. 534 DE FECHA 2 DE DICIEMBRE DE 2010. PUBLICADA EN EL DIARIO OFICIAL TOMO 371, No. 70, DEL 8 DE ABRIL DE 2011. En: <https://www.asamblea.gov.sv>. Disponible en: <https://www.asamblea.gov.sv/decretos/details/493> Consultado el: 5 de marzo de 2019*

o jurídicas que manejen recursos o información pública, como SERTRACEN o los DUICENTROS, porque llevan a cabo servicios públicos.

Incluso, puedes solicitar información sobre los fondos provenientes de convenios o tratados que tenga El Salvador con otros países u organismos internacionales, siempre y cuando en estos documentos no aparezca una cláusula de acceso a la información. En consecuencia, todos los y las servidoras públicas, dentro o fuera del territorio de la República, están obligados al cumplimiento de la LAIP.

También incluye:

Tipos de información: pública, reservada y confidencial La información pública es aquella en poder de los entes obligados contenida en documentos, archivos, datos, bases de datos, comunicaciones y todo tipo de registros que documenten el ejercicio de sus facultades o actividades, que consten en cualquier medio, ya sea impreso, óptico o electrónico, independientemente de su fuente, fecha de elaboración, y que no sea confidencial. Dentro de la información pública se encuentra un subconjunto de información denominado "información oficiosa", la cual debe de ser publicada de forma inmediata sin que ninguna persona lo solicite. Esta información puede estar impresa o colgada en los sitios web de las instituciones y deben entregártela en el mismo momento en que lo solicites.

La información reservada es la información pública cuyo acceso se restringe de manera expresa, según lo estipula la LAIP, en razón de un interés general durante un periodo determinado y por causas justificadas. Por ejemplo, los planes militares secretos, las negociaciones internacionales o cualquier tipo de negociación o discusión que se tenga, mientras no se adopte una decisión definitiva. O toda aquella información que esté relacionada con la investigación o persecución de actos ilícitos o que genere una ventaja indebida en perjuicio de un tercero. Si la información que solicitas es reservada, puedes solicitar una versión pública. Esta es un documento en el cual se tacharán todos los datos que no puedes ver y te permitirá acceder al resto de información pública.

La información confidencial es la información privada en poder del Estado cuyo acceso público se prohíbe por mandato constitucional o legal en razón de un interés personal jurídicamente protegido. Es decir, la información referente a la intimidad personal y familiar, al honor y propia imagen, así como archivos médicos cuya divulgación constituiría una invasión a la privacidad de la persona. A esta información solo tendrán acceso las personas que son dueñas de ella. Dentro de la información confidencial están los datos personales la cual es la información privada de una persona, como por ejemplo su nacionalidad, domicilio, patrimonio, dirección electrónica, número de teléfono o cualquier otra parecida.

Artículos de Interés

Según el artículo 1 de esta ley:

Art. 1. La presente ley tiene como objeto garantizar el derecho de acceso de toda persona a la información pública, a fin de contribuir con la transparencia de las actuaciones de las instituciones del Estado.

El Artículo 2 dice:

Art. 2. Toda persona tiene derecho a solicitar y recibir información generada, administrada o en poder de las instituciones públicas y demás entes obligados de manera oportuna y veraz, sin sustentar interés o motivación alguna.

Los fines de esta ley se citan en el artículo 3:

Art. 3. Son fines de esta ley:

- Facilitar a toda persona el derecho de acceso a la información pública mediante procedimientos sencillos y expeditos.
- Propiciar la transparencia de la gestión pública mediante la difusión de la información que generen los entes obligados.
- Impulsar la rendición de cuentas de las instituciones y dependencias públicas.
- Promoción de la participación ciudadana en el control de la gestión gubernamental y la fiscalización ciudadana al ejercicio de la función pública.
- Modernizar la organización de la información pública.
- Promover la eficiencia de las instituciones públicas.
- Promover el uso de las tecnologías de la información y comunicación y la implementación del gobierno electrónico.
- Proteger los datos personales en posesión de los entes obligados y garantizar su exactitud.
- Contribuir a la prevención y combate de la corrupción.
- Fomentar la cultura de transparencia.
- Facilitar la participación de los ciudadanos en los procesos de toma de decisiones concernientes a los asuntos públicos.

Instituciones bajo esta ley.

Según el artículo 7:

Art. 7. Están obligados al cumplimiento de esta ley los órganos del Estado, sus dependencias, las instituciones autónomas, las municipalidades o cualquier otra entidad u organismo que administre recursos públicos, bienes del Estado o ejecute actos de la administración pública en general.

Se incluye dentro de los recursos públicos aquellos fondos provenientes de Convenios o Tratados que celebre el Estado con otros Estados o con Organismos Internacionales, a menos que el Convenio o Tratado determine otro régimen de acceso a la información. También están obligadas por esta ley las sociedades de economía mixta y las personas naturales o jurídicas que manejen recursos o información pública o ejecuten actos de la función estatal, nacional o local tales como las contrataciones públicas, concesiones de obras o servicios públicos. El ámbito de la obligación de estos entes se limita a permitir el acceso a la información concerniente a la administración de los fondos o información pública otorgados y a la función pública conferida, en su caso. En consecuencia, todos los servidores públicos, dentro o fuera del territorio de la República, y las personas que laboren en las entidades mencionadas en este artículo, están obligados al cumplimiento de la presente ley.

De manera que la Universidad de El Salvador está bajo esta ley según el artículo por defecto, y por ende, todas sus instituciones dependientes en todas sus facultades, incluyendo Administración Académica en la Facultad Multidisciplinaria Oriental.

La información completa se puede leer en la ley completa, cuyo enlace está en la subsección "Enlaces".

2.3.2. Ley de Archivo de la Nación

El archivo general de la nación³² es una ley nacional que se encarga de administrar la documentación cultural, histórica, académica, etc, que concierne al estado en general, de manera de no perder información importante como estudios realizados por persona, acuerdos legislativos, etc.

Esta ley es concerniente a la Alma Mater y en particular, a Administración Académica, ya que estas administran expedientes estudiantiles, así como acuerdos de junta directiva, entre otra información importante, la formación de profesionales es una tarea de suma importancia, y la documentación que respalda al profesional debe ser protegida y respaldada.

El artículo dos de la "Ley de Archivo General de la Nación" dice:

1. Conservar, restaurar, clasificar, describir, investigar e inventariar, los manuscritos históricos y administrativos que datan desde el año de 1660 hasta 1930; así como todos aquellos documentos que no perteneciendo a esa época, por su propia naturaleza así lo ameritaran;
2. Proporcionar a entidades públicas o privadas, la asistencia que le sea solicitada para la conservación, restauración y clasificación de documentos;
3. Elaborar el proyecto del Reglamento para su respectiva aprobación;

³² *DECRETO N° 316. Fecha de Emisión: 15/01/1985. En: <https://www.asamblea.gov.sv>. Disponible en: <http://www.transparencia.ues.edu.sv/sites/default/files/PDF/LEY%20DE%20ARCHIVO.pdf>. Consultado en: 5 de marzo de 2019*

4. Cumplir con las demás atribuciones y funciones que establece esta ley y el reglamento.

La Universidad de El Salvador, y todas sus instituciones, están bajo esta ley, así como otras entidades autóctonas y estatales, según el artículo 7:

Art. 7.- Todos aquellos documentos históricos que de conformidad al Art. 63 de la Constitución, formen parte del tesoro cultural salvadoreño, y que se encuentren en cualquier Institución del Gobierno, en las Alcaldías Municipales o en Instituciones Oficiales Autónomas, deberán ser entregados al Archivo General de la Nación, previo el trámite respectivo y el descargo correspondiente.

Documentación cultural, histórica, artística, académica, etc; proveniente de la Alma Mater, tienen que, según este artículo, ser entregados al Archivo General de la Nación, y ser conservados con los conocimientos archivísticos necesarios.

Otros artículos pertinentes a nuestro proyecto:

Art. 14.- El Archivo General de la nación, deberá mantener intercambio de documentos con otras instituciones u organismos que tengan relación con acontecimientos históricos en el país.

Art. 15.- El Archivo General de la Nación, a iniciativa propia o a petición de organismos interesados, deberá promover cursos de capacitación en técnicas de archivo.

El Archivo General de Nación, podrá dictar seminarios de actualización técnica a todo el personal que administra Archivos en los distintos organismos del Gobierno Central, Municipal y de Instituciones Oficiales Autónomas, a fin de unificar criterios en materia archivística. Podrá prestar asesoría técnica a todas aquellas instituciones religiosas que lo soliciten; a fin de poder preservar la cultura nacional, que se encuentra en los archivos de las mismas.

Art. 16.- Los Archivos del Gobierno Central, Municipal, así como los de las Instituciones Oficiales Autónomas, podrán solicitar la colaboración del Archivo General de la Nación, en la organización de sus fuentes documentales, a fin de implementar un sistema nacional de archivo.

Los Archivos del Gobierno Central, Municipal y de Instituciones Oficiales Autónomas y de la empresa privada, que hayan sido organizados con la asistencia técnica del Archivo General de la Nación, podrán seguir teniendo esa asistencia, si así lo solicitar en y en ese caso, respetarán las decisiones técnicas, que éste haya adoptado en materia archivística.

Art. 17.- Las certificaciones por servicios prestados en el Archivo General de la Nación; serán firmados por el Director, y en ausencia de éste, por el que le siguiere en jerarquía.

Estos son solo algunos de los artículos que conforman la ley, esta misma completa es descargable desde su página oficial en la sección "Enlaces".

2.3.3. Ley de Procedimientos Administrativos.

La Ley de Procedimientos Administrativos³³ Se trata de la ley nacional que define la validez de los procedimientos administrativos de las instituciones públicas y entidades autóctonas.

Los objetivos de esta ley se describen en el artículo 1:

Art. 1.- Las Disposiciones de la presente Ley tienen por objeto regular:

1. Los requisitos de validez y eficacia de las actuaciones administrativas de toda la Administración Pública;
2. Los derechos de los ciudadanos frente a la Administración Pública;
3. El régimen de responsabilidad patrimonial de la Administración Pública y de sus funcionarios; y,
4. El ejercicio de la potestad normativa, así como los principios y garantías del procedimiento administrativo sancionado.

El ámbito de esta ley está descrito en el artículo 2:

Art. 2.- La presente Ley se aplicará al Órgano Ejecutivo y sus dependencias, a las entidades autónomas y demás entidades públicas, aun cuando su Ley de creación se califique de carácter especial; y a las municipalidades, en cuanto a los actos administrativos definitivos o de trámite que emitan y a los procedimientos que desarrollen.

Asimismo, se aplicará a los Órganos Legislativo y Judicial, la Corte de Cuentas de la República, la Procuraduría General de la República, la Procuraduría para la Defensa de los Derechos Humanos, la Fiscalía General de la República, el Consejo Superior de Salud Pública, el Tribunal Supremo Electoral y, en general, a cualquier institución de carácter público, cuando excepcionalmente ejerza potestades sujetas al derecho administrativo.

Esta Ley será aplicable a los concesionarios de la Administración Pública.

Los requisitos generales de la Administración se describen en este siguiente artículo:

Art. 3.- La Administración Pública debe servir con objetividad a los intereses generales, y sus actuaciones están sujetas a los siguientes principios:

³³ *DECRETO N° 856. Fecha de Emisión: 15/01/1985. En: <https://www.asamblea.gov.sv>. Disponible en: <https://www.asamblea.gob.sv/sites/default/files/documents/decretos/8B52B09D-52EC-460C-9C7C-19AC897EB64D.pdf> Consultado en: 5 de marzo de 2019*

1. Legalidad: La Administración Pública actuará con pleno sometimiento al ordenamiento jurídico, de modo que solo puede hacer aquello que esté previsto expresamente en la Ley y en los términos en que ésta lo determine;

2. Proporcionalidad: Las actuaciones administrativas deben ser cualitativamente aptas e idóneas para alcanzar los fines previstos, restringidas en su intensidad a lo que resulte necesario para alcanzar tales fines y limitadas respecto a las personas cuyos derechos sea indispensable afectar para conseguirlos.

En este supuesto, deberá escogerse la alternativa que resulte menos gravosa para las personas y, en todo caso, el sacrificio de éstas debe guardar una relación razonable con la importancia del interés general que se trata de salvaguardar;

3. Antiformalismo: Ningún requisito formal que no sea esencial debe constituir un obstáculo que impida injustificadamente el inicio del procedimiento, su tramitación y su conclusión normal. Asimismo, la Administración debe interpretar los requisitos esenciales en el sentido que posibilite el acceso a los procedimientos y el pronunciamiento de una resolución de fondo;

4. Eficacia: La Administración, antes de rechazar el inicio del procedimiento o recurso, su conclusión anormal o la apertura de un incidente, debe procurar la reparación o subsanación de cualquier defecto que haya advertido, incluso sin necesidad de prevención al interesado;

5. Celeridad e Impulso de Oficio: Los procedimientos deben ser ágiles y con la menor dilación posible y serán impulsados de oficio cuando su naturaleza lo permita;

6. Economía: La actividad administrativa debe desarrollarse de manera que los interesados y la Administración incurran en el menor gasto posible, evitando la realización de trámites o la exigencia de requisitos innecesarios;

7. Coherencia: Las actuaciones administrativas serán congruentes con los antecedentes administrativos, salvo que por las razones que se expliciten por escrito y se motiven adecuadamente, sea pertinente en algún caso apartarse de ellos;

8. Verdad Material: Las actuaciones de la autoridad administrativa deberán ajustarse a la verdad material que resulte de los hechos, aun cuando no hayan sido alegados ni se deriven de pruebas propuestas por los interesados; y,

9. Buena fe: Todos los participantes en el procedimiento deben ajustar sus comportamientos a una conducta honesta, leal y conforme con las actuaciones que podrían esperarse de una persona correcta, la cual se presume respecto de todos los intervinientes.

En el artículo 6 de esta ley se habla sobre la documentación, que es pertinente para la Alma Mater y Administración Académica, ergo, nuestro proyecto:

Art. 6.-Sin perjuicio de lo establecido en la Ley de Acceso a la Información Pública, en cada oficina, los documentos y expedientes administrativos deberán ser agrupados, catalogados y archivados en función de sus características y formatos comunes, con el objeto de facilitar su manejo y comprensión.

Para agilizar la actuación administrativa, en las oficinas públicas deberán racionalizarse los trabajos burocráticos, procurando mecanizarlos y automatizarlos progresivamente.

Con la misma finalidad, cuando los motivos y fundamentos de las resoluciones sean idénticos, deberá mecanizarse la producción en serie de tales resoluciones, siempre que no se lesionen las garantías jurídicas de los interesados.

En lo referido a las actuaciones de los particulares, éstos podrán presentar la información solicitada por la Administración Pública en formularios oficiales, en copias, sistemas electrónicos en línea o mediante cualquier documento que respete el contenido íntegro y la estructura de dichos formularios, y que contenga los aspectos requeridos por la normativa aplicable.

El artículo 11 habla sobre la atención al público, cuyas normas son pertinentes para administración académica y otras oficinas que prestan servicios al público, dentro del campus:

Art. 11. - Los órganos administrativos deberán potenciar los más altos estándares de atención al ciudadano, para lo cual se atenderán a las siguientes normas:

1. No podrá negarse la atención al público antes del cumplimiento de la hora de cierre de la respectiva oficina;
2. El servicio de atención al público deberá prestarse de manera ininterrumpida, dentro del horario establecido;
3. Se deberá respetar el orden de atención de los ciudadanos y aplicar técnicas que eviten la formación de aglomeraciones;
4. Los servicios en línea podrán establecer horarios de atención ininterrumpida, de acuerdo a la capacidad económica y técnica del organismo de que se trate; y,
5. Deberá tratarse a los ciudadanos con el más alto índice de decoro y transparencia.

Los anteriores artículos son destacables para, tanto Administración Académica, como para el proyecto. Para más artículos, vease la ley completa en la subsección "Enlaces".

2.3.4. Reglamento de la Gestión Académico Administrativa de la Universidad de El Salvador

Reglamento de la Gestión Académico Administrativa de la Universidad de El Salvador³⁴, es el reglamento interno de la universidad, que se encarga, según su

³⁴ ACUERDO No. 106/2011-2013 (V). Disponible en: <http://www.jurisprudencia.ues.edu.sv/academica/documentos-de-la-Gestion-Academico-Administrativa-de-la-UES.pdf>. Consultado el: 5 de marzo de 2019

artículo 1:

El objeto del presente Reglamento es normar y desarrollar las disposiciones generales sobre la gestión, organización, administración y funcionamiento académico de la Universidad de El Salvador, así como los procedimientos, medidas y resoluciones académicas y administrativas necesarias para la buena marcha de la misma, desde los procesos de ingreso hasta el otorgamiento de los grados académicos ofrecidos por la Universidad, en concordancia con la Constitución de la República de El Salvador, Ley de Educación Superior y su Reglamento, Ley Orgánica de la Universidad de El Salvador y su Reglamento General y demás normativas pertinentes.

Todo procedimiento, trámite, organización, según el artículo, está regido por la ley en cuestión, esto, obviamente, incluye a Administración Académica, de manera que los expedientes físicos de los estudiantes, así como todos los procedimientos relacionados, están bajo esta ley o reglamento.

La administración académica de la facultad está bajo el ámbito de esta ley según el siguiente artículo:

Artículo 2. El presente Reglamento por su carácter general será de aplicación obligatoria en todas las unidades académicas de la Universidad de El Salvador, y no podrá ser contrariado por normas contenidas en otros Reglamentos específicos en materia académica-administrativa.

En el presente Reglamento, toda referencia a cargos, funciones o condiciones de personas, se entenderá indistintamente en género femenino y masculino.

Según el siguiente artículo, Administración Académica, así como otras entidades de la universidad, serán responsables de cumplir con la presente ley, como se lee a continuación:

Artículo 4. La administración y gestión Académico-Administrativa de la Universidad de El Salvador, estará bajo la responsabilidad de: a) Vicerrectoría Académica. b) Vicedecanatos. c) Secretaría de Asuntos Académicos. d) Comité de Ingreso Universitario. e) Comité de Ingreso Universitario de Facultad. f) Administraciones Académicas de las Facultades. Esta estructura tendrá bajo su responsabilidad los procesos curriculares, la planificación, organización, ejecución y funcionamiento de los procesos: de selección de aspirantes, ingreso, matrícula, traslado, retiro, activación, reingreso, evaluación, egreso y graduación de los profesionales formados en la Universidad de El Salvador, así como los procesos de incorporación de estudiantes graduados en universidades extranjeras y expedición de documentos.

Este reglamento define quien y de que se encarga la secretaría de asuntos internos:

Artículo 12. La Secretaría de Asuntos Académicos será la responsable de administrar y organizar en lo concerniente a los procesos Académico-Administrativos Universitarios, con base a la legislación Universitaria y Acuerdos que emanen del Consejo Superior Universitario.

Dependerá de la Vicerrectoría Académica, será dirigida por el Secretario de Asuntos Académicos quien será nombrado por el Rector de una terna propuesta por el Vicerrector Académico.

En el siguiente artículo se define la organización de la Secretaría de Asuntos Internos, que comprende la Administración Académica Central.

Artículo 13. La Secretaría de Asuntos Académicos, estará organizada internamente en las unidades siguientes: a) Administración Académica Central; b) Unidad Curricular; y c) Unidad de Ingreso Universitario.

El artículo 17 de este reglamento define que es y de que se encarga la Administración Académica Central:

Artículo 17. La Administración Académica Central; es la responsable de velar por la aplicación de las disposiciones generales sobre procedimientos, medidas y resoluciones académicas, contenidas en la Ley Orgánica, este Reglamento y demás Reglamentos pertinentes. La Administración Académica Central será dirigida por un Administrador Académico, quien dependerá jerárquicamente de la Secretaría de Asuntos Académicos, será nombrado por el Rector de una terna propuesta por el Secretario de Asuntos Académicos. Durará en sus funciones el mismo período del Rector, pudiendo ser nombrado un período más y continuara en su cargo hasta su nombramiento y toma de posesión de la persona que deba sustituirlo.

El siguiente artículo define los requisitos para optar por el puesto de administrador académico central:

REQUISITOS PARA SER NOMBRADO ADMINISTRADOR ACADÉMICO CENTRAL. Artículo 18. Para ser Administrador Académico Central, se requiere: a) Ser graduado de la Universidad de El Salvador, con no menos de 3 años en ejercicio profesional; b) De preferencia acreditar experiencia en procesos académicos y/o administrativos en la Universidad de El Salvador, o en otros contextos académicos nacionales o internacionales; c) Ser de reconocida y probada honorabilidad; y d) No haber sido sancionado disciplinariamente en la Universidad de El Salvador o en instancias judiciales fuera de la universidad, en los últimos cinco años, anteriores a su nombramiento.

Ahora en el artículo siguiente se define los objetivos y atribuciones de Administración Académica:

Artículo 19. La Administración Académica Central tendrá las siguientes atribuciones: a) Coordinar, controlar y evaluar las actividades y recursos que estén bajo su responsabilidad; b) Elaborar y proponer a la Secretaría de Asuntos Académicos los calendarios anuales: de Actividades Académico Administrativas y de Graduaciones; c) Coordinar y controlar las incorporaciones y reposiciones de títulos, actas y expedientes de graduación; d) Remitir al Secretario de Asuntos Académicos las certificaciones de calificaciones, Títulos y otras certificaciones o constancias de índole académico para su respectiva firma; e) Dictaminar sobre los asuntos académicos de

su competencia que requiera el Secretario de Asuntos Académicos; f) Coordinar con las Administraciones Académicas de las Facultades las acciones académicas y administrativas de activación, reingreso, retiro oficial, deserción, incorporaciones, cambio de carrera, traslados, egresos y toda actividad académica administrativa relacionada con las graduaciones; g) Coordinar, con las instancias pertinentes, el procesamiento de datos de las acciones académicas administrativas; h) Llevar un registro actualizado de las carreras a nivel de grado y posgrado que se imparten en la Universidad de El Salvador y de sus respectivos Planes y Programas de estudio; i) Gestionar, generar y disponer la estadística académica de la Universidad de El Salvador; j) Colaborar en la planificación y programación del Sistema Académico Administrativo de la Universidad; y k) Las demás atribuciones de índole académico administrativo que le asigne la Secretaria de Asuntos Académicos y la legislación universitaria. En ausencia del Secretario de Asuntos Académicos, asumirá dicho cargo el Vicerrector Académico.

Estos son algunos de los artículos del reglamento concerniente a Administración Académica, el reglamento entero es descargable desde el enlace en el apartado de enlaces.

Enlaces

- <http://api.gobiernoabierto.gob.sv/attachments/10/download>
- <https://www.asamblea.gob.sv/decretos/details/233>
- <https://www.asamblea.gob.sv/decretos/details/3389>
- https://academica.ues.edu.sv/documentos/reglamento_academica.pdf

2.4. Marco Conceptual

2.4.1. Definiciones

BACKUP: Copia de datos que se realiza en un medio de almacenamiento externo, tal como un disquete, cinta o CD-ROM. Como la información almacenada en el disco rígido es susceptible a determinados accidentes o pérdidas, es necesario tener una copia de respaldo actualizada. Es importante decidir en qué medio de almacenamiento se va a realizar el backup. Los disquetes son de uso común y de bajo costo, pero no demasiado seguros, por lo que es aconsejable realizar estas copias en CD-ROMs u otros medios.

BASE DE DATOS: (DataBase). Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos basándose en diferentes criterios, etc.

CAMPO: En algunas aplicaciones (como bases de datos) es el espacio reservado para introducir determinados datos asociados a una categoría de clasificación.

CASCADAS (CSS): (Cascading Style Sheets). Archivo de estilo externo definido para desplegar elementos específicos de HTML desde la versión 4.0 para solventar problemas de visualización. Su nombre deriva de la múltiple definición de estilos en cascada dentro de un solo archivo.

