



**REPUBLICA DEL ECUADOR
INSTITUTO SUPERIOR TECNOLÓGICO
“EL PACÍFICO”**

**TRABAJO DE TITULACIÓN PREVIA A LA OBTENCIÓN DEL
TÍTULO DE TECNÓLOGO SUPERIOR EN
AUTOMATIZACIÓN E INSTRUMENTACIÓN**

TEMA

**PROYECTO TECNOLÓGICO DE CREACIÓN DE UN SISTEMA
DE BASE DE DATOS REPOSITORIO PARA REGISTRO DE
TRABAJOS DE TITULACIÓN DE LAS CARRERAS DEL
INSTITUTO SUPERIOR TECNOLÓGICO “EL PACÍFICO”,
DMQ.**

AUTOR

ALEJANDRO DAVID FLORES BARROS

DIRECTOR

MAGISTER JAVIER FLORES

Agosto 2024

QUITO – ECUADOR

DECLARACIÓN

Yo, **ALEJANDRO DAVID FLORES BARROS** declaro que el presente trabajo es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

El Instituto Tecnológico Superior “El Pacífico”, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

ALEJANDRO DAVID FLORES BARROS

C.C. 171557242-4.

CERTIFICACIÓN

La Directora del Trabajo de Titulación, luego de revisar las sugerencias y correcciones planteadas al señor:

ALEJANDRO DAVID FLORES BARROS

Con el Tema:

PROYECTO TECNOLÓGICO DE CREACIÓN DE UN SISTEMA DE BASE DE DATOS REPOSITORIO PARA REGISTRO DE TRABAJOS DE TITULACIÓN DE LAS CARRERAS DEL INSTITUTO SUPERIOR TECNOLÓGICO “EL PACÍFICO”, DMQ.

Luego de haber cumplido con todas las etapas del trabajo de titulación, el proceso investigativo realizado científicamente.

Declaro apto al presente trabajo para la presentación a la defensa final.

DMQ a los 16 días del mes de agosto del 2024

Director(a)

Msc. Javier Flores

AGRADECIMIENTO

Quiero agradecer a Dios, por permitirme la fuerza y valor necesario para continuar a pesar de todo, por estar presente en mi vida, desde niño hasta hoy, por bendecirme en cada paso.

A mis padres Raúl y Gloria por su empeño y sacrificio, por amarme y ser mi soporte en cada paso, sin ustedes nada hubiera sido posible.

A mis hijos Marcelita, Matías, Benjamín y Valentina, por su simple existencia, todo es llevadero gracias a ustedes.

A mi corazón, Vanne, gracias por elegirme a pesar de tanto y a pesar de todo.

Gracias Instituto Superior Tecnológico “El Pacífico”, por abrirme sus puertas, a su personal administrativo por su paciencia, a sus docentes por su entrega y preocupación para que complete todo lo necesario en el tiempo adecuado, a sus directivos, en especial a la ingeniera Orfa Garcia, por su paciencia, por su sentido del humor y sus regaños, por exigirme un poco más para ser mejor.

A mi prima que es más mi hermana, Evelyn, gracias por ser una de las personas que jamás dudo de mí y que me apoyó hasta el final, en buenas malas y peores.

Y, por último, pero no menos importante, gracias a mí, por ver cada día mi valor, mis virtudes, mis capacidades, por resaltar cada día lo mejor de mí, por buscar una salida, por luchar sin descanso, por caminar esa milla extra para alcanzar mis objetivos, por mantener la sonrisa, por ser paciente con todos y más conmigo.

Alejandro Flores Barros.

DEDICATORIA

A mis hijos, sepan ustedes que no importa que este en su contra, no importa que tiempo les tome, luchen, persigan y si caen, vuélvanse a levantar, que no se trata de ganas, se trata de propósito.

A mis padres, sin su sacrificio, paciencia y amor, no estaría escribiendo estas palabras, esto es más suyo que mío.

A mi corazón, porque me ayudaste y me diste tu mano siempre, porque me mostraste el camino, sin intención ni de recorrerlo por mí, ni de estar ahí a mi lado, sino porque me enseñaste a caminarlo solo, sabiendo que tú estabas dispuesta a darme tu mano y guiarme nuevamente si alguna vez me desviaba un poco.

A quienes creyeron que yo podía, a quienes mostraron interés en verme mejor, a quienes aportaron su granito de arena, revisando, comentando, criticando este trabajo y todo lo que he realizado, a quienes vieron mi potencial y mis virtudes, talentos y atributos, a quienes hicieron este caminar llevadero, a quien preguntaron, a quienes se interesaron.

ÍNDICE GENERAL

CAPÍTULO I	1
GENERALIDADES.....	1
1.1. Identificación y formulación del problema	1
1.1.1. Identificación del tema	1
1.1.2. Formulación del problema.....	2
1.2. Justificación	2
1.3. Objetivos General y Específicos.	2
1.3.1. Objetivo General.....	3
1.3.2. Objetivos Específicos.....	3
1.4. Alineados a la línea de investigación y al Plan Nacional de Desarrollo ..	4
1.5. Factibilidad.....	5
1.6. Metodología.....	5
1.7. Variables independientes y dependientes.....	5
1.7.2. Variables dependientes	5
CAPITULO II	6
FUNDAMENTACIÓN TEÓRICA	6
2.1. La Computadora	6
2.2. Investigación de campo	17
2.3. Muestra.	17
2.4. Calculo de la muestra	18

2.5. Encuesta	19
2.6. Procesamiento de datos.....	22
2.7. FLUJOGRAMA DEL PROCESO.....	31
CAPITULO III	34
INGENIERIA DEL PROYECTO	34
3.4.1 Detalles del Repositorio Digital	95
3.4.2. Ventajas de c-panel como Plataforma	95
3.4.3. Acceso al Repositorio	95
3.4.4. Manual de Usuario	96
EL REPOSITORIO EN IMÁGENES	97
Manual de Mantenimiento (Claves, Accesos y Forma De Realimentación De Textos)	110
CAPITULO IV	114
PRESUPUESTO	114
Mano de Obra	114
Costo directo	114
Costo indirecto	114
Resumen costos y gastos	115
Evaluación del tutor.....	116
CAPITULO V	118
CONCLUSIONES Y RECOMENDACIONES	118

5.1. Conclusiones.....	118
5.1.2. Recomendaciones	120
Bibliografía y Linkografía.....	122

ÍNDICE DE TABLAS

Tabla 1 Proyecto Tecnológico.....	4
Tabla 2 Identificación del encuestado.....	22
Tabla 3 Ocupación del encuestado.....	23
Tabla 4 Medios Tecnológicos más usados.....	24
Tabla 5 Medios Tecnológicos en unidades educativas.....	25
Tabla 6 Herramientas virtuales más usadas.....	26
Tabla 7 Uso de la Tecnología	27
Tabla 8 Aplicación de competencias al usar la tecnología.....	28
Tabla 9 Repositorios bibliográficos.....	29
Tabla 10 Sugerencias de repositorios	30

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Identificación del problema Repositorio de documentos....	1
Ilustración 2 evolución de las computadoras	10
Ilustración 3 Regla de calculo.....	10
Ilustración 4 Maquina tabuladora de HOLLERITH.....	11
Ilustración 5 Ordenadores Mark.....	12
Ilustración 6 computador UNIVAC	13
Ilustración 7 microcomputadores cuarta generación	14
Ilustración 8 Diagrama de flujo	33
Ilustración 9 Estructura general de página web	37
Ilustración 10 Modelo-Vista-Controlador (MVC).....	38
Ilustración 11 Metodología MIDAS/DB.....	42
Ilustración 12 Página de inicio.....	97
Ilustración 13 Botón de inicio	98
Ilustración 14 Botón Proyectos de titulación	98
Ilustración 15 Botón Facultades.....	99
Ilustración 16 Botón Carreras	99
Ilustración 17 Botón Nosotros.....	100
Ilustración 18 Botón Explorar Proyectos.....	100
Ilustración 19 Botón Registrarse	100
Ilustración 20 Formulario de Registro	101
Ilustración 21 Mensaje obligatoriedad.....	101
Ilustración 22 Botón login Estudiante	102
Ilustración 23 Formulario Login Estudiante	102

Ilustración 24 Pantalla de usuario estudiante	102
Ilustración 25 Botón Perfil	103
Ilustración 26 Actualizar detalles.....	104
Ilustración 27 Botón Mis Archivos.....	104
Ilustración 28 Botón Acción.....	105
Ilustración 29 Botón Logout estudiante.....	105
Ilustración 30 Login Administrador.....	106
Ilustración 31 Pantalla de Administrador	106
Ilustración 32 Dashboard.....	107
Ilustración 33 Archivos administrador	107
Ilustración 34 Botón de Acción administrador.....	107
Ilustración 35 Botón estudiantes administrador	108
Ilustración 36 Botón Facultades administrador.....	108
Ilustración 37 Botón Carreras administrador.....	108
Ilustración 38 Botón Usuarios administrador	108
Ilustración 39 Botón Ajustes administrador	109
Ilustración 40 Botón Administrador perfil	109

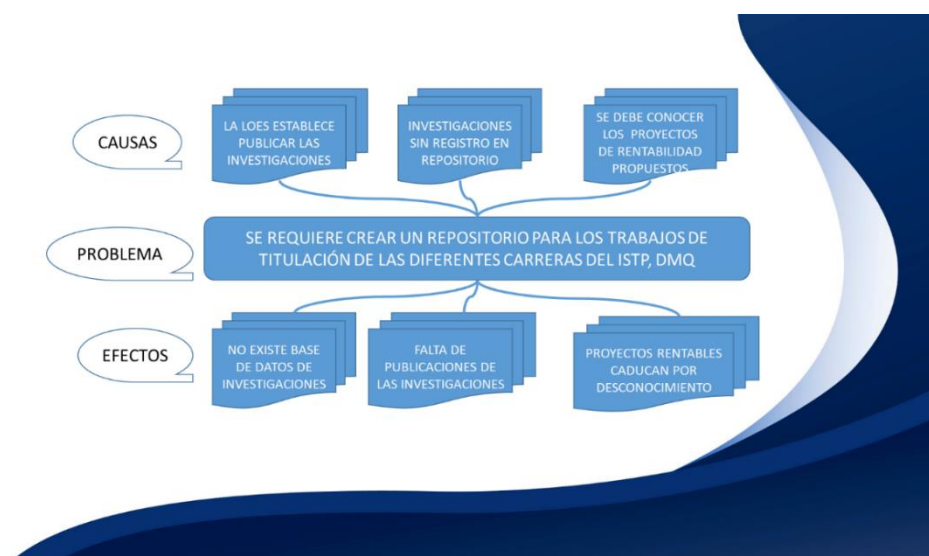
CAPÍTULO I

GENERALIDADES

1.1. Identificación y formulación del problema

1.1.1. Identificación del tema

Ilustración 1 Identificación del problema Repositorio de documentos



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Las funciones de la educación superior se enmarcan en la Docencia, Investigación y las prácticas preprofesionales con la vinculación con la sociedad. Siendo uno de los pilares fundamentales para el perfil profesional del Tecnólogo la investigación aplicada su trayectoria en la Carrera determinada.

Una de las principales preocupaciones del Tecnólogo de Sistema de Automatización e instrumentación es de enfilarse en la innovación del sistema informático, la misma que avanza a pasos del día, mejorando el servicio de la información y la comunicación, los mismos que han desarrollado con el uso de inteligencia artificial.

Al realizarse la investigación esta debe dejar evidencia textual, la misma que servirá de consulta y/o apoyo para nuevas investigaciones

o mejorarla la realizada. En caso de no registrarse la misma correrá el peligro que se pierda la valiosa o información o quede rezagada a en el rincón de la Biblioteca institucional

Con estos antecedentes se ha realizado la evaluación del servicio de la Biblioteca del ISTEP, y se ha detectado el requerimiento de un repositorio que contenga la información de las diferentes investigaciones que realizan los estudiantes al culminar su Carrera de estudios en las diferentes disciplinas y de acuerdo con las líneas de investigación de cada una de ellas.

1.1.2. Formulación del problema

Cuando se determina la necesidad de exponer las diferentes y valiosas investigaciones realizadas por docentes y estudiantes al culminar su carrera, se concluye que es necesario crear una base de datos con la calidad de repositorio de investigaciones, la misma que servirá al público, a la innovación tecnológica y al emprendimiento.

1.2. Justificación

Se cuenta con el material bibliográfico que se incluirá en el repositorio, con sus referencias para la ubicación y sus medios de acceso para la lectura de quien desee consultarlo.

Existe el hardware y software que facilitará el peso en calidad y eficiencia para la base de datos e información a quien alimente la información y a quien requiera acceder a la información digitalizada.

1.3. Objetivos General y Específicos.

1.3.1. Objetivo General

Ejecutar el proyecto tecnológico de creación de un sistema de base de datos repositorio para registro de trabajos de titulación de las carreras del Instituto Superior Tecnológico “El Pacífico”, DMQ.

1.3.2. Objetivos Específicos

- Seleccionar un sistema de gestión base de datos (SGBD);
- Realizar un análisis de contenidos para la exposición
- Programar el sistema con característica dinámica.
- Establecer una auditoria de alimentación y uso periférico
- Adquisición de dominio
- Fortalecer contra ataques de haqueos o daño del sistema.

1.4. Alineados a la línea de investigación y al Plan Nacional de Desarrollo

Tabla 1 Proyecto Tecnológico

TEMA	LINEAMIENTO A CARRERA	ALINEAMIENTO AL P.N.D.	ZONA
PROYECTO TECNOLÓGICO DE CREACIÓN DE UN SISTEMA DE BASE DE DATOS REPOSITORIO PARA REGISTRO DE TRABAJOS DE TITULACIÓN DE LAS CARRERAS DEL INSTITUTO SUPERIOR TECNOLÓGICO “EL PACÍFICO”, DMQ.	Proyecto tecnológico	El Plan de Desarrollo del Sistema de Educación Superior para el periodo 2022-2026, se construye sobre la base de tres objetivos estratégicos: 1. Fortalecer la cobertura en la educación superior bajo los principios del Sistema; 2. Incrementar la contribución al conocimiento para el desarrollo social y económico de la sociedad; y, 3. Fortalecer el aseguramiento de la calidad de la educación superior; que abarcan lo relacionado a cobertura y calidad; y, se enmarcan en los principios del Sistema de Educación Superior, establecidos tanto en la Constitución de la República del Ecuador como en la Ley Orgánica de Educación Superior.	Zona 9, DMQ, PARROQUIA BELISARIO QUEVEDO, LAS CASAS

Fuente: Plan Nacional de Educación Superior

Elaborado por: Alejandro Flores

1.5. Factibilidad

Acumulada experiencia por estudio y aplicación de la programación bases de datos.

Recursos económicos propios y el asesoramiento técnico de profesionales de desarrollo de software.

Deseo de compartir las experiencias obtenidas en investigaciones tecnológicas.

1.6. Metodología

Se realizará el método de análisis de sitios web y desarrollo de Bases de Datos;

La metodología de investigación inductiva- deductiva, recopilación de variables cualitativas de calidad e innovación y que permita tomar decisiones de eficiencia.

La metodología descriptiva, que permitirá mejorar el detalle de procesos para la agilidad y beneficios del sistema de comunicación.

Metodología de análisis de sistemas de información y diseño de bases de datos documentales: aspectos lógicos y funcionales.

Investigación en fuentes primarias relacionadas con la inteligencia artificial.

1.7. Variables independientes y dependientes

1.7.1. Variable independiente

El sistema repositorio

1.7.2. Variables dependientes

- Base de datos documentales.
- Sistema amigable en calidad y eficiencia
- Sistema de comunicación segura y fiable
- Manual de funcionamiento y mantenimiento

CAPITULO II

FUNDAMENTACIÓN TEÓRICA

2.1. La Computadora

Una computadora es capaz de procesar datos a gran velocidad y en grandes cantidades.

Un computador, computadora u ordenador es una máquina digital programable, de funcionamiento electrónico, capaz de procesar grandes cantidades de datos a grandes velocidades. Así obtiene información útil que luego presenta a un operador humano, o transmite a otros sistemas mediante redes informáticas de distinto tipo.

La computadora es la herramienta más versátil, potente y revolucionaria que el ser humano ha creado en su historia reciente. Representa el punto cumbre de la Revolución industrial, científica y tecnológica que presenció el siglo XX después de la Segunda Guerra Mundial.

Su presencia y popularización en nuestro tiempo no sólo cambió para siempre el modo de procesar la información en el mundo, sino también la manera de trabajar y concebir el trabajo, las formas de comunicarse a larga distancia, las formas de ocio, y muchas otras áreas de la vida cotidiana.

Consisten fundamentalmente en un gran número de circuitos integrados, componentes de apoyo y extensiones electrónicas. Sin embargo, las computadoras han cambiado radicalmente a lo largo de su propia y rápida historia, pasando de ser enormes e incómodas instalaciones, a ocupar un lugar tan pequeño como en el bolsillo de nuestros pantalones, en el caso de los teléfonos inteligentes.

La enorme cantidad de componentes de una computadora pueden agruparse en dos categorías separadas, que son:

- **Hardware.** La parte física y tangible del sistema, o sea, sus componentes eléctricos y electrónicos, que cumplen con diversas funciones fundamentales, como la realización de cálculos o la alimentación eléctrica del sistema. De algún modo equivaldría al “cuerpo” de la computadora.

- **Software.** La parte intangible, digital, abstracta del sistema, que se ocupa de las operaciones de tipo conceptual o representacional, normalmente dentro de un entorno virtual simulado, esto es, dentro de una simulación que hace más amable la interacción con el usuario. Esto abarca todo tipo de programas, desde los programas de base (como el Sistema Operativo que mantiene andando el sistema) hasta las aplicaciones posteriormente instaladas. Siguiendo la metáfora, equivaldría a la “mente” del computador.

Tipos de computadora

Las mainframes llevan a cabo millones de cálculos y operaciones por segundo.

Existen muy diversos tipos de computador, atendiendo a rasgos como su tamaño, potencia y utilidad. Entre ellos, destacan:

- **Supercomputadoras.** Los aparatos de mayor potencia del mundo son, en realidad, conjuntos de computadoras distintas integrados en una sola unidad, que pueden potenciarse de manera exponencial. La supercomputadora más grande del mundo se encuentra en la Universidad Nacional de Tecnología de Defensa de China, se llama Tianhe-2 y puede realizar unos 33.48 mil billones de operaciones por segundo.

- **Mainframes.** Se conocen también como macrocomputadoras, y suelen tener un gran tamaño (al menos comparado con los computadores portátiles) y hallarse en habitaciones cuidadosamente refrigeradas dentro de las grandes empresas o instituciones de un país, en donde llevan a cabo millones de cálculos y operaciones por

segundo, alimentando redes y sistemas computarizados enteros con información.

- **Computadoras personales (PCs).** Unidades destinadas al uso de un solo usuario a la vez, permitiéndole realizar muy diversas tareas, entre ellas conectarse a una red informática y enviar y recibir datos a altas velocidades. Poseen un microprocesador de potencia variable y son el tipo de computadoras a los que podemos acceder comercialmente en cualquier tienda de tecnología. Se les conoce también como computadoras de escritorio.

- **Computadores portátiles (laptops, netbooks).** Aunque se trata en efecto de computadoras personales, a este tipo de aparatos los listamos aparte porque se trata de piezas físicamente ligeras, diseñadas para su uso en exteriores o para viajar con nosotros en un maletín. Aunque poseen menos potencia que las PC, lo compensan en practicidad y movilidad.

- **Tabletas y celulares.** La generación más reciente de artefactos tecnológicos (gadgets) son esencialmente computadoras, aunque destinadas a funciones distintas (generalmente más lúdicas o de comunicaciones) y de un tamaño todavía menor a las portátiles. Los celulares “inteligentes” son pequeñas pero potentes computadoras dedicadas a las telecomunicaciones y la navegación por Internet, mientras que las tabletas son pequeñas pantallas táctiles con funciones similares.

Partes de una computadora

A grandes rasgos, un sistema informático o computador se compone numerosos elementos reunidos en tres conjuntos o partes:

- **Unidad Central de Procesamiento (CPU).** CPU son las siglas en inglés de Central Processing Unit, y ese es el nombre que recibe el “cerebro” de la computadora, o sea, su núcleo lógico-electrónico. Allí tienen lugar las operaciones lógicas en los procesadores o microprocesadores del sistema, y están físicamente dispuestas las

unidades fijas de memoria y de almacenamiento. Estas últimas se denominan “disco rígido” o “disco duro”, y usada para contener información. A su vez, el CPU se compone de:

- **La Unidad Aritmeticológica (ALU).** Dedicada a la realización de las operaciones lógicas, matemáticas o formales que sostienen el sistema.
- **La Unidad de Control (UC).** Encargada de la vigilancia del sistema y de garantizar el funcionamiento continuo del mismo.
- **Los registros.** Que son la información que genera el funcionamiento del sistema, y que sirve para procesos de feedback del sistema.
- **Memoria.** La memoria es un espacio electrónico donde se almacena de manera temporal la información que el sistema requiere para trabajar, por lo que se conoce también como memoria de trabajo. Existen dos formas de memoria:
 - **Memoria de Acceso Aleatorio (RAM: Random Access Memory).** Que consiste en una secuencia de celdas de almacenamiento ocupada temporalmente por la información de trabajo. Puede escribirse y reescribirse según se necesite. Al apagar e iniciar el sistema, la RAM se reinicia por completo.
 - **Memoria de Sólo Lectura (ROM: Read Only Memory).** Un espacio en el que viene ya inscrita de fábrica la información mínima fundamental para que el ordenador inicie sus operaciones, como el set up y la configuración mínima que da pie a los procesos más complejos del software. Esta memoria puede leerse millones de veces, pero no puede alterarse a voluntad.
- **Dispositivos periféricos.** Los periféricos son los componentes no centrales de una computadora, o sea, sus accesorios o aditamentos, que pueden cambiarse o sustituirse sin alterar el núcleo de operaciones del sistema. Estos dispositivos permiten ingresar (entrada) o extraer (salida) información del sistema, o realizar ambas

operaciones a la vez (entrada-salida). Son ejemplos de periféricos los monitores, las impresoras, los teclados, los parlantes, etc.

Evolución de las computadoras

Ilustración 2 evolución de las computadoras



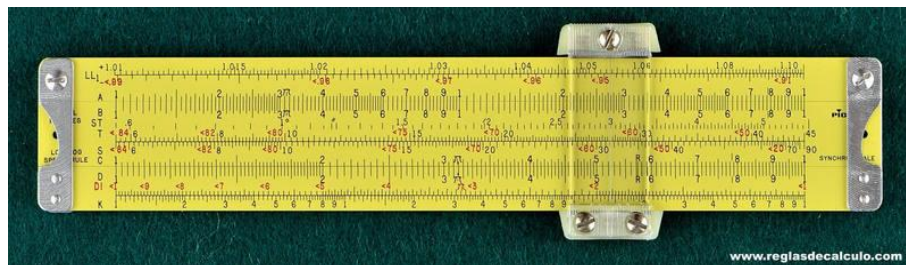
Fuente: <https://concepto.de/computadora/>

La historia de las computadoras es bastante reciente.

Sin embargo, pueden considerarse como antecedentes numerosos aparatos no digitales e incluso no electrónicos para lidiar con grandes volúmenes de información: el ábaco, o los aparatos mecánicos para lidiar con grandes volúmenes de información: el ábaco, o los aparatos mecánicos para calcular que se empleaban antaño, son ejemplo de ello.

Además, debe considerarse lo antiguo de las bases conceptuales de todo sistema informático, como la matemática y los algoritmos (830 d.C.), o las reglas de cálculo (1620 d.C.).

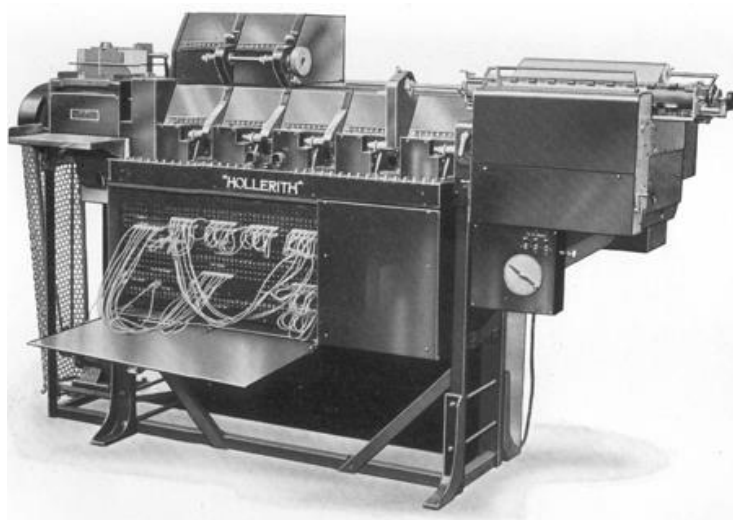
Ilustración 3 Regla de calculo



Fuente: <https://www.reglasdecalculo.com>

Sin embargo, se considera a la “máquina analítica” de Charles Babbage (1833) y la máquina tabuladora de Hermann Hollerith (1890) como los primeros computadores.

Ilustración 4 Maquina tabuladora de HOLLERITH



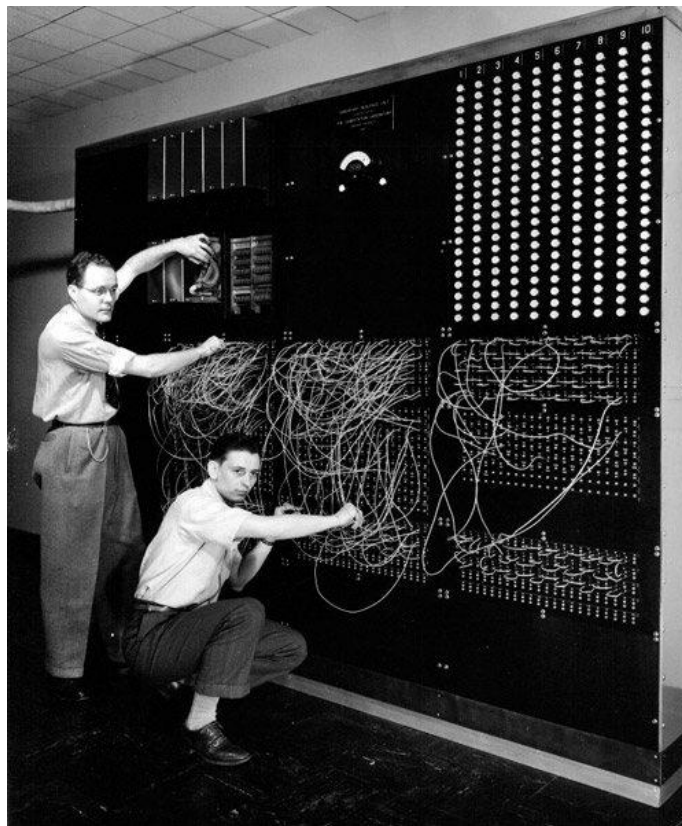
Fuente: <https://www.kubernetica.com/2010/08/21/la-maquina-tabuladora-de-hollerith-y-el-inicio-de-la-informatica/>

No obstante, no eran equiparables aún a los sistemas que comenzaron a aparecer en el siglo XX. Liderados por el matemático inglés Alan Turing (1912-1954), durante la Segunda Guerra Mundial diversos científicos aliados colaboraban en el desarrollo de sistemas automáticos para descifrar los códigos militares enemigos.

El bando contrario también comenzó a implementar esta estrategia, con la creación de la calculadora automática Z1, del ingeniero alemán Konrad Zuse (1910-1995), a la que seguirían las versiones Z2, Z3 y Z4, cuyo éxito pasó inadvertido debido a la guerra.

La primera computadora electromecánica, la Harvard Mark I, nació en 1944, fruto de la empresa estadounidense IBM, y enseguida aparecieron sus sucesoras Colossus Mark I y Colossus Mark 2.

Ilustración 5 Ordenadores Mark

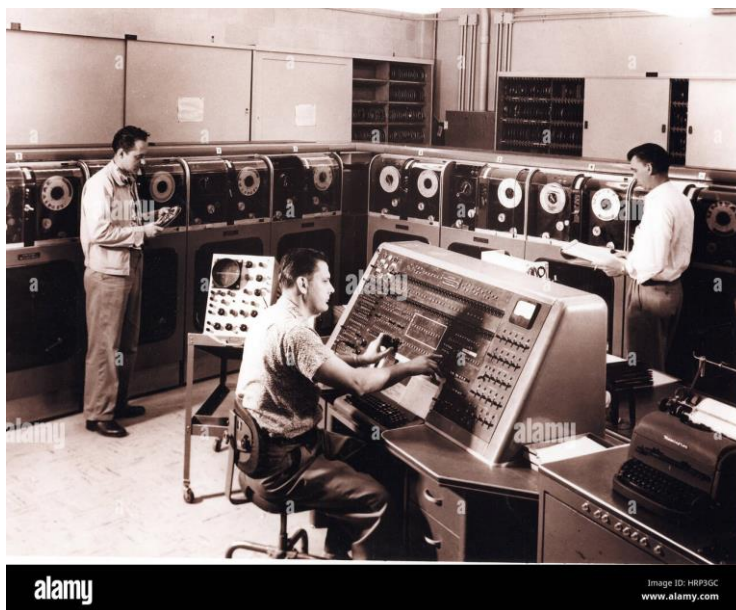


Fuente: <https://www.neoteo.com/la-historia-de-los-ordenadores-mark/>

A partir de entonces, cinco generaciones de computadoras se han desarrollado sucesivamente, logrando cada vez mayores capacidades:

- La primera generación. Aparecida en 1951, se componía de voluminosas máquinas de cálculo dotadas de bulbos, finos tubos de mercurio líquido y tambores magnéticos. Los operadores debían ingresar los programas de control mediante colecciones de tarjetas de cartón perforadas, en las que se cifraba en código binario (presencia o ausencia del agujero) la información. El punto de inicio de esta generación es la comercialización del computador UNIVAC, que pesaba unas 30 toneladas y requería una sala completa.

Ilustración 6 computador UNIVAC



Fuente: <https://www.alamy.es/imagenes/computadora-univac.html?sortBy=relevant>

- La segunda generación. La primera revolución en el mundo de los computadores la produjo la introducción en 1959 de los transistores, que sustituyeron a las válvulas al vacío y permitieron mayor rapidez de cómputo, menor tamaño físico del sistema y menores necesidades de ventilación y enfriamiento. Estas máquinas se beneficiaron de la invención de COBOL, el primer lenguaje de programación de la historia.
- La tercera generación. Surgida a partir de la invención en 1957 de los circuitos integrados (en pastillas de silicio) y su introducción al mundo de la informática en 1964. Esto brindó mayores capacidades logísticas y permitió flexibilizar el uso del computador a través de la multiprogramación, lo cual brindó gran versatilidad a la industria de las computadoras.
- La cuarta generación. Gracias a la miniaturización de los circuitos integrados, la invención del procesador y microprocesador tuvo lugar en 1971 y con ella una nueva revolución informática. Los chips y microchips resultaron ser baratos, potentes y eficaces, y su

tecnología de semiconductores permitió el desarrollo de computadores pequeños, prácticos y potentes. Los computadores comerciales se convierten ya en parte indisoluble de la vida y el trabajo, y comienzan sus primeras conexiones abiertas en red, que darán en 1990 origen a la Internet.