CLAVE DE ACCESO O LOGIN: Es una combinación de letras, números y signos que debe teclearse para obtener acceso a un programa o partes de un programa determinado, un terminal u ordenador personal, un punto en la red, etc.

EXPEDIENTE: (estudiantil) Conjunto de documentos físicos o digitalizados concernientes a un estudiante y que se mantienen en físico en el archivo de la Administración Académica.

GNU: Licencia Publica General. Software desarrollado para distribución sin fines de lucro. El proyecto GNU (GNU es un acrónimo recursivo para "Gnu No es Unix") comenzó en 1984 para desarrollar un sistema operativo tipo Unix completo, que fuera Software Libre. Las variantes del sistema operativo GNU, que utilizan el kernel Linux, son muy utilizadas. La gente a menudo se refiere erróneamente a estos sistemas como "Linux", cuando es más preciso y concreto llamarlos "GNU/Linux".

IDENTIFICACIÓN: Es un código único, utilizado por el sistema, para identificar unívocamente a un usuario en particular.

LINK: Referido a programación, cada uno de los enlaces de un módulo con las librerías que utiliza. Como segunda acepción, enlace desde Internet que permite acceder directamente de un Web a otro pulsando dos veces con el ratón sobre el texto marcado.

OPEN SOURCE: "Software Libre" se refiere a la libertad de los usuarios de correr, copiar, distribuir, estudiar, cambiar y mejorar el software. Open source podría

traducirse como 'código fuente abierto': un programa que ofrece al usuario la posibilidad de poder estudiarlo o modificarlo. Pero no sólo hace referencia al libre acceso al código fuente.

Las condiciones de distribución de un programa open source deben cumplir una serie de criterios. La intención de la 'Definición de open source' es establecer que esos criterios contengan la esencia de lo que los programadores quieren que signifique: que aseguren que los programas distribuidos con 'licencia open source' estarán disponibles para su continua revisión y mejora para que alcancen niveles de fiabilidad que no pueda conseguir ningún programa comercial 'cerrado'.

PARÁMETRO: Denominación de una cantidad de elementos de información que se utilizan en una rutina, subrutina, programa o cálculo matemático, a la que pueden otorgarse diferentes valores cada vez que el proceso se repite. Puede ser cualquier condición para el desarrollo de un programa, que modifica o escinde su forma de funcionar.

REGISTRO: Es una pequeña unidad de almacenamiento destinada a contener cierto tipo de datos. Puede estar en la propia memoria central o en unidades de memoria de acceso rápido.

SITIO: Lugar en la World Wide Web representado por una dirección electrónica, en el que se encuentra ubicada toda la información relacionada con una institución gubernamental, educativa o comercial. Por lo general, la presentación de un sitio está representada por una breve página principal o home page, en donde se coloca el logo o título de la institución o persona a cargo del sitio, además de botones con enlaces hipertextuales a otras páginas.

SISTEMA INFORMÁTICO: Un sistema informático es un sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático.

2.4.2. Supuestos teóricos

Al desarrollar el sistema web de gestión de expedientes digitales para la Administración Académica se podrán registrar los expedientes de miles de estudiantes de la Facultad de manera digital.

Además se podrá gestionar los expedientes digitales de los estudiantes, siendo capaz de ingresar nuevos expedientes, editar los expedientes existentes en el sistema y eliminar expedientes existentes. Agilizando los procesos de la Administración Académica de la Facultad Multidisciplinaria Oriental, permitiendo una consulta eficiente y acceso a la información de los expedientes sin necesidad de buscar manualmente entre miles de archivos físicos.

Con la implementación de dicho sistema informático se tendrá un respaldo de todos los expedientes de forma segura, de manera periódica y remota.

Capítulo 3

Metodología de la investigación

3.1. Tipo de investigación

3.1.1. investigación tecnológica

La investigación tecnológica¹ es la actividad que a través de la aplicación del método científico, está encaminada a descubrir nuevos conocimientos (investigación básica), a la que posteriormente se le buscan aplicaciones prácticas (investigación aplicada) para el diseño o mejoramiento de un producto, proceso industrial o maquinaria y equipo. La comunicación tecnológica también ha aportado grandes avances y es que nos ha llevado a conocer y aprender cosas nuevas que nos han llevado a la simplificación y automatización de las actividades que realizamos diariamente para poder realizar los procesos de una manera más eficaz y eficiente.

La investigación puede cumplir dos propósitos: proporcionar teoría y conocimiento sobre la realidad. Podríamos referirnos a esta como investigación científica. El segundo propósito es resolver problemas cotidianos de la sociedad, a esta podríamos llamarle investigación tecnológica.

La investigación tecnológica es de gran importancia para la sociedad porque gracias a ella podemos resolver problemas de la vida cotidiana, ahorrar esfuerzos para realizar un trabajo o simplemente optimizar ciertas cosas de las actividades humanas. Una de las características de la investigación científica y tecnológica es la utilidad, es por esa razón que una investigación siempre es importante por su utilidad ya que si se dice ser una investigación pero no tiene utilidad deja de ser importante, y como demostración de que la investigación es útil tenemos la historia de los grandes avances de la humanidad, desde los tiempos en que el hombre vivió en las cavernas hasta la actualidad en la que la computación es de gran importancia.

La investigación es un instrumento para poder resolver problemas de la sociedad, la ciencia y tecnología, tienen como causas las necesidades de que el ser humano viva con más comodidad y seguridad. La investigación es necesaria para impulsar el desarrollo de un país. La investigación y la escritura son las actividades más importantes de un científico. Un investigador al redactar un texto científico que trate por ejemplo de su investigación ofrece a los lectores un panorama amplio de su trabajo, la labor de escribir no es solo para un escritor o periodista sino también la de un científico, que debe poder escribir de manera clara y sencilla.

Es por ello que para el desarrollo de el sistema de expedientes para la Administración Académica de la Facultad Multidisciplinaria Oriental, se ha escogido esta clase de metodología de investigación, debido a que es la que mayormente satisface las necesidades de este proyecto de desarrollo, ya que, no solamente se desea realizar una investigación científica, sino que además se desea mejorar y aportar al desarrollo de la Administración Académica, mediante el desarrollo de un sistema informático de gestión de expedientes digitales, y con ésto, contribuir con la automatización y mejoramiento de los procesos internos, y por ende, contribuir al mejoramiento de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

¹ Cegarra Sánchez, J. (2004). *Metodología de la investigación científica y tecnológica* (No. 001.891). Díaz de Santos,.

3.2. Universo y muestreo

3.2.1. Universo

El universo de la investigación estará conformado por todo el personal administrativo que labora en la Administración Académica, y que tengan acceso al sistema de gestión de expedientes, siendo este universo de 5 personas en total; Sin embargo como el sistema informático está orientado al apoyo de la administración académica en la gestión de expedientes estudiantiles, se ve beneficiada de manera indirecta toda la comunidad estudiantil.

3.2.2. Muestra

Debido al tamaño reducido de el universo, no se tomará una muestra significativa; sino que se usará a todas las personas involucradas para la recolección de información.

3.3. Técnicas e instrumentos para la recolección de información

Según Sampieri² y otros grandes conocedores, se define a la técnica de recolección de información como: ".el método de recolección de datos de información pertinente sobre las variables involucradas en la investigación". Lo que el autor trata de explicar es que la técnica no es más que la manera cómo se van a recaudar o a recoger los datos.

Debido a los objetivos que se pretenden alcanzar en la investigación, y especificación de requerimientos, las técnicas de recopilación de información que se han considerado pertinentes para ser parte de esta investigación son: es la entrevista, los cuestionarios y la observación directa.

3.3.1. Entrevista

Una entrevista³ es un intercambio de ideas, opiniones mediante una conversación que se da entre una, dos o más personas donde un entrevistador es el designado para preguntar.

² *Sampieri, R. H. (2018). Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta. McGraw Hill México.*

³ *Hamodi, C., López Pastor, V. M., & López Pastor, A. T. (2015). Medios, técnicas e instrumentos de evaluación formativa y compartida del aprendizaje en educación superior. Perfiles educativos, 37(147), 146-161.*

Una entrevista es recíproca, donde el entrevistado utiliza una técnica de recolección mediante una interrogación estructurada o una conversación totalmente libre; en ambos casos se utiliza un formulario o esquema con preguntas o cuestiones para enfocar la charla que sirven como guía. Es por esto, que siempre encontraremos dos roles claros, el del entrevistador y el del entrevistado (o receptor).

El entrevistador es quien cumple la función de dirigir la entrevista mediante la dominación del diálogo con el entrevistado y el tema a tratar haciendo preguntas y a su vez, cerrando la entrevista. A continuación haremos mención a los dos tipos principales de entrevistas.

3.3.1.1. Entrevista estructurada

En el primer caso hablamos de una entrevista formal y estructurada, que se caracteriza por estar planteada de una manera estandarizada donde se hacen preguntas que previamente fueron pensadas y para un entrevistado en particular que responde concretamente lo que se le está preguntando.

Por esta razón, el entrevistador tiene una libertad limitada a la hora de formular las preguntas pues no pueden nacer de la entrevista en sí misma, sino de un cuestionario realizado de ante mano.

3.3.1.2. Entrevista no estructurada y libre

En el segundo caso hacemos mención a una entrevista no estructurada que es la clara oposición de una entrevista estructurada por diferentes motivos. Es flexible y abierta ya que, por más de que haya un objetivo de investigación (que es lo que rige a las preguntas) no se espera que sus respuestas se vean compuestas de un contenido ordenado y con cierta profundidad.

Si hablamos del rol que le toca al entrevistador, afirmamos que él es el encargado de elaborar preguntas pero (a diferencia de la entrevista formal) no debe seguir un cronograma de orden sobre la forma de llevar las preguntas y la formulación de las mismas.

Tipo de entrevista a utilizar:

El tipo de entrevista a ser utilizada en la presente investigación será la entrevista estructurada, debido al carácter formal y profesional de la institución a la cual se está investigando, además de la formalidad y seriedad necesaria que requiere entrevistar al administrador académico.

Otro de los factores por el cuál se ha optado por la entrevista estructurada, es debido al orden y previa elaboración de las preguntas que han de usarse en la entrevista con el administrador académico, ya que se tocarán temas puntuales acerca del proceso de gestión de expedientes del sistema informático actual, las carencias de este y características que debe poseer el nuevo sistema que vamos a desarrollar, además de diversos factores que rodean esta investigación científica para llevar a cabo todo el proceso de forma óptima.

3.3.2. Cuestionario

Un cuestionario es una herramienta de investigación que consiste en una serie de preguntas y otras indicaciones con el propósito de obtener información de los consultados. Aunque a menudo están diseñados para poder realizar un análisis estadístico de las respuestas, no es siempre así.

El cuestionario es un documento formado por un conjunto de preguntas que deben estar redactadas de forma coherente, y organizadas, secuenciadas y estructuradas de acuerdo con una determinada planificación, con el fin de que sus respuestas nos puedan ofrecer toda la información.

El cuestionario será una de las herramientas a utilizar en la etapa de recolección de información en Administración Académica. El cuestionario será aplicado a todo el personal administrativo, menos al administrador académico, el cual será abordado en entrevista. Las preguntas formuladas están orientadas de tal manera que se conozca de primera mano el proceso de gestión de expedientes actual en Administración académica, así como las mejoras que pueden llevarse a cabo, y en general todos los insumos necesarios que necesitamos para el desarrollo óptimo del sistema para poder hacer más eficiente dichos procesos administrativos.

3.3.3. Observación directa

La observación directa es un método de recolección de datos que consiste en observar al objeto de estudio dentro de una situación particular. Esto se hace sin intervenir ni alterar el ambiente en el que el objeto se desenvuelve. De lo contrario, los datos obtenidos no serían válidos.

Este método de recolección de datos se emplea en ocasiones en las que otros sistemas (como encuestas, cuestionarios, entre otros) no son efectivos. Por ejemplo, es recomendable recurrir a la observación directa cuando lo que se desea es evaluar el comportamiento por un período de tiempo continuo.

La observación directa será de gran importancia en nuestra etapa de recolección de información, ya que, será crucial observar directamente el sistema actual, para poder observar su funcionamiento, sus carencias y todo aquello que puede y debe mejorarse.

3.4. Procedimiento para la validación de instrumentos

Después de haber seleccionado los instrumentos que vamos a utilizar para la recolección de los datos, estos deben ser validados adecuadamente para asegurarnos que éstos nos brindarán la información más fidedigna y adecuada para nuestra investigación y desarrollo. Dicho proceso de validación consta de los siguientes pasos:

1. Elaboración y análisis de preguntas tanto para la entrevista como para el cuestionario.
2. Se presentarán ambos instrumentos al asesor de tesis para la revisión y validación de las preguntas formuladas.

3. En caso de correcciones por parte del asesor, se procederá a la reformulación de preguntas y se volverá al paso anterior.
4. Una vez validados los instrumentos, se procederá a la aplicación de estos en la Administración Académica para la recolección de toda la información.

3.5. Procedimiento para la recolección de información

Una vez validados los instrumentos a utilizar, procederemos a agendar una entrevista con el administrador académico, en la cual se abordarán las preguntas formuladas de manera coherente, y ordenada, como una guía que nos ayudará a obtener una mayor calidad de información.

De la misma manera se procederá a entregar una copia del cuestionario elaborado a cada uno de los integrantes del personal administrativo, para que puedan responder de manera clara y precisa a las preguntas ahí planteadas, las cuales están orientadas a la recolección de información necesaria para el mejoramiento de las características que ya posee el actual sistema.

3.6. Procedimiento para procesar la información

El procesamiento de los datos obtenidos es una de las etapas más importantes y vitales de toda investigación científica. Ya que por medio de este proceso es que se obtendrá la interpretación de toda la información que nos fue proporcionada y que nos servirá de guía para el óptimo desarrollo del sistema informático que se pretende concretar.

Para obtener los mejores resultados, se analizará e interpretará de manera detallada cada una de las preguntas planteadas en cada uno de los instrumentos utilizados, de tal manera que se haga un análisis exhaustivo pregunta a pregunta, para extraer correctamente cada uno de los requerimientos de los usuarios para el sistema a desarrollarse.

3.7. Procedimiento para presentar e interpretar la información

Se hará un análisis y recopilación de requerimientos de software basados en las respuestas de los usuarios a los instrumentos utilizados. Se detallarán los requerimientos extraídos de cada una de las preguntas formuladas, tanto del cuestionario aplicado al personal administrativo como de la entrevista realizada al administrador académico.

3.8. Análisis e interpretación de resultados

3.8.1. Análisis de requerimientos de la entrevista con el Administrador Académico.

1. ¿Desea que el software sea local o remoto?

Respuesta: "Remoto, ya que muchas veces requiere hacer demostraciones a Junta Directiva".

Podemos observar la necesidad de que el sistema a desarrollar tenga la posibilidad de acceder a él de manera remota, ya que facilita el análisis de información en reuniones con la Junta Directiva, además de la facilidad de uso en cualquier PC.

2. ¿Ha pensado en deshacerse de los documentos físicos viejos?

Respuesta: "Sí, pero es una acción que no puede llevarse a cabo por motivos legales dentro de la institución. Tiene que quedar todo documentado en físico también".

No aplica como requerimiento para el sistema.

3. ¿Desea que se implemente seguridad en la aplicación?

Respuesta: "Sí, debe poseer seguridad. Usuarios diferentes con distintos permisos, y que tenga un registro del último usuario que hizo cambios".

Analizando la respuesta podemos observar que el sistema requiere de manejo de usuarios y sesiones dentro de él, teniendo cada usuario con distintos permisos según su rol. Deberá también llevar un registro de qué usuario realizó cambios y en qué momento lo hizo.

4. ¿Desea que cada miembro del personal tenga su cuenta de usuario en el sistema?

Respuesta: "Sí".

Requerimiento extraído de la pregunta 4.

5. ¿Desea que se escanee todos los documentos de cada expediente o solo los del F2?

Respuesta: "Los documentos del F2, además de cualquier documento necesario, incluido expediente de graduación".

Podemos determinar que el sistema de gestión de expedientes, deberá tener la flexibilidad de poder agregar cualquier documento que se considere necesario, pero tener como base siempre los del F2, pero manteniendo la flexibilidad de agregar más.

6. ¿Desearía flexibilidad a la hora de determinar qué documentos se escanearán?

Respuesta: "Claro".

Requerimiento extraído de la pregunta 5.

7. ¿Le gustaría que se maneje cada documento como un archivo a parte o que se tenga uno solo (PDF) para el expediente?

Respuesta: "Hay que manejarlos individualmente".

Podemos concluir que cada documento que sea parte de un expediente, debe poder gestionarse independiente de los demás.

8. ¿Desearía que se liste en pantalla los expedientes digitales?

Respuesta: "Sí, que tenga su apartado".

Podemos concluir que debe haber un módulo donde se pueda ver un listado de todos los expedientes que existen en sistema.

9. ¿Cómo desearía que se listen, por nombre o carnet?

Respuesta: "Ambas, pero también por fecha".

Vemos que se requiere mostrar tanto el nombre completo del estudiante como el carnet en el módulo de expedientes, y que puedan filtrarse por fecha.

10. ¿Desea que se modifique cada expediente?

Respuesta: "Sí, pero solo por ciertos usuarios".

Una vez más vemos que se hace énfasis en que existan usuarios con diferentes permisos en el sistema.

11. ¿Desearía que se permita eliminar cada expediente?

Respuesta: "No realmente, pero si hubiera la posibilidad, que solo pudiera hacerlo el administrador".

Una vez más vemos que se hace énfasis en que existan usuarios con diferentes permisos en el sistema.

12. ¿Qué lista de campos desea que tenga cada expediente, a parte de los documentos?

Respuesta: "Toda la información básica, como nombres, apellidos, teléfonos, dirección, DUE, etc".

El sistema requiere de información complementaria para cada expediente del estudiante, no solo el documento.

13. ¿Le gustaría implementar una copia de seguridad de los expedientes digitales?

Respuesta: "Sí, de ser posible una vez por semana".

El sistema requiere que toda la información se respalde semanalmente.

14. ¿Desearía agregar algo?

Respuesta: "Que el sistema sea lo más simple posible para no tener complicacio-

nes al usarlo, que sea intuitivo y que no esté muy saturado de colores, mantenganlo simple”.

Podemos concluir que el diseño del sistema de gestión de expedientes digitales debe ser sobrio, simple e intuitivo, fácil de manejar y práctico.

3.8.2. Cuestionario

Cuestionario dirigido a los responsables de utilizar el sistema actual y manejar los expedientes físicos

Los encargados de administrar los expedientes de los estudiantes nos informaron sus puntos de vistas y opiniones respecto al sistema actual y lo que esperan de el sistema que se va a desarrollar. Nos comentaron que actualmente se ven con grandes problemas para manejar la cantidad de expedientes iniciales de los estudiantes en físico, aunado a que ahora se adjunta el expediente de graduación, y todo esto debe ser almacenado y archivado en estantes.

Nos explicaron que el departamento cuenta con un sistema informático desde hace un año para ayudar a esa labor y con el cual han comenzado a digitalizar algunos de los expedientes; sin embargo dicho sistema posee múltiples carencias además de ser en extremo limitado, lo cual genera un retraso en la digitalización del acervo existente.

Ellos ven necesario un sistema informático moderno que cumpla con los estándares oficiales de manejo de información, sugirieron que el sistema cuente con autenticación y manejo de sesiones, que incorpore una bitácora de las actividades realizadas, que permita manejar múltiples archivos por expediente, que cuente con respaldo automático, entre otros.

También nos dieron sus opiniones sobre la solución recomendada por nosotros, consideran que el sistema informático es necesario y muy relevante para dicho departamento.

Para ver el cuestionario aplicado a los responsables de Administración Académica, consulte el anexo #2.

Capítulo 4

Desarrollo de software

4.1. Definición del sistema

Desarrollar un sistema informático web para la gestión de expedientes de la Administración Académica en la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador.

4.2. Resumen de problemas del sistema actual

1. Fallos graves de seguridad.
2. No cuenta con manejo de sesiones para distintos usuarios con diferentes privilegios.
3. No posee un mecanismo de respaldo de información.
4. No existe manera de acceder al sistema desde otra PC que no sea en la que está implementado.
5. Tiene una pobre experiencia de usuario debido a su mal diseño, además es poco intuitivo y amigable con el usuario (Botones mal distribuidos en pantalla, algunos de ellos no realizan ninguna acción).
6. Deficiente gestión de los documentos.
7. El expediente carece de información relevante del estudiante.
8. No posee un buscador eficiente de expedientes.
9. No pueden consultarse partes específicas del expediente sin tener que revisarlo todo.
10. No se pueden añadir expedientes adicionales.

4.3. Supuestos

- El sistema web funcionará en todos los dispositivos móviles como tablets y smartphones, además del PC.
- No se requiere que se tenga un sistema operativo determinado para funcionar, funciona en cualquiera de los existentes.
- El sistema mantendrá un registro de los últimos usuarios que agreguen o modifiquen expedientes.
- La red interna deberá estar configurada para el manejo de protocolos TCP/IP, HTTP, HTTPS y DNS, principalmente los aspectos relacionados con desempeño y seguridad.
- Si bien el sistema web funciona en cualquier sistema operativo, es necesario tener un navegador web instalado, que soporte HTML5, CSS3, JAVASCRIPT, además de tener un servidor Apache instalado en el equipo, corriendo MariaDB y PHP 7.0 o posterior.

4.4. Restricciones

- La ocupada agenda del Administrador Académico, que impedirá una retroalimentación óptima.
- El servidor que almacenará el sistema cuenta con espacio reducido.
- Cuenta con un miembro con conocimiento de IT, Jacquelin Paniagua, pero no siempre tiene disponibilidad de tiempo.

4.5. Requerimientos del nuevo sistema

El sistema informático web para gestionar expedientes de estudiantes debe satisfacer las necesidades del personal administrativo y del administrador de Académica. Para ello se aplicaron instrumentos como entrevista y cuestionario, los cuáles nos permitieron tener una visión más clara de los requisitos que debe satisfacer el sistema, los cuáles se detallan a continuación:

4.5.1. Requerimientos funcionales

Los requerimientos funcionales que se implementarán en el sistema son los siguientes:

- Requisitos funcionales del sistema
 - CRUD de expedientes de estudiantes
 - El sistema debe ser capaz de registrar nuevos expedientes en la base de datos.
 - El sistema debe ser capaz de eliminar expedientes existentes en la base de datos.
 - El sistema debe ser capaz de modificar expedientes existentes en la base de datos.
 - El sistema debe permitir consultar información general de los estudiantes asignados a cada expediente.
 - Búsqueda de registros
 - El sistema debe ser capaz de realizar búsqueda de expedientes por carnet, nombre o año de ingreso de los estudiantes.
 - El sistema debe ser capaz de realizar búsqueda de usuarios registrados en el sistema.
 - Gestión de expedientes de estudiantes.
 - El sistema debe permitir editar expedientes por documentos individuales.
 - El sistema debe permitir anexar documentos adicionales a los expedientes de estudiantes.
 - CRUD de Usuarios
 - El sistema debe ser capaz de registrar un nuevo usuario en la base de datos.

- El sistema debe ser capaz de eliminar un usuario existente en la base de datos.
- El sistema debe ser capaz de editar un usuario existente en la base de datos.
- Configuraciones
 - El sistema debe permitir realizar un respaldo de la información.
 - El sistema debe almacenar un registro de los últimos usuarios que hagan movimientos en el sistema.
 - El sistema debe permitir modificar los diferentes tipos de documentos en el sistema (F2, Expediente de graduación, entre otros).
- Requisitos funcionales de seguridad
 - Autenticación de los usuarios, los usuarios requieren iniciar sesión para poder tener acceso a todos los procesos a los que tengan permisos en sistema.
 - El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña
 - El sistema notificará al administrador del sistema cuando ocurra alguno de los siguientes eventos: Modificación de expedientes, eliminación de expedientes.
- Otros requerimientos.
 - El sistema debe generar reportes basados en la cantidad de registros existentes, por año de ingreso, entre otros criterios.
 - El sistema podrá ser utilizado desde cualquier PC con acceso a internet.
 - Implementará mecanismos que aseguren la integridad de los datos.

4.5.2. Requerimientos no funcionales

Los requerimientos no funcionales que se detectaron para implementar en nuestro sistema son los siguientes:

- El sistema podrá ser utilizado a través de cualquier navegador web compatible con HTML5, CSS3 y Javascript.
- El sistema debe presentar mensajes de error que permitan al usuario identificar el tipo de error.
- El administrador del sistema podrá saber qué usuario fue el último en realizar cambios.
- El sistema estará desarrollado mediante orientación a objetos para mayor escalabilidad.
- El sistema debe ser de fácil uso por parte de los usuarios.
- Presentará de forma correcta la redacción y ortografía en las pantallas.

- El sistema no debe permitir el cierre de una operación hasta que todos sus procesos hayan sido terminados y cerrados satisfactoriamente.
- El sistema debe contar con una interfaz de administración de usuarios donde se visualicen todos juntos.
- El login del sistema deberá mostrar mensajes de error bien estilizados.
- El sistema debe validar la información insertada en los formularios de ingreso. Se toma en cuenta aspectos tales como: longitud de caracteres permitidos por campo y manejo de tipos de datos mediante expresiones regulares.
- El software mostrará un menú con todos los módulos horizontalmente arriba de cada página, los menús dependerán de los privilegios de cada usuario.
- El software mostrará un menú de sesión en la esquina superior derecha de cada página para las opciones de configuración, perfil y cierre de sesión de cada usuario.
- El software mostrará un listado de los usuarios, expedientes, departamentos, etc, en sus respectivos módulos.
- Los botones de acción en los listados deberán ser pequeños, de apariencia simple, con íconos y sin texto, serán descriptivos: papelera para borrar un usuario.
- En cada módulo habrá un botón para registrar un elemento nuevo de cada módulo, en la parte superior derecha de ese módulo o sección.
- El formulario de registro de todos los módulos se mostrarán como una pequeña ventana encima de la página, que aparecerá con el botón "Nuevo...", y que desaparecerá, sin necesidad de ir a otra página.
- El formulario de modificación será en otra página con un diseño idéntico a la de registro.
- La eliminación se ejecutará en la misma lista de usuarios, estudiantes, departamentos, etc.
- En el expediente de cada usuario se mostrará una lista de pestañas para cada tipo de documento, y dentro de cada una un listado de los documentos correspondientes.
- Cada elemento del listado tendrá un formulario para subir el archivo pdf, escribir un comentario, ver y descargarlo.
- El botón para subir el archivo será simple, sin texto con un ícono descriptivo, reemplazando el formulario tradicional.

4.6. Estudio de factibilidad

4.6.1. Factibilidad técnica

La factibilidad técnica se analizó en dos categorías: Hardware y Software.

4.6.1.1. Hardware

En la parte de hardware, la Administración Académica actualmente posee solamente una PC destinada a ejecutar el sistema de gestión de expedientes, cuyas características son las óptimas para su función, y podrá ejecutar el sistema. También cuentan con un servidor propio que se encuentra dentro de la propia Administración Académica, este nos permitirá almacenar el nuevo sistema en él de manera satisfactoria. Las características de ambos equipos se muestran a continuación.

Las características del ordenador cliente son:

Característica	Valor
Procesador	Intel core i5-4590 3.30GHz
Memoria	8 GB RAM
Almacenamiento	750 GB disco duro
Sistem operativo	Windows 10 Pro 64 bits
Navegador Web	Google chrome 74 03729.169 64 bits

Cuadro 4.1: Características del ordenador cliente.

Las características del servidor son:

Característica	Valor
Procesador	Intel Xeon X5560 2.8GHz 8 MB 4 nucleos 95 W
Memoria RAM	Hasta 128 GB RAM DDR3
Almacenamiento	12 TB en disco duro
Periféricos	5 ranuras PCI
Gráficos	Matrox G200eW con 8 MB

Cuadro 4.2: Características del servidor.

Como podemos observar, las características del equipo con el que se cuenta son las óptimas, nos permitirán implementar el nuevo sistema de manera satisfactoria.

4.6.1.2. Software

La factibilidad técnica en este apartado es completa, debido a que el sistema será de uso web, compatible con cualquier navegador. Por la tanto la institución no incurrirá en costos adicionales o adquisición de licencias, exceptuando la del sistema operativo que ya viene incluido en el precio del PC, y la de paquetes de Ofimática o Antivirus que se pueden adquirir libres o de pago, según la elección particular que pueda tener el personal de la Administración Académica.

En su desarrollo, el sistema también será completamente factible, debido a que será desarrollado con herramientas libres y/o con licencias GPL, tales como: servidor Apache, MariaDB como gestor de base de datos, PHP7 como lenguaje de desarrollo del lado del servidor, y HTML5, CSS3 y Javascript como lenguajes del lado del cliente que nos permitirán el diseño y creación de interfaces de usuario agradables, utilizables e interactivas.

4.6.2. Factibilidad económica

La factibilidad económica de la implementación del proyecto se considera viable debido a que la inversión inicial para la implementación de este proyecto informático es relativamente nula, ya que la Administración Académica cuenta con un servidor donde será alojado el sistema y la misma Universidad cuenta con un Departamento de tecnologías de la información que proveerá los recursos y personal necesarios. Todo esto aunado a que no se incurrirán en gastos por el desarrollo del sistema informático, permite a la entidad el desarrollo e implementación del nuevo sistema de gestión de expedientes estudiantiles.

A continuación se muestra una tabla con los costos estimados de desarrollo e implementación de dicho sistema, con el fin brindar información sobre un proyecto de esta índole.

Área	Detalle	Costo
Desarrollo del sistema informático	Licencia de Servidor Web Apache 2.4	\$0.00
	Licencia de MariaDB 10.4.7	\$0.00
	Licencia de PHP 7.3.8 + HTML5 + CSS3 + Javascript 6	\$0.00
	Análisis del sistema (Salario de dos meses de una persona)	\$1 600.00
	Diseño del sistema (Salario de dos meses de una persona)	\$1 600.00
	Desarrollo del sistema (Salario de dos meses de dos personas)	\$2 400.00
	Concesión de licencias sobre derechos de propiedad intelectual	\$1 000.00
Subtotal		\$6 600.00
Recurso tecnológico	Servidor Dell PowerEdge T410	\$1 800.00
	Ordenador HP Pavillon	\$800.00
Subtotal		\$2 600.00
Instalación	Instalación y configuración del servidor	\$600.00
	Instalación y configuración del sistema en la red	\$960.00
	Capacitación del personal	\$200.00
Subtotal		\$1 760.00
Total		\$10 960.00

Cuadro 4.3: Costo de desarrollo del proyecto.

NOTA: Adicionalmente se incurrirá en gastos anuales operativos como son el dominio del sitio web, el costo del internet, energía eléctrica, entre otros. Sumando \$600 anuales aproximadamente.

4.6.3. Factibilidad operativa

A pesar que el personal administrativo que labora en Académica no es un personal avanzado en informática, tienen los conocimientos básicos y necesarios del uso del internet y de otros sistemas usados con anterioridad. Todos ellos tienen conocimiento básico de informática, por lo tanto con una capacitación mínima serán capaces de operar perfectamente el sistema informático, que será intuitivo y con una curva de aprendizaje reducida.

A excepción de la Lic. Jacqueline Paniagua, que tiene conocimientos prácticos y teóricos de IT, la cual es de gran ayuda para hacer consultas de resolver dudas, ella consulta directamente con el Departamento de Sistemas Informáticos de la facultad, y es capaz de asesorar a aquel que este cumpliendo horas sociales en el departamento.

En cuanto a la administración y mantenimiento del sistema, se recomienda contratar a una o dos personas expertas en TI que hagan dichas funciones, o la contratación de una empresa externa para el desarrollo de dichas tareas.

4.6.3.1. Perfil de Desarrolladores

Cada elemento del equipo tiene como perfil:

- Conocimientos en desarrollo de sistemas informáticos determinados por clientes.
- Conocimientos en lenguajes como SQL92, PHP7, HTML5, CSS3, JavaScript.
- Conocimiento en metodologías como: Cascada, Scrum, Pressman, etc.
- Manejo de GNU/Linux (nivel intermedio), Windows, Android.
- Conocimientos de IDE como: Netbeans 8.1, Atom 1.3.
- Uso de Latex.
- Administración de bases de datos con MySQL, Phpmyadmin.
- Conocimiento en gestión de bases de datos y servidores web con WAMP, XAMPP, o en sistemas UNIX-like.
- Frameworks como Bootstrap 4.3, jQuery 3.3.

4.6.4. Factibilidad jurídica

Existen una serie de leyes nacionales y universitarias que rigen como se debe tratar la documentación y administración, y todo lo relacionado a estos. Entre ellas se pueden mencionar:

- Ley de Acceso a la Información Pública
- Ley de Archivo de la Nación

- Ley de Procedimientos Administrativos.
- Reglamento de la Gestión Académico Administrativa de la Universidad de El Salvador

Luego de hacer las consultas pertinentes y apegadas al marco jurídico, acorde al rubro de la institución, se concluye que se cumple con todos los requerimientos, y que es viable legalmente la realización del proyecto.

4.7. Herramientas de desarrollo

Como hemos mencionado a lo largo del documento, para el desarrollo del proyecto hemos optado por herramientas de código abierto, es decir libres y sin costo alguno de licencias. A continuación presentamos con más detalle las herramientas, lenguajes y utilidades que hemos seleccionado, y sus versiones correspondientes. Para mayor comprensión hemos dividido las tecnologías según el propósito en el que han de implementarse.

4.7.1. Metodologías de desarrollo.

Para la metodología de desarrollo se optó por el ‘Ciclo de vida de desarrollo de software’, que consiste en cinco etapas según Pressman.

- Etapa I: Análisis de los requisitos del software.
- Etapa II: Diseño.
- Etapa III: Generación de Código.
- Etapa IV: Pruebas.
- Etapa V: Mantenimiento.

Para más información consulte en la referencia en pie de página.

4.7.2. Diseño

HTML 5 Como lenguaje para la definición de la estructura del sistema.

CSS 3 Como lenguaje para la definición del estilo y estética del sistema.

JavaScript (ECMAScript 6) Como lenguaje de programación orientado a aplicar dinamismo a la estructura y estilo del sistema.

4.7.3. Desarrollo

PHP 7.1 Como lenguaje de desarrollo del lado del servidor para implementar todos los procesos y lógica del sistema.

4.7.4. Frameworks y librerías

Bootstrap 4.3 Framework principal CSS y Javascript para facilitar el diseño e interacción del sistema.

jQuery 3.3 Librería Javascript utilizada para el correcto funcionamiento de las utilidades Javascript de Bootstrap.

Subsubsección 2.2.8.1 ‘Ciclo de vida de desarrollo de software’, Página 35

4.7.5. Editor de texto y Entorno de desarrollo integrado (IDE)

Atom 1.3 Editor de texto principal utilizado mayormente para el diseño HTML, CSS y Javascript debido a su versatilidad y variedad de plugins.

Netbeans 8.1 Entorno de desarrollo utilizado para el desarrollo con PHP ya que cuenta con autocompletado inteligente, plugins y creación rápida de proyectos.

4.7.6. Base de datos)

MariaDB 10.1 Motor de base de datos libre y potente, en el cual se van a almacenar y manejar los datos e información del sistema.

4.7.7. Servidor)

Apache 2.4.37 Servidor de código abierto en el cual correrá el sistema de gestión de expedientes.

4.7.8. Software complementario)

XAMPP 7.1 Paquete de software libre, que consta principalmente del sistema de gestión de bases de datos MariaDB, el servidor web Apache y el intérprete para el lenguaje PHP.

MySQL Workbench 8.0.15 Software de código abierto para el diseño y modelado de la base de datos.

phpmyadmin 4.3 Utilidad web para el manejo de la base de datos mediante interfaz gráfica.

4.7.9. Cantidad de archivos promedio a almacenar

Se necesita determinar el tamaño total de archivos que el almacenamiento de la administración es capaz de alojar, para prever a futuro si es capaz de almacenar tanto expediente de futuros alumnos, a parte de los expedientes físicos ya existentes, para ellos se calculó el tamaño promedio de los PDF almacenados actualmente, así como su tamaño máximo y mínimo, y el total almacenado.

Disco Duro en servidor Disco de 761 GB único.

Tamaño máximo existente 22.9 MB, tamaño máximo registrado.

Tamaño mínimo registrado 88.9 kB.

Tamaño promedio de archivos 2.90 MB

Cantidad a almacenar a futuro 268,711 archivos de tamaño promedio.

4.8. Estándares de desarrollo

4.8.1. Estándares para el desarrollo del sistema informático

En el desarrollo del sistema informático web para la gestión de expedientes, se utilizó el ciclo de vida del software como guía de desarrollo, y como paradigma, la programación orientada a objetos.

Las convenciones de codificación ayudan a la uniformidad y legibilidad del código fuente, facilitando su mantenimiento por terceros y eliminando la dependencia del desarrollador. Las siguientes son las convenciones que hemos adoptado para el desarrollo del sistema en el lenguaje php.

Nomenclatura de ficheros: Utilizar caracteres alfanuméricos pero no espacios. En todos los ficheros sólo se utilizarán caracteres alfanuméricos. Los espacios están estrictamente prohibidos. En caso de que un fichero contenga una clase PHP, el nombre del fichero debe ser el mismo nombre de la clase, ambos con inicial mayúscula.

Declaración de variables: Utilizar caracteres alfanuméricos, utilizando como estilo de escritura "Camel case" que consiste en escribir las palabras que componen un nombre de variable una seguida de la otra, sin espacios ni ningún otro símbolo de por medio, con la primer palabra escrita toda en minúsculas, y las subsiguientes con la inicial mayúscula.

Nomenclatura de variables: Utilizar caracteres alfanuméricos comenzando con '\$'. Las variables deben comenzar con el carácter '\$' seguido de caracteres alfanuméricos, incluido el guión bajo.

Alineación: El código debe estar bien indentado, es decir, contener sangría para mayor legibilidad.

Demarcación de código en PHP: Delimitar el código con las correspondientes etiquetas php. El código PHP siempre debe estar delimitado por las etiquetas php, evitando el uso de atajos. Si los ficheros sólo contienen código PHP entonces podemos omitir la etiqueta de cierre.

Concatenación: Concatenar cadenas utilizando el operador ".". La concatenación de cadenas debe realizarse utilizando el operador ".", añadiendo un espacio delante y detrás del operador.

Comillas simples: Usar las comillas simples en los literales. Cuando una cadena es literal (no contiene sustituciones) debe marcarse la cadena entre comillas simples.

Comillas dobles: Usar las comillas dobles en cadenas. Cuando una cadena incluye diferentes apóstrofes debe ir encerrada entre comillas dobles.

Manejo del Array: No utilizar índices negativos en los arrays. Un array unidimensional comienza situando el índice a 0. Cuando el array se declara utilizando el constructor, se debe dejar un espacio tras cada delimitador siguiendo criterios de legibilidad. Si el array es inicializado mediante el constructor cada línea sucesiva debe

alinearse con la superior.

Sustituciones en cadenas: Utilizar los símbolos $o\{y\}$ para las sustituciones. Las sustituciones dentro de una cadena deben realizarse utilizando los símbolos $,o\{ y \}$.

Extensión de ficheros php: Los ficheros terminan con la extensión ".php" Todos los ficheros que contienen código PHP terminan con la extensión ".php", con la excepción de la vista de los scripts.

Documentación de elementos: Documentar los elementos para facilitar el mantenimiento de los mismos. Con esto nos referimos a comentar el código.

Los siguientes elementos deben ser documentados siguiendo una estructura adecuada y utilizando etiquetas específicas:

- Sentencias define once.
- Sentencias include once.
- Funciones.
- Clases.
- Métodos y atributos.
- Variables globales.
- Archivo de código.

4.9. Metodología de programación

Aquí se describe la o las metodologías de desarrollo en las cuáles nos basamos para poder desarrollar el nuevo sistema informático web.

4.9.1. Metodología de programación orientada a objetos

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades y métodos. Esto promueve la escalabilidad, facilita el mantenimiento y promueve la reutilización del código.

4.10. Diseño del sistema informático

4.10.1. Diagramas de casos de uso

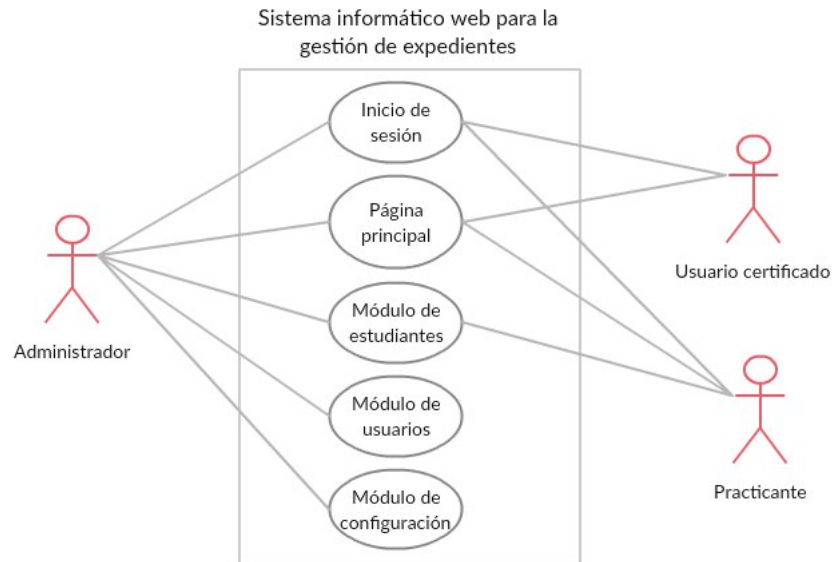


Figura 4.1: Diagrama general de casos de uso

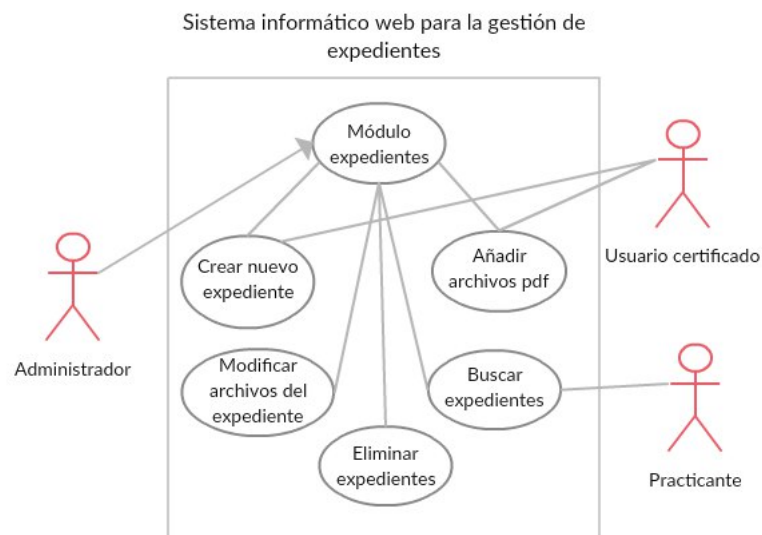


Figura 4.2: Caso de uso de expedientes

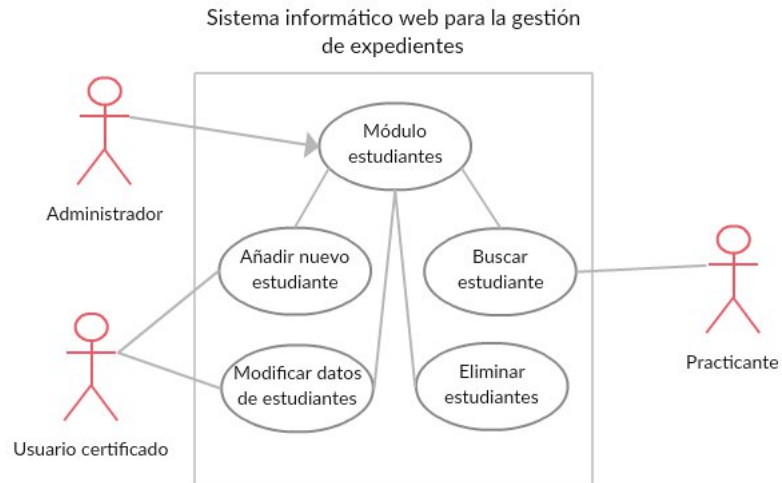


Figura 4.3: Caso de uso de estudiantes



Figura 4.4: Caso de uso de configuración

A continuación se describen algunos de los casos de uso de el sistema informático para el manejo de expedientes.

CASO DE USO:	INICIAR SESIÓN
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Cualquier usuario
Precondición:	El usuario proporciona su nombre y contraseña para acceder al sistema de acuerdo a su nivel de privilegios.
Descripción:	El sistema intenta iniciar una nueva sesión con los datos proporcionados, si los datos son correctos se procede a entrar al sistema.
Secuencia normal:	<p>Acción</p> <p>1- El usuario solicita al sistema crear una nueva sesión.</p> <p>2- El sistema compara los datos con los existentes en la base de datos.</p> <p>3- El sistema muestra el módulo de inicio al usuario identificado.</p>
Postcondición:	
Excepciones:	<p>Acción</p> <p>2- Si las credenciales ingresadas no se encuentran en la base de datos .</p> <p>2.1 - El sistema muestra un mensaje indicando que los datos son incorrectos.</p>

Cuadro 4.4: Caso de uso: Iniciar sesión.

CASO DE USO: CREAR NUEVO EXPEDIENTE	
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Usuario con acceso
Precondición:	El usuario debe tener permisos para poder crear registros.
Descripción:	El usuario con permisos de escritura añade un nuevo registro para un estudiante con sus respectivos datos y documentos.
Secuencia normal:	Acción 1- El usuario entra al formulario para crear un nuevo registro. 2- El usuario ingresa los datos requeridos. 3- Se crea un nuevo expediente.
Postcondición:	
Excepciones:	Acción 1.- Si el usuario no tiene permisos para crear nuevos expedientes 1.1 - El sistema muestra un mensaje indicando que no se puede realizar dicha acción.

Cuadro 4.5: Caso de uso: Crear nuevo expediente.

CASO DE USO: AÑADIR ARCHIVOS PDF	
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Administrador, usuario certificado, practicante
Precondición:	Cualquier usuario registrado tiene acceso.
Descripción:	El usuario escoge un expediente y selecciona desde el ordenador los archivos a añadir en el campo correcto.
Secuencia normal:	Acción 1- El usuario selecciona un estudiante. 2- El usuario escoge la pestaña acorde al tipo de expediente. 3- selecciona el documento a añadir. 4- selecciona el archivo PDF a cargar.
Postcondición:	
Excepciones:	Acción 4.- El tipo de archivo no es permitido 4.1 - El sistema muestra un mensaje indicando que dicho archivo es incompatible.

Cuadro 4.6: Caso de uso: Añadir archivos PDF.

CASO DE USO:	MODIFICAR ARCHIVOS DEL EXPEDIENTE
Prioridad:	Mediana
Tipo:	Indispensable
Actores:	Administrador, usuario certificado, practicante
Precondición:	Seleccionar un expediente con archivos añadidos.
Descripción:	El usuario escoge un expediente y modifica los archivos PDF que contiene
Secuencia normal:	Acción 1- El usuario selecciona un expediente. 2- El usuario escoge que documento modificar 3- Selecciona el nuevo archivo a añadir.
Postcondición:	
Excepciones:	Acción 3.- El tipo de archivo no es permitido 3.1 - El sistema muestra un mensaje indicando que dicho archivo es incompatible.