Ilustración 7 microcomputadores cuarta generación



Fuente: <https://www.alamy.es/seleccion-de-diferentes-dispositivos-de-almacenamiento-informatico-de-datos-e-informacion-incluyendo-un-cd-dvd-disquete-memoria-usb-tarjeta-compact-flash-y-tarjeta-sd-ver-image270197891.html>

- La quinta generación. A partir de este punto, se hace muy difícil identificar propiamente las generaciones de computadoras, ya que hay cambios revolucionarios sucediéndose constantemente. Pero la aparición de veloces soportes de información (CD, DVD, Flash drive), de nuevos protocolos de interconexión, y mucho después de los teléfonos inteligentes, de las tabletas y de todo un mundo informático táctil y Wireless (sin cable), son claros indicios de que a partir de la década del 2000 una nueva generación de computadoras inundó los mercados.

Hay quienes señalan hacia las computadoras cuánticas, o hacia la inteligencia artificial, como sus posibles caminos futuros, pero únicamente el tiempo dirá si tenían razón.

La informática en el tiempo

La Informática es la rama de la Ingeniería que estudia el hardware, las redes de datos y el software necesarios para tratar información de forma automática.

El hardware son los ordenadores de sobremesa, los portátiles, los tablets, los teléfonos móviles, las impresoras, las consolas de videojuegos, los lectores de DVDs, los reproductores de música. Estos aparatos están formados internamente por componentes electrónicos a los que también se les llama hardware: el microprocesador, las tarjetas de memoria, las tarjetas gráficas, los discos duros o los acelerómetros; hay miles de dispositivos llamados sistemas empotrados que ayudan a los coches a tomar mejor las curvas, a los aviones a volar en las peores condiciones atmosféricas o a los semáforos a controlar el tráfico de forma inteligente; también son hardware y también se estudian en Informática.

Es difícil imaginar el hardware aislado. Casi todo el hardware está conectado a través de redes de datos.

El Internet es la mayor red de datos en el mundo. Millones de ordenadores, tablets, teléfonos e incluso automóviles están continuamente conectados a esta red para intercambiar información en tiempo real y hacer más fácil la comunicación.

El software son programas que dicen al hardware qué tiene que hacer: intercambiar un mensaje con un familiar, mostrar la cartelera de cine, encontrar oportunidades de estudio en el extranjero, visualizar las últimas fotos de nuestros amigos en una red social, hacer una videoconferencia, generar las imágenes de un video juego o también intercambiar datos con un teléfono o un tablet.

El hardware son las máquinas y el software lo que nos permite decirles qué queremos que hagan. Hardware y software son las especialidades clásicas de la Informática, pero hay otras que se llaman interdisciplinarias porque la mezclan con otras ramas del conocimiento, ejemplo, las especialidades de Sistemas de Información y Tecnologías de la Información mezclan la Informática con la gestión empresarial; la de Informática Clínica trata sobre la aplicación de la informática y las comunicaciones en el ámbito de la salud; y la de Ingeniería Biomédica la mezcla con la Ingeniería Mecánica y con la Biomedicina.

En el mundo del software hay dos especialidades:

La Ingeniería del Software, en la que se estudia cómo desarrollar software en un contexto empresarial; y, la Computación, en la que se estudian problemas complejos como la inteligencia artificial, el reconocimiento del habla o la búsqueda de información en la Web.

2.2. Investigación de campo. -

Según Stephen **Rhotonⁱ (2023)** Investigación de campo, estudio de campo o trabajo de campo, es el proceso que permite obtener datos de la realidad y estudiarlos tal y como se presentan, sin manipular las variables. Por esta razón, su característica esencial es que se lleva a cabo fuera del laboratorio, en el lugar de ocurrencia del fenómeno.

La investigación de campo usa instrumentos como ficheros o representaciones estadísticas. Combinados con técnicas como la observación o la encuesta, estos permiten recopilar y analizar los datos que se van a estudiar.

Una de sus principales ventajas es que, al llevarse a cabo en el lugar del fenómeno, los datos recolectados son más fiables. Sin embargo, el hecho de trasladarse al terreno puede resultar costoso para un equipo de investigación.

La investigación de campo permite generar nuevos conocimientos aplicando el método científico. Por ello, se considera esencial tanto en las ciencias puras como en las ciencias sociales.

2.3. Muestra. -

Muestra es una porción de la totalidad de un fenómeno, producto o actividad que se considera representativa del total también llamada una muestra representativa.

Muestra viene de mostrar siendo que da a conocer a los interesados o públicos objetivos resultados, productos o servicios que ejemplifican o sirve como demostración de un tipo de evento, calidad o la estandarización.

En estadísticas, la muestra es una porción extraída mediante métodos específicos que representan los resultados de una totalidad llamada población usando la probabilidad como, por ejemplo, “la muestra estadística de 100 personas que se someten a una encuesta para conocer la satisfacción de un producto”.

2.4. Cálculo de la muestra

El repositorio prestará servicio a poblaciones externas del instituto, como a personal docente, estudiantes y administrativo del ISTP, por lo cual se establece una población infinita de usuarios.

Ante esta posición para calcular la muestra de población se utilizará la fórmula para datos infinitos.

$$Muestra = \frac{z^2 PQ}{E^2}$$

Las variables son las siguientes:

PROBABILIDAD DE ÉXITO= P= 0.94

NIVEL DE NO ÉXITO =q= 0.06

VARIANZA= v =1.89

ERROR = e= 0.05

$$Muestra = \frac{1.89^2 0.94 * 0.06}{0.05^2}$$

$$Muestra = \frac{3.5721 * 0.94 * 0.06}{0.0025}$$

$$Muestra = \frac{0.20146}{0.0025}$$

$$Muestra = 80.58$$

Obteniéndose un aproximado de muestra poblacional de 81 personas.

2.5. Encuesta

Las encuestas son un tipo de instrumentos de recopilación de información, que consisten en un conjunto prediseñado de preguntas normalizadas, dirigidas a una muestra socialmente representativa de individuos, con el fin de conocer sus opiniones o visiones respecto de alguna problemática o asunto que les afecta.

Dichas preguntas están siempre diseñadas, de acuerdo a la hipótesis que buscan someter a la comprobación empírica de la opinión pública. Por consiguiente, pueden ser de dos tipos distintos:

- **Abiertas**, cuando el encuestado puede responder a las preguntas con sus propias palabras, novedosas y diferentes.
- **Cerradas**, cuando al encuestado se le ofrece un conjunto de respuestas posibles y se le pide que elija la que mejor se adecúa a su opinión.

Encuesta

Agradezco por el tiempo concedido para responder a esta encuesta, que requiere de su información para elaborar el trabajo de Titulación previo a obtener el Título de Tecnólogo en Sistemas de automatización e instrumental, su información será confidencial y solo servirá para el presente proyecto.

Identificación del Encuestado

Sexo: Masculino __ Femenino ____ Edad: _____ años

Ocupación: _____

Cuestionario:

1 ¿Cuáles de los siguientes medios o recursos tecnológicos utiliza en sus labores escolares?

- ☐ Computadores ☐ Televisores ☐ DVD ☐ Proyectores
- ☐ Equipos de audio ☐ Celular ☐ Software educativo

2 ¿Qué medios tecnológicos existen en la Institución Educativa?

- ☐ Computadores ☐ Televisores ☐ DVD ☐ Proyectores ☐ Equipos de audio ☐ Celular ☐ Software educativo

3 ¿Señale las herramientas virtuales que sabe manejar o utiliza con mayor frecuencia?

- ☐ Computador ☐ Televisor ☐ DVD ☐ Proyector ☐ Celular
- ☐ Equipos de audio

4 ¿Cuál es la utilidad que le das a las nuevas tecnologías de la comunicación TIC?

- ☐ Ayuda en las tareas ☐ Aprender nuevas cosas ☐ Para jugar
- ☐ Para comunicarse con sus amigos o familiares

5 ¿Cuándo desarrollas una investigación técnica que competencias utilizas?

- ☐ Interpretativa, comprende los textos y reconstruye su significado.
- ☐ Argumentativa, organiza y sustenta premisas alrededor del texto.
- ☐ Propositiva, genera hipótesis y da alternativa de soluciones según su relación con el contexto.

6 La utilización de repositorios bibliográficos en la web como contribuye en tus investigaciones

- ☐ Incentiva el proceso investigativo.
- ☐ Facilita el conocimiento respecto al tema investigado
- ☐ Fortalece las capacidades de interpretación de las fuentes documentales.
- ☐ Los sustentos antes logrados fortalecen nuestras conclusiones

7 ¿Qué sugieres que los repositorios expuestos por las instituciones investigadores permitan al nuevo investigador?

- ☐ Contengan sustentos abalizados por profesionales
- ☐ Contengan textos completos en archivos .pdf
- ☐ Codificación acorde al tema y alfabético
- ☐ Se puedan utilizar sin previa autorización

2.6. Procesamiento de datos

Identificación del Encuestado

Sexo: Masculino __ Femenino ____ Edad: _____ años

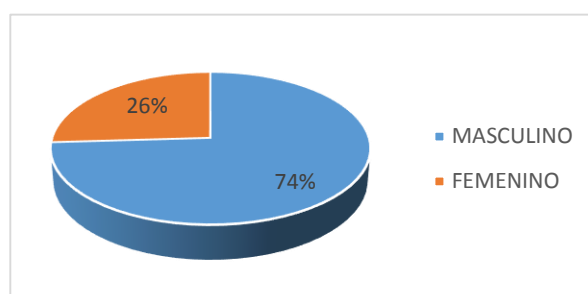
Ocupación:

Tabla 2 Identificación del encuestado

Sexo	Encuestados	Porcentaje
MASCULINO	60	74%
FEMENINO	21	26%
Total	81	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

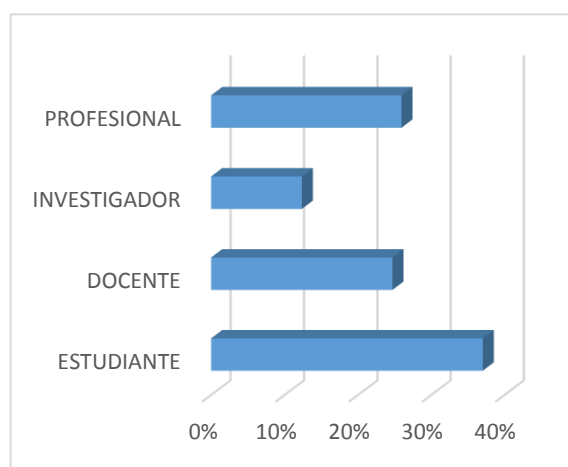
Elaborado por: Alejandro Flores

Tabla 3 Ocupación del encuestado

Ocupación	Encuestado	Porcentaje
ESTUDIANTE	30	37%
DOCENTE	20	25%
INVESTIGADOR	10	12%
PROFESIONAL	21	26%
Total	81	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

Se encuestó al 74% de hombres y el 26% de mujeres, con ocupación diferente, así: estudiantes un 37%, docentes 25%, investigadores el 12% y profesionales un 26%, llegando al 100% de la muestra.

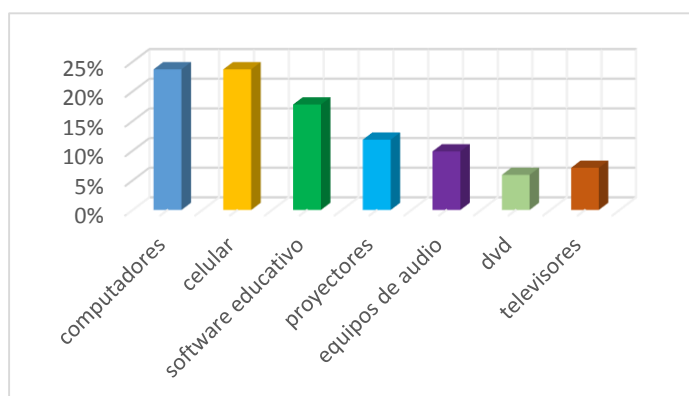
Pregunta No.1 ¿Cuáles de los siguientes medios o recursos tecnológicos utiliza en sus labores escolares?

Tabla 4 Medios Tecnológicos más usados

Variables	Encuestados	Porcentaje
Computadores	60	24%
Celular	60	24%
Software educativo	45	18%
Proyectores	30	12%
Equipos de audio	25	10%
DVD	15	6%
Televisores	18	7%
Total	253	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

De los encuestados manifiestan que en su mayoría ocupan los siguientes medios tecnológicos: computadores un 24% a igual que los celulares un 24%, un 18% utiliza el software, un 12% los infocus y equipos de audio el 10%.

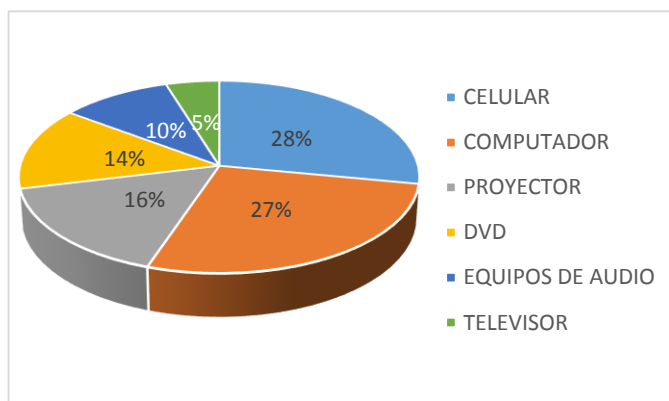
Pregunta No. 2 ¿Qué medios tecnológicos existen en la Institución Educativa?

Tabla 5 Medios Tecnológicos en unidades educativas

Variables	Encuestados	Porcentaje
Celular	70	28%
Computador	68	27%
Proyector	40	16%
Dvd	35	14%
Equipos de audio	25	10%
Televisor	12	5%
Total	250	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

Los encuestados han señalado las herramientas de mayor utilidad, la misma que señala el manejo de varias herramientas, siendo la de mayor conocimiento la computadora y el celular con un 27% y 28% respectivamente, y además otras herramientas, pero en mayor porcentaje por su variación de innovación.

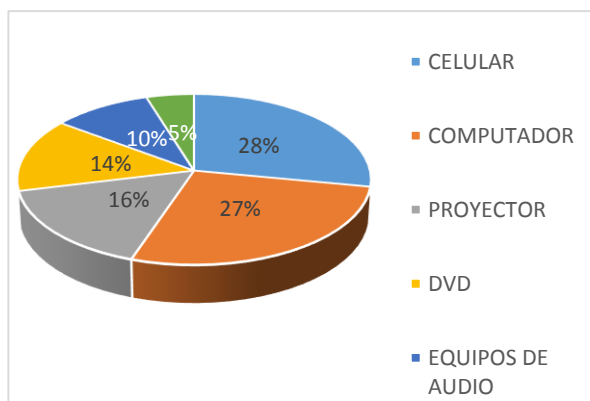
Pregunta No. 3 ¿Señale las herramientas virtuales que sabe manejar o utiliza con mayor frecuencia?

Tabla 6 Herramientas virtuales más usadas

Variables	Encuestados	Porcentaje
CELULAR	70	28%
COMPUTADOR	68	27%
PROYECTOR	40	16%
DVD	35	14%
EQUIPOS DE AUDIO	25	10%
TELEVISOR	12	5%
Total	250	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

La mayoría de los encuestados manifiestan conocer el manejo de varios equipos tecnológicos y en su mayoría en celulares y computación. En menor porcentaje de proyectores, dvd, equipos de audio y televisor.

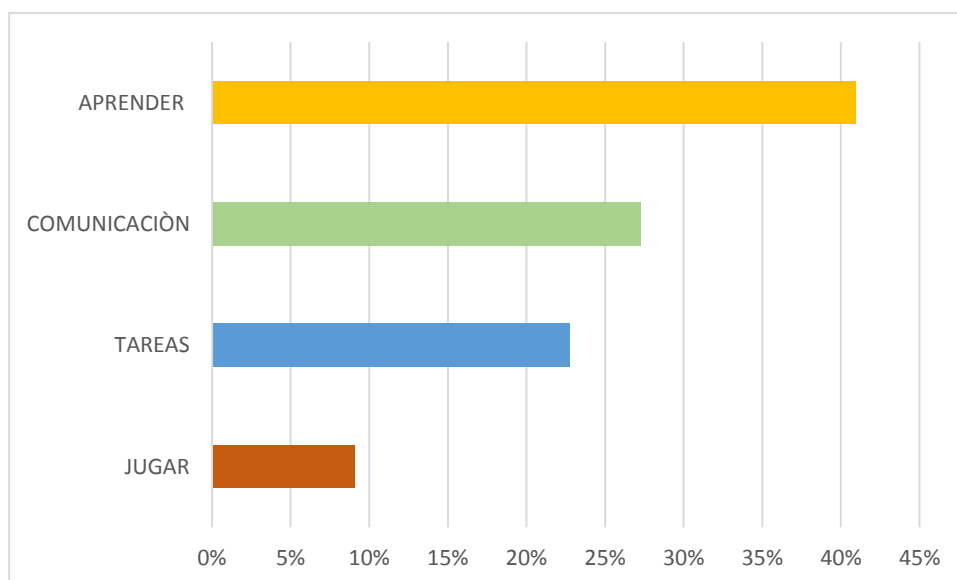
Pregunta No. 4 ¿Cuál es la utilidad que le das a las nuevas tecnologías de la comunicación TIC?

Tabla 7 Uso de la Tecnología

Variables	Encuestados	Porcentaje
JUGAR	10	9%
TAREAS	25	23%
COMUNICACIÓN	30	27%
APRENDER	45	41%
Total	110	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

La mayoría de usuario de la tecnología lo realiza para aprender nuevas cosas, manifestado por 41%, el 27% para comunicarse con familiares, el 23% para realizar tareas escolares. Y el 9% para jugar.

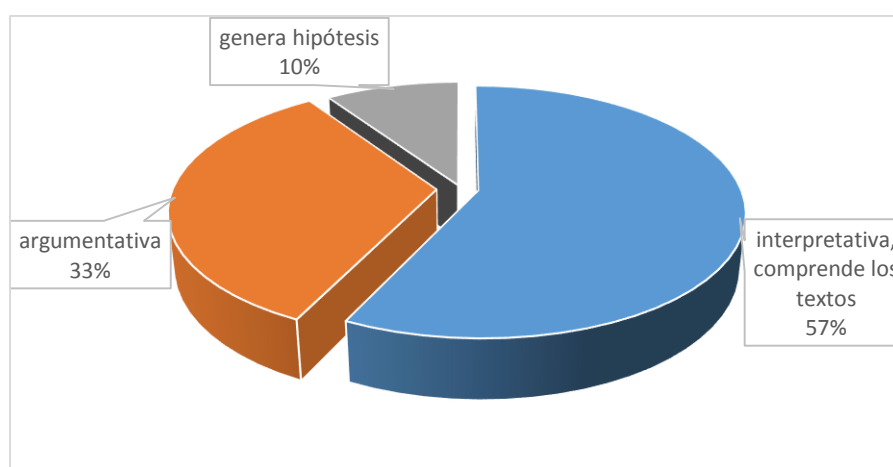
Pregunta No. 5 ¿Cuándo desarrollas una investigación técnica que competencias utilizas?

Tabla 8 Aplicación de competencias al usar la tecnología

Variable	Encuestados	Porcentaje
Interpretativa, comprende los textos	60	57%
Argumentativa	35	33%
Genera hipótesis	10	10%
Total	105	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

El 57% señala que aplica la competencia Interpretativa, comprende los textos y reconstruye su significado, el 33% aplica la competencia Argumentativa, organiza y sustenta premisas alrededor del texto. Y el 10% para generar hipótesis.

Pregunta No. 6 La utilización de repositorios bibliográficos en la web como contribuye en tus investigaciones

Tabla 9 Repositorios bibliográficos

Variable	Encuestados	Porcentaje
Incentiva proceso de investigación	45	25%
Facilita investigación	38	21%
Facilita capacidad de interpretación	20	11%
Fortalece experiencias	80	44%
Total	183	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

Al consultar los repositorios de fuentes bibliográficas contribuye a un 44% en fortalecer los conocimientos, a un 25% incentiva al proceso de investigación, al 21% facilita la investigación a un 11% facilita capacidad de interpretación.

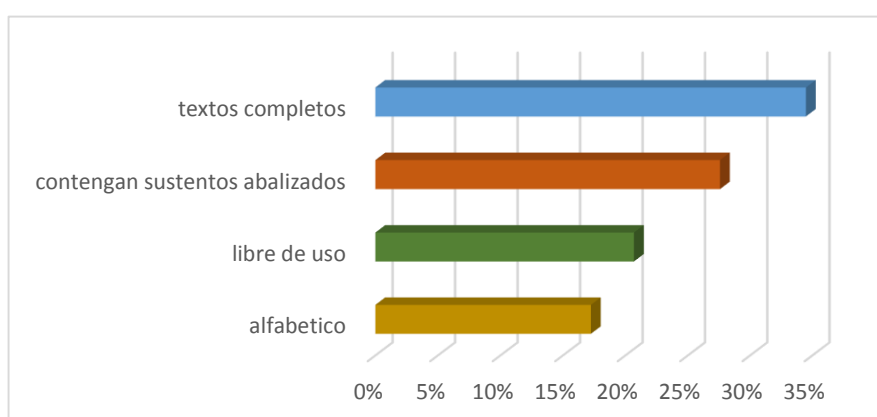
Pregunta No. 7 ¿Qué sugieres que los repositorios expuestos por las instituciones investigadores permitan al nuevo investigador?

Tabla 10 Sugerencias de repositorios

Variable	Encuestados	Porcentaje
alfabético	15	17%
libre de uso	18	21%
contengan sustentos abalizados	24	28%
textos completos	30	34%
Total	87	100%

Fuente: Investigación directa

Elaborado por: Alejandro Flores



Fuente: Investigación directa

Elaborado por: Alejandro Flores

Interpretación

La mayoría de los encuestados solicitan que los textos que están en los repositorios sean con textos completos, 34%, que los textos tengan previo abalización de ser reales (28%) y libre uso el 21%.

2.7. FLUJOGRAMA DEL PROCESO

1. Inicio:

Inicio del proceso: "El usuario ingresa al sitio web".

2. Pantalla Principal:

Se muestran opciones como "Búsqueda por Facultad", "Búsqueda por Carrera", "Búsqueda General", "Nosotros", "Registro", "Login Estudiante", "Login Administrador".

3. Decisiones y Acciones:

Búsqueda por Facultad:

Decisión: "¿Seleccionó una Facultad?"

Acción: "Muestra proyectos de la Facultad seleccionada".

Búsqueda por Carrera:

Decisión: "¿Seleccionó una Carrera?"

Acción: "Muestra proyectos de la Carrera seleccionada".

Búsqueda General:

Decisión: "¿Ingresó los datos correctos?"

Acciones: "Muestra la información solicitada" o "Mensaje de error: No existe información con las palabras buscadas".

Nosotros

Acción: "Muestra información general del instituto".

Registro:

Decisión: "¿El usuario llenó todos los campos?"

Acciones: "Información enviada al administrador" o "Mensaje de error: Correo o contraseña no son válidos".

Login Estudiante:

Decisión: "¿El usuario ha sido validado por el administrador?"

Acciones:

Si es validado: "El estudiante puede cargar proyectos".

Subproceso: "Cargar proyecto con los campos Título, Año, Resumen, Integrantes, Imagen, Documento (solo PDF)".

Decisión: "¿Los datos están completos?"

Acción: "El proyecto se carga para validarse por el administrador".

Si no es validado: "Mensaje de error: Correo o contraseña no son válidos".

Login Administrador:

Decisión: "¿El administrador ingresó el correo y contraseña válidos?"

Acción: "Mostrar pantalla de administración con dashboard".

Opciones en el Dashboard:

Archivos: "Ver proyectos, actualizar estado o eliminarlos".

Estudiantes: "Ver, habilitar, deshabilitar o eliminar estudiantes".

Facultades: "Ver, editar, crear o eliminar facultades".

Carreras: "Ver, editar, crear o eliminar carreras".

Usuarios: "Verificar, autorizar, eliminar o editar permisos y usuarios".

Ajustes: "Verificar o cambiar la información de 'Nosotros'".

4. Finalización:

Final del proceso en cada flujo específico.

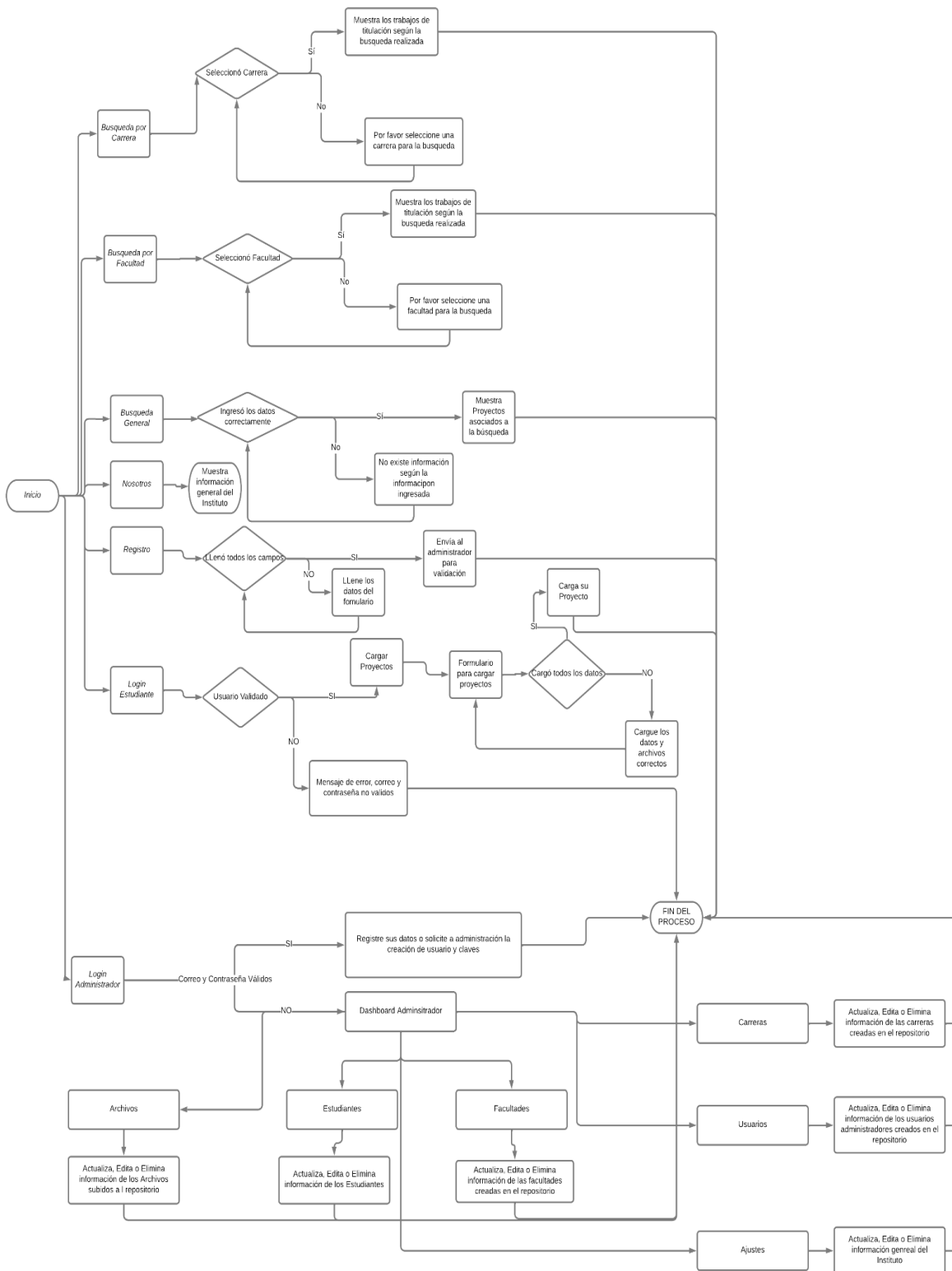


Ilustración 8 Diagrama de flujo

CAPITULO III

INGENIERIA DEL PROYECTO

3.1 Planificación de la ejecución del proyecto

A fin de construir un programa acorde a las necesidades del Instituto, fue necesaria la planificación tanto de recursos, como software, hosting, equipos; así como, de entender las necesidades específicas del usuario a fin de entregar un producto que cumpla con las expectativas esperadas.

En base a lo expuesto, es necesario conocer a breves rasgos conceptos generales de informática, a fin de que la información sea digerible y se logre comprender a perfección las acciones a realizarse, así como las herramientas que se usan para el desarrollo de la aplicación.

Los programas informáticos diseñados para soportar la programación y las aplicaciones, incluidos aquellos que se pueden combinar para completar una tarea, se conocen como herramientas de desarrollo. Estas herramientas son usadas por desarrolladores de software para diversas tareas o funcionalidades. La creación de programas requiere el uso de distintos lenguajes o códigos de programación, que dependen del software que el programador desee desarrollar o de las instrucciones que desee. Es así que se procede a listar algunos lenguajes de programación conocidos y utilizados para la creación de herramientas de sistematización, como páginas web, a fin de brindar mayor comprensión sobre los mismos.

PHP: PHP es el acrónimo recursivo de Hypertext Preprocessor, un lenguaje de programación de código abierto muy utilizado especialmente en el desarrollo de aplicaciones y web, este lenguaje de código puede ser incrustado en HTML5 puesto que es un código

muy variable favorece el enlace entre la interfaz de usuario y los servidores.

Diferenciar el lenguaje de programación PHP con los típicos como pueden ser HTML, CSS o JavaScript es sencillo, estos último son leídos primero por el navegador al momento de acceder a una web, mientras que PHP se ejecuta directamente en el servidor de la página.

MySQL: Es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL). MySQL funciona prácticamente en todas las plataformas, incluyendo Linux, UNIX y Windows. Aunque puede utilizarse en una amplia gama de aplicaciones, MySQL se asocia más a menudo con las aplicaciones web y la publicación en línea.

Framework: Es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software. Utilizar frameworks puede simplificar una tarea o proceso, de ahí que se trate de una de las herramientas habituales que manejan los desarrolladores.

Bootstrap: Es un framework CSS utilizado en aplicaciones frontend — es decir, en la pantalla de interfaz con el usuario— para desarrollar aplicaciones que se adaptan a cualquier dispositivo.

PhpMyadmin: Es una aplicación web que sirve para administrar bases de datos MySQL de forma sencilla y con una interfaz amistosa. Se trata de un software muy popular basado en PHP. La ventaja de usar una aplicación web es que nos permite conectarnos con servidores remotos, a los cuales no siempre se puede acceder usando programas de interfaz gráfica.