Cuadro 4.7: Caso de uso: Modificar archivos del expediente.

CASO DE USO:	BUSCAR EXPEDIENTES
Prioridad:	Mediana
Tipo:	Indispensable
Actores:	Administrador, usuario certificado, practicante.
Precondición:	Existir registros de expedientes
Descripción:	El usuario consulta un expediente por DUE, nombre, u otro criterio de búsqueda del estudiante
Secuencia normal:	Acción 1- El usuario busca un estudiante. 2- Selecciona la pestaña del tipo de expediente 3- Selecciona que documento visualizar.
Postcondición:	
Excepciones:	Acción 1.- No existe registro 3.1 - Se muestra un mensaje indicando que sus criterios de búsqueda no ofrecen resultados.

Cuadro 4.8: Caso de uso: Buscar expediente.

CASO DE USO:	ELIMINAR EXPEDIENTE
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Administrador
Precondición:	Ser administrador.
Descripción:	El administrador elimina un expediente.
Secuencia normal:	Acción 1- El administrador busca y selecciona un expediente. 2- Confirma que desea eliminar permanentemente dicho registro. 3- Se registra dicha acción en la bitácora.
Postcondición:	
Excepciones:	Acción 1.- Fallo en la conexión con el servidor 1.1 - El sistema indica que no se puede conectar exitosamente.

Cuadro 4.9: Caso de uso: Eliminar un expediente.

CASO DE USO:	AÑADIR NUEVO ESTUDIANTE
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Administrador, usuario certificado.
Precondición:	El usuario debe tener permisos para poder crear registros.
Descripción:	El usuario crea un nuevo registro con los datos personales de un estudiante.
Secuencia normal:	Acción 1- El usuario entra al formulario para añadir un estudiante. 2- El usuario ingresa los datos requeridos. 3- Se crea un nuevo registro de estudiante.
Postcondición:	
Excepciones:	Acción 1.- Si el usuario no tiene permisos para añadir nuevos estudiantes 1.1 - El sistema muestra un mensaje indicando que no se puede realizar dicha acción.

Cuadro 4.10: Caso de uso: Añadir nuevo estudiante.

CASO DE USO: MODIFICAR DATOS DE ESTUDIANTES	
Prioridad:	Mediana
Tipo:	Indispensable
Actores:	Administrador, usuario certificado.
Precondición:	Seleccionar un registro de un estudiante.
Descripción:	El usuario escoge un estudiante y modifica sus datos personales
Secuencia normal:	Acción 1- El usuario selecciona un estudiante. 2- El usuario escoge que campo editar 3- Guarda los cambios.
Postcondición:	
Excepciones:	Acción 3.- Los caracteres ingresados son inválidos 3.1 - El sistema muestra un mensaje solicitando que corrija el error.

Cuadro 4.11: Caso de uso: Modificar datos de estudiantes.

CASO DE USO: BUSCAR ESTUDIANTES	
Prioridad:	Mediana
Tipo:	Indispensable
Actores:	Administrador, usuario certificado, practicante.
Precondición:	Existir registros de expedientes
Descripción:	El usuario filtra un estudiante por DUE, nombre, u otro criterio de búsqueda.
Secuencia normal:	Acción 1- El usuario ingresa el criterio de búsqueda. 2- Se ofrecen los resultados.
Postcondición:	
Excepciones:	Acción 1.- No existe registro 3.1 - Se muestra un mensaje indicando que sus criterios de búsqueda no ofrecen resultados.

Cuadro 4.12: Caso de uso: Buscar estudiante.

CASO DE USO: ELIMINAR ESTUDIANTE	
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Administrador
Precondición:	Ser administrador.
Descripción:	El administrador elimina un estudiante.
Secuencia normal:	Acción 1- El administrador busca y selecciona un estudiante. 2- Confirma que desea eliminar permanentemente dicho registro. 3- Se registra dicha acción en la bitácora.
Postcondición:	
Excepciones:	Acción 1.- Fallo en la conexión con el servidor 1.1 - El sistema indica que no se puede conectar exitosamente.

Cuadro 4.13: Caso de uso: Eliminar un estudiante.

CASO DE USO: AÑADIR NUEVOS USUARIOS	
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Administrador
Precondición:	Ser administrador.
Descripción:	El administrador registra nuevos usuarios.
Secuencia normal:	Acción 1- El administrador entra al módulo de configuración. 2- ingresa los datos personales del nuevo usuario. 3- Escoge el nivel de acceso . 4-Crea el nuevo usuario
Postcondición:	
Excepciones:	Acción 1.- Fallo en la conexión con el servidor 1.1 - El sistema indica que no se puede conectar exitosamente.

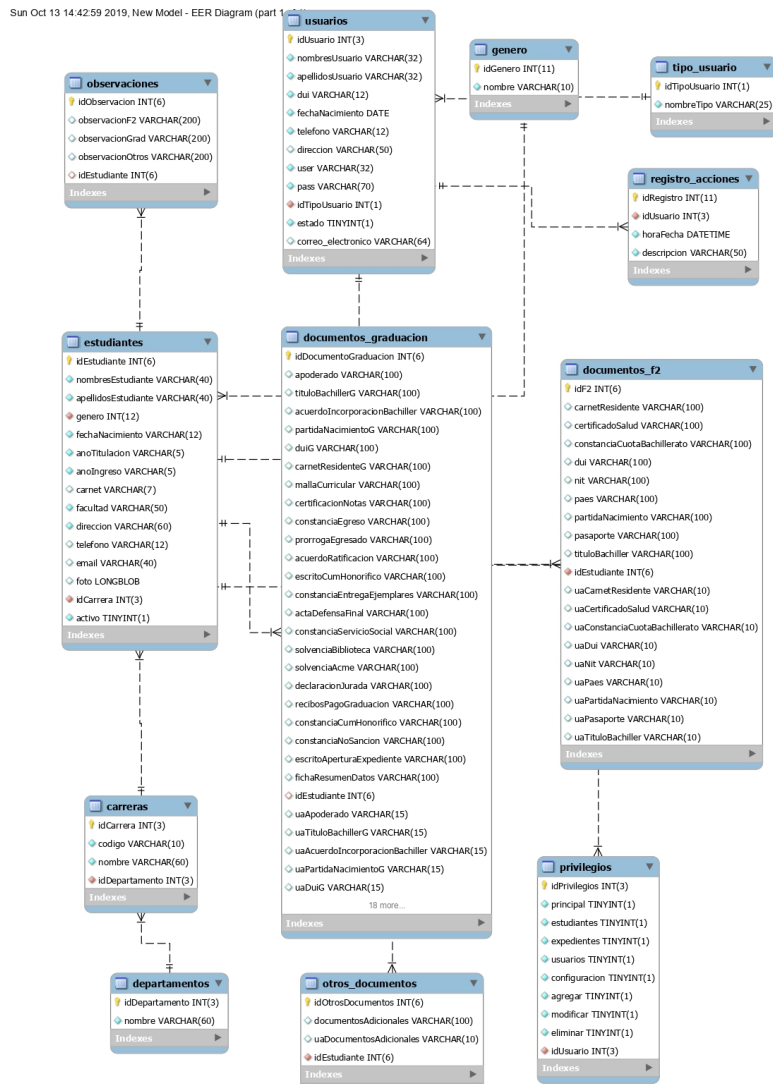
Cuadro 4.14: Caso de uso: Añadir nuevos usuarios.

CASO DE USO:	RESPALDO
Prioridad:	Alta
Tipo:	Indispensable
Actores:	Administrador
Precondición:	Ser administrador.
Descripción:	El administrador realiza un respaldo de la base de datos.
Secuencia normal:	Acción 1- El administrador entra al módulo de configuración. 2- Selecciona la opción Respaldo BD. 3- Escoge las Configuraciones . 4-Realiza el respaldo
Postcondición:	
Excepciones:	Acción 1.- Fallo en la conexión con el servidor 1.1 - El sistema indica que no se puede conectar exitosamente.

Cuadro 4.15: Caso de uso: Respaldo.

4.10.2. Diseño de la base de datos

La base de datos del sistema informático web para la gestión de expedientes es escalable y relacional. A continuación se presenta el diagrama de entidad-relación.



1 of 1

Figura 4.5: Diagrama entidad relación

El diccionario de datos puede ser consultado en el apartado de anexos.

Capítulo 5

Plan de implementación

El Plan de Implantación es donde se sustituyen sistemas antiguos ya sea manuales o mecanizados, y se pone en marcha el nuevo sistema para que pueda ser operado por los usuarios. en este caso, el personal administrativo de la Administración Académica de la Facultad Multidisciplinaria Oriental.

El plan de implantación que se presenta continuación, se divide en 4 partes:

- La planeación.
- El acondicionamiento.
- La implementación.
- Las pruebas.

La planeación: Comprende la manera en la que se pretende realizar la implementación del sistema, y de la división del proceso completo en tareas y actividades a realizar para facilitar el proceso.

El acondicionamiento: Abarca todos los aspectos previos a la implementación, como la actualización del software del servidor, la instalación de paquetes nuevos y la preparación de los aspectos de red como nombre de dominio, ip y acceso a la red.

La implementación: Comprende la puesta en marcha del nuevo sistema, donde todos los archivos y paquetes del nuevo sistema se instalan en el directorio /html para que sea accesible al público objetivo (los usuarios del sistema). También se implementa la base de datos del sistema.

Las pruebas: Es donde el sistema es puesto a prueba por los usuarios para la detección de posibles errores o confirmación de que todo funciona a la perfección.

5.1. Objetivos

5.1.1. Objetivo general

Establecer un plan que permita llevar a cabo la implementación del sistema informático de gestión de expedientes en la Administración Académica de la Facultad

Multidisciplinaria Oriental de la Universidad de El Salvador, tomando en cuenta las actividades de planificación, Acondicionamiento, Implementación y Pruebas.

5.1.2. Objetivos específicos

- Realizar la planificación del plan de implementación, determinando las actividades a realizar.
- Identificar los paquetes de software y complementos necesarios dentro del servidor, y en caso de ser necesario, actualizarlos o instalar software complementario.
- Implementar los paquetes y carpetas del nuevo sistema dentro del directorio /html.
- Efectuar pruebas del nuevo sistema para corroborar que todo haya salido según lo planeado y verificar que no existan errores.

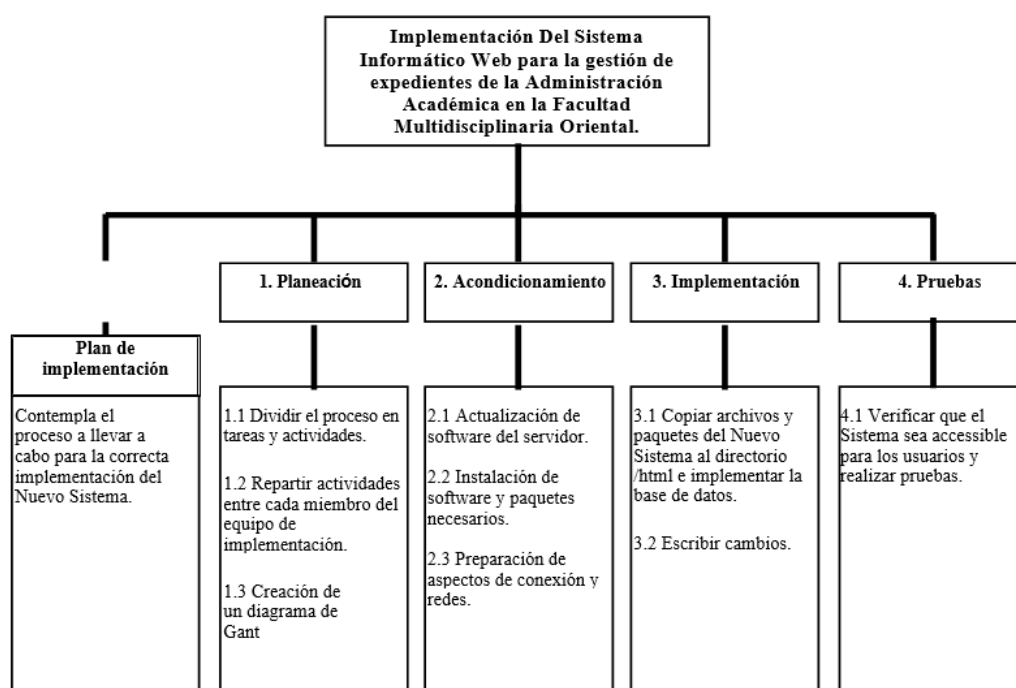


Figura 5.1: Diagrama de desglose analítico

5.2. Planeación

5.2.1. Actividad 1: Dividir el proceso en tareas y actividades

Divide y vencerás en un dicho muy acertado. Más aún en el ámbito de desarrollo de sistemas. En esta etapa de la planeación, se divide todo el proceso en tareas y actividades a realizar para poder sobrellevar de mejor manera todo el proceso.

Las actividades principales se dividen en: Acondicionamiento, Implementación y pruebas, y estas a su vez se subdividen en actividades más específicas, las cuáles se detallan a partir de la sección de acondicionamiento.

5.2.2. Actividad 2: Repartir actividades entre cada miembro del equipo de implementación

Dividir actividades permite a su vez trabajar de manera más eficiente repartiendo esas actividades entre todos los participantes de la implementación del sistema.

5.2.2.1. Responsables de cada actividad

A continuación se mencionan las personas que estarán a cargo de cada actividad:

- Diego Armando Herrera Flores estará a cargo de la supervisión de todas las actividades y realizará todas las actividades relacionadas al acondicionamiento.
- Rafael Elías Mariona Turcios estará a cargo de todas las actividades relacionadas a la implementación.
- Rommel Amadeus Alberto Mejía estará a cargo de las pruebas de acceso y pruebas de funcionamiento del sistema.

5.2.3. Actividad 3: Creación de un diagrama de Gant

Esto permitirá una mejor gestión de las actividades con respecto al tiempo.

Todos los involucrados en el plan de implementación participan en la creación de este diagrama.

5.3. Acondicionamiento

5.3.1. Actividad 1: Actualización de software del servidor

Estas actividades involucran tanto la actualización del sistema mismo (Debian 8.0 en este caso), como paquetes de software y dependencias.

5.3.2. Actividad 2: Instalación de software y paquetes necesarios

Instalación de PHP, MariaDB o apache en caso de ser requerido, o bien alguno de los anteriores en caso de no encontrarse instalado.

5.3.3. Actividad 3: Preparación de aspectos de conexión y redes

Requiere de la participación del personal encargado de redes en la facultad. Esta actividad contempla la asignación de un nombre de dominio para el nuevo sistema, asignación de una ip y recursos de red.

5.4. Implementación

5.4.1. Actividad 1: Copiar archivos y paquetes del Nuevo Sistema al directorio /html.

Esta actividad consiste en copiar todos los archivos del sistema desarrollado a la carpeta /html del servidor para poder utilizarlo dentro de este y que los usuarios puedan acceder a él. En esta actividad también debe implementarse la base de datos diseñada para el sistema.

5.4.2. Actividad 2: Escribir cambios.

Esta actividad asegura que todos los cambios realizados anteriormente han quedado debidamente guardados.

5.5. Pruebas

5.5.1. Actividad 1: Verificar que el Sistema sea accesible para los usuarios y realizar pruebas.

Esta actividad consiste en probar el acceso al sistema por parte de los usuarios, además de realizar pruebas de procesos que realiza el sistema en sí. Esto nos asegura que todo funcione según lo planeado antes de dar por finalizada la implementación.

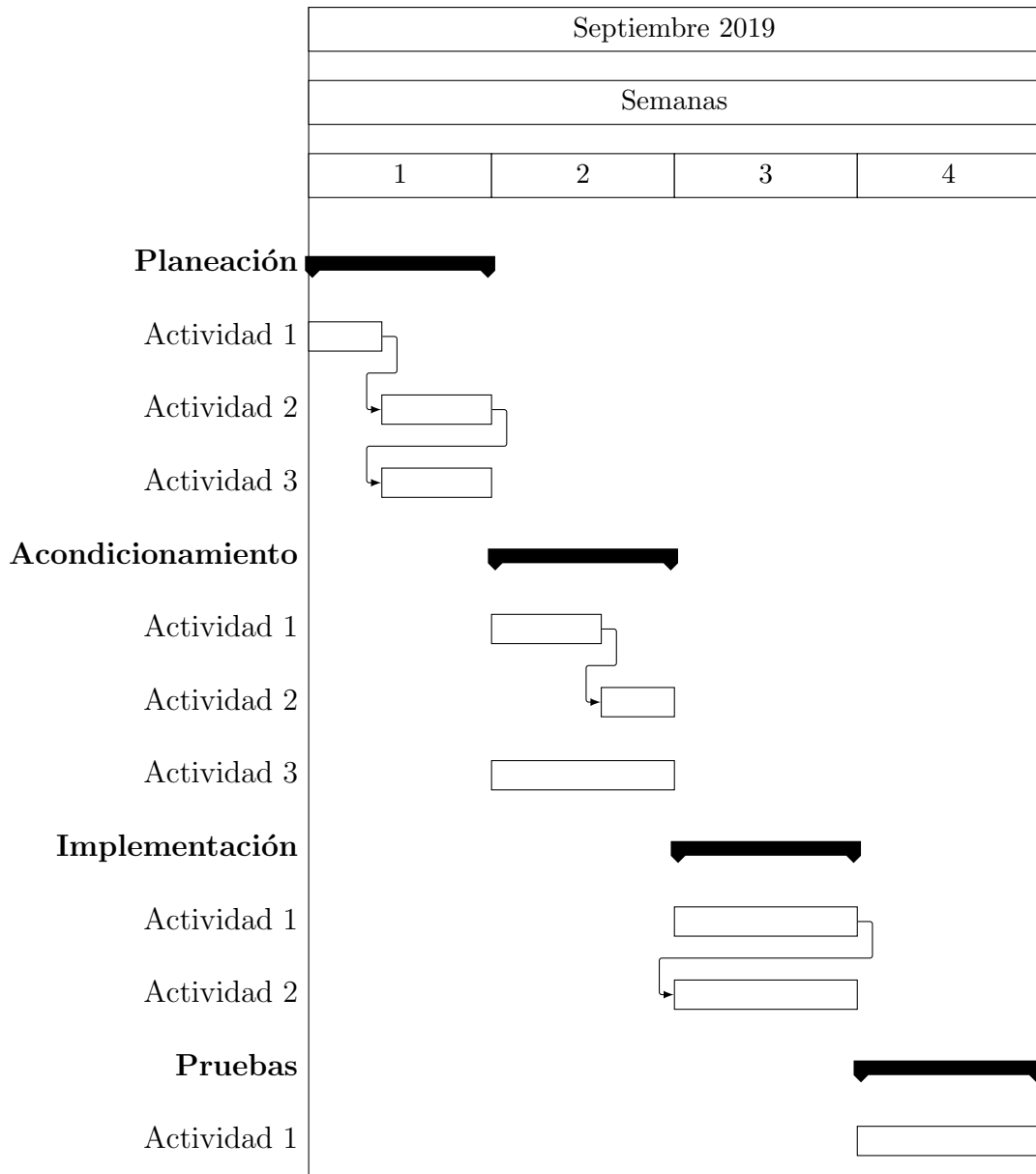


Figura 5.2: Diagrama de Gantt para el plan de implementación.

Capítulo 6

Conclusiones y recomendaciones

6.1. Conclusiones

Se realizó un estudio preliminar para determinar la situación problemática de Administración Académica en relación al manejo de los expedientes estudiantiles y de graduación, consecuentemente se evaluó el sistema anterior que era usado con dicha finalidad; pero con resultados no tan deseables. El anterior sistema carecía de seguridad y facilidad de uso, entre muchas otras cosas detalladas en el capítulo uno de esta investigación.

Se realizó un diagnóstico mediante el cual se llegó a la conclusión que la mejor alternativa es el desarrollo de un nuevo sistema basado en la web para el manejo de los expedientes de Administración Académica.

Mediante el estudio de las factibilidades, se logró comprobar que el sistema web era factible a nivel económico, operativo, legal y técnico.

El sistema fue desarrollado puramente con lenguajes de entorno web (PHP, JavaScript, HTML5 y CSS3) siguiendo los estándares de diseño propuestos y los requerimientos captados de diversas fuentes. Garantizando de esta manera que posea una interfaz intuitiva y una curva de aprendizaje reducida.

Con la implementación de esta alternativa, se automatiza un proceso bastante tedioso y mecánico que realizan los encargados de los procesos académicos de Administración Académica, y se ofrece una solución a la acumulación de expedientes en físico con los problemas inherentes a este proceso.

6.2. Recomendaciones

Se recomienda utilizar la metodología de ciclo de vida de los sistemas para realizar cualquier mejora o cambio en el sistema. También se recomienda respetar los lineamientos de diseño y estándares de desarrollo previamente establecidos, lo cual promueve la correcta escalabilidad de este sistema.

A los encargados de los procesos administrativos de Administración Académica, los cuales serán los encargados de usar el sistema se les recomienda seguir la documentación establecida en el manual de usuario. Y si se desea realizar algún cambio de en su código y/o funciones, verificar el manual de programación proporcionado.

Se recomienda utilizar herramientas de desarrollo bajo licencia de software libre y optar por la que mejor se ajuste a las necesidades del momento.

Todo el cambio o actualización que se realice debe ser documentado para facilitar el soporte y mantenimiento de el mismo.

Se recomienda realizar un respaldo periódico de la información almacenada en la base de datos.

Sin más que agregar, la presente investigación y sistema desarrollado queda a disposición de la Administración Académica de la Facultad Multidisciplinaria Oriental, así como las respectivas actualizaciones, mejoras y soporte que desee darse al mismo.

Bibliografía

- Roger S. Pressman; Ingeniería del Software. Un Enfoque Practico; Sexta Edición; McGraw-Hill
- VERGEL CABRALES, Gustavo. Metodología, Un Manual para la elaboración de diseños y proyectos de investigación. Editorial Mejoras. Tercera edición, Barranquilla, 1997.
- McConnell, Steve; Desarrollo y gestión de proyectos informáticos; McGrawHill, Madrid; 1997
- Carnegie Mellon. SEI Report on Undergraduate Software Engineering Education. Software Engineering Institute, 1990.
- Peter Hughes Paul Fisher, James Mc Daniel. System development life cycle models and methodologies, 2010. Canadian Society for International Health Certificate Course in Health Information Systems Module 3: System Analysis and Database Development Part 3: Life Cycle Models and Methodologies.
- Sergio Luján Mora (2002). Programación de aplicaciones web: historia, principios básicos y clientes web (1ª edición). Editorial Club Universitario.
- M. Domínguez-Dorado, Editorial Iberprensa (Madrid). DL M-13679-2004. Septiembre de 2005. Bases de datos en el cliente con JavaScript DB.
- Kendall, Kenneth & Kendall, Julie. Análisis y Diseño de Sistemas, Tercera. Edición. Prentice Hall. 1997.
- Hernández Sampieri, Roberto; et al. Metodología de la Investigación. 2ª. ed. McGraw-Hill. México, D.F., 2001.
- Trasobares, A. H. (2003). Los sistemas de información: evolución y desarrollo. Proyecto social: Revista de relaciones laborales, (10), 149-165.
- Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., ... & Wolff, S. (1999). Una breve historia de Internet. Revista Novática. Números, 130, 131.
- Morales, K. G. (2003). Fundamentos de programación.
- Elmasri, R. A., & Navathe, S. B. (2007). Fundamentos de sistemas de bases de datos (No. 004.65). Addison Wesley,.

Sitiografía

- https://es.wikipedia.org/wiki/Historia_de_Internet
- https://es.wikipedia.org/wiki/Historia_de_la_World_Wide_Web
- https://es.wikipedia.org/wiki/Historia_de_los_lenguajes_de_programacion
- <https://www.php.net/manual/es/history.php.php>
- <http://api.gobiernoabierto.gob.sv/attachments/10/download>
- <https://www.asamblea.gob.sv/decretos/details/233>
- <https://www.asamblea.gob.sv/decretos/details/3389>
- https://academica.ues.edu.sv/documentos/reglamento_academica.pdf

Anexos

ANEXO #1 FORMATO DE ENTREVISTA DIRIGIDA AL ADMINISTRADOR ACADÉMICO

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



Objetivo: Recopilar información que permita especificar los requerimientos funcionales del nuevo sistema a desarrollar para la gestión de expedientes.

Indicaciones: Responda de forma breve y concisa las siguientes interrogantes.

1. ¿Desea que el software sea local o remoto?
2. ¿Ha pensado en deshacerse de los documentos físicos viejos?
3. ¿Desea que se implemente seguridad en la aplicación?
4. ¿Desea que cada miembro del personal tenga su cuenta de usuario en el software?
5. ¿Desea que se escanee todos los documentos de cada expediente o solo los del F2?
6. ¿Desearía flexibilidad a la hora de determinar que documentos se escanearán?
7. ¿Le gustaría que se maneje cada documento con un archivo aparte o que se tenga un solo (PDF) para el expediente?
8. ¿Desearía que liste en pantalla los expedientes digitales?
9. ¿Como desearía que se liste, por nombre o carnet?
10. ¿Desea que se modifique cada expediente?
11. ¿Desearía que se permita eliminar cada expediente?
12. ¿Que lista de campos desea que tenga cada expedientes, aparte de los documentos?
13. ¿Le gustaría implementar una copia de seguridad de los expedientes digitales?
14. ¿Desearía agregar algo?

ANEXO #2 FORMATO DE CUESTIONARIO DIRIGIDO AL PERSONAL DE LA ADMINISTRACIÓN ACADÉMICA



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

Objetivo: Recopilar información sobre el funcionamiento del sistema actual para la gestión de expedientes y sus necesidades.

Indicaciones: Responda de forma breve y concisa las siguientes interrogantes.

1. ¿Piensa que es de gran importancia en la Administración Académica poder realizar una gestión de los expedientes digitales de los estudiantes?
2. ¿El sistema de gestión de expedientes que posee actualmente la Administración Académica, cumple con todos los requerimientos que se necesitan?
3. ¿En caso de no cumplir todos los requerimientos, podría describir lo que a su criterio le hace falta al actual sistema?
4. ¿Considera que es importante tener la posibilidad de contar con la información de los expedientes en cualquier lugar y en cualquier momento?
5. ¿Cuántas personas hacen uso del sistema de gestión de expedientes actual?
6. A continuación describa los aspectos que NO le gustan del sistema actual.
7. A continuación describa los aspectos que SÍ le gustan del sistema actual.

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



MANUAL DE USUARIO

SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN
DE EXPEDIENTES DE LA ADMINISTRACIÓN
ACADÉMICA EN LA FACULTAD
MULTIDISCIPLINARIA ORIENTAL DE LA
UNIVERSIDAD DE EL SALVADOR

CIUDAD UNIVERSITARIA ORIENTAL, 2019

Índice general

1. Manual de usuario	2
1.1. Introducción	2
2. Pantallas del sistema	3
2.1. Inicio de sesión	3
2.2. Módulo inicio	4
2.3. Ver un estudiante	6
2.4. Administrador de estudiantes	8
2.5. Administrar Expedientes	9
2.6. Usuarios	11
2.7. Configuración	13
2.8. Salir	16

Capítulo 1

Manual de usuario

1.1. Introducción

El presente documento, es una guía de uso sobre el adecuado manejo del sistema informático para la gestión de expedientes de la Administración Académica de la Facultad Multidisciplinaria Oriental de la Universidad de El Salvador. El propósito de esta guía es servir de base para la comprensión y correcta función de todos los procesos que desarrolla el ya mencionado sistema.

Con este sistema informático se pueden realizar pocos pero primordiales procesos, al ser de uso específico, como es el manejo de los expedientes. Explícitamente posee 3 módulos, el de estudiantes, el de expedientes y el de configuración. A continuación se desarrollará de forma más detallada cada uno de ellos y las funciones que permiten realizar.

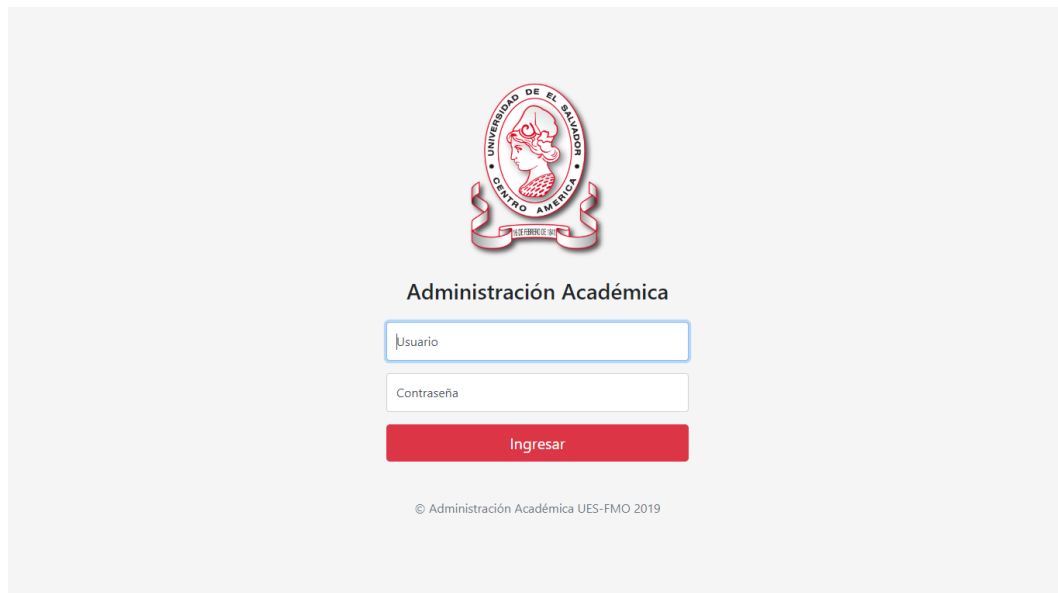
Capítulo 2

Pantallas del sistema

2.1. Inicio de sesión

Al ingresar la dirección del sitio en un navegador web nos lleva a la sección de 'login' o inicio de sesión. Para poder acceder al sistema se debe estar registrado y tener ciertos privilegios. Los pasos a seguir son:

1. Ingresar al sitio web.
2. Ingresar un usuario y contraseña válidos.
3. En caso contrario se muestra una alerta.
4. Listo, se muestra un mensaje de bienvenida acorde al nivel de privilegios del usuario



UNIVERSIDAD DE EL SALVADOR
CENTRO AMÉRICA

Administración Académica

Usuario

Contraseña

Ingresar

© Administración Académica UES-FMO 2019

Figura 2.1: Inicio de sesión

2.2. Módulo inicio

Luego de ingresar correctamente los datos requeridos, se accede al sistema mostrando un mensaje de bienvenida y habilitando los diversos niveles de acceso según el usuario. Estos son:

- **Administrador:** Acceso completo a todos los módulos, permitiendo ver, añadir, modificar, y eliminar cualquier registro.
- **Certificado:** Acceso a los módulos de expediente y estudiante, permitiendo añadir, ver y modificar los registros de ellos. Sin acceso a eliminar registros ni los aspectos de configuración
- **Practicante:** Similar al anterior pero añadiendo la restricción de modificar cualquier registro.



The screenshot shows the 'Administración de expedientes UES-FMO' interface. At the top, there is a navigation menu with 'UES-FMO', 'Inicio', 'Estudiantes', 'Expedientes', 'Usuarios', and 'Configuración'. On the right, it says 'Administrador Principal'. Below the menu is the UES-FMO logo and the title 'Administración de expedientes UES-FMO'. A welcome message 'Bienvenido [Administrador]' is displayed. A red search bar labeled 'Búsqueda por carnet' contains a search input field with the placeholder 'Carnet del estudiante' and a green 'Buscar' button. Below the search bar is a table with the following data:

Carnet	Carrera	Nombres	Apellidos	Acciones	Estado
HF09001	Ing. de Sistemas Informáticos	Diego Armando	Herrera Flores	  	

At the bottom, there is a copyright notice: '© Administración Académica UES-FMO 2019'.

Figura 2.2: Inicio

Además se muestran los últimos estudiantes ingresados al sistema junto con una barra de búsqueda por diversos criterios. Y se activa la pestaña de usuario en la esquina superior derecha mostrando el nombre del usuario activo.

En la columna de estado hay un indicador en forma circular de color rojo si a el expediente le hace falta uno o más expedientes y de color verde si ya posee todos los documentos mínimos requeridos en el sistema.

Si se desea una vista rápida de los documentos ingresados para dicho estudiante o de los requeridos, se da click en el ícono azul y se habilita la pestaña de confrontación, ya se para F2 o expediente de graduación.

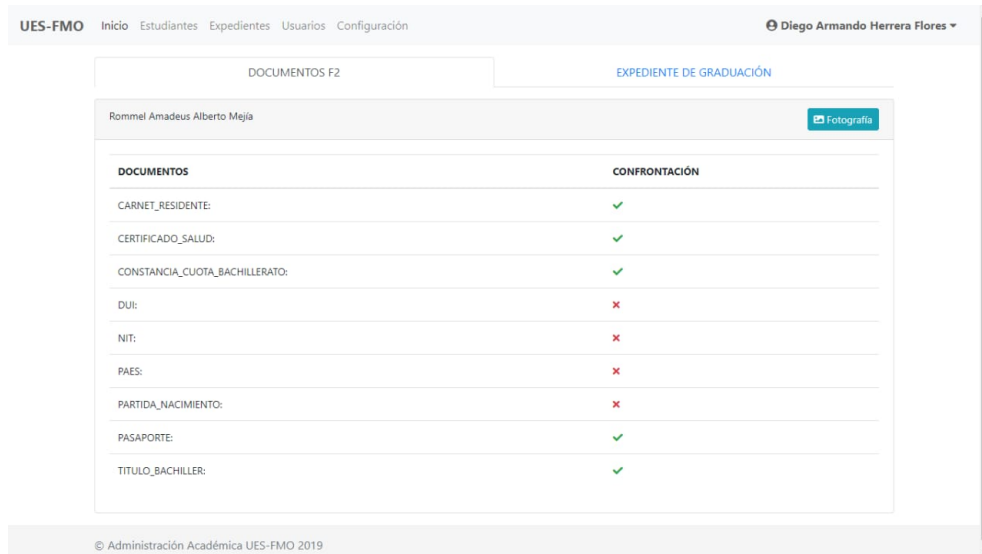


Figura 2.3: Confrontación

Los cheques verdes indican que se ha ingresado correctamente dicho documento y las equis rojas que no se posee registro aún del documento en específico.

2.3. Ver un estudiante

Al hacer click en el ícono celeste con forma de ojo del registro de estudiante que se desea visualizar, se presenta una vista con distintas pestañas incluyendo los datos personales del mismo, su expediente F2, de graduación, y la sección de otros documentos.

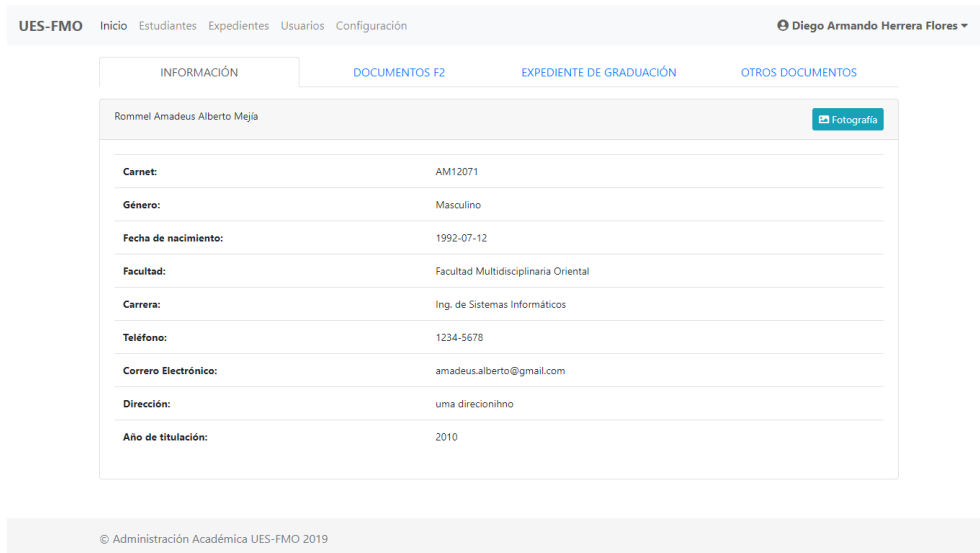


Figura 2.4: Información de un estudiante

Al dar click en la pestaña de 'Documentos F2' se listan los documentos requeridos junto con observaciones y ultima fecha de modificación, en el caso de poseerlas. Permitiendo además ver el documento individual, añadir uno nuevo y guardar el cambio u omitirlo según se desee. De manera similar ocurre con el resto de las pestañas.

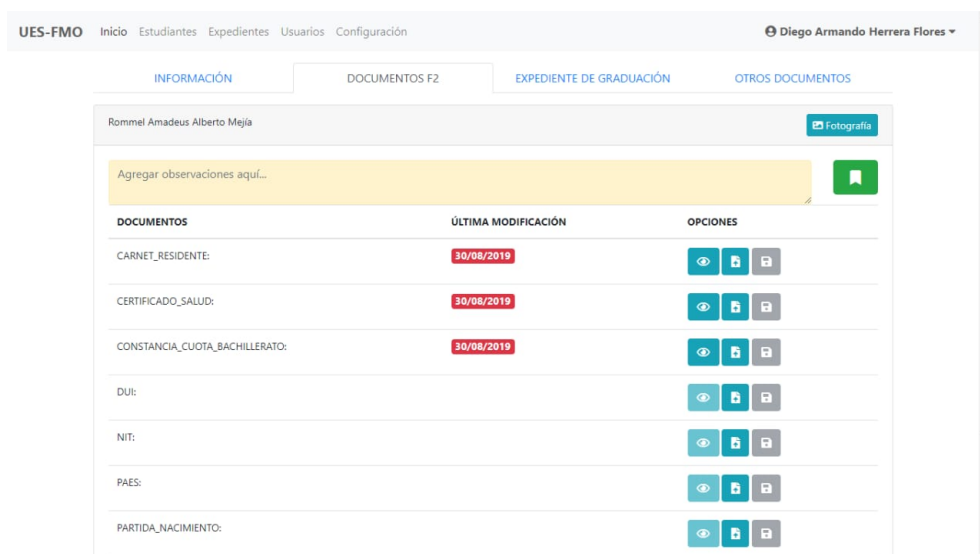


Figura 2.5: Documentos F2

En la misma sección de Expediente F2 en cada uno de los documentos listados aparece un ícono verde con el símbolo de una hoja y una flecha lo que indica subir un documento, la acción genera una ventana emergente en la cual se puede escoger el pdf a subir al sistema.

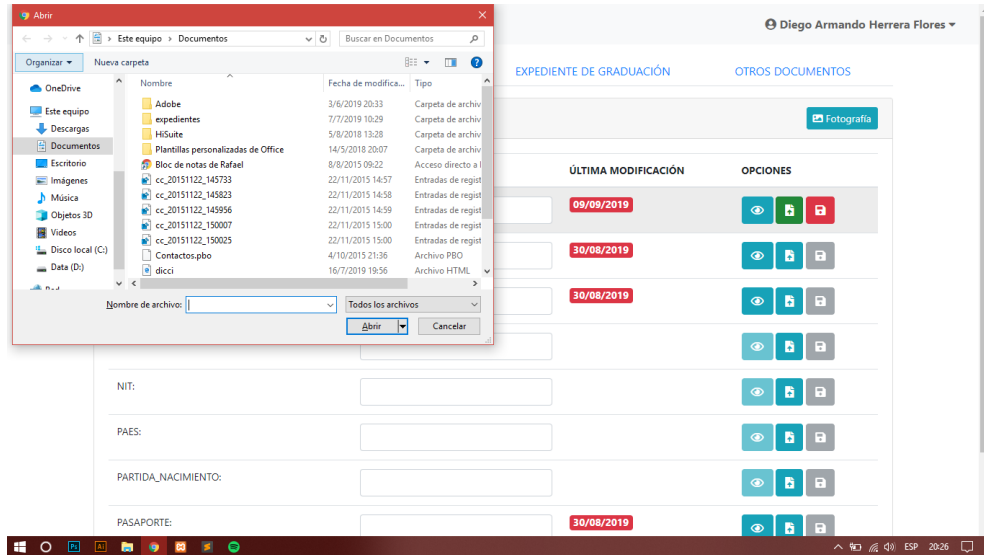


Figura 2.6: Subir PDF

De realizar esta acción correctamente nos muestra el siguiente mensaje.

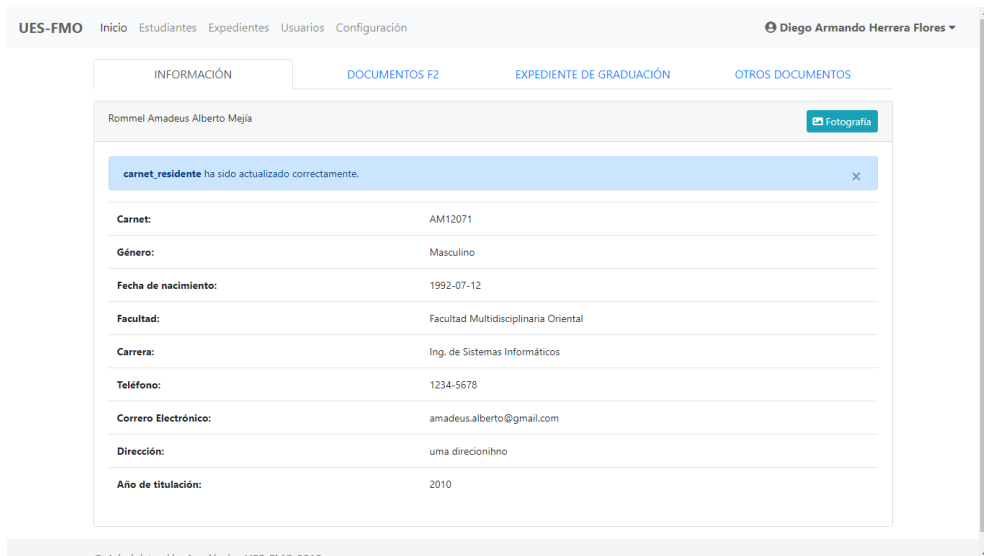
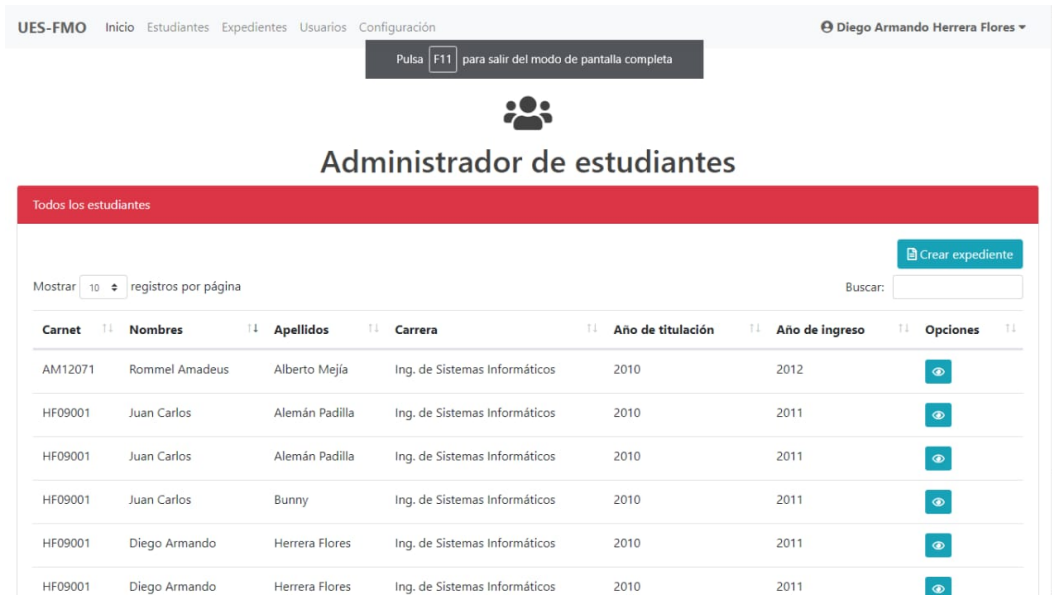


Figura 2.7: Actualizado correctamente

2.4. Administrador de estudiantes

Al escoger de los modulos principales la sección de estudiantes, nos lista todos los registros de estudiantes existentes paginados según el criterio existente y permitiendo la búsqueda entre ellos según diversos filtros. Mostrando el número de registros por página que se desee.

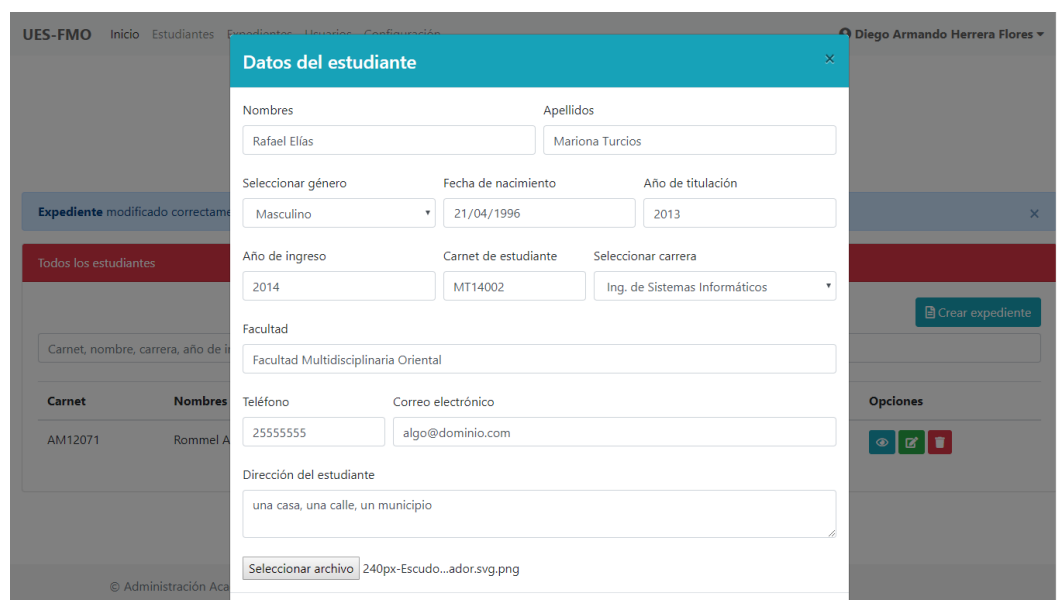


The screenshot shows the 'Administrador de estudiantes' interface. At the top, there is a navigation bar with 'UES-FMO' and links to 'Inicio', 'Estudiantes', 'Expedientes', 'Usuarios', and 'Configuración'. A user profile for 'Diego Armando Herrera Flores' is visible in the top right. Below the navigation, there is a search bar and a 'Crear expediente' button. The main content area displays a table of student records with the following columns: 'Carnet', 'Nombres', 'Apellidos', 'Carrera', 'Año de titulación', 'Año de ingreso', and 'Opciones'. The table contains six rows of data, each with a 'Ver' icon in the 'Opciones' column.