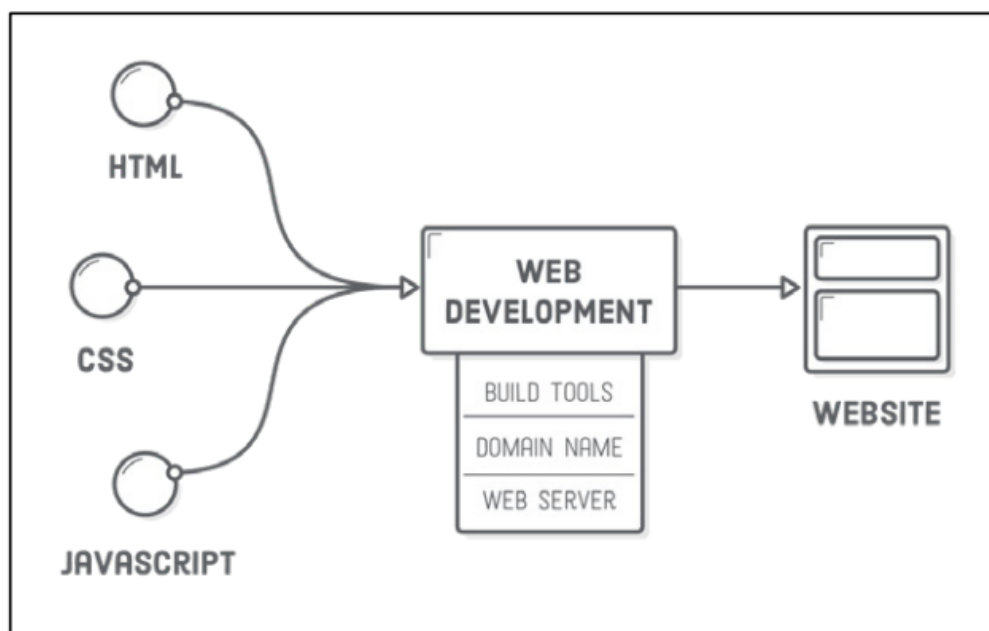
Javascript: Es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.

HTML5: Es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página.

CSS3: Es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas.

Entre los lenguajes enumerados anteriormente, HTML, CSS y JavaScript son herramientas cruciales que desempeñan un papel importante en el desarrollo de sitios web, contribuyendo a la creación de elementos visuales y elementos interactivos. Para lograr su objetivo principal de promover el uso de sitios web y ganar participación de mercado, la estructura general de los sitios web debe ser consistente tanto en el diseño de la información como en la distribución del contenido. La estructura de una página web está directamente ligada a la arquitectura de la información y la distribución de contenidos, con el objetivo de optimizar el uso de la información, priorizando la velocidad y facilidad en las búsquedas, lo que requiere una cuidadosa consideración. Al respecto Valdivia (2020) dice: *“El objetivo principal de casi todos los sitios web es que el usuario sea capaz de encontrar lo que busca rápida y fácilmente, por lo que una buena navegabilidad que invite al usuario a recorrer el sitio web es imprescindible”*. Para mayor claridad de este tema, en la ilustración No. 8 se muestra la relación de los tres lenguajes en el desarrollo web.

Ilustración 9 Estructura general de página web

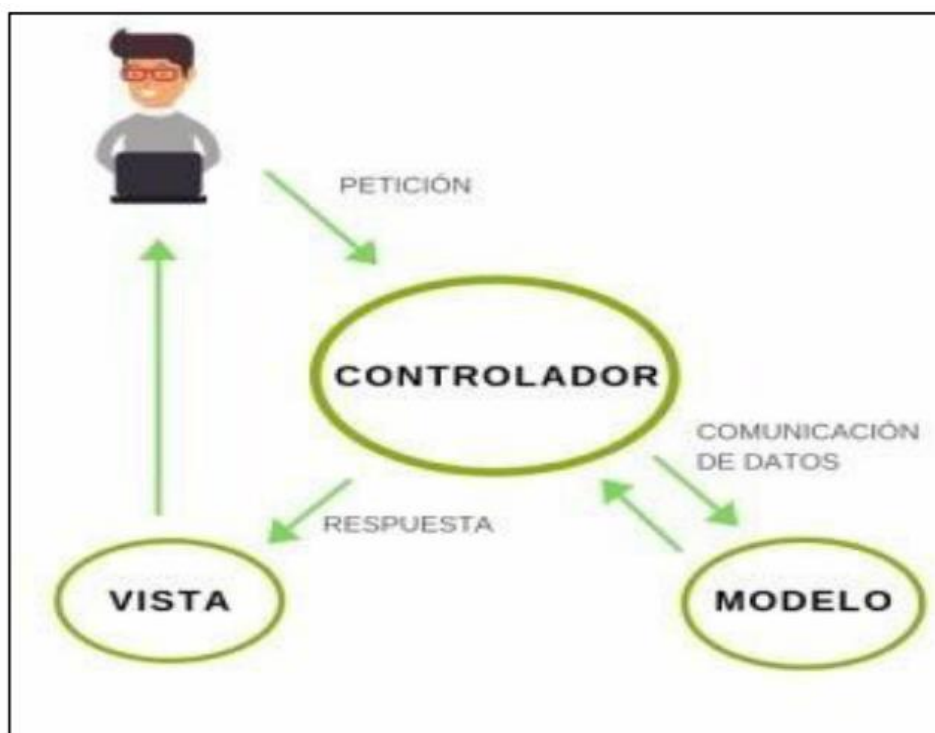


Fuente: "Idiomas versus desarrollo web", <https://internetingshard.netlify.app/index.html> (James, 2017)

Nota. La imagen muestra el modelo de la parte del backend, donde se integran los códigos fuentes HTML, CSS y JS. Como resultado se tiene el frontend, la parte final que el navegador muestra al usuario. Después de comprender la estructura general de una página web, es importante comprender la arquitectura de software que divide los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes interconectados. Esta arquitectura se conoce dentro del medio tecnológico como el Modelo-Vista-Controlador (MVC), y se muestra gráficamente en la Figura 2. El modelo, que incluye una representación de los datos que maneja el sistema, así como la lógica comercial y los mecanismos de persistencia. La Vista, también conocida como interfaz de usuario, es la parte que compone la información que se envía al cliente y los mecanismos por los cuales interactúa con él. El Controlador sirve como intermediario entre el Modelo y la Vista, controlando el flujo de información entre ellos y realizando transformaciones para adaptar los datos a las necesidades de cada uno. El modelo Vista Controlador es un patrón de diseño de

software que divide el programa o la aplicación web en tres partes interconectadas. Cada uno de estos componentes se construye para manejar aspectos de desarrollo específicos de una aplicación. El Controlador gestiona las solicitudes entrantes, envía información y establece las interacciones entre los componentes del Modelo y la Vista. El Modelo es el componente central del patrón, maneja directamente los datos, la validación de datos, la lógica y las reglas de la aplicación. Interactúa con la base de datos. Pasa los datos al Controlador cuando éste lo solicita. La Vista se encarga de presentar la información, representa la interfaz de usuario. Por lo general, renderizará páginas HTML dinámicas basadas en los datos del Modelo.

Ilustración 10 Modelo-Vista-Controlador (MVC)



Fuente: "El apasionante mundo de la programación", <https://amizba-programacion.blogspot.com/2021/01/1-introduccion-laravel.html>

Nota: En esta figura el usuario interactúa con la vista de la página entregada por el navegador, a través de esta se envían peticiones al controlador que acude al modelo para intercambiar información.

Esta información se verá plasmada en la vista como respuesta final a la petición realizada por el usuario.

Para terminar, sólo queda mencionar e ilustrar con la figura 3, una de las metodologías utilizadas en la gestión de proyectos y que fue la base para el desarrollo del presente trabajo, la cual es conocida dentro del medio tecnológico como método MIDAS/DB, la cual se basa en iteraciones o fases, las que se detallan:

• **MIDAS/SD (*Requisitos de sistema*)**

En esta primera iteración se realiza un levantamiento de información, con la intención de conocer la situación actual del Instituto Superior Tecnológico “El Pacífico”, respecto de la manera en la que se presenta y mantiene la documentación digital que posee el Instituto, se realizó el levantamiento de las exigencias que se utilizaron para levantar el sistema, se evidenciaron las necesidades de la institución y se obtuvo el conocimiento de la problemática existente en el Instituto.

Se evidenció que al momento toda la documentación digital que posee el instituto se almacena únicamente en CDs los que reposan junto a las tesis presentadas por el cuerpo colegiado de la institución, lo que hace que esta sea una información vulnerable y susceptible a cualquier alteración o eliminación.

• **MIDAS/HT (*Interfaz de usuario*)**

Una vez verificada la problemática y realizado el análisis correspondiente: y, posterior a la interpretación del análisis, se procedió a realizar la clasificación de la información que se debe mostrar al usuario, como se va a organizar y que funciones se pueden o no realizar el sistema, así también se obtuvo una idea en contexto general de los procesos solicitados para el diseño de la interfaz, los que incluyen, la visualización de los proyectos, los autores, fechas de publicación, resumen y link para visualizar el trabajo completo, entre otra funciones de acuerdo al perfil.

Requisitos funcionales

El sistema web debe controlar y prohibir el acceso de usuarios, considerando que cada usuario registrado cuente con una cuenta y claves diferentes y privadas para el sistema.

Cada usuario administra el área de la cual es responsable en el instituto, sea esta Administración General, Secretaría, Personal docente.

Las claves de los usuarios deben ser encriptadas a fin de que estas no sean reconocidas fácilmente.

Requisitos no funcionales

El realizó el diseño de un sistema capaz de gestionar, almacenar y acceder a documentación digital respecto de tesis presentada por los estudiantes, el modelo de programa diseñado cumple con estos requisitos:

- El sistema realiza estos tres procesos (almacenamiento, muestra y eliminación de datos)
- Cada usuario tiene su perfil con claves privadas, el administrador podrá acceder a todos los permisos.
- Cada proceso debe ser validado previo a ser publicado.

Limitaciones de diseño

- El acceso al sistema es realizado por el administrador, personal de secretaría, docentes y alumnos; y, cada uno de ellos tendrá permisos para realizar acciones específicas.
- Se estima una respuesta de segundos en la apertura de páginas y el contenido.

Hardware

Respecto del hardware requerido para el uso y funcionamiento de este sistema, se estima su uso con requisitos básicos tales como:

- Procesador Intel Core i3 y posteriores
- Memoria (RAM) de 4 GB
- Disco con 250 GB
- Sistema operativo Windows 7 y posteriores

- Exploradores IE v.9, Google Chrome v.22, Mozilla v.34

Los archivos se alojarán dentro del mismo servidor que contiene el sistema operativo de la página web principal del instituto. Servicio obtenido con C-PANEL.

• **MIDAS/DB (Base de datos)**

A fin de conservar usuarios, claves, registros de estudiantes; y, con la intención de ocupar la menor cantidad de espacio en memoria, se mantiene la base de datos existente dentro del instituto, únicamente se incrementan tablas que se relacionan tanto con tablas de usuarios como de claves, se mantienen estándares de nombres de tablas, de atributos y de datos.

• **MIDAS/FC (Funcionalidad)**

Una vez se realizó el diseño de bases de datos, así como de interfaces, se procede con la implementación, para que la información almacenada se encuentre disponible para usuarios internos y externos.

Se utilizó un lenguaje de programación tipo servidor PHP, utilizando plantillas Bootstrap a fin de que la interfaz sea amigable y de fácil manejo.

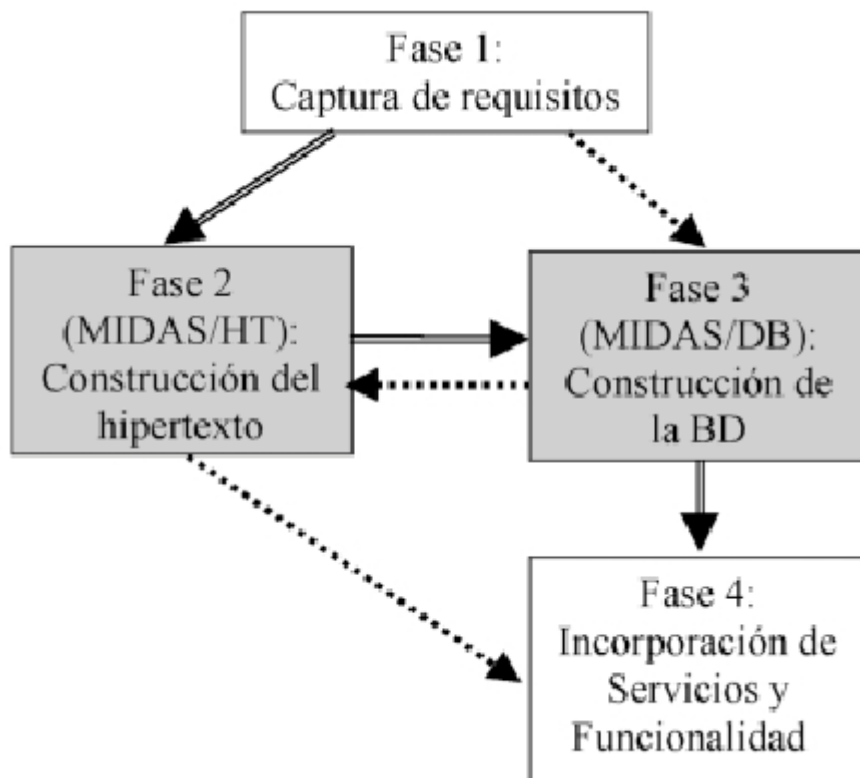
• **MIDAS/TST (Pruebas del Sistema)**

Una vez resueltas las fases anteriores se solicitó un espacio en el almacenamiento (hosting) en los servidores locales proporcionados por el Data Center del Instituto, a fin de alojar el sistema tipo web que se realizó, integrándolo con un subdominio que de igual manera fue proporcionado por el Data Center.

Se realizan pruebas continuas a fin de validar que el sistema cumpla su propósito y en caso de incurrir en errores, estos serán corregidos a la inmediatez posible.

Para una mayor comprensión se incluye una imagen referencial del proceso de aplicación del método MIDAS/DB tal como se aprecian en la ilustración No. 11.

Ilustración 11 Metodología MIDAS/DB



Fuente: El proceso de creación de una base de datos web

https://www.researchgate.net/publication/43309049_El_proceso_de_creacion_de_una_base_de_datos_web

3.2 Selección de software de programación

Como se ha mencionado con anterioridad; y, en razón de que la plataforma virtual del Instituto se ha desarrollado con lenguaje PHP, se ha procedido entonces al desarrollo del presente repositorio en el mismo lenguaje, y manteniendo los parámetros establecidos en el desarrollo de la plataforma.

Por tal razón, es preciso ampliar el conocimiento del lenguaje PHP; así como, sus ventajas, desventajas y características:

¿Qué es PHP?

PHP es, como se mencionó anteriormente es un lenguaje de programación de código abierto muy utilizado especialmente en el desarrollo de aplicaciones y web, este lenguaje se ejecuta en el

servidor web, lo que significa que se abre la posibilidad de realizar cambios y modificaciones a las páginas web o aplicaciones desarrolladas antes de enviar al cliente.

Entre las tareas que permiten a este lenguaje realizar contenidos dinámicos están los accesos a las bases de datos y las conexiones de red, las que hacen que este lenguaje sea tan aplicado en el mundo del desarrollo web.

Según la documentación que se encuentra habilitada en *The PHP Documentation Group (2014)* las características más importantes de PHP son las siguientes:

- Es un software libre, no es necesario comprar licencias.
- Es compatible con la mayoría de los servidores web.
- Maneja sesiones HTTP.
- Soporta la programación POO (Programación Orientada a Objetos).
- Maneja excepciones.
- Mantiene una gran cantidad de módulos y extensiones a fin de ampliar las funcionalidades.
- Maneja protocolos de internet.
- Opera varias técnicas de programación.
- Es compatible con la mayoría de gestores de bases de datos.
- Soporta manejo de cookies.
- Dispone la carga de archivos.
- Es uno de los lenguajes con una rápida ejecución.
- Cuenta con un sinnúmero de frameworks que facilitan el uso de las características del lenguaje.
- Implementa de forma sencilla patrones de diseño como MVC (Modelo Vista Controlador).

De igual manera, según la documentación que se encuentra habilitada en *The PHP Documentation Group (2014)* las ventajas de PHP son las siguientes:

- Es un lenguaje multiplataforma.

- Es un software libre.
- Cuenta con amplia documentación pública en su página web, con funciones, ejemplos de código, generalidades respecto de seguridad.
- Permite el uso de varias técnicas de programación.
- Se puede incluir dentro de HTML el código PHP.
- Cuenta con una gran variedad de api's y extensiones habilitadas.

Respecto de las desventajas del lenguaje de programación, de acuerdo con la documentación que se encuentra habilitada en *The PHP Documentation Group (2014)* las desventajas de PHP son las siguientes:

- La capacidad que tiene el lenguaje para incrustarse en HTML; así como, la aplicación de varias técnicas de programación a veces resulta en un código poco legible y desorganizado.
- No existe el soporte suficiente en ciertas funcionalidades de carácter visual y de interacción con el usuario.

3.3 Programación

Se inició con la elaboración de la página de INICIO, misma que se programó como Index.php, esta página contiene toda la información y los hipervínculos que el usuario puede visualizar al ingresar a la primera página del repositorio.

```

index.php
1  <?php
2
3  ini_set('display_errors', '1');
4  ini_set('display_startup_errors', '1');
5  error_reporting(E_ALL);
6
7  require_once('./config.php');
8
9  ?>
10 <!DOCTYPE html>
11 <html lang="es" style="height: auto;">
12 <head>
13   <meta charset="UTF-8">
14   <meta name="viewport" content="width=device-width, initial-scale=1.0">
15   <title>Bienvenido</title>
16   <style>
17     #header {
18       height: 70vh;
19       width: 100%;
20       position: relative;
21       top: -1em;
22     }
23     #header:before {
24       content: "";
25       position: absolute;
26       height: 100%;
27       width: 100%;
28       background-image: url(<? echo (validate_image($_settings->info("cover")) ?>);
29       background-size: cover;
30       background-repeat: no-repeat;
31       background-position: center center;
32     }
33     #header > div {
34       position: absolute;
35       height: 100%;
36       width: 100%;
37       z-index: 2;
38     }
39
40     #top-Nav a.nav-link.active {
41       color: #001f3f;
42       font-weight: 900;
43       position: relative;
44     }
45     #top-Nav a.nav-link.active:before {
46       content: "";
47       position: absolute;
48       border-bottom: 2px solid #001f3f;
49       width: 33.33%;
50       left: 33.33%;
51       bottom: 0;
52     }
53   </style>
54   <?php require_once('inc/header.php') ?>
55 </head>
56 <body class="layout-top-nav layout-fixed layout-navbar-fixed" style="height: auto;">
57   <div class="wrapper">
58     <?php
59     $page = isset($_GET['page']) ? $_GET['page'] : 'home';
60     require_once('inc/topBarNav.php');

```

```

61     if ($_settings->chk_flashdata('success')): ?>
62         <script>
63             alert_toast("<?> htmlspecialchars($_settings->flashdata('success'),
64                 ENT_QUOTES, 'UTF-8') ?>", 'success');
65         </script>
66     <?php endif; ?>
67     <div class="content-wrapper pt-5" style="background-color: white;">
68         <?php if ($page == "home" || $page == "about_us"): ?>
69             <div id="header" class="shadow mb-4">
70                 <div class="d-flex justify-content-center h-100 w-100
71                     align-items-center flex-column px-3">
72                     <h1 class="w-100 text-center site-title"><?> htmlspecialchars($_
73                         _settings->info('name'), ENT_QUOTES, 'UTF-8') ?></h1>
74                     <a href="."?page=projects" class="btn btn-lg btn-light rounded-pill
75                         w-25" id="enrollment"><b>Explorar Proyectos</b></a>
76                 </div>
77             </div>
78         <?php endif; ?>
79         <section class="content">
80             <div class="container">
81                 <?php
82                 if (!file_exists($page . ".php") && !is_dir($page)) {
83                     include '404.html';
84                 } else {
85                     if (is_dir($page)) {
86                         include $page . '/index.php';
87                     } else {
88                         include $page . '.php';
89                     }
90                 }
91             </div>
92         </section>
93     </div>
94
95     <div class="modal fade" id="confirm_modal" role="dialog">
96         <div class="modal-dialog modal-md modal-dialog-centered" role="document">
97             <div class="modal-content">
98                 <div class="modal-header">
99                     <h5 class="modal-title">Confirmación</h5>
100                 </div>
101                 <div class="modal-body">
102                     <div id="delete_content"></div>
103                 </div>
104                 <div class="modal-footer">
105                     <button type="button" class="btn btn-primary" id="confirm">
106                         Continuar</button>
107                     <button type="button" class="btn btn-secondary" data-dismiss="modal"
108                         ">Cerrar</button>
109                 </div>
110             </div>
111         </div>
112     </div>
113
114     <div class="modal fade" id="uni_modal" role="dialog">
115         <div class="modal-dialog modal-md modal-dialog-centered" role="document">
116             <div class="modal-content">
117                 <div class="modal-header">
118                     <h5 class="modal-title"></h5>
119                 </div>
120                 <div class="modal-body"></div>
121                 <div class="modal-footer">
122                     <button type="button" class="btn btn-primary" id="submit" onclick="
123                         document.querySelector('#uni_modal form').submit()">Guardar</
124                         button>
125                     <button type="button" class="btn btn-secondary" data-dismiss="modal"
126                         ">Cancelar</button>
127                 </div>
128             </div>
129         </div>
130     </div>
131 </div>

```

```

121 <div class="modal fade" id="uni_modal_right" role="dialog">
122 <div class="modal-dialog modal-full-height modal-md" role="document">
123 <div class="modal-content">
124 <div class="modal-header">
125 <h5 class="modal-title"></h5>
126 <button type="button" class="close" data-dismiss="modal" aria-label
    ="Close">
    <span class="fa fa-arrow-right"></span>
    </button>
    </div>
    <div class="modal-body"></div>
    </div>
    </div>
    </div>
133 <div class="modal fade" id="viewer_modal" role="dialog">
134 <div class="modal-dialog modal-md" role="document">
135 <div class="modal-content">
136 <button type="button" class="btn-close" data-dismiss="modal"><span
    class="fa fa-times"></span></button>
    <img src="" alt="">
    </div>
    </div>
    </div>
141 </div>
142 <?php require_once('inc/footer.php') ?>
143 </body>
144 </html>
145

```

Posteriormente se realizó el formulario de ingreso, mismo que se denominó LOGIN.php, donde el usuario debe ingresar sus credenciales de usuario y contraseña válidas a fin de que pueda acceder al repositorio.

```

login.php
1 <?php require_once('./config.php'); ?>
2 <!DOCTYPE html>
3 <html lang="en" style="height: auto;">
4 <?php require_once('inc/header.php'); ?>
5 <body class="hold-transition">
6 <script>
7 start_loader()
8 </script>
9 <style>
10 html, body {
11 height: calc(100%) !important;
12 width: calc(100%) !important;
13 }
14 body {
15 background-image: url("<?php echo validate_image($_settings->info('cover')); ?>");
16 background-size: cover;
17 background-repeat: no-repeat;
18 }
19 .login-title {
20 text-shadow: 2px 2px black;
21 }
22 #login {
23 display: flex;
24 direction: rtl;
25 }
26 #login > * {
27 direction: ltr;
28 }
29 #logo-img {
30 height: 150px;
31 width: 150px;
32 object-fit: scale-down;
33 object-position: center center;
34 border-radius: 100%;
35 }
36 .card {
37 margin: auto;
38 }

```

```

39 </style>
40 <?php if ($_settings->chk_flashdata('success')): ?>
41     <script>
42         alert_toast("<?php echo $_settings->flashdata('success'); ?>", 'success');
43     </script>
44 <?php endif; ?>
45
46 <div class="h-100 d-flex align-items-center w-100" id="login">
47     <div class="col-7 h-100 d-flex align-items-center justify-content-center">
48         <div class="w-100">
49             <center></center>
50             <h1 class="text-center py-5 login-title"><b><?php echo $_settings->info('name'); ?>
50             </b></h1>
51         </div>
52     </div>
53     <div class="col-5 h-100 bg-gradient bg-primary d-flex justify-content-center
53     align-items-center">
54         <div class="card card-outline card-primary rounded-0 shadow col-lg-10 col-md-10
54         col-sm-5">
55             <div class="card-header">
56                 <h5 class="card-title text-center text-dark"><b>Iniciar Sesión</b></h5>
57             </div>
58             <div class="card-body">
59                 <form action="" id="slogin-form">
60                     <div class="form-group">
61                         <input type="email" name="email" id="email" placeholder="Correo" class="
61                         form-control form-control-border" required>
62                     </div>
63                     <div class="form-group">
64                         <input type="password" name="password" id="password" placeholder="Contraseña"
64                         class="form-control form-control-border" required>
65                     </div>
66                     <div class="form-group text-right">
67                         <button class="btn btn-default bg-navy btn-flat">Ingresar</button>
68                     </div>
69                 </form>
70             </div>
71         </div>
72     </div>
73 </div>
74
75 <!-- jQuery -->
76 <script src="plugins/jquery/jquery.min.js"></script>
77 <!-- Bootstrap 4 -->
78 <script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
79 <!-- AdminLTE App -->
80 <script src="dist/js/adminlte.min.js"></script>
81 <!-- Select2 -->
82 <script src="<?php echo base_url ?>plugins/select2/js/select2.full.min.js"></script>
83
84 <script>
85     $(document).ready(function() {
86         end_loader();
87         // Login Form Submit
88         $('#slogin-form').submit(function(e) {
89             e.preventDefault();
90             var _this = $(this);
91             $('#pop-msg').remove();
92             $('#password').removeClass("is-invalid");
93             var el = $("<div>");
94             el.addClass("alert pop-msg my-2");
95             el.hide();
96             start_loader();
97             $.ajax({
98                 url: base_url_ + "classes/Login.php?f=student_login",
99                 method: 'POST',
100                 data: _this.serialize(),
101                 dataType: 'json',
102                 error: function(err) {
103                     console.log(err);
104                     el.text("Ocurrió un error al guardar los datos");
105                     el.addClass("alert-danger");
106                     _this.prepend(el);
107                     el.show('slow');
108                     end_loader();

```

```

109     },
110     success: function(resp) {
111         if (resp.status === 'success') {
112             location.href = ".";
113         } else if (resp.msg) {
114             el.text(resp.msg);
115             el.addClass("alert-danger");
116             this.prepend(el);
117             el.show('show');
118         } else {
119             el.text("Ocurrió un error al guardar los datos");
120             el.addClass("alert-danger");
121             this.prepend(el);
122             el.show('show');
123         }
124     },
125     end_loader();
126     $('html, body').animate({scrollTop: 0}, 'fast');
127 });
128 });
129 });
130 </script>
131 </body>
132 </html>

```

Una vez que se realizó el formulario de ingreso y la página de inicio, se crearon las páginas que conforman el header, es decir las pestañas que redirigen a las diferentes secciones del repositorio.

Se realiza la creación de la página ACERCA DE, misma que se programa como ABOUT_US.html, esta página muestra la información del Instituto, su información de contacto, horarios y ubicación.

```

<!-- fragment -->
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>About Us</title>
</head>
<body>
<div class="heading-primary-wrap" style="line-height: 1; color: rgb(105, 105, 105);
font-family: &quot;Geeza Pro&quot;; font-size: 14px; text-align: center;"><h2 class=
"heading-primary elementor-heading-title" style="margin-right: 0px; margin-bottom: 0px;
margin-left: 0px; clear: both; letter-spacing: 0em; line-height: 1.4em; font-size:
40px; padding: 0px; display: inline-block;">Tu talento + nuestro método educativo =
un profesional de éxito</h2></div><div class="heading-description-wrap" style="
margin-top: 25px; color: rgb(105, 105, 105); font-family: &quot;Geeza Pro&quot;;
font-size: 14px; text-align: center;"><div class="heading-description" style="
display: inline-block; margin-bottom: 0px; font-size: 15px; line-height: 26px;"><p
style="margin-bottom: 0px;">29 años de experiencia sumado al talento humano de nuestro
equipo administrativo y docente, suman esfuerzos para darte la mejor calidad educativa,
para entregarte las herramientas necesarias para un futuro profesional prometedor.</p></div>
</div>
</body>
</html>

```

```

<?php
header('Content-Type: text/html; charset=utf8');
?>
<h3 class="text-center">Sobre Nosotros</h3>
<hr>
<div class="content">
<?php
$aboutUsFile = "about_us.html";

if (file_exists($aboutUsFile) && is_readable($aboutUsFile)) {
    include($aboutUsFile);
} else {
    echo "<p>No se pudo cargar la información sobre nosotros en este momento. Por
    favor, inténtelo de nuevo más tarde.</p>";
}
?>
</div>

```

```

about.php
1 <?php
2 $email = $_settings->info('email') ?? 'Correo no disponible';
3 $contact = $_settings->info('contact') ?? 'Número de contacto no disponible';
4 $address = $_settings->info('address') ?? 'Dirección no disponible';
5 $aboutUsContent = file_exists("about_us.html") ? file_get_contents("about_us.html") : 'Información
  sobre nosotros no disponible';
6
7
8 <div class="col-12">
9   <div class="row my-5">
10     <div class="col-md-5">
11       <div class="card card-outline card-navy rounded-0 shadow">
12         <div class="card-header">
13           <h4 class="card-title">Contacto</h4>
14         </div>
15         <div class="card-body rounded-0">
16           <dl>
17             <dt class="text-muted"><i class="fa fa-envelope"></i> Correo electrónico</dt>
18             <dd class="pr-4"><?= htmlspecialchars($email, ENT_QUOTES, 'UTF-8') ?></dd>
19             <dt class="text-muted"><i class="fa fa-phone"></i> Nº Contacto</dt>
20             <dd class="pr-4"><?= htmlspecialchars($contact, ENT_QUOTES, 'UTF-8') ?></dd>
21             <dt class="text-muted"><i class="fa fa-map-marked-alt"></i> Ubicación</dt>
22             <dd class="pr-4"><?= htmlspecialchars($address, ENT_QUOTES, 'UTF-8') ?></dd>
23           </dl>
24         </div>
25       </div>
26     </div>
27     <div class="col-md-7">
28       <div class="card rounded-0 card-outline card-navy shadow">
29         <div class="card-body rounded-0">
30           <h2 class="text-center">Sobre Nosotros</h2>
31           <center><hr class="bg-navy border-navy w-25 border-2"></center>
32           <div>
33             <?= $aboutUsContent ?>
34           </div>
35         </div>
36       </div>
37     </div>
38   </div>
39 </div>

```

Se creó también la página de proyectos publicados con el nombre PROJECTS.php, la misma que muestra todos los proyectos de titulación, permitiendo un filtro por facultad, carrera o tema.

```

projects.php
1 <?php
2 // Determinar la paginación
3 $limit = 10;
4 $page = isset($_GET['p']) ? $_GET['p'] : 1;
5 $offset = 10 * ($page - 1);
6 $paginate = " LIMIT {$limit} OFFSET {$offset}";
7
8 // Verificar si hay una búsqueda
9 $isSearch = isset($_GET['q']) ? "&q={$_GET['q']}" : "";
10 $search = "";
11
12 // Si hay búsqueda, construir la cláusula WHERE correspondiente
13 if (isset($_GET['q'])) {
14   $keyword = $conn->real_escape_string($_GET['q']);
15   $search = " AND (title LIKE '%{$keyword}%' OR abstract LIKE '%{$keyword}%' OR members LIKE '%{$keyword}%' OR curriculum_id IN (SELECT id FROM curriculum_list WHERE name LIKE '%{$keyword}%' OR description LIKE '%{$keyword}%' OR curriculum_id IN (SELECT id FROM curriculum_list WHERE department_id IN (SELECT id FROM department_list WHERE name LIKE '%{$keyword}%' OR description LIKE '%{$keyword}%'))) ";
16 }
17
18 // Consultar la cantidad total de registros
19 $countAllQuery = $conn->query("SELECT * FROM archive_list WHERE `status` = 1 {$search}");
20 $countAll = $countAllQuery->num_rows;
21 $pages = ceil($countAll / $limit);
22
23 // Consultar los archivos con paginación y búsqueda
24 $archivesQuery = $conn->query("SELECT * FROM archive_list WHERE `status` = 1 {$search} ORDER BY UNIX_TIMESTAMP(date_created) DESC {$paginate}");
25
26
27
28 <div class="content py-2">
29   <div class="col-12">
30     <div class="card">
31       <div class="card-body rounded-0">
32         <h2>Documentos</h2>
33         <hr class="">
34         <?php if (!empty($isSearch)) : ?>
35           <h3 class="text-center"><b>Resultados de búsqueda para "<?= $keyword ?>"</b></h3>

```

```

36     <?php endif ?>
37     <div class="list-group">
38         <?php while ($row = $archivesQuery->fetch_assoc()) : ?>
39             <?php
40             // Eliminar etiquetas HTML y decodificar entidades HTML en el resumen del
41             $row['abstract'] = strip_tags(html_entity_decode($row['abstract']));
42             ?>
43             <a href="./?page-view_archive&id=<?=$row['id'] ?>" class="
44             text-decoration-none text-dark list-group-item list-group-item-action">
45                 <div class="row">
46                     <div class="col-lg-4 col-md-5 col-sm-12 text-center">
47                         
49                     </div>
50                     <div class="col-lg-8 col-md-7 col-sm-12">
51                         <h3 class="text-blue"><b><?=$row['title'] ?></b></h3>
52                         <small class="text-muted">Por <b class="text-info"><?=$isset($
53                         student_arr[$row['student_id']]) ? $student_arr[$row['
54                         student_id']] : "N/A" ?></b></small>
55                         <p class="truncate-5"><?=$row['abstract'] ?></p>
56                     </div>
57                 </div>
58             </a>
59         <?php endwhile; ?>
60     </div>
61     <div class="card-footer clearfix rounded-0">
62         <div class="col-12">
63             <div class="row">
64                 <div class="col-md-6"><span class="text-muted">Mostrando <?=$archivesQuery
65                 ->num_rows ?></span></div>
66                 <div class="col-md-6">
67                     <ul class="pagination pagination-sm m=0 float-right">
68                         <li class="page-item"><a class="page-link" href="./?page-projects<?=$
69                         $isSearch ?>&p=<?=$page - 1 ?>" <?=$page == 1 ? 'disabled' :
70                         '' ?>></a></li>
71                         <?php for ($i = 1; $i <= $pages; $i++) : ?>
72                         <li class="page-item"><a class="page-link <?=$page == $i ? '
73                         active' : '' ?>" href="./?page-projects<?=$isSearch ?>&p=
74                         <?=$i ?>"><?=$i ?></a></li>
75                     </ul>
76                 </div>
77             </div>
78         </div>
79     </div>

```

Página para ver las tesis por facultad.

```

1  <?php
2  // Verificar si se ha proporcionado un ID
3  if (isset($_GET['id'])) {
4      $qry = $conn->query("SELECT * FROM curriculum_list WHERE `status` = 1 AND id = '{$_GET['id']}' "
5      );
6      if ($qry->num_rows > 0) {
7          foreach ($qry->fetch_assoc() as $k => $v) {
8              // Validar si la clave no es numérica para evitar valores no deseados en el arreglo
9              if (!is_numeric($k)) {
10                  $curriculum[$k] = $v;
11              }
12          }
13      } else {
14          echo "<script> alert('ID de materia desconocido'); location.replace('./') </script>";
15          exit; // Salir del script después de redireccionar
16      }
17      echo "<script> alert('Se requiere ID de la materia'); location.replace('./') </script>";
18      exit; // Salir del script después de redireccionar
19  }
20  ?>
21  <div class="content py-2">
22      <div class="col-12">
23          <div class="card">
24              <div class="card-body rounded-0">
25                  <h2>Archivos de <?=$isset($curriculum['name']) ? $curriculum['name'] : "" ?></h2>
26                  <p><small><?=$isset($curriculum['description']) ? $curriculum['description'] : "" ?>
27                  </small></p>
28                  <hr class="">
29                  <?php
30                  // Definir el ID y la paginación
31                  $id = isset($_GET['id']) ? $_GET['id'] : '';
32                  $limit = 10;
33                  $page = isset($_GET['p']) ? $_GET['p'] : 1;
34                  $offset = 10 * ($page - 1);
35                  $paginate = " LIMIT {$limit} OFFSET {$offset}";
36                  $wherecid = " AND curriculum_id = '{ $id }' ";
37
38                  // Consultar la lista de estudiantes
39                  $students = $conn->query("SELECT * FROM `student_list` WHERE id IN (SELECT
40                  student id FROM archive list WHERE `status` = 1 { $wherecid })");

```

```

39 $student_arr = array_column($students->fetch_all(MYSQLI_ASSOC), 'email', 'id');
40
41 // Contar la cantidad total de archivos
42 $count_all = $conn->query("SELECT * FROM archive_list WHERE `status` = 1 {$wherecid}
43   ")->num_rows;
44 $pages = ceil($count_all / $limit);
45
46 // Consultar los archivos con paginación
47 $archives = $conn->query("SELECT * FROM archive_list WHERE `status` = 1 {$wherecid}
48   ORDER BY UNIX_TIMESTAMP(date_created) DESC {$paginate}");
49
50 <div class="list-group">
51   <?php while ($row = $archives->fetch_assoc()) : ?>
52     <?php
53     // Eliminar etiquetas HTML y decodificar entidades HTML en el resumen del
54     // archivo
55     $row['abstract'] = strip_tags(html_entity_decode($row['abstract']));
56     <a href="./?page=view_archive&id=<?=$row['id'] ?>" class="
57       text-decoration-none text-dark list-group-item list-group-item-action">
58       <div class="row">
59         <div class="col-lg-4 col-md-5 col-sm-12 text-center">
60           
62         </div>
63         <div class="col-lg-8 col-md-7 col-sm-12">
64           <h3 class="text-blue"><b><?=$row['title'] ?></b></h3>
65           <small class="text-muted">Por <b class="text-info"><?=$isset($
66             student_arr[$row['student_id']]) ? $student_arr[$row['
67             student_id']] : "N/A" ?></b></small>
68           <p class="truncate-5"><?=$row['abstract'] ?></p>
69         </div>
70       </div>
71     </a>
72   <?php endwhile; ?>
73 </div>
74 <div class="card-footer clearfix rounded-0">
75   <div class="col-12">
76     <div class="row">
77       <div class="col-md-6"><span class="text-muted">Mostrando: <?=$archives->
78
79       <div class="col-md-6">
80         <ul class="pagination pagination-sm m-0 float-right">
81           <li class="page-item"><a class="page-link" href="./?
82             page=projects_per_curriculum&id=<?=$id ?>&p=<?=$page - 1 ?>" <?=$
83             page == 1 ? 'disabled' : '' ?>></a></li>
84           <?php for ($i = 1; $i <= $pages; $i++) : ?>
85             <li class="page-item"><a class="page-link" <?=$page == $i ? '
86               active' : '' ?>" href="./?page=projects_per_curriculum&id=
87               <?=$id ?>&p=<?=$i ?>"><?=$i ?></a></li>
88           <?php endfor; ?>
89           <li class="page-item"><a class="page-link" href="./?
90             page=projects_per_curriculum&id=<?=$id ?>&p=<?=$page + 1 ?>" <?=$
91             page == $pages || $pages <= 1 ? 'disabled' : '' ?>></a></li>
92         </ul>
93       </div>
94     </div>
95   </div>
96 </div>
97 </div>
98 </div>
99 </div>

```

Página para ver las proyectos de titulación por carrera.

```

1 <?php
2 // Verificar si se ha proporcionado un ID de departamento
3 if (isset($_GET['id'])) {
4     $qry = $conn->query("SELECT * FROM department_list WHERE `status` = 1 AND id = '{$_GET['id']}' "
5     );
6     if ($qry->num_rows > 0) {
7         foreach ($qry->fetch_assoc() as $k => $v) {
8             // Validar si la clave no es numérica para evitar valores no deseados en el arreglo
9             if (!is_numeric($k)) {
10                 $department[$k] = $v;
11             }
12         }
13     } else {
14         echo "<script> alert('ID de departamento desconocido'); location.replace('./') </script>";
15         exit; // Salir del script después de redireccionar
16     }
17 } else {
18     echo "<script> alert('Se requiere ID del departamento'); location.replace('./') </script>";
19     exit; // Salir del script después de redireccionar
20 }
21
22 <div class="content py-2">
23     <div class="col-12">
24         <div class="card">
25             <div class="card-body rounded-0">
26                 <h2>Archivos de <?= isset($department['name']) ? $department['name'] : "" ?> </h2>
27                 <p><small><?= isset($department['description']) ? $department['description'] : "" ?>
28                 </small></p>
29                 <hr class="">
30                 <?php
31                 // Definir el ID y la paginación
32                 $id = isset($_GET['id']) ? $_GET['id'] : '';
33                 $limit = 10;
34                 $page = isset($_GET['p']) ? $_GET['p'] : 1;
35                 $offset = 10 * ($page - 1);
36                 $paginate = " LIMIT {$limit} OFFSET {$offset}";
37                 $wheredid = " WHERE department_id = '{$id}' ";
38
39                 // Consultar la lista de estudiantes
40                 $students = $conn->query("SELECT * FROM `student_list` WHERE id IN (SELECT
41                     student_id FROM archive_list WHERE `status` = 1 AND curriculum_id IN (SELECT id
42                     student_id FROM archive_list WHERE `status` = 1 AND curriculum_id IN (SELECT id
43                     FROM curriculum_list {$wheredid} ))");
44                 $student_arr = array_column($students->fetch_all(MYSQLI_ASSOC), 'email', 'id');
45
46                 // Contar la cantidad total de archivos
47                 $count_all = $conn->query("SELECT * FROM archive_list WHERE `status` = 1 AND
48                     curriculum_id IN (SELECT id FROM curriculum_list {$wheredid} )->num_rows;
49                 $pages = ceil($count_all / $limit);
50
51                 // Consultar los archivos con paginación
52                 $archives = $conn->query("SELECT * FROM archive_list WHERE `status` = 1 AND
53                     curriculum_id IN (SELECT id FROM curriculum_list {$wheredid} ) ORDER BY
54                     UNIX_TIMESTAMP(date_created) DESC {$paginate}");
55
56                 <?php while ($row = $archives->fetch_assoc()) : ?>
57                     <?php
58                     // Eliminar etiquetas HTML y decodificar entidades HTML en el resumen del
59                     // archivo
60                     $row['abstract'] = strip_tags(html_entity_decode($row['abstract']));
61                     <?php
62                     <a href="./?page=view_archive&id=<?= $row['id'] ?>" class="
63                     text-decoration-none text-dark list-group-item list-group-item-action">
64                         <div class="row">
65                             <div class="col-lg-4 col-md-5 col-sm-12 text-center">
66                                 
68                             </div>
69                             <div class="col-lg-8 col-md-7 col-sm-12">
70                                 <h3 class="text-blue"><b><?= $row['title'] ?></b></h3>
71                                 <small class="text-muted">Por <b class="text-info"><?= isset($
72                                 student_arr[$row['student_id']] ? $student_arr[$row['
73                                 student_id']] : "N/A" ?></b></small>
74                                 <p class="truncate-5"><?= $row['abstract'] ?></p>
75                             </div>
76                         </div>
77                     </a>
78                 <?php endwhile; ?>
79
80 </div>
81 <div class="card-footer clearfix rounded-0">

```

```

70         <div class="col-12">
71             <div class="row">
72                 <div class="col-md-6"><span class="text-muted">Mostrando: <?= $archives->
num_rows ?></span></div>
73                 <div class="col-md-6">
74                     <ul class="pagination pagination-sm m=0 float-right">
75                         <li class="page-item"><a class="page-link" href="."/?.
page=projects_per_department&id=<?= $id ?>&p=<?= $page - 1 ?>" <?= $
page == 1 ? 'disabled' : '' ?></a></li>
76                         <?php for ($i = 1; $i <= $pages; $i++) : ?>
77                             <li class="page-item"><a class="page-link" <?= $page == $i ? '
active' : '' ?>" href="."/?.page=projects_per_department&id=
<?= $id ?>&p=<?= $i ?>"><?= $i ?></a></li>
78                         <?php endfor; ?>
79                         <li class="page-item"><a class="pagelink" href="."/?.
page=projects_per_department&id=<?= $id ?>&p=<?= $page + 1 ?>" <?= $
page == $pages || $pages <= 1 ? 'disabled' : '' ?>></a></li>
80                     </ul>
81                 </div>
82             </div>
83         </div>
84     </div>
85 </div>
86 </div>
87 </div>
88 </div>

```

Una vez realizada la programación de las páginas que componen los hipervínculos del header, se procedió con la creación de los hipervínculos de los usuarios; esto es, el botón de ingreso de estudiantes y administrativos y el formulario de registro de estudiantes.

```

register.php
1 <?php
2 // Incluir el archivo de configuración
3 require_once('./config.php');
4
5 // Incluir el encabezado
6 require_once('inc/header.php');
7 ?>
8
9 <body class="hold-transition">
10
11 <script>
12     start_loader();
13 </script>
14
15 <style>
16     html,
17     body {
18         height: calc(100%) !important;
19         width: calc(100%) !important;
20     }
21
22     body {
23         background-image: url("<?= validate_image($_settings->info('cover')) ?>");
24         background-size: cover;
25         background-repeat: no-repeat;
26     }
27
28     .login-title {
29         text-shadow: 2px 2px black;
30     }
31
32     #login {
33         direction: rtl;
34     }
35
36     #login* {
37         direction: ltr;
38     }
39
40     #logo-img {
41         height: 150px;

```

```

40     #logo-img {
41         height: 150px;
42         width: 150px;
43         object-fit: scale-down;
44         object-position: center center;
45         border-radius: 100%;
46     }
47 </style>
48
49 <div class="h-100 d-flex align-items-center w-100" id="login">
50     <div class="col-7 h-100 d-flex align-items-center justify-content-center">
51         <div class="w-100">
52             <center></center>
53             <h1 class="text-center py-5 login-title"><b><? $settings->info('name') ?> - Admin</b></h1>
54         </div>
55     </div>
56     <div class="col-5 h-100 bg-gradient bg-primary">
57         <div class="w-100 d-flex justify-content-center align-items-center h-100 text-navy">
58             <div class="card card-outline card-primary rounded-0 shadow col-lg-10 col-md-10 col-sm-5">
59                 <div class="card-header">
60                     <h5 class="card-title text-center text-dark"><b>Registro</b></h5>
61                 </div>
62                 <div class="card-body">
63                     <form action="" id="registration-form">
64                         <input type="hidden" name="id">
65                         <div class="row">
66                             <div class="col-lg-6">
67                                 <div class="form-group">
68                                     <input type="text" name="firstname" id="firstname" autofocus placeholder="Nombre" class="form-control form-control-border" required>
69                                 </div>
70                             </div>
71                             <div class="col-lg-6">
72                                 <div class="form-group">
73                                     <input type="text" name="middlename" id="middlename" placeholder="Segundo Nombre (opcional)" class="form-control form-control-border">
74                                 </div>
75                             </div>
76                         </div>
77                     </div>
78                     <div class="row">
79                         <div class="col-lg-6">
80                             <div class="form-group">
81                                 <input type="text" name="lastname" id="lastname" placeholder="Apellido" class="form-control form-control-border" required>
82                             </div>
83                         </div>
84                         <div class="row">
85                             <div class="form-group col-auto">
86                                 <div class="custom-control custom-radio">
87                                     <input class="custom-control-input" type="radio" id="genderMale" name="gender" value="Male" required checked>
88                                     <label for="genderMale" class="custom-control-label">Masculino</label>
89                                 </div>
90                             </div>
91                             <div class="form-group col-auto">
92                                 <div class="custom-control custom-radio">
93                                     <input class="custom-control-input" type="radio" id="genderFemale" name="gender" value="Female">
94                                     <label for="genderFemale" class="custom-control-label">Femenino</label>
95                                 </div>
96                             </div>
97                         </div>
98                     <div class="row">
99                         <div class="col-lg-12">
100                             <div class="form-group">
101                                 <span class="text-navy"><small>Facultad</small></span>
102                                 <select name="department_id" id="department_id" class="form-control form-control-border select2" data-placeholder="Seleccionar el departamento" required>
103                                     <option value="" disabled></option>
104                                 <?php
105                                     // Consultar la lista de facultades
106                                     $department = $conn->query("SELECT * FROM `department_list` where status = 1 order by `name` asc");
107                                     while($row = $department->fetch_assoc()):
108                                     ?>

```

```

109         <option value="<?= $row['id'] ?>"><?= ucwords($row['name']) ?></option>
110     <?php endwhile; ?>
111 </select>
112 </div>
113 </div>
114 </div>
115 <div class="row">
116 <div class="col-lg-12">
117 <div class="">
118 <div class="form-group">
119 <span class="text-navy"><small>Carrera</small></span>
120 <select name="curriculum_id" id="curriculum_id" class="form-control
121 form-control-border select2" data-placeholder="Seleccionar materia" required>
122 <option value="" disabled selected>Primero seleccionar facultad</option>
123 <?php
124 // Consultar la lista de planes de estudio
125 $curriculum = $conn->query("SELECT * FROM `curriculum_list` where status = 1
126 order by `name` asc");
127 $cur_arr = [];
128 while($row = $curriculum->fetch_assoc()){
129     $row['name'] = ucwords($row['name']);
130     $cur_arr[$row['department_id']][] = $row;
131 }
132 </select>
133 </div>
134 </div>
135 </div>
136 <div class="row">
137 <div class="col-lg-12">
138 <div class="form-group">
139 <input type="email" name="email" id="email" placeholder="Correo" class="
140 form-control form-control-border" required>
141 </div>
142 </div>
143 <div class="row">
144 <div class="col-lg-12">
145 <div class="form-group">
146 <input type="password" name="password" id="password" placeholder="Contraseña"
147 class="form-control form-control-border" required>
148 </div>
149 </div>
150 <div class="row">
151 <div class="col-lg-12">
152 <div class="form-group">
153 <input type="password" id="cpassword" placeholder="Confirmar contraseña" class=
154 "form-control form-control-border" required>
155 </div>
156 </div>
157 <div class="row">
158 <div class="col-lg-12">
159 <div class="form-group text-right">
160 <button class="btn btn-default bg-navy btn-flat"> Registrarse</button>
161 </div>
162 </div>
163 </div>
164 </form>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170
171 <!-- jQuery -->
172 <script src="plugins/jquery/jquery.min.js"></script>
173 <!-- Bootstrap 4 -->
174 <script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
175 <!-- AdminLTE App -->
176 <script src="dist/js/adminlte.min.js"></script>
177 <!-- Select2 -->
178 <script src="<?php echo base_url ?>plugins/select2/js/select2.full.min.js"></script>
179
180 <script>
181     var cur_arr = <?=> json_encode($cur_arr) ?>;
182     $(document).ready(function () {
183         end_loader();
184         $('#select2').select2({

```

```

185         width: "100%"
186     });
187     $('#department_id').change(function () {
188         var did = $(this).val();
189         $('#curriculum_id').html("");
190         if (!!cur_arr[did]) {
191             Object.keys(cur_arr[did]).map(k => {
192                 var opt = $("<option>");
193                 opt.attr('value', cur_arr[did][k].id);
194                 opt.text(cur_arr[did][k].name);
195                 $('#curriculum_id').append(opt);
196             });
197         }
198         $('#curriculum_id').trigger("change");
199     });
200
201     // Registro del formulario de registro
202     $('#registration-form').submit(function (e) {
203         e.preventDefault();
204         var _this = $(this);
205         $('#pop-msg').remove();
206         $('#password, #cpassword').removeClass("is-invalid");
207         var el = $("<div>");
208         el.addClass("alert pop-msg my-2");
209         el.hide();
210         if ($('#password').val() != $('#cpassword').val()) {
211             el.addClass("alert-danger");
212             el.text("La contraseña no coincide.");
213             $('#password, #cpassword').addClass("is-invalid");
214             $('#cpassword').after(el);
215             el.show('slow');
216             return false;
217         }
218         start_loader();
219         $.ajax({
220             url: _base_url + "classes/Users.php?f=save_student",
221             method: 'POST',
222             data: _this.serialize(),
223             dataType: 'json',
224             success: function (resp) {
225                 error: function (err) {
226                     console.log(err);
227                     el.text("Ocurrió un error al guardar los datos");
228                     el.addClass("alert-danger");
229                     _this.prepend(el);
230                     el.show('slow');
231                     end_loader();
232                 },
233                 success: function (resp) {
234                     if (resp.status == 'success') {
235                         location.href = "/login.php";
236                     } else if (!!resp.msg) {
237                         el.text(resp.msg);
238                         el.addClass("alert-danger");
239                         _this.prepend(el);
240                         el.show('show');
241                     } else {
242                         el.text("Ocurrió un error al guardar los datos");
243                         el.addClass("alert-danger");
244                         _this.prepend(el);
245                         el.show('show');
246                     }
247                     end_loader();
248                     $('html, body').animate({ scrollTop: 0 }, 'fast');
249                 }
250             });
251         });
252     </script>
253 </body>
254 </html>

```

Debido a que el presente repositorio fue creado con el modelo MVC, el mismo se ha dividido en diferentes carpetas a fin de que se obtenga lo que se necesita en específico de cada parte del código; en tal razón, la parte del administrador se ha incluido en una sola carpeta.

Se crea el formulario de LOGIN.php del administrador, mismo que debe tener todos, o casi todos los permisos dentro del repositorio.

```

login.php
1 <?php require_once('../config.php'); ?>
2 <!DOCTYPE html>
3 <html lang="en">
4 <?php require_once('inc/header.php'); ?>
5 <body class="hold-transition">
6 <script>
7     start_loader(); // Inicia el loader cuando se carga la página
8 </script>
9 <style>
10     html, body{
11         height: 100%; /* Se establece la altura del cuerpo y el html al 100% */
12         width: 100%; /* Se establece el ancho del cuerpo y el html al 100% */
13     }
14     body{
15         background-image: url("php echo htmlspecialchars(validate_image($_settings-&gt;info('cover')),
16             ENT_QUOTES, 'UTF-8'); ?&gt;"); /* Se muestra la imagen de fondo */
17         background-size: cover; /* Se ajusta el tamaño de la imagen de fondo */
18         background-repeat: no-repeat; /* Se establece la repetición de la imagen de fondo */
19     }
20     .login-title{
21         text-shadow: 2px 2px black; /* Se agrega un efecto de sombra al texto */
22     }
23     #login{
24         flex-direction: column !important; /* Se establece la dirección de la caja flexible a columna */
25     }
26     #logo-img{
27         height: 150px;
28         width: 150px;
29         object-fit: scale-down;
30         object-position: center center;
31         border-radius: 100%; /* Se establece un borde circular para la imagen */
32     }
33     #login .col-7, #login .col-5{
34         width: 100% !important;
35         max-width: unset !important; /* Se establece el ancho de las columnas */
36     }
37 &lt;/style&gt;
38 &lt;div class="h-100 d-flex align-items-center w-100" id="login"&gt;
39     &lt;div class="col-7 h-100 d-flex align-items-center justify-content-center"&gt;
40         &lt;div class="w-100"&gt;
</pre

```

```

40         'UTF-8'); ?>" alt="" id="logo-img"></center> <!-- Se muestra el logotipo -->
41         <h1 class="text-center py-5 login-title"><b><?php echo htmlspecialchars($_settings->info('
42             name'), ENT_QUOTES, 'UTF-8'); ?></b></h1> <!-- Se muestra el nombre -->
43     </div>
44     <div class="col-5 h-100 bg-gradient">
45         <div class="d-flex w-100 h-100 justify-content-center align-items-center">
46             <div class="card col-sm-12 col-md-6 col-lg-3 card-outline card-primary">
47                 <div class="card-header">
48                     <h4 class="text-purple text-center"><b>Iniciar Sesión</b></h4>
49                 </div>
50                 <div class="card-body">
51                     <form id="login_frm" action="" method="post">
52                         <div class="input-group mb-3">
53                             <input type="text" class="form-control" autofocus name="username" placeholder="
54                                 Nombre de usuario">
55                             <div class="input-group-append">
56                                 <div class="input-group-text">
57                                     <span class="fas fa-user"></span>
58                                 </div>
59                             </div>
60                         </div>
61                         <div class="input-group mb-3">
62                             <input type="password" class="form-control" name="password" placeholder="Contraseña">
63                             <div class="input-group-append">
64                                 <div class="input-group-text">
65                                     <span class="fas fa-lock"></span>
66                                 </div>
67                             </div>
68                         </div>
69                         <div class="row">
70                             <div class="col-8">
71                                 <a href="<?php echo htmlspecialchars(base_url, ENT_QUOTES, 'UTF-8'); ?>">ir al
72                                 inicio</a> <!-- Se muestra el enlace al inicio -->
73                             </div>
74                             <div class="col-4">
75                                 <button type="submit" class="btn btn-primary btn-block">Ingresa</button> <!-- Se
76                                 muestra el botón de ingreso -->
77                             </div>
78                         </div>
79                     </form>
80                 </div>
81             </div>
82         </div>
83     </div>
84 </div>

```

```

75         muestra el botón de ingreso -->
76     </div>
77     <!-- /.col -->
78 </div>
79 </form>
80 </div>
81 </div>
82 </div>
83 </div>
84
85 <!-- jQuery -->
86 <script src="plugins/jquery/jquery.min.js"></script>
87 <!-- Bootstrap 4 -->
88 <script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
89 <!-- AdminLTE App -->
90 <script src="dist/js/adminlte.min.js"></script>
91
92 <script>
93     $(document).ready(function(){
94         end_loader(); // Finaliza el loader cuando el documento está listo
95     });
96 </script>
97 </body>
98 </html>
99

```

Se procede con la creación de la página de inicio del administrador, que como se evidencia en capturas de pantalla adjuntas al tema repositorio en imágenes es diferente a la página de inicio del repositorio.

```

index.php
1 <?php require_once('..../config.php'); ?>
2 <!DOCTYPE html>
3 <html lang="en">
4 <?php require_once('inc/header.php'); ?>
5 <body class="sidebar-mini layout-fixed control-sidebar-slide-open layout-navbar-fixed" style="height
  : auto;">
6 <div class="wrapper">
7 <?php require_once('inc/topBarNav.php'); ?>
8 <?php require_once('inc/navigation.php'); ?>
9 <?php if($ _settings->chk_flashdata('success')): ?>
10 <script>
11     alert_toast("<?php echo htmlspecialchars($_settings->flashdata('success'),
    ENT_QUOTES, 'UTF-8'); ?>","success");
12 </script>
13 <?php endif; ?>
14 <?php
15     $page = isset($_GET['page']) ? $_GET['page'] : 'home';
16     $page_path = $page . '.php';
17     if (!file_exists($page_path) && !is_dir($page)) {
18         include '404.html';
19     } else {
20         include (is_dir($page) ? $page . '/index.php' : $page . '.php');
21     }
22 >
23 <?php require_once('inc/footer.php'); ?>
24 </div>
25 <div class="modal fade" id="confirm_modal" role="dialog">
26 <div class="modal-dialog modal-md modal-dialog-centered" role="document">
27 <div class="modal-content">
28 <div class="modal-header">
29 <h5 class="modal-title">Confirmación</h5>
30 </div>
31 <div class="modal-body">
32 <div id="delete_content"></div>
33 </div>
34 <div class="modal-footer">
35 <button type="button" class="btn btn-primary btn-flat" id='confirm' onclick="">
    Continuar</button>
36 <button type="button" class="btn btn-secondary btn-flat" data-dismiss="modal">
    Cerrar</button>
37 </div>

```

```

38     </div>
39   </div>
40 </div>
41 <div class="modal fade" id="uni_modal" role='dialog'>
42   <div class="modal-dialog modal-md modal-dialog-centered" role="document">
43     <div class="modal-content">
44       <div class="modal-header">
45         <h5 class="modal-title"></h5>
46       </div>
47       <div class="modal-body">
48       </div>
49       <div class="modal-footer">
50         <button type="button" class="btn btn-primary btn-flat" id='submit' onclick="$('
51           #uni_modal form').submit()'>Guardar</button>
52         <button type="button" class="btn btn-secondary btn-flat" data-dismiss="modal">
53           Cancelar</button>
54       </div>
55     </div>
56   </div>
57 </div>
58 <div class="modal fade" id="uni_modal_right" role='dialog'>
59   <div class="modal-dialog modal-full-height modal-md" role="document">
60     <div class="modal-content">
61       <div class="modal-header">
62         <h5 class="modal-title"></h5>
63         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
64           <span class="fa fa-arrow-right"></span>
65         </button>
66       </div>
67       <div class="modal-body">
68       </div>
69     </div>
70   </div>
71 </div>
72 <div class="modal fade" id="viewer_modal" role='dialog'>
73   <div class="modal-dialog modal-md" role="document">
74     <div class="modal-content">
75       <button type="button" class="btn-close" data-dismiss="modal"><span class="fa
76         fa-times"></span></button>
77       <img src="" alt="">
78     </div>
79   </div>
80 </div>
81
60     <h5 class="modal-title"></h5>
61     <button type="button" class="close" data-dismiss="modal" aria-label="Close">
62       <span class="fa fa-arrow-right"></span>
63     </button>
64   </div>
65   <div class="modal-body">
66   </div>
67 </div>
68 </div>
69 <div class="modal fade" id="viewer_modal" role='dialog'>
70   <div class="modal-dialog modal-md" role="document">
71     <div class="modal-content">
72       <button type="button" class="btn-close" data-dismiss="modal"><span class="fa
73         fa-times"></span></button>
74       <img src="" alt="">
75     </div>
76   </div>
77 </div>
78 </body>
79 </html>
80
81

```

Se creó la página de home del administrador donde se visualizan los pendientes y puede realizar los cambios o subir los archivos necesarios.

```

home.php
1 <h1><?php echo htmlspecialchars($_settings->info('name'), ENT_QUOTES, 'UTF-8'); ?></h1>
2 <hr class="border-info">
3 <div class="row">
4 <?php
5 // Lista de cajas de información
6 $info_boxes = [
7
8     'icon_class' => 'fas fa-th-list',
9     'query' => "SELECT * FROM `department_list` WHERE status = 1",
10    'text' => 'Lista de facultades'
11 ],
12
13    'icon_class' => 'fas fa-scroll',
14    'query' => "SELECT * FROM `curriculum_list` WHERE `status` = 1",
15    'text' => 'Lista de Carreras'
16 ],
17
18    'icon_class' => 'fas fa-users',
19    'query' => "SELECT * FROM `student_list` WHERE `status` = 1",
20    'text' => 'Estudiantes verificados'
21 ],
22
23    'icon_class' => 'fas fa-users',
24    'query' => "SELECT * FROM `student_list` WHERE `status` = 0",
25    'text' => 'Estudiantes no verificados'
26 ],
27
28    'icon_class' => 'fas fa-archive',
29    'query' => "SELECT * FROM `archive_list` WHERE `status` = 1",
30    'text' => 'Archivos verificados'
31 ],
32
33    'icon_class' => 'fas fa-archive',
34    'query' => "SELECT * FROM `archive_list` WHERE `status` = 0",
35    'text' => 'Archivos no verificados'
36 ],
37 ];
38
39 foreach ($info_boxes as $box) :
40     $result = $conn->query($box['query']);
41     ?>

```

```

37 ];
38
39 foreach ($info_boxes as $box) :
40     $result = $conn->query($box['query']);
41     ?>
42     <div class="col-12 col-sm-12 col-md-6 col-lg-3">
43         <div class="info-box bg-light shadow">
44             <span class="info-box-icon bg-info elevation-1"><i class="<?php echo $box['
45                 icon_class']; ?>"></i></span>
46
47             <div class="info-box-content">
48                 <span class="info-box-text"><?php echo htmlspecialchars($box['text'], ENT_QUOTES
49                     , 'UTF-8'); ?></span>
50                 <span class="info-box-number text-right"><?php echo $result->num_rows; ?></span>
51             </div>
52             <!-- /.info-box-content -->
53         </div>
54         <!-- /.info-box -->
55     </div>
56 <?php endforeach; ?>

```

Dentro de la funcionalidad del administrador está la de manejar los archivos subidos, los usuarios creados, dar el mantenimiento a las carreras y facultades, por lo que se crearon las diversas pantallas para realizar estas actividades.