Carnet	Nombres	Apellidos	Carrera	Año de titulación	Año de ingreso	Opciones
AM12071	Rommel Amadeus	Alberto Mejía	Ing. de Sistemas Informáticos	2010	2012	
HF09001	Juan Carlos	Alemán Padilla	Ing. de Sistemas Informáticos	2010	2011	
HF09001	Juan Carlos	Alemán Padilla	Ing. de Sistemas Informáticos	2010	2011	
HF09001	Juan Carlos	Bunny	Ing. de Sistemas Informáticos	2010	2011	
HF09001	Diego Armando	Herrera Flores	Ing. de Sistemas Informáticos	2010	2011	
HF09001	Diego Armando	Herrera Flores	Ing. de Sistemas Informáticos	2010	2011	

Figura 2.8: Administrar estudiantes

Además en este mismo módulo, arriba a la derecha en color azul, se encuentra el botón para registrar un nuevo estudiante, al seleccionarlo nos muestra el siguiente formulario.



The screenshot shows the 'Datos del estudiante' form. The form is divided into several sections with input fields for the following information:

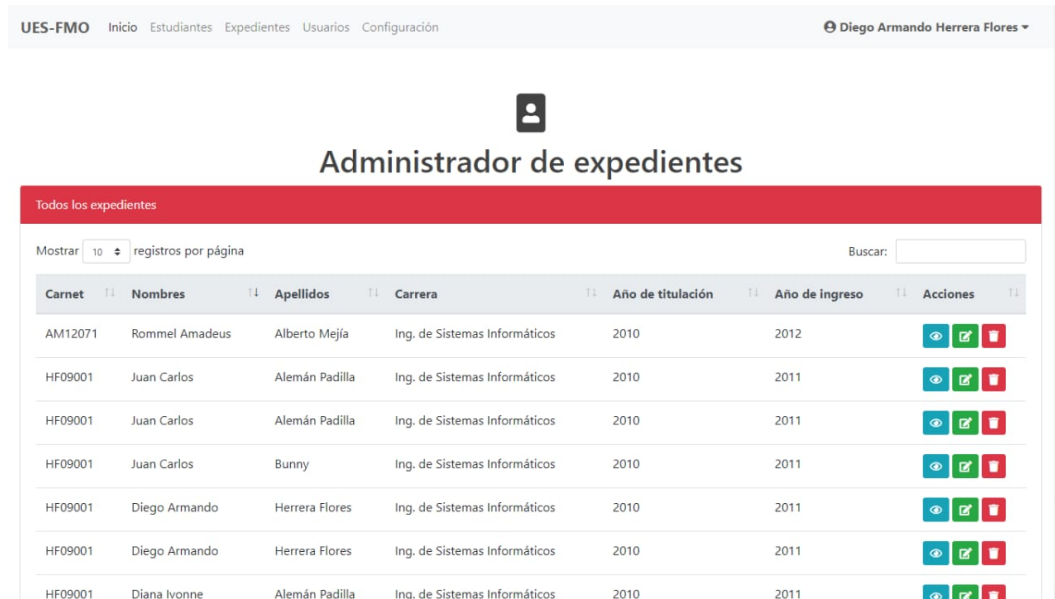
- Nombres:** Rafael Elías
- Apellidos:** Mariona Turcios
- Seleccionar género:** Masculino
- Fecha de nacimiento:** 21/04/1996
- Año de titulación:** 2013
- Año de ingreso:** 2014
- Carnet de estudiante:** MT14002
- Seleccionar carrera:** Ing. de Sistemas Informáticos
- Facultad:** Facultad Multidisciplinaria Oriental
- Teléfono:** 25555555
- Correo electrónico:** algo@dominio.com
- Dirección del estudiante:** una casa, una calle, un municipio
- Seleccionar archivo:** 240px-Escudo...ador.svg.png

Figura 2.9: Añadir nuevo estudiante

Al ingresar todos los campos requeridos correctamente y crear el nuevo registro correctamente se muestra un mensaje indicándolo.

2.5. Administrar Expedientes

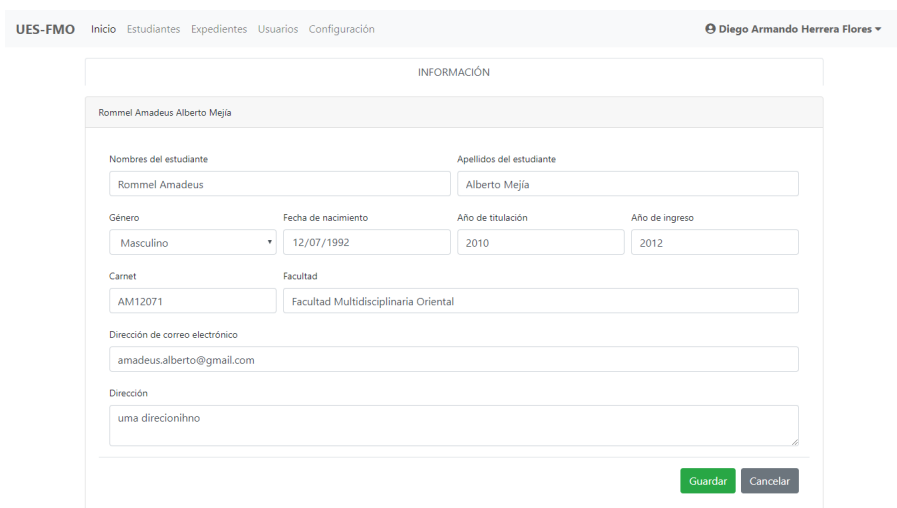
Dentro de los módulos principales tenemos la sección de expedientes, al hacer click veremos la siguiente pantalla listando los expedientes registrados para todos los estudiantes, estos pueden ser tanto como documentos F2 o expediente de graduación. Se muestra de la siguiente manera.



Carnet	Nombres	Apellidos	Carrera	Año de titulación	Año de ingreso	Acciones
AM12071	Rommel Amadeus	Alberto Mejía	Ing. de Sistemas Informáticos	2010	2012	[Ver] [Editar] [Eliminar]
HF09001	Juan Carlos	Alemán Padilla	Ing. de Sistemas Informáticos	2010	2011	[Ver] [Editar] [Eliminar]
HF09001	Juan Carlos	Alemán Padilla	Ing. de Sistemas Informáticos	2010	2011	[Ver] [Editar] [Eliminar]
HF09001	Juan Carlos	Bunny	Ing. de Sistemas Informáticos	2010	2011	[Ver] [Editar] [Eliminar]
HF09001	Diego Armando	Herrera Flores	Ing. de Sistemas Informáticos	2010	2011	[Ver] [Editar] [Eliminar]
HF09001	Diego Armando	Herrera Flores	Ing. de Sistemas Informáticos	2010	2011	[Ver] [Editar] [Eliminar]
HF09001	Diana Ivonne	Alemán Padilla	Ing. de Sistemas Informáticos	2010	2011	[Ver] [Editar] [Eliminar]

Figura 2.10: Administrador de expedientes

También en esta misma sección se puede modificar los datos personales de los estudiantes ya registrados en el sistema, si el usuario activo posee dichos privilegios, claro está. A la derecha de el nombre aparece un ícono verde representando un lapiz sobre una hoja. Al hacer click nos muestra lo siguiente.



INFORMACIÓN

Rommel Amadeus Alberto Mejía

Nombres del estudiante: Rommel Amadeus

Apellidos del estudiante: Alberto Mejía

Género: Masculino

Fecha de nacimiento: 12/07/1992

Año de titulación: 2010

Año de ingreso: 2012

Carnet: AM12071

Facultad: Facultad Multidisciplinaria Oriental

Dirección de correo electrónico: amadeus.alberto@gmail.com




Dirección: una direccionihno

Guardar Cancelar

Figura 2.11: Editar datos de estudiante

Luego de editar la información deseada se procede a guardar los cambios y aparece el siguiente mensaje en caso de ser una operación exitosa.

The screenshot displays the 'Administrador de estudiantes' interface. At the top, there is a navigation bar with 'UES-FMO' and menu items: 'Inicio', 'Estudiantes', 'Expedientes', 'Usuarios', and 'Configuración'. The user's name 'Diego Armando Herrera Flores' is shown in the top right. Below the navigation bar is a header with a group of people icon and the title 'Administrador de estudiantes'. A blue notification banner at the top of the main content area reads 'Expediente modificado correctamente.' with a close button. Below this is a red bar with the text 'Todos los estudiantes'. A search bar contains the placeholder text 'Carnet, nombre, carrera, año de ingreso, año de egreso'. To the right of the search bar is a 'Crear expediente' button. Below the search bar is a table with the following data:

Carnet	Nombres	Apellidos	Carrera	Opciones
AM12071	Rommel Amadeus	Alberto Mejía	Ing. de Sistemas Informáticos	  





At the bottom of the page, there is a footer with the text '© Administración Académica UES-FMO 2019'.

Figura 2.12: Modificación exitosa

2.6. Usuarios

En el módulo usuarios al que solo se puede acceder con privilegios de administrador, se permite añadir, eliminar y modificar los registros de las personas que poseen acceso al sistema. Su vista es la siguiente.

The screenshot shows the 'Administrador de usuarios' interface. At the top, there is a navigation bar with 'UES-FMO' and menu items: 'Inicio', 'Estudiantes', 'Expedientes', 'Usuarios', and 'Configuración'. On the right, the user 'Diego Armando Herrera Flores' is logged in. The main heading is 'Administrador de usuarios' with a user icon. Below this is a red bar labeled 'Todos los usuarios' and a green 'Nuevo' button. A search box for 'Nombre de usuario' is present. The main content is a table with the following data:

Nombres	Apellidos	Usuario	Opciones
Diego Armando	Herrera Flores	diego	 
Rommel Amadeus	Alberto Mejia	amadeus	 

At the bottom, there is a footer: '© Administración Académica UES-FMO 2019'.

Figura 2.13: Administrar usuarios

Se listan los usuarios actuales y se habilitan dos botones a su derecha que permiten editar y eliminar respectivamente a dicho usuario. Al hacer click el botón verde con el ícono de editar se muestra el siguiente formulario.

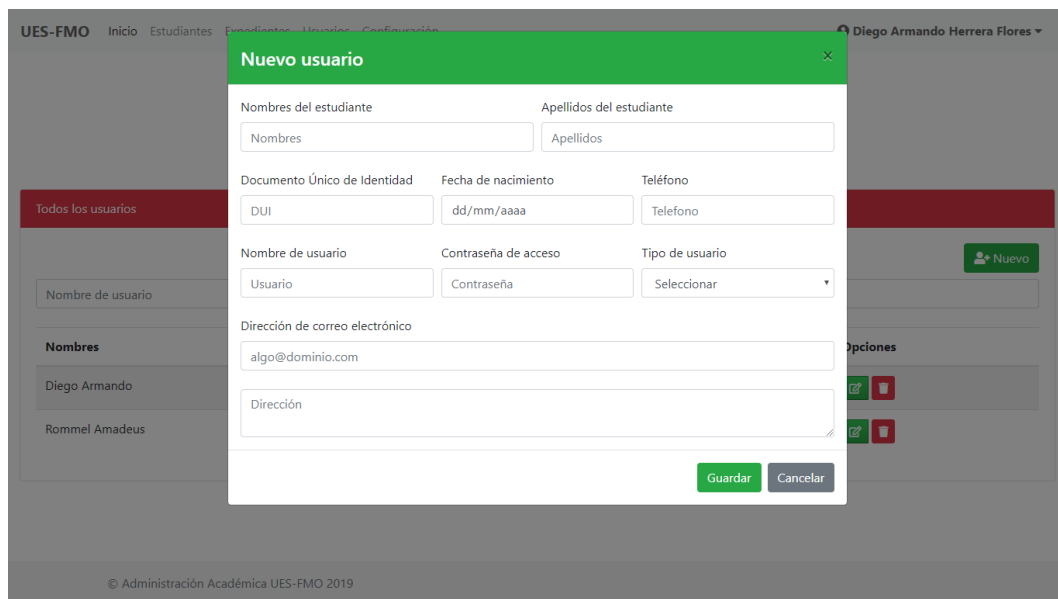
The screenshot shows the 'INFORMACIÓN' form for editing user data. The navigation bar is the same as in Figure 2.13. The form title is 'INFORMACIÓN' and the user name is 'Rommel Amadeus Alberto Mejia'. The form contains the following fields:

- Nombres del usuario: Rommel Amadeus
- Apellidos del usuario: Alberto Mejia
- Documento Único de Identidad: 04671802-3
- Fecha de nacimiento: 12/07/1992
- Teléfono: 76317091
- Nombre de usuario: amadeus
- Contraseña de acceso: Dejar vacío para no cambiar
- Tipo de usuario: Administrador
- Dirección de correo electrónico: amadeus.alberto@gmail.com
- Dirección: 17° calle poniente #423 Barrio San Nicolas, San Mi

At the bottom right, there are 'Guardar' and 'Cancelar' buttons.

Figura 2.14: Editar información de usuarios

Además arriba a la derecha se encuentra un botón verde con la leyenda nuevo usuario, permitiendo crear un nuevo perfil para la persona que usará el sistema, al seleccionarlo se muestra la siguiente pantalla.



The image shows a web application interface for UES-FMO. A modal window titled "Nuevo usuario" is open, displaying a form for creating a new user profile. The form includes the following fields and controls:

- Nombres del estudiante:** Input field labeled "Nombres".
- Apellidos del estudiante:** Input field labeled "Apellidos".
- Documento Único de Identidad:** Input field labeled "DUI".
- Fecha de nacimiento:** Input field with a date mask "dd/mm/aaaa".
- Teléfono:** Input field labeled "Telefono".
- Nombre de usuario:** Input field labeled "Usuario".
- Contraseña de acceso:** Input field labeled "Contraseña".
- Tipo de usuario:** A dropdown menu labeled "Seleccionar".
- Dirección de correo electrónico:** Input field containing the text "algo@dominio.com".
- Dirección:** A larger input field for the address.

At the bottom right of the form are two buttons: "Guardar" (green) and "Cancelar" (grey). The background shows a sidebar with a "Nuevo" button and a list of users including "Diego Armando" and "Rommel Amadeus". The footer of the page reads "© Administración Académica UES-FMO 2019".

Figura 2.15: Añadir nuevo usuario

Ingrese los campos requeridos de forma correcta y proceda a guardar el registro.

2.7. Configuración

En el módulo de configuración se encuentran las generalidades del sistema, estas ofrecen diversas funciones. Permite añadir nuevos departamentos y carreras que sirven a la hora de registrar un nuevo estudiante, también se encuentra la función para el respaldo de toda la información de la base de datos en un solo archivo, y además se encuentra una bitácora con el registro de toda la actividad realizada en el sistema de gestión de expedientes.

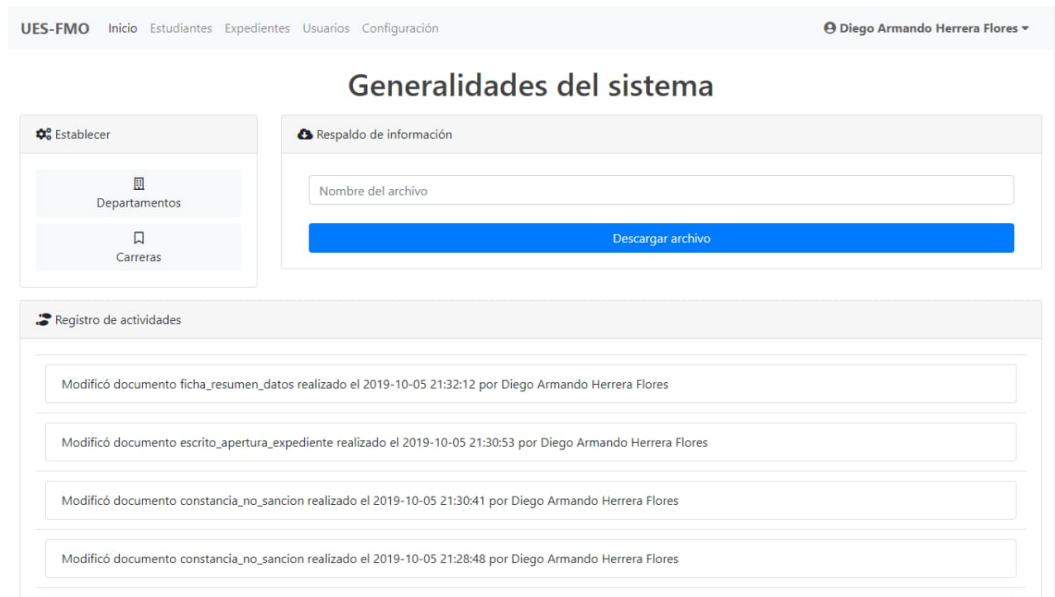


Figura 2.16: Configuración del sistema

Al seleccionar el botón con el símbolo y la leyenda departamentos se carga la siguiente vista en la que se puede añadir, editar, y eliminar dichos registros.

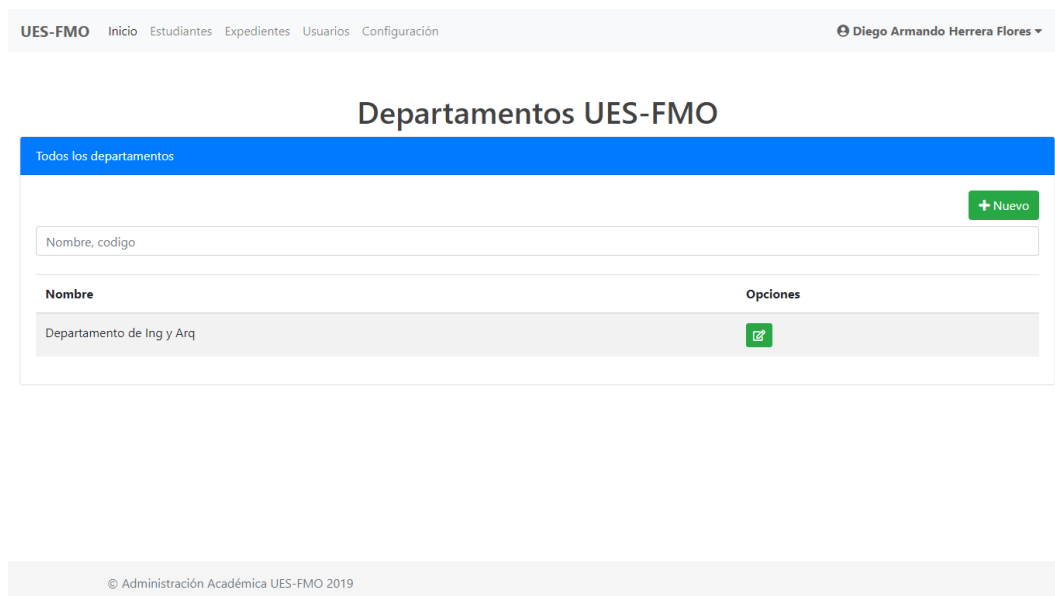


Figura 2.17: Departamentos

De manera similar sucede con el apartado carreras.



Figura 2.18: Carreras

En la que podemos añadir nuevas carreras de acorde a sus departamentos respectivos. Esto garantiza la escalabilidad del sistema.



Figura 2.19: Carreras

Se puede acceder a la bitácora completa del sistema con el botón que lo indica en la parte inferior. E incluye un filtro de búsqueda para encontrar una acción en específico.

UES-FMO Inicio Estudiantes Expedientes Usuarios Configuración Diego Armando Herrera Flores

Registro de actividades

Mostrar 10 registros por página Buscar:

Acción	Fecha y hora	Responsable
Creó carrera	realizado el 2019-10-08 22:06:36	por Diego Armando Herrera Flores
Modificó estudiante HF09001	realizado el 2019-10-08 21:54:43	por Diego Armando Herrera Flores
Modificó documento ficha_resumen_datos	realizado el 2019-10-05 21:32:12	por Diego Armando Herrera Flores
Modificó documento escrito_apertura_expediente	realizado el 2019-10-05 21:30:53	por Diego Armando Herrera Flores
Modificó documento constancia_no_sancion	realizado el 2019-10-05 21:30:41	por Diego Armando Herrera Flores
Modificó documento constancia_no_sancion	realizado el 2019-10-05 21:28:48	por Diego Armando Herrera Flores
Modificó documento constancia_cum_honorífico	realizado el 2019-10-05 21:20:53	por Diego Armando Herrera Flores
Modificó documento recibos_pago_graduacion	realizado el 2019-10-05 21:20:33	por Diego Armando Herrera Flores
Modificó documento declaracion_jurada	realizado el 2019-10-05 21:20:15	por Diego Armando Herrera Flores
Modificó documento solvencia_acme	realizado el 2019-10-04 22:09:29	por María Elena Gutierrez Portillo

Mostrando 67 de 67 Registros

Anterior 1 2 3 4 5 6 7 Siguiente

Figura 2.20: Bitácora

2.8. Salir

En todo momento dentro del sistema de gestión de expedientes se muestra en la esquina superior derecha el nombre del usuario con la sesión activa, al hacer click se muestra el siguiente menú desplegable.

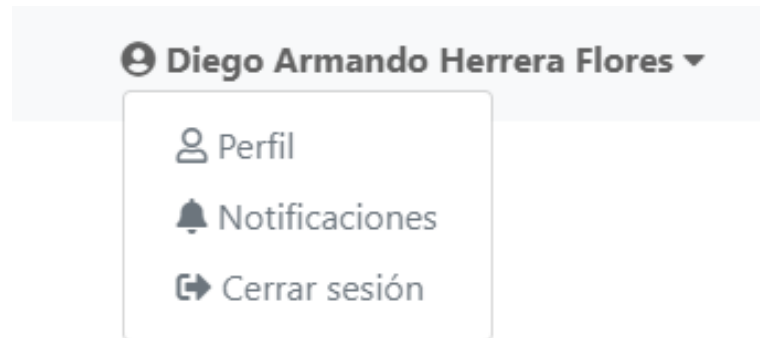


Figura 2.21: Perfil

En ese apartado se puede manejar el perfil, las notificaciones existentes y cerrar la sesión. Al seleccionar cerrar sesión se cierra el sistema regresando a la pantalla principal, para evitar que alguien sin autorización acceda al mismo.

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



MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

SISTEMA INFORMÁTICO WEB PARA LA
GESTIÓN DE EXPEDIENTES DE LA
ADMINISTRACIÓN ACADÉMICA EN LA
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DE LA UNIVERSIDAD DE EL SALVADOR

CIUDAD UNIVERSITARIA ORIENTAL, OCTUBRE 2019

Índice

1. Instalación del software necesario en Windows	2
1.1. Instalación de Netbeans 8.2	2
1.2. Instalación de XAMPP	6
2. Configuración de GNU/Linux	11
2.1. Instalación de Apache	11
2.2. Instalación y configuración de Mysql	11
2.3. Instalación de PHP	12
2.4. Instalar phpmyadmin	12

1. Instalación del software necesario en Windows

1.1. Instalación de Netbeans 8.2

Como se ha optado por el uso de Netbeans como IDE para escribir el proyecto, se procede a la descarga del mismo desde la página oficial, seleccionándose necesariamente la versión PHP de Netbeans, o la versión completa, que trae para desarrollar para PHP, HTML5, CSS3, JavaScript, etc. Una vez termine la descarga hacemos doble clic en el archivo y seleccionamos 'Abrir como administrador'.

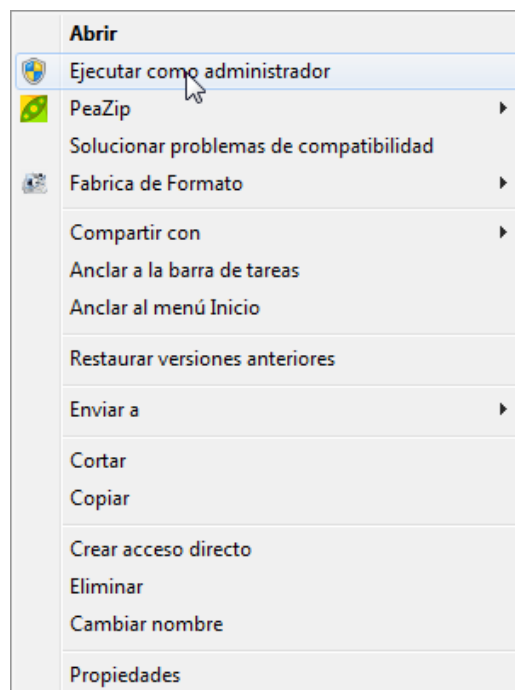


Figura 1:

Ahora se abre la ventana de instalación, solo haga clic en "Next".

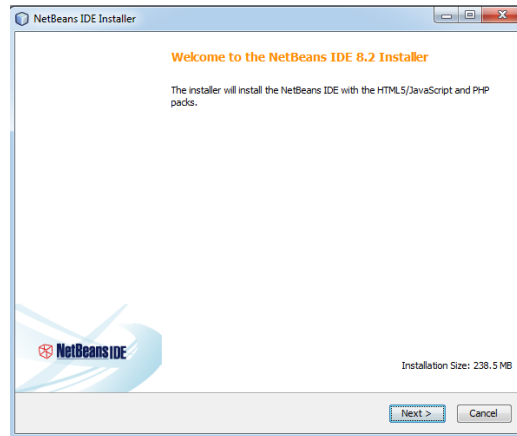


Figura 2:

Haga clic en aceptar los términos de la licencia, y luego en Siguiente.

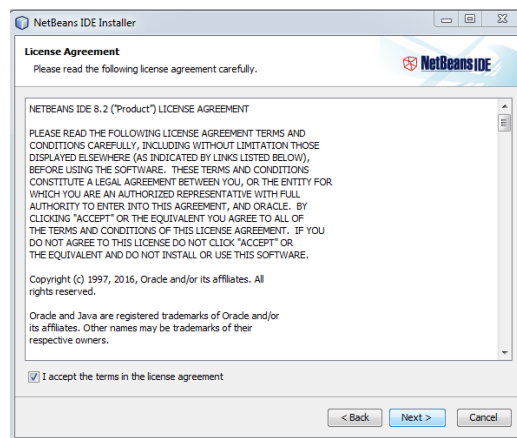


Figura 3:

Hoy, seleccione la carpeta a instalar, o deje la carpeta que esta por defecto.

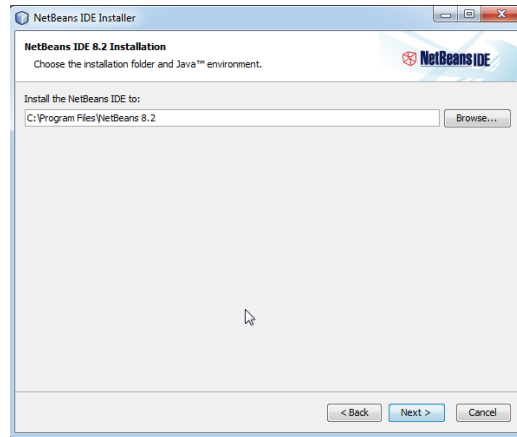


Figura 4:

Verifique la ruta especificada y si desea presione en "Verificar Actualizaciones".

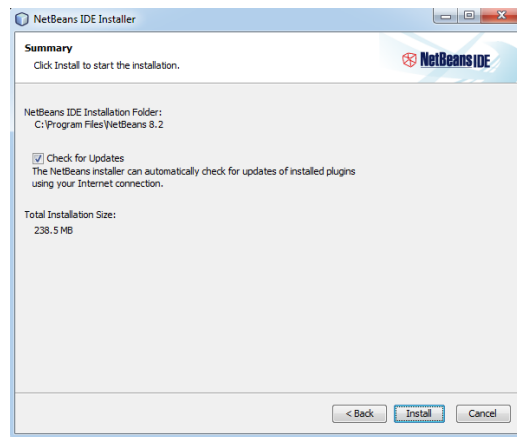


Figura 5:

Espera a que la instalación finalice, tomará unos minutos.

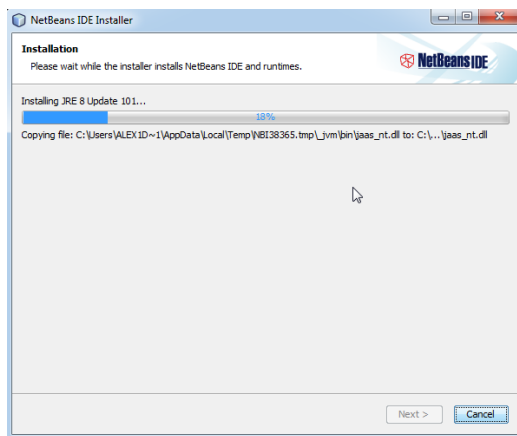


Figura 6:

Por último, de clic en finalizar, luego de la espera, y la instalación estará completa.

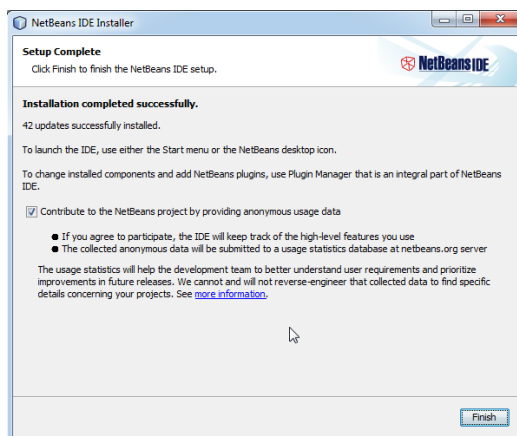


Figura 7:

1.2. Instalación de XAMPP

Primero haga clic derecho sobre el instalador de XAMPP, y seleccione: Abrir como administrador.

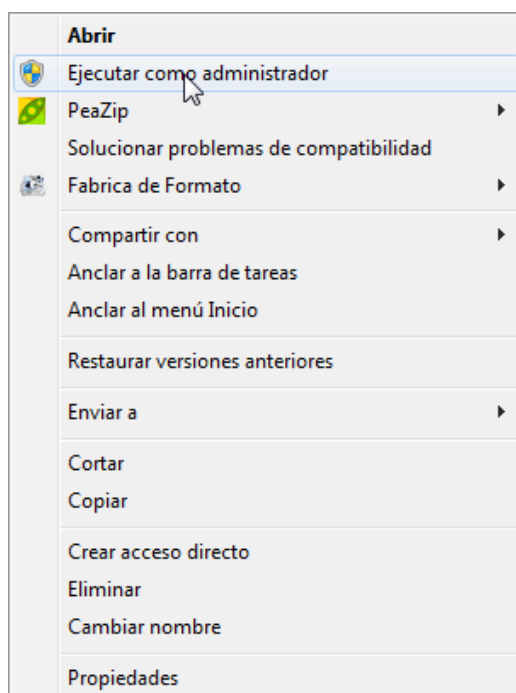


Figura 8:

Una vez se abre, haga clic en siguiente en la ventana abierta.

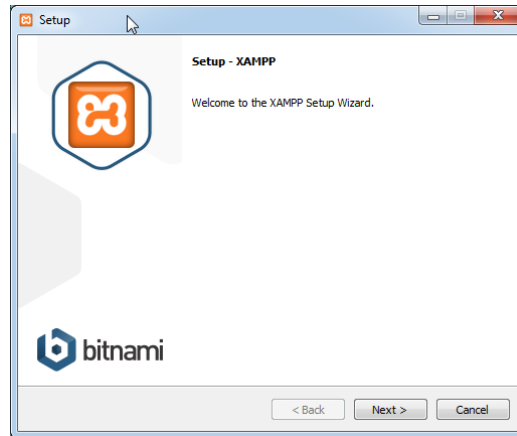


Figura 9:

Seleccione los elementos que quiere instalar.

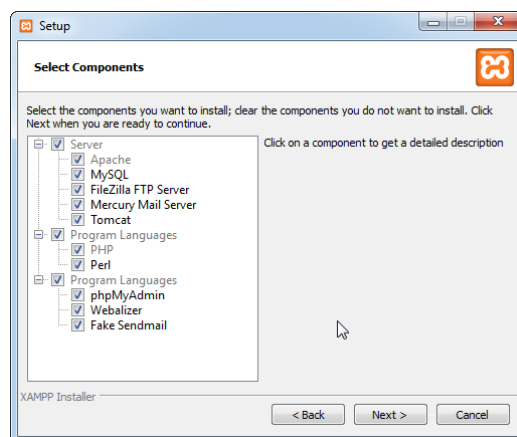


Figura 10:

Seleccione el raíz de la instalación.

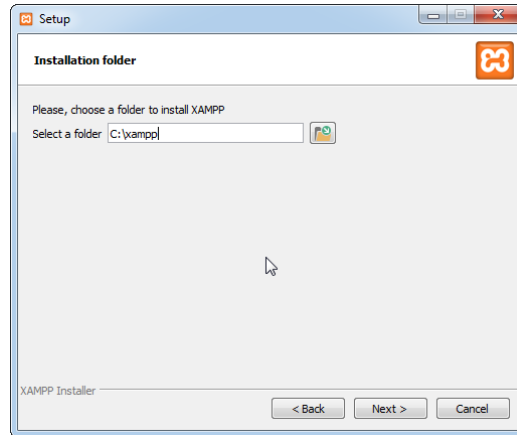


Figura 11:

Ahora solo haga clic en siguientes.

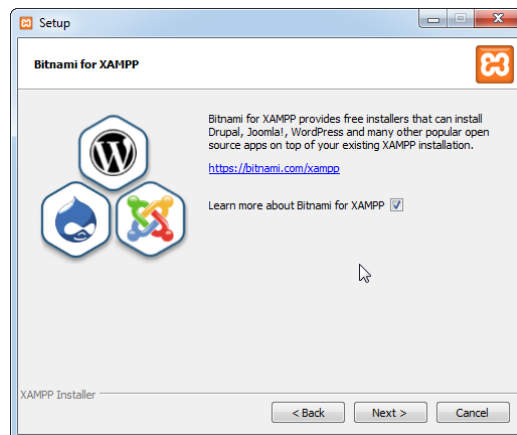


Figura 12:

Ahora solo haga clic en siguientes.

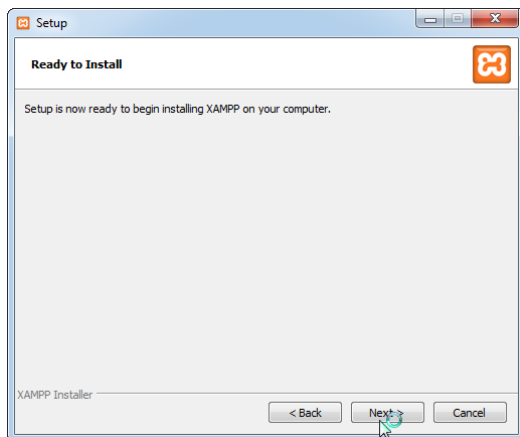


Figura 13:

Ahora espere a que se efectúe la instalación, tardará unos minutos.



Figura 14:

Ahora que se ha instalado, de clic en finalizar.

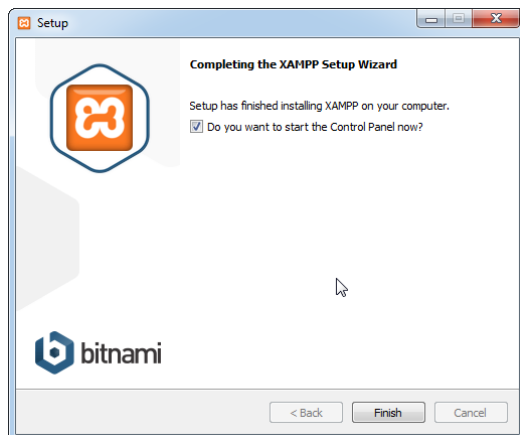


Figura 15:

Para ejecutar, de doble clic sobre el icono de xampp en el escritorio y seleccione el idioma, en este caso ingles.



Figura 16:

2. Configuración de GNU/Linux

En los repositorios de cualquier distro hay software disponible y necesario para configurar cualquier servidor, en nuestro caso un servidor web. GNU/Linux es un sistema operativo perfecto para servidores web, y de hecho es el más usado en el mundo para este fin. A continuación se verá una serie de pasos para instalar y configurar un servidor web para nuestro software en GNU/Linux.

2.1. Instalación de Apache

Ejecutamos el siguiente comando:

```
$ sudo apt update
$ sudo apt install apache2
```

Figura 17:

Apache 2 será instalado en el equipo, aunque sin MySQL ni PHP, por si solo no ejecutará lo que determinara la necesidad de instalar más software.

2.2. Instalación y configuración de MySQL

Ejecutamos el siguiente comando:

```
$ sudo apt install mysql-server
```

Figura 18:

Para configurar MySQL, es necesario ejecutar este comando:

```
$ sudo mysql_secure_installation
```

Figura 19:

Solo es necesario seguir los pasos indicados en el comando, para restaurar o determinar la contraseña para el usuario root de MySQL, para luego no tener problemas en la configuración del servidor al conectarse a la base de datos.

2.3. Instalación de PHP

Ahora instalaremos PHP, con este comando:

```
$ sudo apt install php libapache2-mod-php php-mysql
```

Figura 20:

Aquí instalamos el interprete PHP, además de las librerías que lo unen a Apache2 (libapache2-mod-php) y para unirse a MySQL (php-mysql). Con esto basta para instalar PHP.

2.4. Instalar phpmyadmin

Para facilitar la administración y creación de la base de datos, una herramienta útil es Phpmyadmin, que de forma gráfica, a través de la web, nos permite gestionar tablas, relaciones, y claro esta, sus contenidos. Será especialmente útil en este software para la rápida solución de problemas que se presentarían en relación a la base de datos.

Ejecutamos este comando:

```
$ sudo apt-get update
$ sudo apt-get install phpmyadmin
```

Figura 21:

A continuación, nos hará una serie de preguntas como nombre de usuario y contraseña para la base de datos phpmyadmin. En caso de no hacer preguntas, y que no permite phpmyadmin el acceso con usuario y contraseña, puede configurar phpmyadmin con el comando: `dpkg-reconfigure phpmyadmin`, donde seleccionaría como usuario por defecto para phpmyadmin, el usuario root, procure ejecutar el comando `mysql_secure_installation`, para reconfigurar todo, incluyendo la contraseña para el root de mysql. Por último, en un navegador, escriba 'localhost/phpmyadmin' (o 127.0.0.1 en para la ip local) y procure iniciar sesión con el root y la contraseña que configuró para el root, si accede, está listo para usar phpmyadmin.

En phpmyadmin. crearemos una base de datos con el nombre de 'expedientes', una vez creada, buscamos la opción 'importar', e importamos la base de datos 'expedientes' en mysql, tenemos así la fuente y almacenamiento de información para nuestro software.

Ahora ya está listo para instalar el software en el servidor, basta con copiar el directorio en la ruta '/var/www/html/'.

Ejecutando este comando lo haría:

```
'cp -r [directorio a copiar] /var/www/html/'
```

Puede configurar el usuario dueño del directorio para que sea apache:

```
'chown www-data:www-data -R /var/www/html/[directorio del servidor]'
```

Por último, otorguemos permisos para evitar problemas a futuro:

```
'chmod 777 -R /var/www/html/[directorio del servidor]'
```

Ahora podemos acceder a nuestro software escribiendo en la barra de direcciones la dirección:

```
'localhost/expedientes'
```

Tenemos la ventana de inicio de sesión, escribimos el usuario y contraseña y estamos adentro.

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



MANUAL DE PROGRAMACIÓN

SISTEMA INFORMÁTICO WEB PARA LA
GESTIÓN DE EXPEDIENTES DE LA
ADMINISTRACIÓN ACADÉMICA EN LA
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DE LA UNIVERSIDAD DE EL SALVADOR

CIUDAD UNIVERSITARIA ORIENTAL, 2019

Índice

1. Introducción	2
2. Estructura de directorios	3
3. Directorio 'core'	4
4. Directorios de recurso varios	18
5. Conclusión	20

1. Introducción

El software para administración de expedientes de Administración Académica, es un software que será de gran importancia para la institución, ya que facilitará el manejo de documentos relacionados con los estudiantes, es decir, los expedientes. Este, por ende, debería ser capaz de ser mejorado o corregido a futuro de ser necesario, es por ello que se ha desarrollado el software de manera escalable, usando archivos controladores separados de las vistas que funcionan como entrada y salida, además de el uso de programación orientado a objetos, de frameworks e IDEs correctos para la construcción de este trabajo.

2. Estructura de directorios

Aquí se hablará de la estructura de directorios y archivos que componen el proyecto, para entender su funcionamiento.

La estructura es la siguiente:

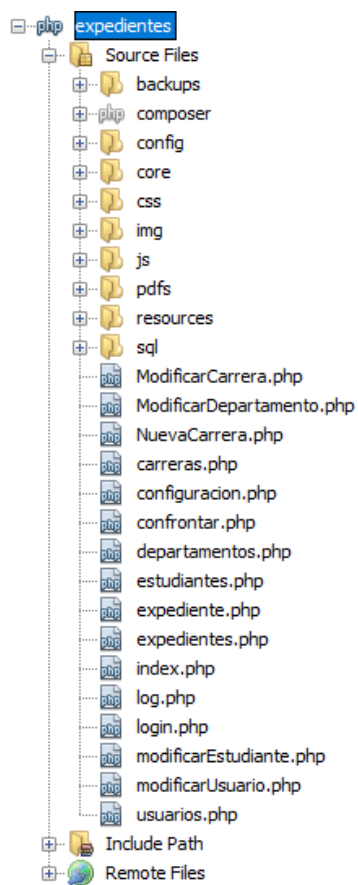


Figura 1:

Los dos directorios que mas toman atencion son core y el directorio raiz. En el directorio core se encuentran los archivos con las clases que funcionan como controladores, y demas archivos que influncian directamente en la funcionalidad del software.

Los archivos en el directorio raiz sirve como las páginas visibles para el usuario, Tenemos las páginas que muestran listados de usuarios, alumnos, expedientes, configuracion con departamentos y carreras, asi como sus respectivas páginas para CRUD, aparte tenemos la venta de login e index, que lógicamente es donde empezamos a usar el software.

3. Directorio 'core'

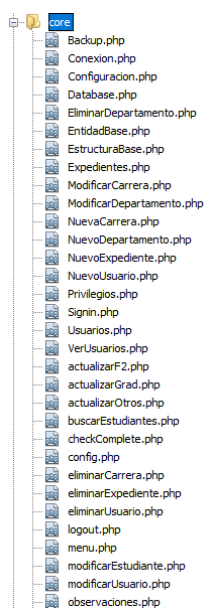


Figura 2:

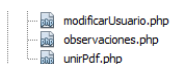


Figura 3:

En el directorio 'core', se encuentran los siguientes archivos:

- Backup.php archivo que contiene la clase para guardar el backup de la base de datos al mandar la orden desde el software.

```
<?php
/*
 * Clase PHP que nos permite generar un respaldo de la Base de datos.
 */

require_once 'config.php';
require_once 'Database.php';
require_once '../composer/vendor/autoload.php';
use Illuminate\Database as Illuminate;
class Backup extends Database {

    public function Backup() {
        parent::__construct();
    }

    public function backupBd() {
        session_start();

        try {
            $nombreBackup = $_POST['nombreBackup'];
            date_default_timezone_set('America/El_Salvador');
            $backupFile = $nombreBackup . "_" . date("Y-m-d-H-i-s") . ".sql";
        }
    }
}
```

Figura 4:

```
try {
    $nombreBackup = $_POST['nombreBackup'];
    date_default_timezone_set('America/El_Salvador');
    $backupFile = $nombreBackup . "_" . date("Y-m-d-H-i-s") . ".sql";

    $dump = new Illuminate\Database\MySqlDump($mysqlHost, $mysqlUser, $mysqlPass, $mysqlDatabase, $root, '');
    $dump->start($backupFile, $backupFile);
    header('Location: ../backup/' . $backupFile);
    unlink($backupFile);
} catch (Exception $e) {
    echo "mysqlDump.php error: " . $e->getMessage();
}

$backup = (new Backup());
$backup->Backup();
```

Figura 5:

- Conexion.php: Aquí se encuentra la clase que define la configuración base de la conexión a la base de datos.

```
class Conexion {
    private $driver, $host, $user, $pass, $database, $charset;

    public function __construct() {
        $db_config = require_once 'database.php';
        $this->driver = $db_config['driver'];
        $this->host = $db_config['host'];
        $this->user = $db_config['user'];
        $this->pass = $db_config['pass'];
        $this->database = $db_config['database'];
        $this->charset = $db_config['charset'];
    }
}
```

Figura 6:

```

public function conectar() {
    if ($this->driver == 'mysql' || $this->driver == null) {
        $con = new mysqli($this->host, $this->user, $this->pass, $this->db);
        $con->query("SET NAMES '".$this->charset."'");
    }
    return $con;
}
}

```

Figura 7:

- Configuracion.php: Aquí se alojan la clase con los métodos que configuran el software en general.

```

/*
 * Archivo que contiene la clase que muestra todos los usuarios registrado
 */
require_once("Database.php");

class Configuracion extends Database
{
    public function Configuracion()
    {
        parent::__construct();
    }
    public function mostrarCarreras()
    {
        try {
            $query = "SELECT * FROM carreras";
            $stmt = $this->con->prepare($query);
            $stmt->execute();
        }
    }
}

```

Figura 8:

```

$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
$stmt->closeCursor();
$this->conn = null;

$carreras = '';
foreach ($result as $key) {
    $carreras .= '
    <tr>
        <td scope="row">'.$key['codigo'].'</td>
        <td>'.$key['nombre'].'</td>
        <td>'. Configuracion::mostrarDeptoById($key['id']).'</td>
        <td>
            <a href="ModificarCarrera.php?idCarrera='.$key['id'].'">
                <!--<button onclick="eliminarCarrera('.$key['id'].'>
            </td>
        </tr>';
}
//mysqli_close($db);
return $carreras;

```

Figura 9:

Otros archivos en 'core' son:

- Database.php: La clase que crea la conexión a la base de datos y que define métodos de consulta, de aquí se heredan las de mas clases.

- EliminarDepartamento.php: Archivo que contiene una clase que se encarga de hacer la consulta en la base de datos, para remover el registro de determinado departamento, en la tabla ‘departamentos’.
- EndidadBase.php: Clase para ser heredada en la clase ‘VerUsuarios’, para tener funcionalidades de mostrar una lista de usuarios.
- EstructuraBase.php: Clase para definir los cabezales y menús en todas las páginas front-end, así como las opciones y menús disponibles en estas basadas en los privilegios del usuario.
- Expedientes.php: Archivo que contiene la clase que maneja la información en general de los estudiantes, es la clase más grande y que requiere más atención del programador.
- Logout: Clase que maneja el cierre de sesión.
- ModificarCarrera.php: Archivo que contiene la clase que modifica un registro de carrera en la tabla ‘carreras’, en la base de datos, es llamada desde el archivo ModificarCarreras.php que esta en el directorio raíz del sitio.
- ModificarDepartamento.php: Archivo que recibe del formulario ModificarDepartamento los datos para editar en la tabla ‘departamentos’, el registro determinado por el parámetro ‘idDepartamento’, este archivo se encarga de la ejecución de la consulta en base a los datos enviados desde el archivo nombrado ya, mediante una clase.
- NuevaCarrera.php, NuevoDepartamento.php, NuevoExpediente.php, NuevoUsuario.php: Archivo que contiene la clase que ingresa en la base de datos, el nuevo registro de una carrera, departamento, Estudiante y usuario respectivamente; son clases independientes heredadas de “Database”.