A fin de realizar las diversas actividades con los proyectos de tesis publicados o por publicar, se crearon las pantallas de índice, estado y detalles.

```

1 <?php
2 require_once('../config.php');
3
4 // Obtener datos de archivos y materias
5 $qry = $conn->query("SELECT * FROM archive_list ORDER BY `year` DESC, `title` DESC ");
6 $curriculum = $conn->query("SELECT * FROM curriculum_list WHERE id IN (SELECT curriculum_id FROM `
7     archive_list`)");
8
9 // Crear un array asociativo de materias con el ID como clave y el nombre como valor
10 $cur_arr = array_column($curriculum->fetch_all(MYSQLI_ASSOC), 'name', 'id');
11
12 <style>
13     .img-avatar {
14         width: 45px;
15         height: 45px;
16         object-fit: cover;
17         object-position: center center;
18         border-radius: 100%;
19     }
20 </style>
21 <div class="card card-outline card-primary">
22     <div class="card-header">
23         <h3 class="card-title">Lista de archivos</h3>
24     </div>
25     <div class="card-body">
26         <div class="container-fluid">
27             <div class="container-fluid">
28                 <table class="table table-hover table-striped">
29                     <colgroup>
30                         <col width="5%">
31                         <col width="15%">
32                         <col width="15%">
33                         <col width="20%">
34                         <col width="20%">
35                         <col width="10%">
36                         <col width="10%">
37                     </colgroup>
38                     <thead>
39                         <tr>
40                             <th>#</th>
41                             <th>Fecha de creación</th>
42                             <th>Código de archivo</th>
43                             <th>Título del proyecto</th>
44                             <th>Materia</th>
45                             <th>Estado</th>
46                             <th>Acción</th>
47                         </tr>
48                     </thead>
49                     <tbody>
50                         <?php
51                         $i = 1;
52                         while($row = $qry->fetch_assoc()):
53                         <?>
54                             <tr>
55                                 <td class="text-center"><?php echo $i++; ?></td>
56                                 <td class=""><?php echo date("Y-m-d H:i", strtotime($row['
57                                     date_created'])) ?></td>
58                                 <td><?php echo htmlspecialchars($row['archive_code']) ?></td>
59                                 <td><?php echo ucwords($row['title']) ?></td>
60                                 <td><?php echo $cur_arr[$row['curriculum_id']] ?></td>
61                                 <td class="text-center">
62                                     <?php
63                                     switch($row['status']){
64                                         case '1':
65                                             echo "<span class='badge badge-success
66                                                 badge-pill'>Publicado</span>";
67                                             break;
68                                         case '0':
69                                             echo "<span class='badge badge-secondary
70                                                 badge-pill'>No Publicado</span>";
71                                             break;
72                                     }
73                                 <?>
74                                 <td align="center">
75                                     <button type="button" class="btn btn-flat btn-default btn-sm
76                                         dropdown-toggle dropdown-icon" data-toggle="dropdown">
77                                         Acción
78                                         <span class="sr-only">Toggle Dropdown</span>
79                                     </button>

```

```

76         <div class="dropdown-menu" role="menu">
77             <a class="dropdown-item" href="<?php echo $row['id'] ?>" target="
              page-view_archive&id=<?php echo $row['id'] ?>">Ver</a>
78             <div class="dropdown-divider"></div>
79             <a class="dropdown-item update_status" href="
              javascript:void(0)" data-id="<?php echo $row['id'] ?>"
              data-status="<?php echo $row['status'] ?>"><span class="fa
              fa-check text-dark"></span> Actualizar estado</a>
80             <div class="dropdown-divider"></div>
81             <a class="dropdown-item delete_data" href="javascript:void(
              0)" data-id="<?php echo $row['id'] ?>"><span class="fa
              fa-trash text-danger"></span> Eliminar</a>
82         </div>
83     </td>
84 </tr>
85 <?php endwhile; ?>
86 </tbody>
87 </table>
88 </div>
89 </div>
90 </div>
91 </div>
92 <script>
93 $(document).ready(function(){
94     $('.delete_data').click(function(){
95         _conf("¿Estás seguro de eliminar este proyecto de forma permanente?", "delete_archive", [
96             $(this).attr('data-id')]
97     )
98     $('.update_status').click(function(){
99         uni_modal("Actualizar detalles", "archives/update_status.php?id="+$(this).attr('data-id')
100             +"&status="+$(this).attr('data-status'))
101     })
102     $('.table td, .table th').addClass('py-1 px-2 align-middle')
103     $('.table').dataTable({
104         columnDefs: [
105             { orderable: false, targets: 5 }
106         ],
107     });

```

```

102         columnDefs: [
103             { orderable: false, targets: 5 }
104         ],
105     });
106 })
107 function delete_archive($id){
108     start_loader();
109     $.ajax({
110         url: base_url+"classes/Master.php?f=delete_archive",
111         method: "POST",
112         data: {id: $id},
113         dataType: "json",
114         error: err => {
115             console.log(err)
116             alert_toast("Ocurrió un error.", 'error');
117             end_loader();
118         },
119         success: function(resp){
120             if(typeof resp == 'object' && resp.status == 'success'){
121                 location.reload();
122             }else{
123                 alert_toast("Ocurrió un error.", 'error');
124                 end_loader();
125             }
126         }
127     })
128 }
129 </script>
130

```

```

1 <div class="container-fluid">
2 <form action="" id="update_status_form">
3 <input type="hidden" name="id" value="<?=$ htmlspecialchars($ _GET['id'] ?? '') ?>">
4 <div class="form-group">
5 <label for="status" class="control-label text-navy">Estado</label>
6 <select name="status" id="status" class="form-control form-control-border" required>
7 <option value="0" <?=$ ($ _GET['status'] ?? '') == '0' ? 'selected' : '' ?>>Despublicar</option>
8 <option value="1" <?=$ ($ _GET['status'] ?? '') == '1' ? 'selected' : '' ?>>Publicar</option>
9 </select>
10 </div>
11 </form>
12 </div>
13 <script>
14 $(function(){
15 $('#update_status_form').submit(function(e){
16 e.preventDefault();
17 start_loader();
18 var el = $('<div>');
19 el.addClass("pop-msg alert");
20 el.hide();
21 $.ajax({
22 url: _base_url_ + "classes/Master.php?f=update_status_",
23 method: "POST",
24 data: $(this).serialize(),
25 dataType: "json",
26 error: function(err){
27 console.log(err);
28 alert_toast("Ocurrió un error al guardar los datos.", "error");
29 end_loader();
30 },
31 success: function(resp){
32 console.log(resp);
33 if(resp.status === 'success'){
34 location.reload();
35 } else if(resp.msg){
36 el.addClass("alert-danger");
37 el.text(resp.msg);
38 $('#update_status_form').prepend(el);
39 } else {
40 } else if(resp.msg){
41 el.addClass("alert-danger");
42 el.text(resp.msg);
43 $('#update_status_form').prepend(el);
44 } else {
45 el.addClass("alert-danger");
46 el.text("Ocurrió un error debido a una razón desconocida.");
47 $('#update_status_form').prepend(el);
48 }
49 el.show('slow');
50 end_loader();
51 }
52 }
53 });
54 });
55 </script>

```

```

1 <?php
2 // Verificar si se proporciona un ID en la URL y recuperar los detalles del alumno
3 if (isset($_GET['id'])) {
4     $qry = $conn->query("SELECT e.*,CONCAT(lastname, ' ', firstname,' ',middlename) as fullname,p.
5         name,p.description,p.training_duration from `enrollee_list` e inner join package_list p on e
6         .package_id = p.id where e.id = '{$_GET['id']}'");
7     if ($qry->num_rows > 0) {
8         $res = $qry->fetch_assoc();
9         // Asignar los valores a las variables correspondientes
10        foreach ($res as $k => $v) {
11            if (!is_numeric($k)) {
12                $$k = $v;
13            }
14        }
15    }
16    // Calcular el saldo del alumno
17    $balance = isset($cost) ? $cost : 0;
18    ?>
19
20 <div class="container-fluid">
21     <div class="card card-outline card-navy rounded-0 shadow">
22         <div class="card-header rounded-0">
23             <h5 class="card-title">Detalles del alumno</h5>
24         </div>
25         <div class="card-body rounded-0">
26             <div id="outprint">
27                 <div class="row">
28                     <div class="col-md-6">
29                         <div>
30                             <!-- Detalles personales del alumno -->
31                         </div>
32                     </div>
33                     <div class="col-md-6">
34                         <div>
35                             <!-- Detalles del paquete de entrenamiento -->
36                         </div>
37                     </div>
38                 </div>
39                 <?php if (in_array($status, range(1, 3))): ?>
40                     <!-- Historial de pagos del alumno -->
41                     <?php endif; ?>
42                 </div>
43             </div>
44             <div class="card-footer rounded-0 text-center">
45                 <!-- Botones de acción -->
46             </div>
47         </div>
48     </div>
49
50 <script>
51     $(function(){
52         // Evento al hacer clic en el botón "Update Status"
53         $('#update_status').click(function(){
54             uni_modal("Actualizar detalles","enrollees/update_status.php?id=<?=$id ?>&status=<?=$
55                 status ?>");
56         });
57         // Evento al hacer clic en el botón "Payment"
58         $('#payment').click(function(){
59             uni_modal("Nuevo pago","enrollees/payment.php?id=<?=$id ?>&balance=<?=$balance ?>");
60         });
61         // Evento al hacer clic en el botón "Delete Payment"
62         $('#delete_payment').click(function(){
63             _conf("¿Seguro que deseas eliminar este pago permanentemente?","delete_payment",[(this)
64                 .attr('data-id')]);
65         });
66         // Evento al hacer clic en el botón "Print"
67         $('#print').click(function(){
68             // Lógica para imprimir los detalles del alumno
69         });
70     });
71
72     // Función para eliminar un pago
73     function delete_payment($id){
74         start_loader();
75         $.ajax({
76             url: _base_url_ + "classes/Master.php?f=delete_payment",
77             method: "POST",

```

```

//      url: _base_url_ + "classes/Master.php?it=delete_payment",
78      method: "POST",
79      data: {id: $id},
80      dataType: "json",
81      error: function(err){
82          console.log(err);
83          alert_toast("Ocurrió un error.", 'error');
84          end_loader();
85      },
86      success: function(resp){
87          if (typeof resp == 'object' && resp.status == 'success'){
88              location.reload();
89          } else {
90              alert_toast("Ocurrió un error.", 'error');
91              end_loader();
92          }
93      }
94  });
95  }
96  </script>
97

```

Así mismo, con la finalidad de realizar las diversas actividades con las carreras habilitadas, se crearon las pantallas de índice, ver carreras y editar carreras.

```

<<▶ index.php ×
1  <style>
2      .img-avatar{
3          width:45px;
4          height:45px;
5          object-fit:cover;
6          object-position:center center;
7          border-radius:100%;
8      }
9  </style>
10
11  <div class="card card-outline card-primary">
12      <div class="card-header">
13          <h3 class="card-title">Lista de Carreras</h3>
14          <div class="card-tools">
15              <a href="javascript:void(0)" id="create_new" class="btn btn-flat btn-sm btn-primary"><
16                  <span class="fas fa-plus"></span> Nuevos</a>
17          </div>
18      </div>
19      <div class="card-body">
20          <div class="container-fluid">
21              <table class="table table-hover table-striped">
22                  <colgroup>
23                      <col width="5%">
24                      <col width="20%">
25                      <col width="25%">
26                      <col width="25%">
27                      <col width="15%">
28                      <col width="10%">
29                  </colgroup>
30                  <thead>
31                      <tr>
32                          <th>#</th>
33                          <th>Fecha de creación</th>
34                          <th>Facultad</th>
35                          <th>Nombre</th>
36                          <th>Estado</th>
37                          <th>Acción</th>
38                      </tr>
39                  </thead>
40                  <tbody>
41                      <?php

```

```

41         $i = 1;
42         $qry = $conn->query("SELECT c.*, d.name as department from `curriculum_list`
         `c` inner join `department_list` d on c.department_id = d.id order by
         c.`name` asc");
43         while($row = $qry->fetch_assoc()):
44             ??
45             <tr>
46                 <td class="text-center"><?php echo $i++; ?></td>
47                 <td class=""><?php echo date("Y-m-d H:i", strtotime($row['date_created']
48                     )); ?></td>
49                 <td class=""><?php echo $row['department']; ?></td>
50                 <td><?php echo ucwords($row['name']); ?></td>
51                 <td class="text-center">
52                     <?php
53                         switch($row['status']){
54                             case '1':
55                                 echo "<span class='badge badge-success
56                                     badge-pill'>Activo</span>";
57                                 break;
58                             case '0':
59                                 echo "<span class='badge badge-secondary
60                                     badge-pill'>Inactivo</span>";
61                                 break;
62                         }
63                     </td>
64                 <td align="center">
65                     <div class="dropdown">
66                         <button class="btn btn-flat btn-default btn-sm dropdown-toggle
67                             dropdown-icon" type="button" id="dropdownMenuButton"
68                             data-toggle="dropdown" aria-haspopup="true" aria-expanded="
69                             false">
70                             Acción
71                         </button>
72                         <div class="dropdown-menu" aria-labelledby="dropdownMenuButton"
73                             >
74                             <a class="dropdown-item view_data" href="javascript:void(0)"
75                                 " data-id="<?php echo $row['id']; ?>"><span class="fa
76                                 fa-eye text-dark"></span> Ver</a>
77                             <div class="dropdown-divider"></div>
78                             <a class="dropdown-item edit_data" href="javascript:void(0)"
79                                 " data-id="<?php echo $row['id']; ?>"><span class="fa
80                                 fa-edit text-primary"></span> Editar</a>
81                             <div class="dropdown-divider"></div>
82                             <a class="dropdown-item delete_data" href="javascript:void(0)"
83                                 " data-id="<?php echo $row['id']; ?>"><span class="fa
84                                 fa-trash text-danger"></span> Eliminar</a>
85                         </div>
86                     </td>
87             </tr>
88         <?php endwhile; ?>
89     </tbody>
90 </table>
91 </div>
92 </div>
93 </div>
94 <script>
95     $(document).ready(function(){
96         // Evento para crear una nueva materia
97         $('#create_new').click(function(){
98             uni_modal("Detalles de la materia","curriculum/manage_curriculum.php")
99         });
100         // Evento para editar una materia existente
101         $('.edit_data').click(function(){
102             uni_modal("Detalles de la materia","curriculum/manage_curriculum.php?id="+$(this).attr(
103                 'data-id'))
104         });
105         // Evento para eliminar una materia
106         $('.delete_data').click(function(){
107             _conf("¿Estás seguro de eliminar la materia de forma permanente?", "delete_curriculum",[
108                 $(this).attr('data-id')]
109             );
110         });
111         // Evento para ver los detalles de una materia
112         $('.view_data').click(function(){
113             uni_modal("Detalles de la materia","curriculum/view_curriculum.php?id="+$(this).attr(
114                 'data-id'))
115         });
116     });
117 </script>

```

```

103         uni_modal("Detalles de la materia","curriculum/view_curriculum.php?id="+$(this).attr('
            data-id'))
104     });
105
106     // Agregar clases CSS a las celdas de la tabla
107     $('table td,table th').addClass('py-1 px-2 align-middle');
108
109     // Inicializar el plugin DataTables
110     $('#table').dataTable({
111         columnDefs: [
112             { orderable: false, targets: 5 }
113         ],
114     });
115
116     // Función para eliminar una materia
117     function delete_curriculum($id){
118         start_loader();
119         $.ajax({
120             url: _base_url_ + "classes/Master.php?f=delete_curriculum",
121             method: "POST",
122             data: {id: $id},
123             dataType: "json",
124             error: function(err){
125                 console.log(err);
126                 alert_toast("Ocurrió un error.", 'error');
127                 end_loader();
128             },
129             success: function(resp){
130                 if(typeof resp == 'object' && resp.status == 'success'){
131                     location.reload();
132                 }else{
133                     alert_toast("Ocurrió un error.", 'error');
134                     end_loader();
135                 }
136             }
137         });
138     }
139
140 </script>

```

```

manage_curriculum.php
1 <?php
2 // Incluye el archivo de configuración
3 require_once('.../config.php');
4
5 // Verifica si se proporcionó un ID y recupera los datos del currículo correspondiente
6 if(isset($_GET['id'])){
7     $qry = $conn->query("SELECT * FROM `curriculum_list` where id = '{$_GET['id']}'");
8     if($qry->num_rows > 0){
9         $res = $qry->fetch_array();
10         foreach($res as $k => $v){
11             if(!is_numeric($k))
12                 $k = $v;
13         }
14     }
15 }
16 ?>
17
18 <div class="container-fluid">
19     <form action="" id="curriculum-form">
20         <!-- Campo oculto para almacenar el ID -->
21         <input type="hidden" name="id" value="<?php echo isset($id) ? $id : '' ?>">
22
23         <!-- Selector de departamento -->
24         <div class="form-group">
25             <label for="department_id" class="control-label">Departamento</label>
26             <select name="department_id" id="department_id" class="form-control form-control-border"
27                 required data-placeholder="Seleccionar departamento">
28                 <!-- Opción predeterminada vacía -->
29                 <option <?= isset($department_id) == 1 ? "selected" : "" ?>></option>
30                 <?php
31                 // Consulta para obtener la lista de departamentos activos
32                 $department = $conn->query("SELECT * FROM `department_list` where `status` = 1.(
33                     isset($department_id) ? "OR id = '{$_GET['id']}'" : "" )." order by `name`
34                     asc");
35                 while($row = $department->fetch_assoc()){
36                     <!-- Opciones del selector de departamento -->
37                     <option value="<?= $row['id'] ?>" <?= isset($department_id) && $department_id == $
38                         row['id'] ? "selected" : "" ?>><?= ucwords($row['name']) ?>></option>
39                 <?php endwhile; ?>
40             </select>

```

```

38     </div>
39
40     <!-- Campo de nombre -->
41     <div class="form-group">
42         <label for="name" class="control-label">Nombre</label>
43         <input type="text" name="name" id="name" class="form-control form-control-border"
44             placeholder="Nombre de la materia" value = "<?php echo isset($name) ? $name : '' ?>"
45             required>
46     </div>
47
48     <!-- Campo de descripción -->
49     <div class="form-group">
50         <label for="description" class="control-label">Descripción</label>
51         <textarea rows="3" name="description" id="description" class="form-control
52             form-control-border" placeholder="Escriba la descripción" required><?php echo isset($
53             description) ? $description : '' ?></textarea>
54     </div>
55
56     <!-- Selector de estado -->
57     <div class="form-group">
58         <label for="" class="control-label">Estado</label>
59         <select name="status" id="status" class="form-control form-control-border" required>
60             <option value="1" <? = isset($status) && $status == 1 ? "selected" : "" ?>>Activo</
61             option>
62             <option value="0" <? = isset($status) && $status == 0 ? "selected" : "" ?>>Inactivo
63             </option>
64         </select>
65     </div>
66 </form>
67 </div>
68
69 <script>
70     $(function(){
71         // Inicializa el plugin Select2 para el selector de departamento
72         $('#department_id').select2({
73             width: "100%",
74             dropdownParent: $("#uni_modal")
75         })
76     })

```

```

69     })
70
71     // Evento de envío del formulario
72     $('#uni_modal #curriculum-form').submit(function(e){
73         e.preventDefault();
74         var _this = $(this);
75         $('#pop-msg').remove();
76         var el = $('<div>');
77         el.addClass("pop-msg alert")
78         el.hide();
79         start_loader();
80         $.ajax({
81             url: _base_url + "classes/Master.php?f=save_curriculum",
82             data: new FormData(_this[0]),
83             cache: false,
84             contentType: false,
85             processData: false,
86             method: 'POST',
87             type: 'POST',
88             dataType: 'json',
89             error: function(err){
90                 console.log(err);
91                 alert_toast("Ocurrió un error.", 'error');
92                 end_loader();
93             },
94             success: function(resp){
95                 if(resp.status == 'success'){
96                     location.reload();
97                 }else if(!resp.msg){
98                     el.addClass("alert-danger")
99                     el.text(resp.msg)
100                     _this.prepend(el)
101                 }else{
102                     el.addClass("alert-danger")
103                     el.text("Ocurrió un error debido a una razón desconocida.")
104                     _this.prepend(el)
105                 }
106             }
107         });
108         el.show('slow')
109         end_loader();

```

```

87         type: 'POST',
88         datatype: 'json',
89         error: function(err){
90             console.log(err)
91             alert_toast("Ocurrió un error.", 'error');
92             end_loader();
93         },
94         success: function(resp){
95             if(resp.status == 'success'){
96                 location.reload();
97             }else if(!resp.msg){
98                 el.addClass("alert-danger")
99                 el.text(resp.msg)
100                 _this.prepend(el)
101             }else{
102                 el.addClass("alert-danger")
103                 el.text("Ocurrió un error debido a una razón desconocida.")
104                 _this.prepend(el)
105             }
106             el.show('slow')
107             end_loader();
108         }
109     }
110 })
111 })
112 }
113 </script>
114

```

```

view_curriculum.php
1 <?php
2 // Verificar si se ha proporcionado un ID en la URL
3 if(isset($_GET['id'])){
4     // Realizar la consulta para obtener los datos del plan de estudios y su departamento
5     // correspondiente
6     $qry = $conn->query("SELECT c.*, d.name as department FROM `curriculum_list` c INNER JOIN `
7     department_list` d ON c.department_id = d.id WHERE c.id = '{$_GET['id']}'");
8     // Verificar si se encontraron resultados en la consulta
9     if($qry->num_rows > 0){
10         // Recorrer los resultados y asignar cada valor a una variable con el mismo nombre de la
11         // columna
12         $res = $qry->fetch_array();
13         foreach($res as $k => $v){
14             if(!is_numeric($k))
15                 $$k = $v;
16         }
17     }
18 }
19 <style>
20 /* Ocultar el pie de página del modal */
21 #uni_modal .modal-footer{
22     display:none !important;
23 }
24 </style>
25 <div class="container-fluid">
26 <!-- Mostrar los detalles del plan de estudios -->
27 <dl>
28     <dt class="text-muted">Facultad</dt>
29     <dd class="pl-4 fs-4 fw-bold"><?= isset($department) ? $department : '' ?></dd>
30     <dt class="text-muted">Nombre</dt>
31     <dd class="pl-4 fs-4 fw-bold"><?= isset($name) ? $name : '' ?></dd>
32     <dt class="text-muted">Descripción</dt>
33     <dd class="pl-4">
34         <p class=""><small><?= isset($description) ? $description : '' ?></small></p>
35     </dd>
36     <dt class="text-muted">Estado</dt>
37     <dd class="pl-4">
38         <?php
39         // Mostrar el estado del plan de estudios como una etiqueta de distintivo
40         if(isset($status)){
41             switch($status){
42                 case '1':
43                     echo "<span class='badge badge-success badge-pill'>Activo</span>";
44                     break;
45                 case '0':
46                     echo "<span class='badge badge-secondary badge-pill'>Inactivo</span>";
47                     break;
48             }
49         }
50     </dd>
51 </dl>
52 <!-- Botón para cerrar el modal -->
53 <div class="col-12 text-right">
54     <button class="btn btn-flat btn-sm btn-dark" type="button" data-dismiss="modal"><i class="fa
55     fa-times"></i> Cerrar</button>
56 </div>
57

```

```

37 // Mostrar el estado del plan de estudios como una etiqueta de distintivo
38 if(isset($status)){
39     switch($status){
40         case '1':
41             echo "<span class='badge badge-success badge-pill'>Activo</span>";
42             break;
43         case '0':
44             echo "<span class='badge badge-secondary badge-pill'>Inactivo</span>";
45             break;
46     }
47 }
48 }
49 </dd>
50 </dl>
51 <!-- Botón para cerrar el modal -->
52 <div class="col-12 text-right">
53     <button class="btn btn-flat btn-sm btn-dark" type="button" data-dismiss="modal"><i class="fa
54     fa-times"></i> Cerrar</button>
55 </div>
56

```

Así mismo, con la finalidad de realizar las diversas actividades con las facultades habilitadas, se crearon las pantallas de índice, ver facultades y editar facultades.

```

1  <style>
2    .img-avatar {
3      width: 45px;
4      height: 45px;
5      object-fit: cover;
6      object-position: center center;
7      border-radius: 100%;
8    }
9  </style>
10
11 <div class="card card-outline card-primary">
12   <div class="card-header">
13     <h3 class="card-title">Lista de Facultades</h3>
14     <div class="card-tools">
15       <a href="javascript:void(0)" id="create_new" class="btn btn-flat btn-sm btn-primary">
16         <span class="fas fa-plus"></span> Nuevo
17       </a>
18     </div>
19   </div>
20   <div class="card-body">
21     <div class="container-fluid">
22       <table class="table table-hover table-striped">
23         <colgroup>
24           <col width="5%">
25             <col width="20%">
26             <col width="20%">
27             <col width="30%">
28             <col width="15%">
29             <col width="10%">
30           </colgroup>
31           <thead>
32             <tr>
33               <th>#</th>
34               <th>Fecha de creación</th>
35               <th>Nombre</th>
36               <th>Descripción</th>
37               <th>Estado</th>
38               <th>Acción</th>
39             </tr>
40           </thead>
41           <tbody>

```

```

42   <?php
43     // Inicializar contador
44     $i = 1;
45
46     // Realizar la consulta para obtener la lista de departamentos
47     $qry = $conn->query("SELECT * FROM `department_list` ORDER BY `name` ASC");
48
49     // Recorrer los resultados de la consulta
50     while($row = $qry->fetch_assoc()):
51   <?>
52   <tr>
53     <td class="text-center"><?=$i++; ?></td>
54     <td><?=$date("Y-m-d H:i", strtotime($row['date_created'])) ?></td>
55     <td><?=$ucwords($row['name']) ?></td>
56     <td class="truncate-1"><?=$row['description'] ?></td>
57     <td class="text-center">
58       <?php
59         // Mostrar el estado del departamento
60         echo match($row['status']) {
61           '1' => "<span class='badge badge-success badge-pill'>Activo</span>",
62           '0' => "<span class='badge badge-secondary badge-pill'>Inactivo</span>",
63           default => "<span class='badge badge-warning badge-pill'>Desconocido</span>",
64         };
65       <?>
66     </td>
67     <td align="center">
68       <button type="button" class="btn btn-flat btn-default btn-sm dropdown-toggle dropdown-icon" data-toggle="dropdown">
69         Acción
70       <span class="sr-only">Toggle Dropdown</span>
71     </button>
72     <div class="dropdown-menu" role="menu">
73       <a class="dropdown-item view_data" href="javascript:void(0)"
74         data-id="<?=$row['id'] ?>"><span class="fa fa-eye text-dark"></span> Ver</a>
75       <div class="dropdown-divider"></div>
76       <a class="dropdown-item edit_data" href="javascript:void(0)"

```

```

74         <div class="dropdown-divider"></div>
75         <a class="dropdown-item edit_data" href="javascript:void(0)"
           data-id="<?= $row['id'] ?>"><span class="fa fa-edit text-primary">
</span> Editar</a>
76         <div class="dropdown-divider"></div>
77         <a class="dropdown-item delete_data" href="javascript:void(0)"
           data-id="<?= $row['id'] ?>"><span class="fa fa-trash text-danger">
</span> Eliminar</a>
78     </div>
79 </td>
80 </tr>
81 <?php endwhile; ?>
82 </tbody>
83 </table>
84 </div>
85 </div>
86 </div>
87
88 <script>
89     $(document).ready(function() {
90         // Botón para crear un nuevo departamento
91         $('#create_new').click(function() {
92             uni_modal("Detalles del Departamento", "departments/manage_department.php");
93         });
94
95         // Botón para editar un departamento existente
96         $('.edit_data').click(function() {
97             uni_modal("Detalles del Departamento", "departments/manage_department.php?id=" + $(this)
98                 .attr('data-id'));
99         });
100
101         // Botón para eliminar un departamento existente
102         $('.delete_data').click(function() {
103             _conf("¿Está seguro de eliminar este departamento de forma permanente?", "
104                 delete_department", [$(this).attr('data-id')]);
105         });
106
107         // Botón para ver detalles de un departamento
108         $('.view_data').click(function() {
109             uni_modal("Detalles del Departamento", "departments/view_department.php?id=" + $(this)
110                 .attr('data-id'));
111         });
112     });
113 </script>

```

```

117         uni_modal("Detalles del Departamento", "departments/view_department.php?id=" + $(this)
118             .attr('data-id'));
119     });
120
121     // Agregar clases de estilo a las celdas de la tabla
122     $('.table td, .table th').addClass('py-1 px-2 align-middle');
123
124     // Inicializar DataTable con configuración personalizada
125     $('.table').dataTable({
126         columnDefs: [
127             { orderable: false, targets: 5 }
128         ],
129     });
130
131 // Función para eliminar un departamento
132 function delete_department(id) {
133     start_loader();
134     $.ajax({
135         url: _base_url + "classes/Master.php?f=delete_department",
136         method: "POST",
137         data: { id: id },
138         dataType: "json",
139         error: function(err) {
140             console.log(err);
141             alert_toast("Ocurrió un error.", 'error');
142             end_loader();
143         },
144         success: function(resp) {
145             if (typeof resp === 'object' && resp.status === 'success') {
146                 location.reload();
147             } else {
148                 alert_toast("Ocurrió un error.", 'error');
149                 end_loader();
150             }
151         }
152     });
153 }
154 </script>
155

```

```

1 <?php
2 require_once('../config.php');
3
4 // Verificar si el parámetro 'id' está presente en la solicitud GET
5 if (isset($_GET['id'])) {
6     // Usar consultas preparadas para evitar inyección SQL
7     $stmt = $conn->prepare("SELECT * FROM `department_list` WHERE id = ?");
8     $stmt->bind_param("i", $_GET['id']);
9     $stmt->execute();
10    $result = $stmt->get_result();
11
12    // Verificar si se encontró algún resultado
13    if ($result->num_rows > 0) {
14        // Obtener los resultados como un array asociativo
15        $res = $result->fetch_assoc();
16        foreach ($res as $k => $v) {
17            // Asignar variables dinámicas
18            if (is_numeric($k)) {
19                $$k = $v;
20            }
21        }
22    }
23    $stmt->close();
24 }
25 ?>
26
27 <div class="container-fluid">
28     <form action="" id="department-form">
29         <input type="hidden" name="id" value="<?= isset($id) ? htmlspecialchars($id) : '' ?>">
30         <div class="form-group">
31             <label for="name" class="control-label">Nombre</label>
32             <input type="text" name="name" id="name" class="form-control form-control-border"
33                 placeholder="Escriba el nombre" value="<?= isset($name) ? htmlspecialchars($name) : '' ?>"
34                 required>
35         </div>
36         <div class="form-group">
37             <label for="description" class="control-label">Descripción</label>
38             <textarea rows="3" name="description" id="description" class="form-control
39                 form-control-border" placeholder="Escriba la descripción." required<?= isset($
40                 description) ? htmlspecialchars($description) : '' ?></textarea>
41         </div>

```

```

42         <div class="form-group">
43             <label for="status" class="control-label">Estado</label>
44             <select name="status" id="status" class="form-control form-control-border" required>
45                 <option value="1" <?= isset($status) && $status == 1 ? "selected" : "" ?>>Activo</
46                 <option value="0" <?= isset($status) && $status == 0 ? "selected" : "" ?>>Inactivo</
47                 </select>
48             </div>
49         </form>
50     </div>
51     <script>
52         $(function() {
53             // Manejar la presentación del formulario
54             $('#uni_modal #department-form').submit(function(e) {
55                 e.preventDefault(); // Prevenir el comportamiento por defecto del formulario
56                 var _this = $(this);
57                 $('#pop-msg').remove(); // Eliminar mensajes anteriores
58                 var el = $('#div').addClass("pop-msg alert").hide();
59
60                 start_loader(); // Iniciar cargador
61
62                 // Realizar la solicitud AJAX
63                 $.ajax({
64                     url: _base_url_ + "classes/Master.php?f=save_department",
65                     data: new FormData(_this[0]),
66                     cache: false,
67                     contentType: false,
68                     processData: false,
69                     method: 'POST',
70                     dataType: 'json',
71                     error: function(err) {
72                         console.log(err);
73                         alert_toast("Ocurrió un error", 'error');
74                         end_loader();
75                     },
76                     success: function(resp) {
77                         if (resp.status === 'success') {
78                             location.reload();
79                         }
80                     }
81                 });
82             });
83         });
84     </script>
85

```

```

86         location.reload();
87     } else if (resp.msg) {
88         el.addClass("alert-danger").text(resp.msg);
89         _this.prepend(el);
90     } else {
91         el.addClass("alert-danger").text("Ocurrió un error debido a una razón
92             desconocida.");
93         _this.prepend(el);
94     }
95     el.show('slow');
96     end_loader();
97 }
98
99
100
101
102
103
104
105
106
107
108
109

```

```

view_department.php
1 <?php
2 require_once('../config.php');
3
4 // Verificar si el parámetro 'id' está presente en la solicitud GET
5 if (isset($_GET['id'])) {
6     // Usar consultas preparadas para evitar inyección SQL
7     $stmt = $conn->prepare("SELECT * FROM `department_list` WHERE id = ?");
8     $stmt->bind_param("i", $_GET['id']);
9     $stmt->execute();
10    $result = $stmt->get_result();
11
12    // Verificar si se encontró algún resultado
13    if ($result->num_rows > 0) {
14        // Obtener los resultados como un array asociativo
15        $res = $result->fetch_assoc();
16        foreach ($res as $k => $v) {
17            // Asignar variables dinámicas
18            if (!is_numeric($k)) {
19                $$k = $v;
20            }
21        }
22    }
23    $stmt->close();
24 }
25 ?>
26 <style>
27 /* Estilo para ocultar el pie de página del modal */
28 #uni_modal .modal-footer {
29     display: none !important;
30 }
31 </style>
32 <div class="container-fluid">
33     <dl>
34         <dt class="text-muted">Nombre</dt>
35         <dd class="p1-4 fs-4 fw-bold"><?= isset($name) ? htmlspecialchars($name) : '' ?></dd>
36         <dt class="text-muted">Descripción</dt>
37         <dd class="p1-4">
38             <p class=""><small><?= isset($description) ? htmlspecialchars($description) : '' ?></small></p>
39         </dd>
40     </dl>
41     <dt class="text-muted">Estado</dt>
42     <dd class="p1-4">
43         <?php if (isset($status)): ?>
44             <?php switch ($status):
45                 case '1': ?>
46                     <span class="badge badge-success badge-pill">Activo</span>
47                     <?php break; ?>
48                 <?php case '0': ?>
49                     <span class="badge badge-secondary badge-pill">Inactivo</span>
50                     <?php break; ?>
51             <?php endswitch; ?>
52         <?php endif; ?>
53     </dd>
54 </dl>
55 <div class="col-12 text-right">
56     <button class="btn btn-flat btn-sm btn-dark" type="button" data-dismiss="modal"><i class="fa
57     fa-times"></i> Cerrar</button>
58 </div>
59 </div>

```

Así también, con la finalidad de realizar las diversas actividades con los usuarios, se crearon las pantallas de índice, ver detalles y actualizar usuarios.

```

index.php
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Lista de estudiantes</title>
7 <style>
8 /* Estilos para los avatares de los usuarios */
9 .img-avatar {
10 width: 45px;
11 height: 45px;
12 object-fit: cover;
13 object-position: center center;
14 border-radius: 100%;
15 }
16 </style>
17 </head>
18 <body>
19 <div class="card card-outline card-primary">
20 <div class="card-header">
21 <h3 class="card-title">Lista de estudiantes</h3>
22 </div>
23 <div class="card-body">
24 <div class="container-fluid">
25 <table class="table table-hover table-striped">
26 <thead>
27 <tr>
28 <th>#</th>
29 <th>Avatar</th>
30 <th>Nombre</th>
31 <th>Correo</th>
32 <th>Estado</th>
33 <th>Acción</th>
34 </tr>
35 </thead>
36 <tbody>
37 <?php
38 // Inicializa el contador de filas
39 $i = 1;
40 // Realiza la consulta a la base de datos

```

```

40 // Realiza la consulta a la base de datos
41 $qry = $conn->query("SELECT *, CONCAT(lastname, ' ', firstname, ' ',
42 middlename) as name FROM 'student_list' ORDER BY CONCAT(lastname,
43 ' ', firstname, ' ', middlename) ASC");
44 // Recorre los resultados de la consulta
45 while($row = $qry->fetch_assoc()):
46 <?
47 <tr>
48 <td class="text-center"><?=$i++; ?></td>
49 <td class="text-center"></td>
52 <td><?=$ucwords($row['name']) ?></td>
53 <td><?=$truncate($row['email'], 1) ?></td>
54 <td class="text-center">
55 <?php if($row['status'] == 1): ?>
56 <span class="badge badge-pill badge-success">Verificado</span>
57 <?php else: ?>
58 <span class="badge badge-pill badge-primary">No Verificado</span>
59 <?php endif; ?>
60 <td align="center">
61 <button type="button" class="btn btn-flat btn-default btn-sm
62 dropdown-toggle dropdown-icon" data-toggle="dropdown">
63 Acción
64 <span class="sr-only">Toggle Dropdown</span>
65 </button>
66 <div class="dropdown-menu" role="menu">
67 <a class="dropdown-item view_details" href="javascript:void
68 (0)" data-id="<?=$row['id'] ?>"><span class="fa fa-eye
69 text-dark"></span> Ver</a>
70 <div class="dropdown-divider"></div>
71 <?php if($row['status'] != 1): ?>
72 <a class="dropdown-item verify_user" href="
73 javascript:void(0)" data-id="<?=$row['id'] ?>"
74 data-name="<?=$row['email'] ?>"><span class="fa
75 fa-check text-primary"></span> Verificar</a>

```

```

67         <?php if($row['status'] != 1): ?>
68         <a class="dropdown-item verify_user" href="
        javascript:void(0)" data-id="<?=$row['id'] ?>"
        data-name="<?=$row['email'] ?>"><span class="fa
        fa-check text-primary"></span> Verificar</a>
69         <div class="dropdown-divider"></div>
70         <?php endif; ?>
71         <a class="dropdown-item delete_data" href="javascript:void(
        0)" data-id="<?=$row['id'] ?>" data-name="<?=$row['email']
        ?>"><span class="fa fa-trash text-danger"></span>
        Eliminar</a>
72         </div>
73     </td>
74 </tr>
75 <?php endwhile; ?>
76 </tbody>
77 </table>
78 </div>
79 </div>
80 </div>
81
82 <!-- Scripts para manejar las acciones de los botones -->
83 <script>
84     document.addEventListener('DOMContentLoaded', function() {
85         // Maneja la acción de eliminar un estudiante
86         document.querySelectorAll('.delete_data').forEach(function(button) {
87             button.addEventListener('click', function() {
88                 const name = this.getAttribute('data-name');
89                 const id = this.getAttribute('data-id');
90                 _conf('¿Estás seguro de eliminar <b>${name}</b> de la lista de estudiantes de
                        forma permanente?', "delete_user", [id]);
91             });
92         });
93
94         // Maneja la acción de verificar un estudiante
95         document.querySelectorAll('.verify_user').forEach(function(button) {
96             button.addEventListener('click', function() {
97                 const name = this.getAttribute('data-name');
98                 const id = this.getAttribute('data-id');
99                 _conf('¿Estás seguro de verificar <b>${name}</b>', "verify_user", [id]);
100             });
101         });
102
103         // Maneja la acción de ver detalles de un estudiante
104         document.querySelectorAll('.view_details').forEach(function(button) {
105             button.addEventListener('click', function() {
106                 const id = this.getAttribute('data-id');
107                 uni_modal('Detalles del estudiante', 'students/view_details.php?id=${id}', '
                        mid-large');
108             });
109         });
110
111         // Inicializa DataTables para la tabla
112         $('table').dataTable();
113     });
114
115     // Función para eliminar un estudiante
116     function delete_user(id) {
117         start_loader();
118         $.ajax({
119             url: _base_url + "classes/Users.php?f=delete_student",
120             method: "POST",
121             data: { id },
122             dataType: "json",
123             error: function(err) {
124                 console.error(err);
125                 alert_toast("Ocurrió un error.", 'error');
126                 end_loader();
127             },
128             success: function(resp) {
129                 if (typeof resp === 'object' && resp.status === 'success') {
130                     location.reload();
131                 } else {
132                     alert_toast("Ocurrió un error.", 'error');
133                     end_loader();
134                 }
135             }
136         });
137     }
138

```

```

137
138
139 // Función para verificar un estudiante
140 function verify_user(id) {
141     start_loader();
142     $.ajax({
143         url: base_url + "classes/Users.php?f=verify_student",
144         method: "POST",
145         data: { id },
146         datatype: "json",
147         error: function(err) {
148             console.error(err);
149             alert_toast("Ocurrió un error.", 'error');
150             end_loader();
151         },
152         success: function(resp) {
153             if (typeof resp === 'object' && resp.status === 'success') {
154                 location.reload();
155             } else {
156                 alert_toast("Ocurrió un error.", 'error');
157                 end_loader();
158             }
159         }
160     });
161 }
162 </script>
163 </body>
164 </html>
165
166

```

```

view_details.php
1 <?php
2 require_once('.../config.php');
3
4 // Verificar si se ha proporcionado un ID válido
5 if(isset($_GET['id']) && $_GET['id'] > 0){
6     // Obtener los datos del estudiante
7     $user_query = $conn->query("SELECT s.*, d.name as department, c.name as curriculum, CONCAT(
8         lastname, ' ', firstname, ' ', middlename) as fullname FROM student_list s INNER JOIN
9         department_list d ON s.department_id = d.id INNER JOIN curriculum_list c ON s.curriculum_id
10         = c.id WHERE s.id = '{$_GET['id']}'");
11
12     // Verificar si se encontró un estudiante con el ID proporcionado
13     if($user_query->num_rows > 0) {
14         $user_data = $user_query->fetch_assoc();
15
16         // Extraer los datos del estudiante en variables individuales
17         foreach($user_data as $k => $v){
18             $$k = $v;
19         }
20     } else {
21         // Si no se encuentra un estudiante con el ID proporcionado, redireccionar o mostrar un
22         // mensaje de error
23         // Ejemplo: header("Location: error.php");
24     }
25 }
26 >>
27 <style>
28     #uni_modal .modal-footer{
29         display:none
30     }
31     .student-img{
32         object-fit:scale-down;
33         object-position:center center;
34     }
35 </style>
36 <div class="container-fluid">
37     <div class="col-md-12">
38         <div class="row">
39             <div class="col-6">
40                 <center>
41                     
39     </div>
40     <div class="col-6">
41         <dl>
42             <dt class="text-navy">Nombre del estudiante:</dt>
43             <dd class="pl-4"><? ucwords($fullname) ?></dd>
44             <dt class="text-navy">Sexo:</dt>
45             <dd class="pl-4"><? ucwords($gender) ?></dd>
46             <dt class="text-navy">Correo:</dt>
47             <dd class="pl-4"><? $email ?></dd>
48             <dt class="text-navy">Facultad:</dt>
49             <dd class="pl-4"><? ucwords($department) ?></dd>
50             <dt class="text-navy">Carrera:</dt>
51             <dd class="pl-4"><? ucwords($curriculum) ?></dd>
52             <dt class="text-navy">Estado de la cuenta:</dt>
53             <dd class="pl-4">
54                 <?php if($status == 1): ?>
55                     <span class="badge badge-pill badge-success">Verificado</span>
56                 <?php else: ?>
57                     <span class="badge badge-pill badge-primary">No Verificado</span>
58                 <?php endif; ?>
59             </dd>
60         </dl>
61     </div>
62 </div>
63 <div class="row">
64     <div class="col-12 text-right">
65         <button class="btn btn-dark btn-flat btn-sm" data-dismiss="modal" type="button"><i
66             class="fa fa-times"></i> Cerrar</button>
67     </div>
68 </div>
69 </div>
70

```

```

manage_user.php
1 <?php
2 // Verifica si se ha pasado un ID válido por GET y recupera los datos del usuario si es así
3 if(isset($_GET['id']) && $_GET['id'] > 0){
4     $user = $conn->query("SELECT * FROM users WHERE id='{$_GET['id']}'");
5     foreach($user->fetch_array() as $k => $v){
6         $meta[$k] = $v;
7     }
8 }
9 >
10 <?php if($_settings->chk_flashdata('success')): ?>
11     <script>
12         // Muestra una notificación de éxito si existe un mensaje flash
13         alert_toast("<?php echo $_settings->flashdata('success') ?>", 'success');
14     </script>
15 <?php endif; ?>
16
17 <div class="card card-outline card-primary">
18     <div class="card-body">
19         <div class="container-fluid">
20             <div id="msg"></div>
21             <form action="" id="manage-user" enctype="multipart/form-data">
22                 <input type="hidden" name="id" value="<?php echo isset($meta['id']) ? $meta['id'] : '' ?>">
23                 <div class="form-group col-6">
24                     <label for="name">Nombre</label>
25                     <input type="text" name="firstname" id="firstname" class="form-control" value="
26                         <?php echo isset($meta['firstname']) ? $meta['firstname'] : '' ?>" required>
27                 </div>
28                 <div class="form-group col-6">
29                     <label for="name">Apellido</label>
30                     <input type="text" name="lastname" id="lastname" class="form-control" value="
31                         <?php echo isset($meta['lastname']) ? $meta['lastname'] : '' ?>" required>
32                 </div>
33                 <div class="form-group col-6">
34                     <label for="username">Nombre de Usuario</label>
35                     <input type="text" name="username" id="username" class="form-control" value="
36                         <?php echo isset($meta['username']) ? $meta['username'] : '' ?>" required
37                     autocomplete="off">
38                 </div>
39                 <div class="form-group col-6">
40                     <label for="password">Contraseña</label>

```

```

35     <div class="form-group col-6">
36         <label for="password">Contraseña</label>
37         <input type="password" name="password" id="password" class="form-control" value
38             ="" autocomplete="off" <?php echo isset($meta['id']) ? "" : 'required' ?>
39         <?php if(isset($GET['id'])): ?>
40             <small class="text-info"><i>Deje esto en blanco si no desea cambiar la
41             contraseña.</i></small>
42         <?php endif; ?>
43     </div>
44     <div class="form-group col-6">
45         <label for="type">Tipo de usuario</label>
46         <select name="type" id="type" class="custom-select" required>
47             <option value="1" <?php echo isset($meta['type']) && $meta['type'] == 1 ? '
48                 selected': '' ?>>Administrador</option>
49             <option value="2" <?php echo isset($meta['type']) && $meta['type'] == 2 ? '
50                 selected': '' ?>>Staff</option>
51         </select>
52     </div>
53     <div class="form-group col-6">
54         <label for="" class="control-label">Avatar</label>
55         <div class="custom-file">
56             <input type="file" class="custom-file-input rounded-circle" id="customFile"
57                 name="img" onchange="displayImg(this,$(this))">
58             <label class="custom-file-label" for="customFile">Elija el archivo</label>
59         </div>
60     </div>
61     <div class="form-group col-6 d-flex justify-content-center">
62         
64     </div>
65 </form>
66 </div>
67 <div class="card-footer">
68     <div class="col-md-12">
69         <div class="row">
70             <button class="btn btn-sm btn-primary mr-2" form="manage-user">Guardar</button>
71             <a class="btn btn-sm btn-secondary" href="./?page=user/list">Cancelar</a>
72         </div>
73     </div>
74 </div>
75 </div>

```

```

76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 <style>
82     img#cimg{
83         height: 15vh;
84         width: 15vh;
85         object-fit: cover;
86         border-radius: 100% 100%;
87     }
88 </style>
89 <script>
90     $(function(){
91         // Inicializa el plugin Select2 para los selects
92         $('select').select2({
93             width: 'resolve'
94         });
95     });
96
97     // Función para mostrar la imagen seleccionada
98     function displayImg(input, this) {
99         if (input.files && input.files[0]) {
100             var reader = new FileReader();
101             reader.onload = function (e) {
102                 $('#cimg').attr('src', e.target.result);
103             };
104             reader.readAsDataURL(input.files[0]);
105         }
106     }
107
108     // Envío del formulario mediante AJAX
109     $('#manage-user').submit(function(e){
110         e.preventDefault();
111         var _this = $(this);
112         start_loader();
113         $.ajax({
114             url: _base_url+'classes/Users.php?f=save',
115             data: new FormData($('form')[0]),
116             cache: false,

```

```

101     $('#manage-user').submit(function(e){
102         e.preventDefault();
103         var _this = $(this);
104         start_loader();
105         $.ajax({
106             url: _base_url+'classes/Users.php?f=save',
107             data: new FormData(_this[0]),
108             cache: false,
109             contentType: false,
110             processData: false,
111             method: 'POST',
112             type: 'POST',
113             success: function(resp){
114                 if(resp == 1){
115                     location.href = './?page=user/list';
116                 } else {
117                     $('#msg').html('<div class="alert alert-danger">El nombre de usuario ya
118                         existe</div>');
119                     $("html, body").animate({ scrollTop: 0 }, "fast");
120                 }
121             }
122         });
123     });
124 </script>
125

```

Como parte de las clases que se construyeron a fin de que se integre el sistema se realizó la programación de las siguientes pantallas: conexión a base de datos, login master, master, configuración del sistema, usuarios master.

```

DBConnection.php
1 <?php
2 // Verifica si la constante DB_SERVER está definida
3 if (!defined('DB_SERVER')) {
4     // Si no está definida, incluye el archivo "initialize.php" que debería definirla
5     require_once("../initialize.php");
6 }
7
8 class DBConnection {
9     // Define las propiedades para la conexión a la base de datos
10    //private string $host = "localhost";
11    //private string $username = "skiol5ax@localhost";
12    private string $password = "admin123";
13    private string $database = "skiol5ax_repositoriootesis";
14
15    private $host = DB_SERVER;
16    private $username = DB_USERNAME;
17    private $password = DB_PASSWORD;
18    private $database = DB_NAME;
19
20    public $conn; // Variable para la conexión
21
22    // Constructor de la clase
23    public function __construct() {
24        // Verifica si la conexión no está establecida
25        if (!isset($this->conn)) {
26            // Intenta establecer la conexión
27            $this->conn = new mysqli($this->host, $this->username, $this->password, $this->database);
28
29            // Verifica si la conexión no pudo establecerse
30            if ($this->conn->connect_error) {
31                // Muestra un mensaje de error y termina el script
32                echo 'No se puede conectar al servidor de la base de datos';
33                exit;
34            }
35        }
36    }
37
38    // Destructor de la clase, se encarga de cerrar la conexión
39    public function __destruct() {
40        // Cierra la conexión a la base de datos
41        $this->conn->close();
42    }
43 }
44
45

```

```

Login.php
1 <?php
2 // Incluye el archivo de configuración
3 require_once '../config.php';
4
5 class Login extends DBConnection {
6     private $settings;
7
8     // Constructor de la clase
9     public function __construct(){
10         global $ settings;
11         $this->settings = $ settings;
12
13         // Llama al constructor de la clase padre
14         parent::__construct();
15
16         // Configura la visualización de errores
17         ini_set('display_error', 1);
18     }
19
20     // Destructor de la clase
21     public function __destruct(){
22         // Llama al destructor de la clase padre
23         parent::__destruct();
24     }
25
26     // Método para mostrar un mensaje de acceso denegado
27     public function index(){
28         echo "<h1>Acceso denegado</h1> <a href='".$base_url.">Volver.</a>";
29     }
30
31     // Método para iniciar sesión de usuario
32     public function login(){
33         // Extrae los datos del formulario de inicio de sesión
34         extract($_POST);
35
36         // Consulta a la base de datos para verificar las credenciales del usuario
37         $qry = $this->conn->query("SELECT * FROM users WHERE username = '$username' AND password =
38             md5('$password')");
39
40         // Verifica si se encontró al menos un usuario con las credenciales proporcionadas
41         if($qry->num_rows > 0){
42             $res = $qry->fetch_array();
43
44             // Verifica si el estado del usuario es activo
45             if($res['status'] != 1){
46                 return json_encode(array('status'=>'notverified'));
47             }
48
49             // Almacena los datos del usuario en la sesión
50             foreach($res as $k => $v){
51                 if(!is_numeric($k) && $k != 'password'){
52                     $this->settings->set_userdata($k,$v);
53                 }
54             }
55
56             // Establece el tipo de inicio de sesión
57             $this->settings->set_userdata('login_type',1);
58
59             return json_encode(array('status'=>'success'));
60         } else {
61             // Si las credenciales son incorrectas, devuelve un mensaje de error
62             return json_encode(array('status'=>'incorrect','last_qry'=>"SELECT * FROM users WHERE
63                 username = '$username' AND password = md5('$password')"));
64         }
65     }
66
67     // Método para cerrar sesión de usuario
68     public function logout(){
69         // Cierra la sesión y redirige al usuario a la página de inicio de sesión
70         if($this->settings->sess_des()){
71             redirect('admin/login.php');
72         }
73     }
74
75     // Método para iniciar sesión de estudiante
76     public function student_login(){
77         // Extrae los datos del formulario de inicio de sesión de estudiante
78         extract($_POST);
79
80         // Consulta a la base de datos para verificar las credenciales del estudiante

```

```

77
78 // Consulta a la base de datos para verificar las credenciales del estudiante
79 $qry = $this->conn->query("SELECT *, CONCAT(lastname,', ',firstname,', ',middlename) AS
    fullname FROM student_list WHERE email = '$email' AND `password` = md5('$password') ");
80
81 // Verifica si se encontró al menos un estudiante con las credenciales proporcionadas
82 if($qry->num_rows > 0){
83     $res = $qry->fetch_array();
84
85     // Verifica si la cuenta del estudiante está verificada
86     if($res['status'] == 1){
87         foreach($res as $k => $v){
88             $this->settings->set_userdata($k,$v);
89         }
90         // Establece el tipo de inicio de sesión
91         $this->settings->set_userdata('login_type',2);
92         $resp['status'] = 'success';
93     } else {
94         // Devuelve un mensaje de error si la cuenta del estudiante no está verificada
95         $resp['status'] = 'failed';
96         $resp['msg'] = "Tu cuenta no se ha verificado aún.";
97     }
98 } else {
99     // Devuelve un mensaje de error si las credenciales son incorrectas
100    $resp['status'] = 'failed';
101    $resp['msg'] = "Correo electrónico o contraseña no válidos.";
102 }
103
104 return json_encode($resp);
105 }
106
107 // Método para cerrar sesión de estudiante
108 public function student_logout(){
109     // Cierra la sesión del estudiante y redirige a la página principal
110     if($this->settings->sess_des()){
111         redirect('/');
112     }
113 }
114 }
115
116 // Crea una instancia de la clase Login

```

```

115
116 // Crea una instancia de la clase Login
117 $auth = new Login();
118
119 // Determina la acción a realizar según el parámetro 'f' en la URL
120 $action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
121
122 // Ejecuta la acción correspondiente
123 switch ($action) {
124     case 'login':
125         echo $auth->login();
126         break;
127     case 'logout':
128         echo $auth->logout();
129         break;
130     case 'student_login':
131         echo $auth->student_login();
132         break;
133     case 'student_logout':
134         echo $auth->student_logout();
135         break;
136     default:
137         echo $auth->index();
138         break;
139 }
140 ?>

```

```

Master.php
1 <?php
2 // Incluye el archivo de configuración
3 require_once('../config.php');
4
5 class Master extends DBConnection {
6     private $settings;
7
8     // Constructor de la clase
9     public function __construct(){
10         global $settings;
11         $this->settings = $settings;
12         parent::__construct();
13     }
14
15     // Destructor de la clase
16     public function __destruct(){
17         parent::__destruct();
18     }
19
20     // Método para capturar errores de la conexión
21     function capture_err(){
22         if(!$this->conn->error)
23             return false;
24         else{
25             $resp['status'] = 'failed';
26             $resp['error'] = $this->conn->error;
27             return json_encode($resp);
28             exit;
29         }
30     }
31
32     // Método para guardar un departamento
33     function save_department(){
34         extract($_POST);
35         $data = "";
36         foreach($_POST as $k => $v){
37             if(!in_array($k, array('id'))){
38                 if(!is_numeric($v))
39                     $v = $this->conn->real_escape_string($v);
40                 if(empty($data)) $data .= " ";
41                 $data .= "{'$k'}='{$v}' ";
42             }
43         }
44         if(empty($id)){
45             $sql = "INSERT INTO `department_list` SET {$data} ";
46         }else{
47             $sql = "UPDATE `department_list` SET {$data} WHERE id = '{$id}' ";
48         }
49         $check = $this->conn->query("SELECT * FROM `department_list` WHERE `name`='{$name}' ".($id
50             > 0 ? " AND id != '{$id}'" : ""))>num_rows;
51         if($check > 0){
52             $resp['status'] = 'failed';
53             $resp['msg'] = "El nombre del departamento ya existe.";
54         }else{
55             $save = $this->conn->query($sql);
56             if($save){
57                 $rid = empty($id) ? $id : $this->conn->insert_id;
58                 $resp['status'] = 'success';
59                 if(empty($id))
60                     $resp['msg'] = "Detalles del departamento agregados con éxito.";
61                 else
62                     $resp['msg'] = "Los detalles del departamento se han actualizado con éxito.";
63             }else{
64                 $resp['status'] = 'failed';
65                 $resp['msg'] = "Ocurrió un error.";
66                 $resp['err'] = $this->conn->error."[{$sql}]";
67             }
68         }
69         if($resp['status'] == 'success')
70             $this->settings->set_flashdata('success',$resp['msg']);
71         return json_encode($resp);
72     }
73
74     // Método para eliminar un departamento
75     function delete_department(){
76         extract($_POST);
77         $del = $this->conn->query("DELETE FROM `department_list` WHERE id = '{$id}'");
78         if($del){
79             $resp['status'] = 'success';
80             $this->settings->set_flashdata('success',"El departamento ha sido eliminado con éxito.");
81         }
82     }
83 }

```

```

80         }else{
81             $resp['status'] = 'failed';
82             $resp['error'] = $this->conn->error;
83         }
84         return json_encode($resp);
85     }
86
87     // Método para guardar un plan de estudios
88     function save_curriculum(){
89         extract($_POST);
90         $data = "";
91         foreach($_POST as $k => $v){
92             if(!in_array($k, array('id'))){
93                 if(!is_numeric($v)){
94                     $v = $this->conn->real_escape_string($v);
95                     if(!empty($data)) $data .= ",";
96                     $data .= " { '$k' : '$v' } ";
97                 }
98             }
99             if(empty($id)){
100                 $sql = "INSERT INTO `curriculum_list` SET {$data} ";
101             }else{
102                 $sql = "UPDATE `curriculum_list` SET {$data} WHERE id = '{$id}' ";
103             }
104             $check = $this->conn->query("SELECT * FROM `curriculum_list` WHERE `name`='{$name}' AND `department_id` = '{$department_id}' " . ($id > 0 ? " AND id != '{$id}'" : "") . ">num_rows;");
105             if($check > 0){
106                 $resp['status'] = 'failed';
107                 $resp['msg'] = "El nombre de la materia ya existe.";
108             }else{
109                 $save = $this->conn->query($sql);
110                 if($save){
111                     $rid = empty($id) ? $id : $this->conn->insert_id;
112                     $resp['status'] = 'success';
113                     if(empty($id))
114                         $resp['msg'] = "Detalles de la materia agregados con éxito.";
115                     else
116                         $resp['msg'] = "Los detalles de la materia se han actualizado con éxito.";
117                 }else{
118                     $resp['status'] = 'failed';
119                     $resp['msg'] = "Ocurrió un error.";