- `privilegios.php`: clase que verifica los privilegios de cada usuario para permitir o restringir al usuario, dependiendo de su tipo.
- `Signin.php`: clase que verifica la existencia de el usuario y si su contraseña es correcta. Este archivo es importado desde la página de inicio de sesión.
- `Usuarios.php`: Clase que muestra y gestiona otros elementos de los usuarios como su modificación, listado de elementos de otras tablas relacionadas, etc. Importante para el modulo usuarios.
- `VerUsuarios.php`: muestra una lista de usuarios.
- `ActualizarF2.php`: Archivo que tiene la clase que se encarga de subir los documentos del f2 a la base de datos (ruta del archivo) y en los directorios de la aplicación en el servidor.
- `ActualizarGrad.php`: clase para subir los documentos del expediente de graduación a la aplicación, actualizando la base de datos con la ruta de los archivos.
- `config.php`: clase para la configuración ara la conexión con la base de datos.
- `menu.php`: clase para mostrar la opciones del menu basade en los privilegios del usuario. Todas las páginas front-end llaman a este archivo.
- `modificarEstudiante.php`, `modificarUsuario.php`: clases para llevar o controlar la modificación del estudiante y usuario respectivamente; que son invocadas desde los formularios de modificar estudiante u usuarios.
- `observaciones.php`: clase para llevar a cabo los registros de las observaciones del administrador en la base de datos.

La clase mas importante aquí es expediente, ya que es esta la que gestiona loa alumnos de forma general, a continuación una explicación de esta.

El archivo ‘expedientes’, tiene métodos para mostrar un listado de alumnos, información de cada alumno por id para la modificación, métodos para mostrar la carrera del alumno asi como su genero, también métodos para el listado de carreras y géneros para los formularios de modificación y registro de expedientes.

Aquí el listado de métodos de la clase expedientes:

- mostrar(): método que muestra un listado de alumno en HTML con tres botones para revisar el expedientes, modificar y eliminar.

```
public function mostrar()
{
    try {
        $query = "SELECT * FROM estudiantes";
        $stmt = $this->con->prepare($query);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $this->conn = null;

        $estudiantes = '';
        foreach ($result as $key) {
            $estudiantes .= '
            <tr>
                <td>'. $key['carnet']. '</td>
                <td>'. $key['nombresEstudiante']. '</td>
                <td>'. $key['apellidosEstudiante']. '</td>
                <td>'. Expedientes::mostrarCarreraById($key['idCarrera']). '</td>
                <td>'. $key['anoTitulacion']. '</td>
                <td>'. $key['anoIngreso']. '</td>
                <td>
                    <a href="expediente.php?idEstudiante="'. $key["idEstudiante"]. '"
                    "><button class="btn btn-info btn-sm" type="button"
                    name="button" data-toggle="tooltip" title="Ver expediente"><i
                    class="far fa-eye"></i></button></a>
                </td>
            </tr>';
        }
        //mysqli_close($db);
        return $estudiantes;
    }
}
```

Figura 19:

- mostrarAlumno(): método para mostrar información de cada alumno para agregarlo al formulario de modificar el estudiante.

```
public function mostrarEstudiante($idEstudiante)
{
    try {
        $query = "SELECT * FROM estudiantes WHERE idEstudiante=$idEstudiante";
        $stmt = $this->con->prepare($query);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $this->conn = null;
        return $result;
    } catch (Exception $e) {
        print "Error: " . $e->getMessage() . "<br>";
        die();
    }
}
```

Figura 20:

- mostrarIndex(): método para enseñar en el index el listado de alumnos pero con botones diferentes, destaca el botón “confrontar”, que muestra los documentos que se permiten registrar, y los documentos que el

alumno ya tiene.

```
$estudiantes = '';  
foreach ($result as $key) {  
    $estudiantes .= '  
        <tr>  
            <td class="font-weight-bold">'. $key['carnet']. '</td>  
            <td>'. Expedientes::mostrarCarreraById($key['idCarrera']). '</td>  
            <td>'. $key['nombresEstudiante']. '</td>  
            <td>'. $key['apellidosEstudiante']. '</td>  
  
            <td>  
                <a href="expediente.php?idEstudiante=' . $key["idEstudiante"] . '  
                    "><button class="btn btn-info btn-sm" type="button"  
                        name="button" data-toggle="tooltip" title="Ver expediente"><i  
                            class="far fa-eye"></i></button></a>  
                <a href="confrontar.php?idEstudiante=' . $key["idEstudiante"] . '  
                    "><button class="btn btn-success btn-sm" type="button"  
                        name="button" data-toggle="tooltip" title="Confrontar"><i  
                            class="far fa-check-square"></i></button></a>  
                <button class="btn btn-danger btn-sm" type="button" name="button"  
                    data-toggle="tooltip" title="Descargar"><i class="fas  
                        fa-download"></i></button>  
  
            </td>  
        </tr>';  
    }  
}
```

Figura 21:

- mostrarF2(): Muestra los formularios para subir y/o modificar los documentos del f2, es un método back-end y devuelve un texto HTML.

Aquí está la consulta a la base de datos:

```
$query = "SELECT * FROM documentos_f2 WHERE idEstudiante=:idEstudiante";  
$stmt = $DB->conn->prepare($query);  
$stmt->bindParam(':idEstudiante', $idEstudiante);  
$stmt->execute();  
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);  
$stmt->closeCursor();  
$DB->conn = null;  
$tablaF2 = '';
```

Figura 22:

Aquí la como será el código html para cada documento del F2.

```

if(strlen($key['carnetResidente']) < 2){
    $stabaF2 = '
    <tr>
    <td>CARNET_RESIDENTE:</td>
    <td><input type="text" value="" name="carnetResidente" /></td>
    </tr>
    <tr>
    <td><input type="button" value="Subir" /></td>
    </tr>
    </table>
    '
}
else{
    $stabaF2 = '
    <tr>
    <td>CARNET_RESIDENTE:</td>
    <td><input type="text" value="" name="carnetResidente" /></td>
    </tr>
    <tr>
    <td><input type="button" value="Subir" /></td>
    </tr>
    </table>
    '
}

```

Figura 23:

- confrontarF2(): este método es invocado para llevar a cabo la comparación que se mencionaba en el punto anterior, no es front end, es invocado desde la página del front end.

```

if(strlen($key['carnetResidente']) < 2){
    $stabaF2 = '
    <tr>
    <td>CARNET_RESIDENTE:</td>
    <td><input type="text" value="" name="carnetResidente" /></td>
    </tr>
    <tr>
    <td><input type="button" value="Subir" /></td>
    </tr>
    </table>
    '
}
else{
    $stabaF2 = '
    <tr>
    <td>CARNET_RESIDENTE:</td>
    <td><input type="text" value="" name="carnetResidente" /></td>
    </tr>
    <tr>
    <td><input type="button" value="Subir" /></td>
    </tr>
    </table>
    '
}

```

Figura 24:

- mostrarDocsGrad(): Muestra los formularios para subir los documentos pertenecientes al Expediente de graduación.

```

if(strlen($key['apoderado']) < 2){
    $stabaFgrad = '
    <tr>
    <td>APODERADO:</td>
    <td><input type="text" value="" name="apoderado" /></td>
    </tr>
    <tr>
    <td><input type="button" value="Subir" /></td>
    </tr>
    </table>
    '
}
else{
    $stabaFgrad = '
    <tr>
    <td>APODERADO:</td>
    <td><input type="text" value="" name="apoderado" /></td>
    </tr>
    <tr>
    <td><input type="button" value="Subir" /></td>
    </tr>
    </table>
    '
}

```

Figura 25:

- `confrontarDocsGrad()`: compara los documentos del expediente de graduación que el alumno ya tiene con los que el software permite, es back-end.

Esta es la consulta a la base de datos:

```
public function confrontarDocsGrad($IDestudiante)
{
    try {
        $query = "SELECT * FROM documentos_graduacion WHERE IDestudiante=:IDestudiante";
        $stmt = $DB->con->prepare($query);
        $stmt->bindParam(':IDestudiante', $IDestudiante);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $DB->con = null;
    }
}
```

Figura 26:

Esta es la plantilla de cada documento a confrontar:

```
$tablaGrad .= "
foreach ($result as $key) {
    $tablaGrad .= "
    <table class='table table-hover'>
    <thead class='>
    <tr class='>
    <th>DOCUMENTOS</th>
    <th>CONFRONTACIÓN</th>
    </tr>
    </thead>
    <tbody>";
    if(strlen($key['apoderado']) < 2){
        $tablaGrad .= "
        <tr>
        <td>APODERADO:</td>
        <td><i class='fas fa-times text-danger'></i></td>
        </tr>
        <tr>
        }
    else{
        $tablaGrad .= "
        <tr>
        <td>APODERADO:</td>
        <td><i class='fas fa-check text-success'></i></td>
        </tr>
        <tr>
    }
}
```

Figura 27:

- `mostrarObservaciones()`: método que consulta de la base de datos las observaciones escritas por el administrador académico si las hay, y las muestra en el area del expediente de cada alumno. Es back-end.

```
public function mostrarObservaciones($idEstudiante)
{
    try {
        $query = "SELECT * FROM observaciones WHERE idEstudiante=:idEstudiante";
        $stmt = $this->con->prepare($query);
        $stmt->bindParam(':idEstudiante', $idEstudiante);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $this->conn = null;
        //mysqli_close($db);
        return $result;
    } catch (Exception $e) {
        print "Errorj: " . $e->getMessage() . "<br>";
        die();
    }
}
```

Figura 28:

- `mostrarCarreras()`. método para mostrar un listado de carreras disponibles en la base de datos, en los formularios de registro y modificación de estudiantes.

```
public function mostrarCarreras()
{
    try {
        $query = "SELECT * FROM carreras";
        $stmt = $this->con->prepare($query);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $this->conn = null;
        //mysqli_close($db);
        return $result;
    } catch (Exception $e) {
        print "Errorj: " . $e->getMessage() . "<br>";
        die();
    }
}
```

Figura 29:

- `mostrarCarrerasById()`: método usado para mostrar información de las carreras de cada alumno en el área del expediente de cada alumno.

```
public function mostrarCarreraById($idCarrera)
{
    try {
        $query = "SELECT nombre FROM carreras WHERE idCarrera = :idCarrera";

        $stmt = $this->con->prepare($query);
        $stmt->bindParam(':idCarrera', $idCarrera);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $this->conn = null;
        //mysql_close($db);
        return $result['nombre'];
    } catch (Exception $e) {
        print "Error: " . $e->getMessage() . "<br>";
        die();
    }
}
```

Figura 30:

- `mostrarGeneroById()`: Muestra el género específico del alumno en la página de información del expediente de cada alumno.

```
public function mostrarGeneroById($idGenero)
{
    try {
        $query = "SELECT * FROM genero WHERE idGenero=:idGenero";

        $stmt = $this->con->prepare($query);
        $stmt->bindParam(':idGenero', $idGenero);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        $stmt->closeCursor();
        $this->conn = null;
        //mysql_close($db);
        return $result['nombre'];
    } catch (Exception $e) {
        print "Error: " . $e->getMessage() . "<br>";
        die();
    }
}
```

Figura 31:

- `getNombre()`: Muestra el nombre completo (Con apellidos) de cada estudiante.

```
public function getNombre($idEstudiante)
{
    try {
        $query = "SELECT nombresEstudiante, apellidosEstudiante FROM estudiantes WHERE
        idEstudiante=:idEstudiante";

        $stmt = $this->con->prepare($query);
        $stmt->bindParam(":idEstudiante", $idEstudiante);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
        $nombre="";
        foreach ($result as $key) {
            $nombre .= $key['nombresEstudiante']." ".$key['apellidosEstudiante'];
        }
        $stmt->closeCursor();
        $this->con = null;
        return $nombre;
    } catch (Exception $e) {
        print "Error: " . $e->getMessage() . "<br>";
        die();
    }
}
```

Figura 32:

4. Directorios de recurso varios

El directorio 'css' y 'js' contienen los archivos CSS (con bootstrap) y JavaScript respectivamente.

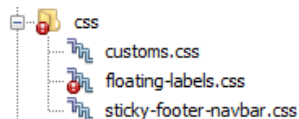


Figura 33:



Figura 34:

También tenemos el directorio resources que contiene a Bootstrap, fontawesome, y JQuery.

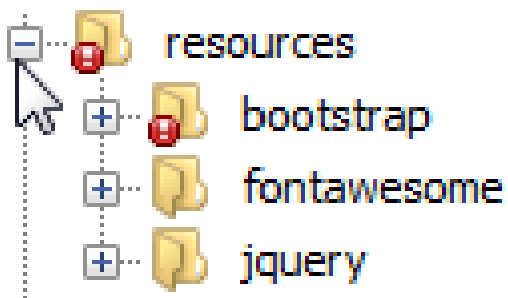


Figura 35:

La carpeta 'sql' guarda sentencias SQL para importar la base de datos, así como el dumpeo o respaldo de esta. Contiene diagramas y otros archivos relevantes a la base de datos.

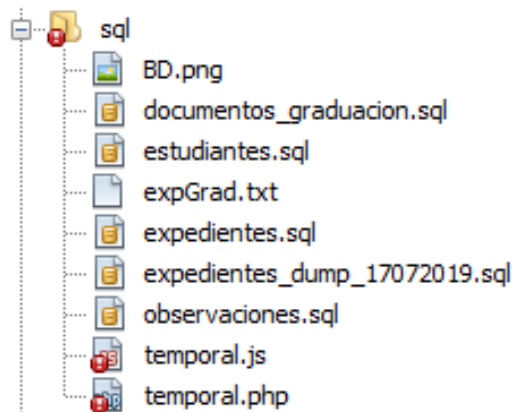


Figura 36:

La carpeta 'img' contiene los gráficos para decorar el sitio web, como el logo de la universidad.



Figura 37:

Por ultimo tenemos el directorio 'pdf', en el que se subiran todos documentos escaneados del f2, expediente de graduación y otros documentos para cada alumno. Se subirán los archivos y la base de datos guardará una ruta para cada archivo, en los registros relacionados a cada estudiantes.

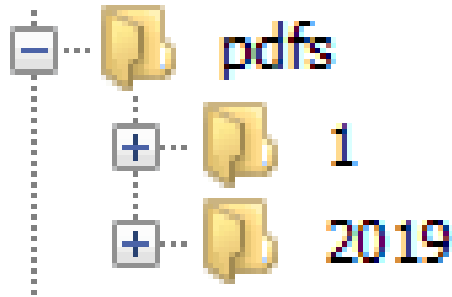


Figura 38:

5. Conclusión

Los formularios están en el directorio raíz, estos para efectuar el CRUD, llaman a los archivos del directorio 'core', que contiene las clases que 'controlan' las tareas esenciales de la aplicación, donde la clase mas importante será 'expediente', donde los referente a manejar estudiantes se maneja allí, y se deberá mejorar, ya se agregando metodos o cambiando otros, esta clase mayoritariamente, si se desea.

El enfoque orientado a objetos, así como la división de front-end y back-end mediante el directorio 'core', hace de este software escalable, permitiendo la modificación por parte de personas ajenas a este proyecto.

Diccionario de datos

A continuación se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema.

carreras

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idCarrera (<i>Primaria</i>)	int(3)	No				
codigo	varchar(10)	No				
nombre	varchar(60)	No				
idDepartamento	int(3)	No		departamentos -> idDepartamento		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idCarrera	0	A	No	
idDepartamento	BTREE	No	No	idDepartamento	0	A	No	

Figura 6.1: Tabla carreras

departamentos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idDepartamento (<i>Primaria</i>)	int(3)	No				
nombre	varchar(60)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idDepartamento	0	A	No	

Figura 6.2: Tabla departamentos

documentos_f2

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idF2 (<i>Primaria</i>)	int(6)	No				
carnetResidente	varchar(100)	Sí	NULL			
certificadoSalud	varchar(100)	Sí	NULL			
constanciaCuotaBachillerato	varchar(100)	Sí	NULL			
dui	varchar(100)	Sí	NULL			
nit	varchar(100)	Sí	NULL			
paes	varchar(100)	Sí	NULL			
partidaNacimiento	varchar(100)	Sí	NULL			
pasaporte	varchar(100)	Sí	NULL			
tituloBachiller	varchar(100)	Sí	NULL			
idEstudiante	int(6)	No		estudiantes -> idEstudiante		
uaCarnetResidente	varchar(10)	Sí	NULL			
uaCertificadoSalud	varchar(10)	Sí	NULL			
uaConstanciaCuotaBachillerato	varchar(10)	Sí	NULL			
uaDui	varchar(10)	Sí	NULL			
uaNit	varchar(10)	Sí	NULL			
uaPaes	varchar(10)	Sí	NULL			
uaPartidaNacimiento	varchar(10)	Sí	NULL			
uaPasaporte	varchar(10)	Sí	NULL			
uaTituloBachiller	varchar(10)	Sí	NULL			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idF2	19	A	No	
documentos_f2_ibfk_1	BTREE	No	No	idEstudiante	19	A	No	

Figura 6.3: Tabla documentos f2

documentos_graduacion

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idDocumentoGraduacion (<i>Primaria</i>)	int(6)	No				
apoderado	varchar(100)	Sí	NULL			
tituloBachillerG	varchar(100)	Sí	NULL			
acuerdoIncorporacionBachiller	varchar(100)	Sí	NULL			
partidaNacimientoG	varchar(100)	Sí	NULL			
duiG	varchar(100)	Sí	NULL			
carnetResidenteG	varchar(100)	Sí	NULL			
mallCurricular	varchar(100)	Sí	NULL			
certificacionNotas	varchar(100)	Sí	NULL			
constanciaEgreso	varchar(100)	Sí	NULL			
prorrogaEgresado	varchar(100)	Sí	NULL			
acuerdoRatificacion	varchar(100)	Sí	NULL			
escritoCumHonorifico	varchar(100)	Sí	NULL			
constanciaEntregaEjemplares	varchar(100)	Sí	NULL			
actaDefensaFinal	varchar(100)	Sí	NULL			
constanciaServicioSocial	varchar(100)	Sí	NULL			
solvenciaBiblioteca	varchar(100)	Sí	NULL			
solvenciaAcme	varchar(100)	Sí	NULL			
declaracionJurada	varchar(100)	Sí	NULL			
recibosPagoGraduacion	varchar(100)	Sí	NULL			
constanciaCumHonorifico	varchar(100)	Sí	NULL			

uaApoderado	varchar(15)	Sí	NULL				
uaTituloBachillerG	varchar(15)	Sí	NULL				
uaAcuerdoIncorporacionBachiller	varchar(15)	Sí	NULL				
uaPartidaNacimientoG	varchar(15)	Sí	NULL				
uaDuiG	varchar(15)	Sí	NULL				
uaCarnetResidenteG	varchar(15)	Sí	NULL				
uaMallaCurricular	varchar(15)	Sí	NULL				
uaCertificacionNotas	varchar(15)	Sí	NULL				
uaConstanciaEgreso	varchar(15)	Sí	NULL				
uaProrrogaEgresado	varchar(15)	Sí	NULL				
uaAcuerdoRatificacion	varchar(15)	Sí	NULL				
uaEscritoCumHonorifico	varchar(15)	Sí	NULL				
uaConstanciaEntregaEjemplares	varchar(15)	Sí	NULL				
uaActaDefensaFinal	varchar(15)	Sí	NULL				
uaConstanciaServicioSocial	varchar(15)	Sí	NULL				
uaSolvenciaBiblioteca	varchar(15)	Sí	NULL				
uaSolvenciaAcme	varchar(15)	Sí	NULL				
uaDeclaracionJurada	varchar(15)	Sí	NULL				
uaRecibosPagoGraduacion	varchar(15)	Sí	NULL				
uaConstanciaCumHonorifico	varchar(15)	Sí	NULL				
uaConstanciaNoSancion	varchar(15)	Sí	NULL				
uaEscritoAperturaExpediente	varchar(15)	Sí	NULL				
uaFichaResumenDatos	varchar(15)	Sí	NULL				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idDocumentoGraduacion	19	A	No	
documentos_graduacion_ibfk_1	BTREE	No	No	idEstudiante	19	A	Sí	

Figura 6.4: Tabla documentos de graduación

estudiantes

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idEstudiante (Primaria)	int(6)	No				
nombresEstudiante	varchar(40)	No				
apellidosEstudiante	varchar(40)	No				
genero	int(12)	No		genero -> idGenero		
fechaNacimiento	varchar(12)	No				
anoTitulacion	varchar(5)	No				
anoIngreso	varchar(5)	No				
carnet	varchar(7)	Sí	NULL			
facultad	varchar(50)	No				
direccion	varchar(60)	No				
telefono	varchar(12)	Sí	NULL			
email	varchar(40)	Sí	NULL			
foto	longblob	Sí	NULL			
idCarrera	int(3)	No		carreras -> idCarrera		
activo	tinyint(1)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idEstudiante	22	A	No	
idCarrera	BTREE	No	No	idCarrera	2	A	No	
genero	BTREE	No	No	genero	4	A	No	
genero_2	BTREE	No	No	genero	4	A	No	

Figura 6.5: Tabla estudiantes

genero

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idGenero (<i>Primaria</i>)	int(11)	No				
nombre	varchar(10)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idGenero	2	A	No	

Figura 6.6: Tabla genero

observaciones

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idObservacion (<i>Primaria</i>)	int(6)	No				
observacionF2	varchar(200)	Sí	NULL			
observacionGrad	varchar(200)	Sí	NULL			
observacionOtros	varchar(200)	Sí	NULL			
idEstudiante	int(6)	Sí	NULL	estudiantes -> idEstudiante		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idObservacion	0	A	No	
observaciones_ibfk_1	BTREE	No	No	idEstudiante	0	A	Sí	

Figura 6.7: Tabla observaciones

otros_documentos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idOtrosDocumentos (<i>Primaria</i>)	int(6)	No				
documentosAdicionales	varchar(100)	Sí	NULL			
uaDocumentosAdicionales	varchar(10)	Sí	NULL			
idEstudiante	int(6)	No		estudiantes -> idEstudiante		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idOtrosDocumentos	0	A	No	
otros_documentos_ibfk_1	BTREE	No	No	idEstudiante	0	A	No	

Figura 6.8: Tabla otros documentos

privilegios

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idPrivilegios (<i>Primaria</i>)	int(3)	No				
principal	tinyint(1)	No				
estudiantes	tinyint(1)	No				
expedientes	tinyint(1)	No				
usuarios	tinyint(1)	No				
configuracion	tinyint(1)	No				
agregar	tinyint(1)	No				
modificar	tinyint(1)	No				
eliminar	tinyint(1)	No				
idUsuario	int(3)	No		usuarios -> idUsuario		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idPrivilegios	3	A	No	
privilegios_ibfk_1	BTREE	No	No	idUsuario	3	A	No	

Figura 6.9: Tabla privilegios

registro_acciones

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idRegistro (<i>Primaria</i>)	int(11)	No				
idUsuario	int(3)	No		usuarios -> idUsuario		
horaFecha	datetime	No				
descripcion	varchar(50)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idRegistro	78	A	No	
idUsuario	BTREE	No	No	idUsuario	7	A	No	

Figura 6.10: Tabla registro de acciones

tipo_usuario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idTipoUsuario (<i>Primaria</i>)	int(1)	No				
nombreTipo	varchar(25)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idTipoUsuario	3	A	No	

Figura 6.11: Tabla tipo de usuario

usuarios

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idUsuario (<i>Primaria</i>)	int(3)	No				
nombresUsuario	varchar(32)	No				
apellidosUsuario	varchar(32)	No				
dui	varchar(12)	No				
fechaNacimiento	date	No				
telefono	varchar(12)	No				
direccion	varchar(50)	Sí	NULL			
user	varchar(32)	No				
pass	varchar(70)	No				
idTipoUsuario	int(1)	No		tipo_usuario -> idTipoUsuario		
estado	tinyint(1)	No				
correo_electronico	varchar(64)	Sí	NULL			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idUsuario	3	A	No	
idTipoUsuario	BTREE	No	No	idTipoUsuario	3	A	No	

Figura 6.12: Tabla usuarios