```

```

121     }
122     if($resp['status'] == 'success')
123         $this->settings->set_flashdata('success', $resp['msg']);
124     return json_encode($resp);
125 }
126
127 // Método para eliminar un plan de estudios
128 function delete_curriculum(){
129     extract($_POST);
130     $del = $this->conn->query("DELETE FROM `curriculum_list` WHERE id = '{$id}'");
131     if($del){
132         $resp['status'] = 'success';
133         $this->settings->set_flashdata('success', "La materia se ha eliminado correctamente.");
134     }else{
135         $resp['status'] = 'failed';
136         $resp['error'] = $this->conn->error;
137     }
138     return json_encode($resp);
139 }
140
141 // Método para guardar un archivo
142 function save_archive(){
143     if(empty($_POST['id'])){
144         $pref = date("Ym");
145         $code = sprintf("%'04d", 1);
146         while(true){
147             $check = $this->conn->query("SELECT * FROM `archive_list` WHERE archive_code = '{$pref}{$code}'")->num_rows;
148             if($check > 0){
149                 $code = sprintf("%'04d", abs($code)+1);
150             }else{
151                 break;
152             }
153         }
154         $_POST['archive_code'] = $pref.$code;
155         $_POST['student_id'] = $this->settings->userdata('id');
156         $_POST['curriculum_id'] = $this->settings->userdata('curriculum_id');
157     }
158     if(isset($_POST['abstract']))
159         $_POST['abstract'] = htmlentities($_POST['abstract']);
160

```

```

159         if(isset($_POST['abstract']))
160             $_POST['abstract'] = htmlentities($_POST['abstract']);
161         if(isset($_POST['members']))
162             $_POST['members'] = htmlentities($_POST['members']);
163         extract($_POST);
164         $data = "";
165         if(isset($_FILES['pdf']) && !empty($_FILES['pdf']['tmp_name'])) {
166             $type = mime_content_type($_FILES['pdf']['tmp_name']);
167             if($type != "application/pdf") {
168                 $resp['status'] = "failed";
169                 $resp['msg'] = "Tipo de archivo de documento";
170                 $resp['msg'] = "Tipo de archivo de documento no válido.";
171                 return json_encode($resp);
172             }
173         }
174         foreach($_POST as $k => $v) {
175             if(!in_array($k, array('id')) && !is_array($_POST[$k])) {
176                 if(!is_numeric($v))
177                     $v = $this->conn->real_escape_string($v);
178                 if(!empty($data)) $data .= ",";
179                 $data .= "{'$k'='$v' }";
180             }
181         }
182         if(empty($id)) {
183             $sql = "INSERT INTO `archive_list` SET {$data} ";
184         } else {
185             $sql = "UPDATE `archive_list` SET {$data} WHERE id = '{$id}' ";
186         }
187         $save = $this->conn->query($sql);
188         if($save) {
189             $aid = empty($id) ? $id : $this->conn->insert_id;
190             $resp['status'] = 'success';
191             $resp['id'] = $aid;
192             if(empty($id))
193                 $resp['msg'] = "El archivo se envió con éxito";
194             else
195                 $resp['msg'] = "Los detalles del archivo se actualizaron correctamente.";
196
197             if(isset($_FILES['img']) && $_FILES['img']['tmp_name'] != '') {
198                 $fname = 'uploads/banners/archive-'. $aid .'.png';
199                 $dir_path = base_app . $fname;
200
201                 $fname = 'uploads/banners/archive-'. $aid .'.png';
202                 $dir_path = base_app . $fname;
203                 $upload = $_FILES['img']['tmp_name'];
204                 $type = mime_content_type($upload);
205                 $allowed = array('image/png', 'image/jpeg');
206                 if(!in_array($type, $allowed)) {
207                     $resp['msg'] = " Pero la imagen no se pudo cargar debido a un tipo de archivo no válido.";
208                 } else {
209                     $new_height = 720;
210                     $new_width = 1280;
211
212                     list($width, $height) = getimagesize($upload);
213                     $t_image = imagecreatetruecolor($new_width, $new_height);
214                     imagealphablending($t_image, false);
215                     imagesavealpha($t_image, true);
216                     $gdImg = ($type == 'image/png') ? imagecreatefrompng($upload) :
217                         imagecreatefromjpeg($upload);
218                     imagecopyresampled($t_image, $gdImg, 0, 0, 0, 0, $new_width, $new_height, $width, $height);
219                     if($gdImg) {
220                         if(is_file($dir_path))
221                             unlink($dir_path);
222                         $uploaded_img = imagepng($t_image, $dir_path);
223                         imagedestroy($gdImg);
224                         imagedestroy($t_image);
225                     } else {
226                         $resp['msg'] = " Pero la imagen no se pudo cargar debido a un motivo desconocido.";
227                     }
228                 }
229             }
230             if(isset($uploaded_img)) {
231                 $this->conn->query("UPDATE archive_list SET `banner_path` = CONCAT('{$fname}',
232                     '?v=', unix_timestamp(CURRENT_TIMESTAMP)) WHERE id = '{$aid}' ");
233             }
234         }
235         if(isset($_FILES['pdf']) && $_FILES['pdf']['tmp_name'] != '') {
236             $fname = 'uploads/pdf/archive-'. $aid .'.pdf';
237             $dir_path = base_app . $fname;
238             $upload = $_FILES['pdf']['tmp_name'];
239             $type = mime_content_type($upload);

```

```

232         $upload = $_FILES['pdf']['tmp_name'];
233         $type = mime_content_type($upload);
234         $allowed = array('application/pdf');
235         if(!in_array($type,$allowed)){
236             $resp['msg'] .= " Pero el archivo del documento no se pudo cargar debido a un
                tipo de archivo no válido.";
237         }else{
238             $uploaded = move_uploaded_file($_FILES['pdf']['tmp_name'],$dir_path);
239         }
240         if(isset($uploaded)){
241             $this->conn->query("UPDATE archive_list SET `document_path` = CONCAT('{$_fname}'
                ,'?v=',unix_timestamp(CURRENT_TIMESTAMP)) WHERE id = '{$_aid}' ");
242         }
243     }else{
244         $resp['status'] = 'failed';
245         $resp['msg'] = "Ocurrió un error.";
246         $resp['err'] = $this->conn->error."[{$sql}]";
247     }
248 }
249 if($resp['status'] == 'success')
250     $this->settings->set_flashdata('success',$resp['msg']);
251 return json_encode($resp);
252 }
253
254 // Método para eliminar un archivo
255 function delete_archive(){
256     extract($_POST);
257     $get = $this->conn->query("SELECT * FROM `archive_list` WHERE id = '{$_id}'");
258     $del = $this->conn->query("DELETE FROM `archive_list` WHERE id = '{$_id}'");
259     if($del){
260         $resp['status'] = 'success';
261         $this->settings->set_flashdata('success',"El archivo de registros se eliminó con éxito.
                ");
262         if($get->num_rows > 0){
263             $res = $get->fetch_array();
264             $banner_path = explode("?", $res['banner_path'])[0];
265             $document_path = explode("?", $res['document_path'])[0];
266             if(is_file(base_app.$banner_path))
267                 unlink(base_app.$banner_path);
268             if(is_file(base_app.$document_path))
269                 unlink(base_app.$document_path);

```

```

269             if(is_file(base_app.$document_path))
270                 unlink(base_app.$document_path);
271         }else{
272             $resp['status'] = 'failed';
273             $resp['error'] = $this->conn->error;
274         }
275         return json_encode($resp);
276     }
277
278 // Método para actualizar el estado de un archivo
279 function update_status(){
280     extract($_POST);
281     $sql = "UPDATE `archive_list` SET status = '{$_status}' WHERE id = '{$_id}'";
282     $update = $this->conn->query($sql);
283     if($update){
284         $resp['status'] = 'success';
285         $resp['msg'] = "El estado del archivo se ha actualizado correctamente.";
286     }else{
287         $resp['status'] = 'failed';
288         $resp['msg'] = "Ocurrió un error. Error: " . $this->conn->error;
289     }
290
291     if($resp['status'] == 'success')
292         $this->settings->set_flashdata('success',$resp['msg']);
293     return json_encode($resp);
294 }
295
296 $Master = new Master();
297 $action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
298 $sysset = new SystemSettings();
299 switch ($action) {
300     case 'save_department':
301         echo $Master->save_department();
302         break;
303     case 'delete_department':
304         echo $Master->delete_department();
305         break;
306     case 'save_curriculum':
307         echo $Master->save_curriculum();
308

```

```

308         echo $Master->delete_department();
309         break;
310     case 'save_curriculum':
311         echo $Master->save_curriculum();
312         break;
313     case 'delete_curriculum':
314         echo $Master->delete_curriculum();
315         break;
316     case 'save_archive':
317         echo $Master->save_archive();
318         break;
319     case 'delete_archive':
320         echo $Master->delete_archive();
321         break;
322     case 'update_status':
323         echo $Master->update_status();
324         break;
325 }

```

```

1 <?php
2 // Verifica si la clase DBConnection ya está definida, de lo contrario, incluye los archivos
  necesarios
3 if (!class_exists('DBConnection')) {
4     require_once('../config.php');
5     require_once('DBConnection.php');
6 }
7
8 class SystemSettings extends DBConnection
9 {
10     public function __construct()
11     {
12         parent::__construct();
13     }
14
15     // Función para verificar la conexión a la base de datos
16     public function check_connection()
17     {
18         return ($this->conn);
19     }
20
21     // Función para cargar la información del sistema en la sesión
22     public function load_system_info()
23     {
24         // Verifica si la información del sistema ya está cargada en la sesión
25         // if (isset($_SESSION['system_info'])) {
26         $sql = "SELECT * FROM system_info";
27         $qry = $this->conn->query($sql);
28         while ($row = $qry->fetch_assoc()) {
29             $_SESSION['system_info'][$row['meta_field']] = $row['meta_value'];
30         }
31         // }
32     }
33
34     // Función para actualizar la información del sistema en la sesión
35     public function update_system_info()
36     {
37         $sql = "SELECT * FROM system_info";
38         $qry = $this->conn->query($sql);
39         while ($row = $qry->fetch_assoc()) {
40             if (isset($_SESSION['system_info'][$row['meta_field']])) unset($_SESSION['system_info'][$row['meta_field']]);
41             $_SESSION['system_info'][$row['meta_field']] = $row['meta_value'];
42         }
43         return true;
44     }
45
46     // Función para actualizar la información de configuración del sistema
47     public function update_settings_info()
48     {
49         // Actualizar valores en la base de datos según los datos enviados por el formulario
50         // Opcional: Guardar contenido HTML en archivos
51
52         // Actualizar logo si se envía un archivo
53         // Actualizar portada si se envía un archivo
54
55         $update = $this->update_system_info();
56         $flash = $this->set_flashdata('success', 'Información del sistema actualizada con éxito.');

```

```

83
84 // Función para establecer un mensaje flash en la sesión
85 public function set_flashdata($flash = '', $value = '')
86 {
87     if (empty($flash) && empty($value)) {
88         $_SESSION['flashdata'][$flash] = $value;
89         return true;
90     }
91 }
92
93 // Función para verificar si existe un mensaje flash en la sesión
94 public function chk_flashdata($flash = '')
95 {
96     if (isset($_SESSION['flashdata'][$flash])) {
97         return true;
98     } else {
99         return false;
100     }
101 }
102
103 // Función para obtener y eliminar un mensaje flash de la sesión
104 public function flashdata($flash = '')
105 {
106     if (empty($flash)) {
107         $_tmp = $_SESSION['flashdata'][$flash];
108         unset($_SESSION['flashdata'][$flash]);
109         return $_tmp;
110     } else {
111         return false;
112     }
113 }
114
115 // Función para destruir los datos de usuario de la sesión
116 public function sess_des()
117 {
118     if (isset($_SESSION['userdata'])) {
119         unset($_SESSION['userdata']);
120         return true;
121     }
122     return true;
123 }

```

```

124
125 // Función para obtener información del sistema
126 public function info($field = '')
127 {
128     if (empty($field)) {
129         if (isset($_SESSION['system_info'][$field]))
130             return $_SESSION['system_info'][$field];
131         else
132             return false;
133     } else {
134         return false;
135     }
136 }
137
138 // Función para establecer información del sistema
139 public function set_info($field = '', $value = '')
140 {
141     if (empty($field) && empty($value)) {
142         $_SESSION['system_info'][$field] = $value;
143     }
144 }
145
146 // Crear una instancia de la clase SystemSettings
147 $settings = new SystemSettings();
148 $settings->load_system_info();
149
150 // Verificar la acción solicitada y llamar a la función correspondiente
151 $action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
152 $sysset = new SystemSettings();
153
154 switch ($action) {
155     case 'update_settings':
156         echo $sysset->update_settings_info();
157         break;
158     default:
159         // Manejar el caso predeterminado aquí, si es necesario
160         break;
161 }
162 }
163 >>

```

```

1 <?php
2 require_once('../config.php');
3
4 class Users extends DBConnection {
5     private $settings;
6
7     public function __construct(){
8         global $_settings;
9         $this->settings = $_settings;
10        parent::__construct();
11    }
12
13    public function __destruct(){
14        parent::__destruct();
15    }
16
17    public function save_users(){
18        if(!isset($_POST['status']) && $this->settings->userdata('login_type') == 1){
19            $_POST['status'] = 1;
20        }
21
22        extract($_POST);
23        $id = $id;
24        $data = '';
25
26        if(isset($oldpassword)){
27            if(md5($oldpassword) != $this->settings->userdata('password')){
28                return 4;
29            }
30        }
31
32        $chk = $this->conn->query("SELECT * FROM `users` where username='{$username}' ".($id>0? "
33            and id= '{$id}' " : ""))->num_rows;
34
35        if($chk > 0){
36            return 3;
37            exit;
38        }
39
40        foreach($_POST as $k => $v){
41            if(in_array($k,array('firstname','middlename','lastname','username','type'))){
42                if(!empty($data)) $data .= " , ";
43                $data .= "{$k} = '{$v}' ";
44            }
45        }
46
47        if(!empty($password)){
48            $password = md5($password);
49            if(!empty($data)) $data .= " , ";
50            $data .= " password = MD5('{$password}') ";
51        }
52
53        if(empty($id)){
54            $qry = $this->conn->query("INSERT INTO users set {$data}");
55            if($qry){
56                $id = $this->conn->insert_id;
57                $this->settings->set_flashdata('success','Detalles del usuario guardados con éxito.
58                ');
59                $resp['status'] = 1;
60            }else{
61                $resp['status'] = 2;
62            }
63        }else{
64            $qry = $this->conn->query("UPDATE users set {$data} where id = '{$id}'");
65            if($qry){
66                $this->settings->set_flashdata('success','Detalles del usuario actualizados con
67                éxito. ');
68                if($id == $this->settings->userdata('id')){
69                    foreach($_POST as $k => $v){
70                        if($k != 'id'){
71                            if(!empty($data)) $data .= " , ";
72                            $this->settings->set_userdata($k,$v);
73                        }
74                    }
75                }
76                $resp['status'] = 1;
77            }else{
78                $resp['status'] = 2;
79            }
80        }
81    }
82 }

```

```

79         if(isset($FILES['img']) && $FILES['img']['tmp_name'] != ''){
80             $fname = 'uploads/avatar-'. $id.'.png';
81             $dir_path = base_app. $fname;
82             $upload = $FILES['img']['tmp_name'];
83             $type = mime_content_type($upload);
84             $allowed = array('image/png','image/jpeg');
85             if(!in_array($type,$allowed)){
86                 $resp['msg'] = " Pero la imagen no se pudo cargar debido a un tipo de archivo no
87                 válido.";
88             }else{
89                 $new_height = 200;
90                 $new_width = 200;
91
92                 List($width, $height) = getimagesize($upload);
93                 $t_image = imagecreatetruecolor($new_width, $new_height);
94                 imagealphablending( $t_image, false );
95                 imagesavealpha( $t_image, true );
96                 $gdimg = ($type == 'image/png')? imagecreatefrompng($upload) : imagecreatefromjpeg(
97                     $upload);
98                 imagecopyresampled($t_image, $gdimg, 0, 0, 0, 0, $new_width, $new_height, $width, $
99                     height);
100                 if($gdimg){
101                     if(is_file($dir_path))
102                         unlink($dir_path);
103                     $uploaded_img = imagepng($t_image,$dir_path);
104                     imagedestroy($gdimg);
105                     imagedestroy($t_image);
106                 }else{
107                     $resp['msg'] = " Pero la imagen no se pudo cargar debido a un motivo desconocido.";
108                 }
109             }
110             if(isset($uploaded_img)){
111                 $this->conn->query("UPDATE users set `avatar` = CONCAT('{ $fname}', '?v=',
112                     unix_timestamp(CURRENT_TIMESTAMP)) where id = '{ $id}' ");
113                 if($id == $this->settings->userdata('id')){
114                     $this->settings->set_userdata('avatar', $fname);
115                 }
116             }
117             if(isset($resp['msg']))
118                 $this->settings->set_flashdata('success', $resp['msg']);
119
120         }
121
122         public function delete_users(){
123             extract($_POST);
124             $avatar = $this->conn->query("SELECT avatar FROM users where id = '{ $id}'")->fetch_array()[
125                 'avatar'];
126             $qry = $this->conn->query("DELETE FROM users where id = $id");
127             if($qry){
128                 $avatar = explode("?", $avatar)[0];
129                 $this->settings->set_flashdata('success', 'Detalles de usuario eliminados con éxito.');
```

```

150     }
151
152     foreach($_POST as $k => $v){
153         if(!in_array($k,array('id','oldpassword','cpassword','password'))){
154             if(!empty($data)) $data .= " , ";
155             $data .= " {$k} = '{$v}' ";
156         }
157     }
158
159     if(!empty($password)){
160         $password = md5($password);
161         if(!empty($data)) $data .= " , ";
162         $data .= " `password` = '{$password}' ";
163     }
164
165     if(empty($id)){
166         $qry = $this->conn->query("INSERT INTO student_list set {$data}");
167         if($qry){
168             $id = $this->conn->insert_id;
169             $this->settings->set_flashdata('success','Los detalles del usuario del estudiante
170             se guardaron con éxito.');
```

```

180         }else{
181             $resp['status'] = "failed";
182             $resp['msg'] = "Ocurrió un error al guardar los datos. Error: ". $this->conn->error
183             ;
184         }
185     }
186
187     }else{
188         $qry = $this->conn->query("UPDATE student_list set {$data} where id = {$id}");
189         if($qry){
190             $this->settings->set_flashdata('success','Detalles de usuario del estudiante
191             actualizados con éxito.');
```

```

221         imagejpeg($img,$image);
222     }else{
223         $resp['msg'] .= " Pero la imagen no se pudo cargar debido a un motivo desconocido.";
224     }
225 }
226 if(isset($uploaded_img)){
227     $this->conn->query("UPDATE student_list set `avatar` = CONCAT('{{ $fname }}', '?v=',
228         unix_timestamp(CURRENT_TIMESTAMP)) where id = '{{ $id }}' ");
229     if($id == $this->settings->userdata('id')){
230         $this->settings->set_userdata('avatar',$fname);
231     }
232 }
233 }
234 return json_encode($resp);
235 }
236
237 public function delete_student(){
238     extract($_POST);
239     $avatar = $this->conn->query("SELECT avatar FROM student_list where id = '{{ $id }}'")->
240         fetch_array()['avatar'];
241     $qry = $this->conn->query("DELETE FROM student_list where id = $id");
242     if($qry){
243         $avatar = explode("?", $avatar)[0];
244         $this->settings->set_flashdata('success', 'Detalles de usuario del estudiante
245             eliminados con éxito. ');
246         if(is_file(base_app, $avatar))
247             unlink(base_app, $avatar);
248         $resp['status'] = 'success';
249     }else{
250         $resp['status'] = 'failed';
251     }
252     return json_encode($resp);
253 }
254
255 public function verify_student(){
256     extract($_POST);
257     $update = $this->conn->query("UPDATE `student_list` set `status` = 1 where id = $id");
258     if($update){
259         $this->settings->set_flashdata('success', 'La cuenta de estudiante se ha verificado con
260             éxito. ');
261         $resp['status'] = 'success';
262     }else{
263         $resp['status'] = 'failed';
264     }
265     return json_encode($resp);
266 }
267
268 $users = new Users();
269 $action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
270 switch ($action) {
271     case 'save':
272         echo $users->save_users();
273         break;
274     case 'delete':
275         echo $users->delete_users();
276         break;
277     case 'save_student':
278         echo $users->save_student();
279         break;
280     case 'delete_student':
281         echo $users->delete_student();
282         break;
283     case 'verify_student':
284         echo $users->verify_student();
285         break;
286     default:
287         // echo $sysset->index();
288         break;
289 }
290 }
291 }

```

Adicionalmente, se creó la página de error 404, a fin de que si se presenta algún inconveniente con el sistema este despliegue la ventana de error.

```

404.html
1 <div class="error-content">
2 <h3><i class="fas fa-exclamation-triangle text-danger"></i> ¡Ups! Página no encontrada.</h3>
3
4 <p>
5     No pudimos encontrar la página que estabas buscando.
6 </p>
7
8 </div>

```

```

index.php
1  <?php
2
3  ini_set('display_errors', '1');
4  ini_set('display_startup_errors', '1');
5  error_reporting(E_ALL);
6
7  require_once('./config.php');
8
9  ?>
10 <!DOCTYPE html>
11 <html lang="es" style="height: auto;">
12 <head>
13     <meta charset="UTF-8">
14     <meta name="viewport" content="width=device-width, initial-scale=1.0">
15     <title>Bienvenido</title>
16     <style>
17         #header {
18             height: 70vh;
19             width: 100%;
20             position: relative;
21             top: -1em;
22         }
23         #header:before {
24             content: "";
25             position: absolute;
26             height: 100%;
27             width: 100%;
28             background-image: url(<? echo (validate_image($_settings->info("cover")) ?>);
29             background-size: cover;
30             background-repeat: no-repeat;

```

```

index.php
1  <?php
2
3  ini_set('display_errors', '1');
4  ini_set('display_startup_errors', '1');
5  error_reporting(E_ALL);
6
7  require_once('./config.php');
8
9  ?>
10 <!DOCTYPE html>
11 <html lang="es" style="height: auto;">
12 <head>
13     <meta charset="UTF-8">
14     <meta name="viewport" content="width=device-width, initial-scale=1.0">
15     <title>Bienvenido</title>
16     <style>
17         #header {
18             height: 70vh;
19             width: 100%;
20             position: relative;
21             top: -1em;
22         }
23         #header:before {
24             content: "";
25             position: absolute;
26             height: 100%;
27             width: 100%;
28             background-image: url(<? echo (validate_image($_settings->info("cover")) ?>);
29             background-size: cover;
30             background-repeat: no-repeat;

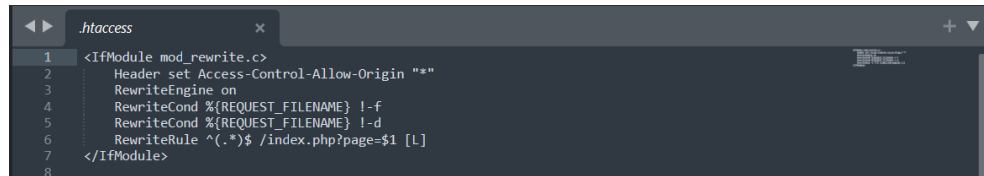
```

```

404.html
1  <div class="error-content">
2      <h3><i class="fas fa-exclamation-triangle text-danger"></i> ¡Ups! Página no encontrada.</h3>
3
4      <p>
5          No pudimos encontrar la página que estabas buscando.
6          Mientras tanto, puedes <a href=".">volver al panel de control</a> o intentar usar el
7          formulario de búsqueda.
8      </p>
9  </div>

```

Es importante mencionar que se creó la página htaccess, a fin de que permita que se modifique la configuración del sistema en directorios individuales sin afectar la configuración global.

A screenshot of a code editor showing the contents of a .htaccess file. The editor has a dark theme and a tab labeled '.htaccess'. The code is as follows:

```
1 <IfModule mod_rewrite.c>
2   Header set Access-Control-Allow-Origin "*"
3   RewriteEngine on
4   RewriteCond %{REQUEST_FILENAME} !-f
5   RewriteCond %{REQUEST_FILENAME} !-d
6   RewriteRule ^(.*)$ /index.php?page=$1 [L]
7 </IfModule>
8
```

3.4 Medio de publicación del repositorio

La implementación del repositorio digital se realizará en la plataforma c-panel con el propósito de almacenar las tesis y trabajos de titulación de sus estudiantes. Este proyecto, tiene como objetivo principal facilitar el acceso y la preservación de los trabajos académicos producidos por la comunidad estudiantil.

3.4.1 Detalles del Repositorio Digital:

- **Contenido:** El repositorio albergará proyectos de titulación y trabajos de los estudiantes en formato PDF, acompañados de extractos y resúmenes para una rápida revisión.

3.4.2. Ventajas de c-panel como Plataforma:

- **Accesibilidad:** c-panel ofrece una interfaz amigable y accesible para el personal administrativo del Instituto, facilitando la navegación y búsqueda de contenido.
- **Gestión de Contenido:** La plataforma permite organizar y estructurar los archivos de manera eficiente, garantizando una fácil localización de las tesis y trabajos almacenados.
- **Interactividad:** c-panel brinda opciones interactivas que promueven la participación y colaboración en la comunidad educativa.

3.4.3. Acceso al Repositorio:

- **Usuarios:** El personal administrativo del Instituto Tecnológico Superior "El Pacífico" tendrá acceso al repositorio para revisar y utilizar el contenido académico almacenado.
- **Credenciales:** Se requieren credenciales para cargar contenido en el repositorio, asegurando la integridad y seguridad de los trabajos académicos.

3.4.4. Manual de Usuario:

- Se proporcionará un manual de usuario junto con esta tesis, detallando los pasos para acceder y utilizar el repositorio de manera eficiente.

Con la implementación de este repositorio digital en c-panel se promueve la difusión y preservación de la producción académica de sus estudiantes.

EL REPOSITORIO EN IMÁGENES



Ilustración 12 Página de inicio

Página de inicio

Dentro de la página de inicio, se visualiza lo siguiente:

- Números de contacto
- Opciones de “Registro”, “Login Estudiante”, “Login Administrador”.
- Logo con nombre abreviado de Repositorio (REPO)
- Botón de “Inicio”
- Botón de “Proyectos de titulación”
- Botón de “Facultades”
- Botón de “Carreras”
- Botón de “Nosotros”
- Botón de Búsqueda (Lupa)
- Título de la página “Repositorio de trabajos de titulación y documentos”
- Botón “Explorar Proyectos”

A fin de comprender que acción realiza cada botón, se procederá con la explicación de la funcionalidad de cada botón.

▪ Botón de Inicio

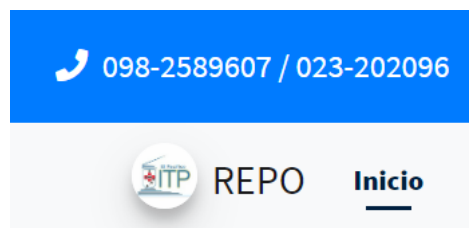


Ilustración 13 Botón de inicio

El botón de Inicio tiene como funcionalidad realizar la redirección hacia la página principal del repositorio si se encuentra en otra página y desea ir a la página principal de manera rápida.

▪ Botón Proyectos de titulación

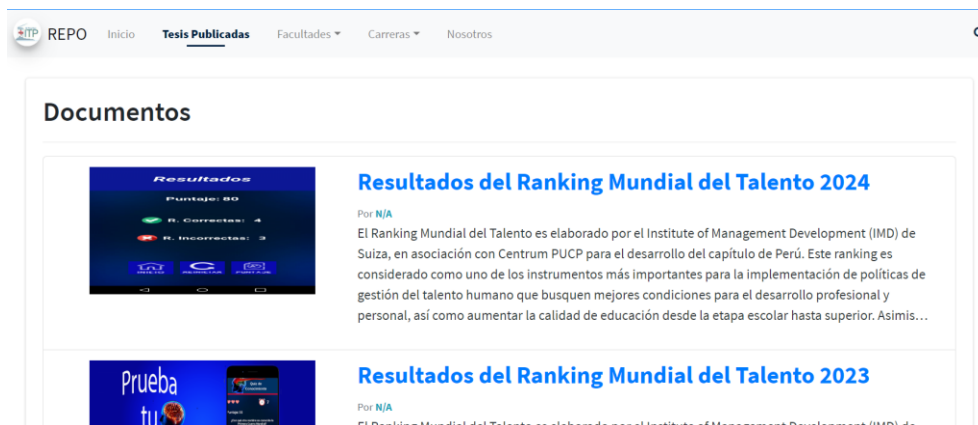


Ilustración 14 Botón Proyectos de titulación

El botón de Proyectos de titulación envía al usuario a la página que almacena los proyectos de titulación por los estudiantes para poder visualizarlas.

▪ Botón Facultades



Ilustración 15 Botón Facultades

El botón Facultades, muestra una lista de opciones donde se encuentran listadas todas las facultades que posee el Instituto, con la finalidad de realizar un filtro de proyectos de titulación por Facultad.

▪ Botón Carreras



Ilustración 16 Botón Carreras

El botón Carreras, muestra una lista de opciones donde se encuentran listadas todas las carreras que posee el Instituto, con la finalidad de realizar un filtro de proyectos de titulación por carrera.

▪ Botón Nosotros

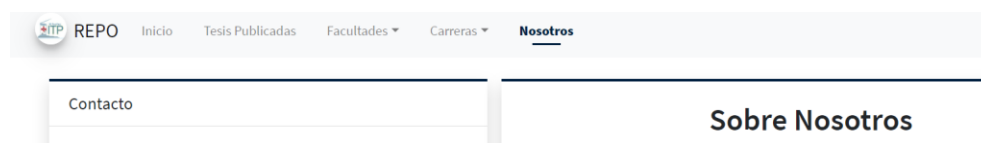


Ilustración 17 Botón Nosotros

El botón Nosotros, muestra la información general del Instituto, como dirección, números de contacto, dirección, horarios de atención y un mensaje del Instituto.

▪ Botón Explorar Proyectos

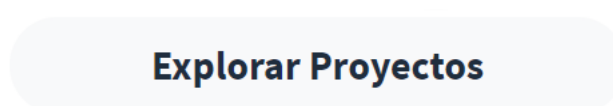


Ilustración 18 Botón Explorar Proyectos

El botón Explorar Proyectos, envía al usuario a la página que almacena los proyectos de titulación por los estudiantes para poder visualizarlas.

▪ Botón Registrarse



Ilustración 19 Botón Registrarse

El botón Registrarse, accede a un formulario para que el usuario se registre, este registro debe ser primero autorizado por el administrador, caso contrario, el usuario que se registra no podrá realizar ninguna acción.

■ Formulario de Registro

Nombre Segundo Nombre (opcional)

Apellido

☒ Masculino ☐ Femenino

Facultad
FACULTAD DE ADMINISTRACIÓN

Carrera
Seleccionar materia

Correo

Contraseña

Confirmar contraseña

Registrarse

Ilustración 20 Formulario de Registro

El Formulario de registro tiene los siguientes campos: Nombre, Segundo Nombre (opcional), Apellido, Sexo, Facultad a la que pertenece, Carrera a la que pertenece, correo electrónico, contraseña, confirmar contraseña y botón de Registrarse.

Cada campo debe ser llenado obligatoriamente, caso contrario se muestra un mensaje de error que le indica que está pendiente llenar un campo del formulario.

Nombre Segundo Nombre (opcional)

Completa este campo

☒ Masculino ☐ Femenino

Ilustración 21 Mensaje obligatoriedad

- Botón Login Estudiante

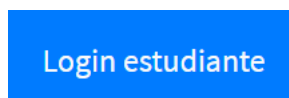


Ilustración 22 Botón login Estudiante

El botón de Login Estudiante, dirige al usuario (estudiante) a la página de login que le habilita ciertos accesos.

- Formulario de Login Estudiante

Ilustración 23 Formulario Login Estudiante

El formulario de login de estudiante solicita las credenciales antes ingresadas en el formulario de registro, una vez que se habilita el acceso del estudiante a la plataforma, este ingresa a login estudiante, ingresa su correo electrónico y contraseña e ingresa a la plataforma.

- Pantalla de usuario estudiante.



Ilustración 24 Pantalla de usuario estudiante

La pantalla de usuario estudiante tiene las siguientes opciones:

Página de inicio

Dentro de la página de inicio, se visualiza lo siguiente:

- Números de contacto
- Instituto Superior Tecnológico El Pacífico
- Botón de “Inicio”
- Botón de “Proyectos de titulación”
- Botón de “Facultades”
- Botón de “Carreras”
- Botón de “Nosotros”
- Botón de “Perfil”
- Botón de “Subir Archivos”
- Botón de Búsqueda (Lupa)
- Título de la página “Repositorio de archivos de Tesis/Proyectos”
- Botón “Explorar Proyectos”

Es importante mencionar que anteriormente se indicó las funcionalidades de las opciones Números de contacto, Logo con el nombre abreviado del repositorio (REPO), botones de Inicio, Proyectos de titulación, Carreras, Nosotros, búsqueda, Título de la página “Repositorio de archivos de Tesis/Proyectos”; y, botón Explorar Proyectos, por lo que se indicará el funcionamiento de los botones Perfil y Subir Archivos.

▪ Botón Perfil

Tu información:		Mis archivos	Actualizar cuenta
	Nombre del estudiante: Perez, Jhoel Sexo: Male Correo: joel@gmail.com Facultad: FACULTAD DE DISEÑO Carrera: Administración De Empresas		

Ilustración 25 Botón Perfil

El botón perfil, accede a la información almacenada por el usuario, presenta sus datos personales mismos que se obtienen del formulario de registro, en esta pantalla también se presentan 2 botones adicionales, Mis archivos y Actualizar cuenta.

■ Actualizar Detalles

Actualizar Detalles

Nombre	Segundo Nombre
Jhoel	Segundo nombre (opcional)

Apellido

Perez

Sexo ☒ Masculino ☐ Femenino

Correo

joel@gmail.com

Elija una imagen

Ningún archivo seleccionado

Ilustración 26 Actualizar detalles

La pantalla de Actualizar detalles presenta la siguiente información: Nombre, Segundo Nombre, Apellido, Sexo, Correo, Opción para cambiar la imagen y la opción para cambiar la contraseña.

Una vez se ha actualizado la información, el usuario da clic en Actualizar y la información cambia, afectando la base de datos.

■ Botón Mis archivos

Mis proyectos enviados

Show entries Search:

#	Fecha de creación	Código de archivo	Título de la Tesis	Carrera	Estado	Acción
1	2022-08-10 18:47	2022080001	Resultados Del Ranking Mundial Del Talento 2021	Administración de Empresas	Publicado	Acción ▼
2	2022-08-11 17:28	2022080002	Resultados Del Ranking De Competitividad Mundial 2022	Administración de Empresas	Publicado	Acción ▼
3	2022-08-15 19:08	2022080003	Resultados Del Ranking Mundial Del Talento 2023	Administración de Empresas	Publicado	Acción ▼

Showing 1 to 3 of 3 entries Previous **1** Next

Ilustración 27 Botón Mis Archivos

El botón mis archivos se presentan todos los proyectos de tesis o trabajos que se hayan cargado a la plataforma, estos trabajos estarán siempre con estado “pendiente”, hasta que el personal administrativo autorice la publicación del archivo. Una vez autorizado el proyecto

cambia de estado de pendiente a Publicado  y la tesis será visible.

▪ Botón de Acción



Ilustración 28 Botón Acción

El botón de acción permite ver el documento subido o eliminar el mismo en caso de que sea necesario.

▪ Botón de Logout

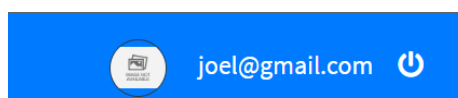


Ilustración 29 Botón Logout estudiante

El botón de logout permite al usuario, en este caso estudiante salir de la plataforma.

▪ Botón de Login admin

El botón de login de administrador permite a los administradores del repositorio ingresar a la plataforma y realizar las validaciones de los usuarios así como de las tesis o información general del sistema.



Ilustración 30 Login Administrador

Pantalla con formulario de LOGIN de administrador, esta pantalla solicita al administrador sus credenciales a fin de permitir el ingreso a la pantalla de administración.

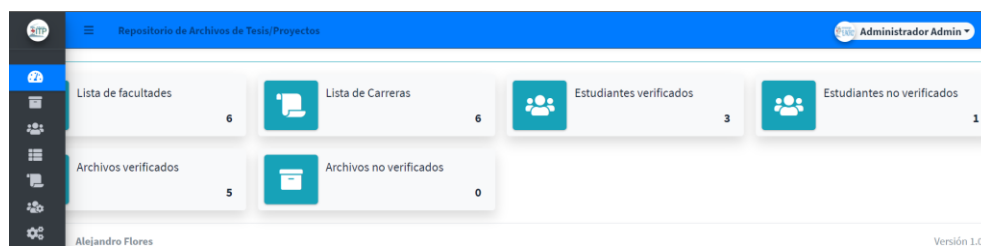


Ilustración 31 Pantalla de Administrador

Pantalla de administración, esta pantalla presenta un dashboard con cajones la información relacionada con usuarios, carreras; y, tesis. Esta pantalla ofrece lo siguiente:

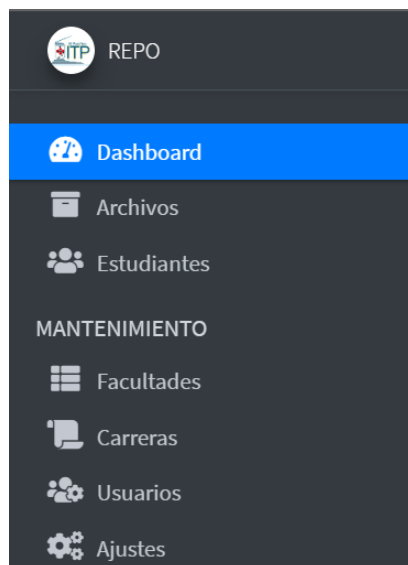


Ilustración 32 Dashboard

Un Dashboard que es la pantalla, un botón de archivos donde se pueden editar la información respecto de los proyectos de tesis subidos.

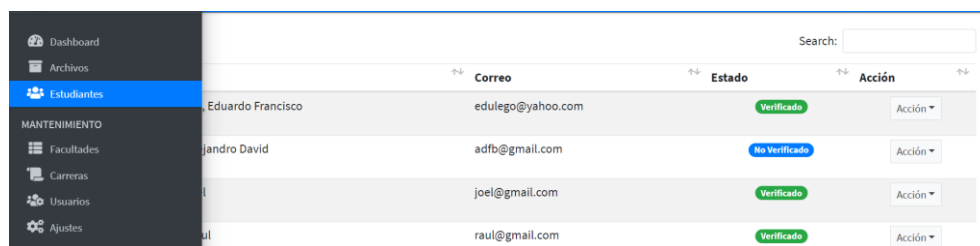
Fecha de creación	Código de archivo	Título del proyecto	Materia	Estado	Acción
2024-07-01 14:05	2024070001	Tesis1	Administración de Empresas	Publicado	Acción
2022-08-15 19:33	2022080004	Resultados Del Ranking Mundial Del Talento 2024	Diseño Gráfico	Publicado	Acción
2022-08-15 19:08	2022080003	Resultados Del Ranking Mundial Del Talento 2023	Administración de Empresas	Publicado	Acción
2022-08-11 17:28	2022080002	Resultados Del Ranking De Competitividad Mundial 2022	Administración de Empresas	Publicado	Acción
2022-08-10 18:47	2022080001	Resultados Del Ranking Mundial Del Talento 2021	Administración de Empresas	Publicado	Acción

Ilustración 33 Archivos administrador

Fecha de creación	Código de archivo	Título del proyecto	Materia	Estado	Acción
2024-07-01 14:05	2024070001	Tesis1	Administración de Empresas	Publicado	Acción
2022-08-15 19:33	2022080004	Resultados Del Ranking Mundial Del Talento 2024	Diseño Gráfico	Publicado	Ver
2022-08-15 19:08	2022080003	Resultados Del Ranking Mundial Del Talento 2023	Administración de Empresas	Publicado	Actualizar estado
		Resultados Del Ranking De			Eliminar

Ilustración 34 Botón de Acción administrador

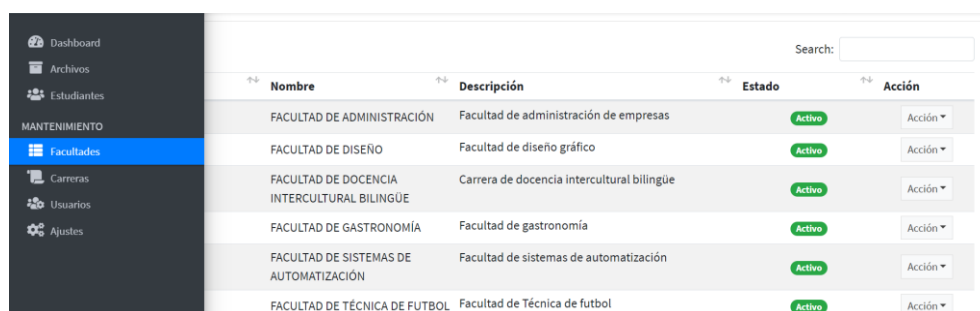
El botón de acción permite actualizar el estado de la tesis de publicado a no publicado y permite eliminar la tesis.



Correo	Estado	Acción
Eduardo Francisco edulego@yahoo.com	Verificado	Acción ▼
Andro David adfb@gmail.com	No Verificado	Acción ▼
joel@gmail.com	Verificado	Acción ▼
raul@gmail.com	Verificado	Acción ▼

Ilustración 35 Botón estudiantes administrador

El botón de estudiantes que permite actualizar el estado y habilitar a los estudiantes registrados.



Nombre	Descripción	Estado	Acción
FACULTAD DE ADMINISTRACIÓN	Facultad de administración de empresas	Activo	Acción ▼
FACULTAD DE DISEÑO	Facultad de diseño gráfico	Activo	Acción ▼
FACULTAD DE DOCENCIA INTERCULTURAL BILINGÜE	Carrera de docencia intercultural bilingüe	Activo	Acción ▼
FACULTAD DE GASTRONOMÍA	Facultad de gastronomía	Activo	Acción ▼
FACULTAD DE SISTEMAS DE AUTOMATIZACIÓN	Facultad de sistemas de automatización	Activo	Acción ▼
FACULTAD DE TÉCNICA DE FUTBOL	Facultad de Técnica de futbol	Activo	Acción ▼

Ilustración 36 Botón Facultades administrador

El botón de facultades que permite actualizar o eliminar las facultades con las que cuenta el Instituto.

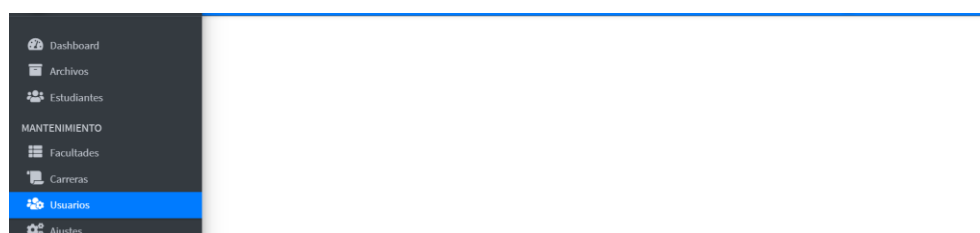


Facultad	Nombre	Estado	Acción
FACULTAD DE ADMINISTRACIÓN	Administración De Empresas	Activo	Acción ▼
FACULTAD DE DISEÑO	Diseño Gráfico	Activo	Acción ▼
FACULTAD DE GASTRONOMÍA	Gastronomía	Activo	Acción ▼
FACULTAD DE DOCENCIA INTERCULTURAL BILINGÜE	Intercultural Bilingüe	Activo	Acción ▼
FACULTAD DE SISTEMAS DE AUTOMATIZACIÓN	Sistemas De Instrumentación Y Automatización	Activo	Acción ▼
FACULTAD DE TÉCNICA DE FUTBOL	Técnica De Futbol	Activo	Acción ▼

Previous 1 Next

Ilustración 37 Botón Carreras administrador

El botón carreras que permite actualizar o eliminar las carreras con las que cuenta el Instituto.



Usuarios

Ilustración 38 Botón Usuarios administrador

El botón usuarios que permite verificar los usuarios con permisos de administrador, activarlos, desactivarlos o eliminarlos.

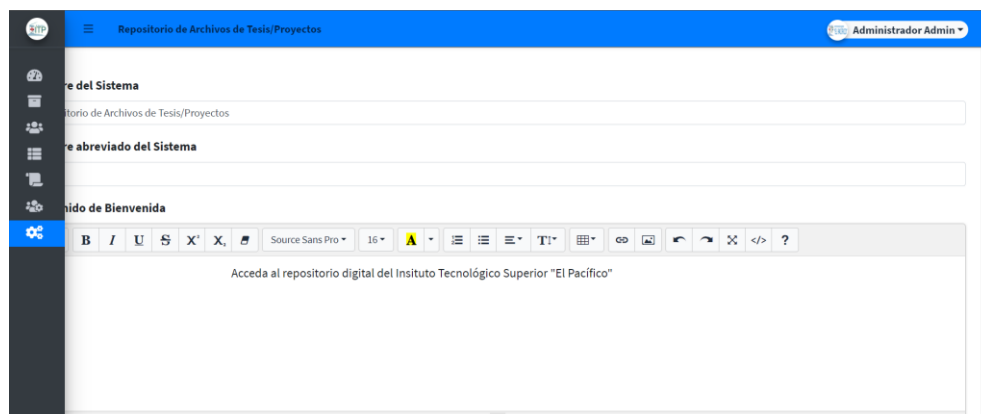


Ilustración 39 Botón Ajustes administrador

El botón de ajustes que permite modificar la información que se muestra en la pantalla de inicio del repositorio.

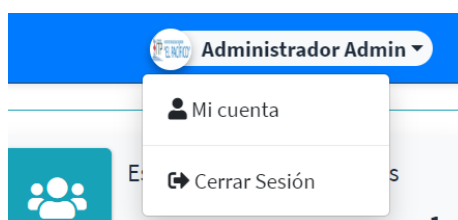


Ilustración 40 Botón Administrador perfil

El botón del administrador le permite ingresar a su perfil y editar su información; y, el botón de cerrar sesión que saca al administrador del sistema.

Manual de Mantenimiento (Claves, Accesos y Forma De Realimentación De Textos)

1. Claves y Accesos

- 1.1. Creación de Usuarios
- 1.2. Asignación de Roles
- 1.3. Cambio de Contraseñas

2. Realimentación de Textos

- 2.1. Revisión y Edición de Textos
- 2.2. Comentarios y Sugerencias
- 2.3. Aprobación y Publicación

3. Mantenimiento del Sistema

- 3.1. Actualización de Software
- 3.2. Respaldo de Datos
- 3.3. Monitoreo y Optimización del Rendimiento

4. Seguridad y Buenas Prácticas

- 4.1. Políticas de Seguridad
- 4.2. Protección de Datos Personales
- 4.3. Buenas Prácticas de Uso

1. Claves y Accesos

1.1. Creación de Usuarios

Para crear un nuevo usuario en el sistema:

1. Iniciar sesión con una cuenta de administrador.
2. Navegar a la sección de administración de usuarios.
3. Hacer clic en "Crear nuevo usuario".
4. Introducir la información requerida: nombre, correo electrónico, y elegir la carrera a la que pertenece.

5. Asignar una contraseña temporal y enviar al nuevo usuario.
6. Asignar el rol correspondiente (estudiante, revisor, administrador).

1.2. Asignación de Roles

Para asignar o modificar roles de usuario:

1. Iniciar sesión con una cuenta de administrador.
2. Navegar a la sección de administración de usuarios.
3. Seleccionar el usuario al que se desea cambiar el rol.
4. Elegir el nuevo rol del menú desplegable y guardar los cambios.

1.3. Cambio de Contraseñas

Para cambiar la contraseña:

1. Iniciar sesión en el sistema.
2. Navegar a la configuración de cuenta.
3. Seleccionar "Cambiar contraseña".
4. Introducir la contraseña actual y la nueva contraseña dos veces.
5. Guardar los cambios.

2. Realimentación de Textos

2.1. Revisión y Edición de Textos

1. Los revisores acceden a los trabajos pendientes de revisión desde su panel de control.
2. Descargan el documento, lo revisan y hacen las correcciones necesarias.
3. Suben el documento corregido al sistema y marcan la revisión como completada.

2.2. Comentarios y Sugerencias

1. Los revisores pueden añadir comentarios y sugerencias directamente en el documento o en el sistema.

2. Los estudiantes reciben notificaciones de las revisiones y comentarios.

3. Los estudiantes pueden responder a los comentarios y realizar las correcciones necesarias.

2.3. Aprobación y Publicación

1. Una vez que el trabajo ha sido revisado y corregido, el revisor puede aprobarlo.

2. El trabajo aprobado se marca como "Listo para publicación".

3. Los administradores verifican la aprobación final y publican el trabajo en el repositorio.

3. Mantenimiento del Sistema

3.1. Actualización de Software

1. Verificar regularmente las actualizaciones del software del sistema.

2. Planificar las actualizaciones fuera de las horas pico para minimizar el impacto.

3. Realizar una copia de seguridad completa antes de iniciar cualquier actualización.

4. Seguir las instrucciones del proveedor para la actualización del sistema.

3.2. Respaldo de Datos

1. Configurar respaldos automáticos diarios del sistema.

2. Almacenar las copias de seguridad en una ubicación segura y diferente del servidor principal.

3. Verificar periódicamente la integridad de las copias de seguridad.

3.3. Monitoreo y Optimización del Rendimiento

1. Utilizar herramientas de monitoreo para supervisar el rendimiento del sistema.
2. Identificar y solucionar cuellos de botella o problemas de rendimiento.
3. Realizar mantenimiento preventivo regularmente, como la limpieza de registros temporales y la optimización de bases de datos.

4. Seguridad y Buenas Prácticas

4.1. Políticas de Seguridad

1. Establecer políticas de seguridad claras para el uso del sistema.
2. Implementar autenticación multifactor para accesos administrativos.
3. Mantener el software actualizado para proteger contra vulnerabilidades.

4.2. Protección de Datos Personales

1. Asegurarse de que los datos personales estén cifrados tanto en tránsito como en reposo.
2. Limitar el acceso a los datos personales solo a aquellos usuarios que realmente lo necesiten.
3. Cumplir con las normativas locales e internacionales de protección de datos.

4.3. Buenas Prácticas de Uso

1. Capacitar a los usuarios en las buenas prácticas de uso del sistema.
2. Fomentar la creación de contraseñas seguras y el cambio regular de las mismas.
3. Monitorizar el uso del sistema para detectar y prevenir usos indebidos.

CAPITULO IV

PRESUPUESTO

Mano de Obra

No. de horas	Valor Hora	Valor Total
50	3.3	165

Costo directo

Producto	Cantidad	Valor Unitario	Valor Total
Energía eléctrica	50 Horas	0.125/hora	6.25
Internet	50 horas	0.1875/hora	9.375
Total			15.625

Costo indirecto

Producto	Cantidad	Valor Unitario	Valor Total
Software	1licencia	100	100
Depreciación equipos informático	33.3%	700	23.310
Arriendo	250 mensual	1.42/hora	52.1
Total			175.41

Resumen costos y gastos

RUBRO	VALOR USD
Costo directo	15,625
Costo Mano de Obra	165,000
Costo Primo	180,625
Costo Indirecto	175,410
subtotal total	356,035
30% Innovación	106,8105
Costo total	462,85

Cálculo del precio

$$Precio = \frac{\text{Costo Unitario}}{(1 - \text{Utilidad deseada})}$$

Considerando que el proyecto se utilizará como software para una institución educativa y considerando la disposición de la Ley de que sus precios debe tener un descuento, se considera una utilidad del 20%.

$$Precio = \frac{462,85}{(1 - 0.2)}$$

$$Precio = \frac{462,85}{0.8}$$

$$Precio = \text{USD}579,00$$

Evaluación del tutor

CRITERIOS DE EVALUACIÓN		Ponderación
CRITERIO I: INNOVACION		
○ Problemática debidamente justificada y orientada a la propuesta de innovación, problema central, causas y efectos coherentes y claros. ○ 25% Grado de innovación de la solución propuesta, contenido tecnológico de la misma, nivel de invención y posibilidad de patentamiento. ○ Existencia de la brecha tecnológica a partir de la descripción del estado del arte de la innovación propuesta		
CRITERIO II: FACTIBILIDAD TÉCNICA		
○ Existe coherencia entre el problema central identificado y el objetivo del proyecto. 35% Existe coherencia entre actividades, objetivos específicos (componentes), objetivo general (propósito) del proyecto. ○ Los resultados finales e intermedios están respaldados por indicadores claramente definidos. ○ La metodología está correctamente descrita, permite esperar la obtención de los resultados esperados y dar respuesta a la propuesta de innovación planteada. ○ Pertinencia del equipamiento, bienes, materiales e infraestructura, consultorías, servicios de terceros y pasantías (si las hubiera) que se destinarán a la ejecución del proyecto. ○ El presupuesto presentado para equipamiento, materiales y servicios de terceros responde a las actividades y objetivos del proyecto, asimismo es considerado en la descripción de la metodología		
CRITERIO III: IMPACTOS		
○ Identificación de posibles impactos en los factores económicos, sociales, tecnológicos y ambientales como consecuencia de la ejecución del proyecto.		

CRITERIO IV: CAPACIDAD DE EJECUCIÓN Y COMPROMISO DEL EQUIPO DE INVESTIGADORES	
○ Composición, calificaciones y experiencia en investigación e innovación de los miembros del equipo técnico de investigadores, actividades y experiencias previas relacionadas a la temática del proyecto. ○ 15% Complementariedad del equipo técnico de investigadores para la propuesta multidisciplinaria.	
CATEGORÍA V: ASPECTOS DE MERCADO, MODELO DE NEGOCIO Y PROPUESTA DE VALOR.	
○ Situación favorable del entorno de mercado, clara identificación del mercado real y potencial. ○ Grado de diferenciación sobre la oferta actual en el segmento de mercado elegido. 15% El modelo de negocio y estrategia empresarial aseguran el ingreso exitoso de la innovación al mercado. ○ Ventajas competitivas a partir de la diferenciación de la propuesta de valor. Consistencia del modelo de negocio ○ Capacidad de escalamiento de la propuesta de innovación. ○ Potencial, respaldo y redes para llevar adelante la propuesta de innovación.	
NOTA FINAL	%

Evaluador(f): _____

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- **Eficiencia en la gestión de usuarios y datos:**

La implementación de tres tipos de usuarios (estudiante, administrador y administrativo) facilita la gestión y el control de acceso, permitiendo que cada rol tenga privilegios específicos. Esto contribuye a una administración más segura y eficiente del repositorio. Además, los filtros por facultad, carrera y filtro general optimizan la búsqueda y el acceso a los trabajos de titulación, mejorando la experiencia del usuario y facilitando la recuperación de información relevante.

- **Uso de tecnologías estandarizadas:**

El uso de Bootstrap para el desarrollo del frontend del repositorio garantiza un diseño responsivo y una interfaz de usuario coherente y moderna. Bootstrap, al ser un framework ampliamente utilizado, también facilita la colaboración y el mantenimiento del código, ya que muchos desarrolladores están familiarizados con él. Esto contribuye a la sostenibilidad a largo plazo del proyecto.

- **Solidez de la base de datos:**

La elección de MySQL como sistema de gestión de bases de datos proporciona una plataforma robusta y fiable para almacenar y gestionar los datos de los trabajos de titulación. MySQL es conocido por su estabilidad y rendimiento, lo cual es esencial para aplicaciones que manejan grandes volúmenes de datos. Además, su integración

con PHP es directa, lo que facilita las operaciones de bases de datos dentro del aplicativo.

- **Autenticación de ingreso:**

La implementación de un sistema de autenticación asegura que solo usuarios autorizados puedan acceder al repositorio, lo cual es crucial para proteger la integridad y confidencialidad de los datos almacenados. Esto no solo aumenta la seguridad, sino que también refuerza la confianza de los usuarios en el sistema.

- **Documentación técnica y de usuario bien desarrollada:**

La existencia de documentación técnica y manuales de usuario demuestra un enfoque profesional en el desarrollo del sistema. Esto no solo facilita la comprensión y el uso del repositorio por parte de los usuarios, sino que también asegura que otros desarrolladores puedan mantener y mejorar el sistema con mayor facilidad en el futuro.

5.1.2. Recomendaciones

- **Implementar pruebas unitarias y de integración:**

Es altamente recomendable implementar un conjunto de pruebas unitarias y de integración para garantizar que todas las partes del sistema funcionen correctamente y de manera conjunta. Las pruebas unitarias permiten verificar que cada componente individual funcione como se espera, mientras que las pruebas de integración aseguran que los diferentes componentes interactúen correctamente. Herramientas como PHPUnit pueden ser útiles para este propósito.

- **Mejorar la seguridad:**

Aunque se ha implementado autenticación, es importante considerar otras medidas de seguridad. Implementar HTTPS para encriptar las comunicaciones entre el cliente y el servidor, así como el uso de prácticas seguras para el manejo de contraseñas (como el hashing con bcrypt) y la protección contra ataques comunes como SQL injection y Cross-Site Scripting (XSS). Recomendaciones de seguridad adicionales incluyen el uso de prepared statements en SQL y la validación/sanitización de datos de entrada.

- **Optimización y escalabilidad:**

Considerar la optimización y escalabilidad del sistema, especialmente si se espera un crecimiento en el número de usuarios y en la cantidad de datos. Esto puede incluir la optimización de consultas SQL, el uso de índices en la base de datos para acelerar las búsquedas, y la implementación de caché para reducir la carga del servidor. Además, evaluar opciones de escalabilidad como la partición de bases de datos y el uso de balanceadores de carga.

- **Mantenimiento de la documentación técnica:**

Dado que ya existe documentación técnica y manuales de usuario, es importante mantenerlos actualizados a medida que el sistema evoluciona. Cualquier cambio en la arquitectura, nuevas

características o modificaciones en el flujo de trabajo deben reflejarse en la documentación. Esto asegurará que la información siga siendo relevante y útil tanto para los usuarios finales como para los desarrolladores.

- **Monitoreo y mantenimiento continuo:**

Implementar herramientas de monitoreo para supervisar el rendimiento y la disponibilidad del sistema. Herramientas como New Relic o Nagios pueden ser útiles para detectar y resolver problemas antes de que afecten a los usuarios. Además, establecer un plan de mantenimiento regular que incluya la actualización de dependencias, la revisión de seguridad, y la limpieza de datos obsoletos.

Bibliografía y Linkografía

<https://www.significados.com/investigacion-de-campo/>

<https://www.significados.com/muestra/>

<https://concepto.de/encuesta/#ixzz8MGeuZxCT>

Borges, M., Valente, M.T., & Bigonha, M. (2012). Analyzing the Evolution of PHP Web Systems.

<https://scholar.google.com/scholar?q=php+web+systems+evolution>

Atzeni, P., Jensen, C.S., Orsi, G., Ram, S., Tanca, L., & Torlone, R. (2013).

A survey on Web application development. ACM Transactions on the Web, 7(3), 1-43.

<https://scholar.google.com/scholar?q=A+survey+on+Web+application+development>

Mark Otto & Jacob Thornton (2011). Bootstrap Documentation.

<https://getbootstrap.com>

Reinman, J., & Barnett, J. (2013). Learning Bootstrap: Understanding and Implementing One of the Web's Most Popular Front-End Frameworks.

<https://scholar.google.com/scholar?q=bootstrap+front-end+framework>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software.

<https://scholar.google.com/scholar?q=Design+Patterns+MVC>

Fowler, M. (2002). Patterns of Enterprise Application Architecture

<https://scholar.google.com/scholar?q=Patterns+of+Enterprise+Application+Architecture>

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), 377-387.

<https://scholar.google.com/scholar?q=relational+model>

Melton, J. (2001). SQL:1999 Understanding Relational Language Components. Morgan Kaufmann.

<https://scholar.google.com/scholar?q=SQL+1999>

Fox, Geoffrey C. (1998). The use of the MIDAS system for database management.

<https://scholar.google.com/scholar?q=MIDAS+database+management>

<http://midas-database.org/>

Flanagan, D. (2006). JavaScript: The Definitive Guide (5th ed.). O'Reilly Media.

<https://scholar.google.com/scholar?q=JavaScript+The+Definitive+Guide>

Zakas, N. C. (2010). Professional JavaScript for Web Developers.

<https://scholar.google.com/scholar?q=Professional+JavaScript+for+Web+Developers>

Musciano, C., & Kennedy, B. (2006). HTML & XHTML: The Definitive Guide (6th ed.). O'Reilly Media.

<https://scholar.google.com/scholar?q=HTML+The+Definitive+Guide>

Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley.

<https://scholar.google.com/scholar?q=HTML+and+CSS+Design+and+Build+Websites>

Meyer, E.A. (2007). CSS: The Definitive Guide (3rd ed.). O'Reilly Media.

<https://scholar.google.com/scholar?q=CSS+The+Definitive+Guide>

Weyl, S. (2015). Responsive CSS: Developing Scalable Web Pages. Addison-Wesley Professional.

<https://scholar.google.com/scholar?q=Responsive+CSS+Developing+Scalable+Web+Pages>

Galloway, J., Wilson, B., Matson, K., & Hazzard, D. (2013). Professional ASP.NET MVC 4. Wrox.

<https://scholar.google.com/scholar?q=Professional+ASP.NET+MVC+4>

Freeman, A. (2014). Pro ASP.NET MVC 5. Apress.

<https://scholar.google.com/scholar?q=Pro+ASP.NET+MVC+5>).

Stallings, W. (2011). Computer Organization and Architecture* (9th ed.). Prentice Hall.

<https://scholar.google.com/scholar?q=Computer+Organization+and+Architecture>

Tanebaum, A. S., & Wetherall, D. J. (2013). Computer Networks (5th ed.). Prentice Hall.

<https://scholar.google.com/scholar?q=Computer+Networks>

Ceruzzi, P. E. (2003). A History of Modern Computing. MIT Press.

<https://scholar.google.com/scholar?q=A+History+of+Modern+Computing>

Martin Campbell-Kelly & William Aspray (2004). Computer: A History of the Information Machine. Basic Books.

<https://scholar.google.com/scholar?q=Computer+A+History+of+the+Information+Machine>