



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
CARRERA DE SOFTWARE

Tema:

**APLICACIÓN WEB MULTIPLATAFORMA PARA LA GESTIÓN DE LA
CARPETA ACADEMICA DE LOS ESTUDIANTES EN LA FISEI DE LA
UNIVERSIDAD TÉCNICA DE AMBATO.**

Trabajo de titulación modalidad: Proyecto de Investigación, presentado previo a la
obtención del Título de Ingeniero de Software.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Paulo Alexis Martínez Altamirano

TUTOR: Ing. Fernando Ibarra Mg.

Ambato – Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN WEB MULTIPLATAFORMA PARA LA GESTIÓN DE LA CARPETA ACADEMICA DE LOS ESTUDIANTES EN LA FISEI DE LA UNIVERSIDAD TÉCNICA DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Paulo Alexis Martínez Altamirano, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024

.....
Ing. Óscar Fernando Ibarra Torres, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN WEB MULTIPLATAFORMA PARA LA GESTIÓN DE LA CARPETA ACADEMICA DE LOS ESTUDIANTES EN LA FISEI DE LA UNIVERSIDAD TÉCNICA DE AMBATO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



.....
Paulo Alexis Martínez Altamirano

C.C. 185095427-0

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Paulo Alexis Martínez Altamirano

C.C. 1850954270

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Paulo Alexis Martinez Altamirano, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado APLICACIÓN WEB MULTIPLATAFORMA PARA LA GESTIÓN DE LA CARPETA ACADEMICA DE LOS ESTUDIANTES EN LA FISEI DE LA UNIVERSIDAD TÉCNICA DE AMBATO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Elsa Pilar Urrutia Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Mg. Hernán Naranjo Avalos

PROFESOR CALIFICADOR

Ing. Mg. Sandra Carrillo Ríos

PROFESOR CALIFICADOR

DEDICATORIA

Este proyecto está dedicado mis padres, Jorge y Narcisa quienes siempre me brindaron su apoyo incondicional a lo largo de este arduo camino académico a pesar de las dificultades presentadas. Su amor, paciencia y sabiduría fueron la luz que iluminó cada paso de mi trayectoria para poder alcanzar uno mis muchos objetivos.

A mis hermanos y hermanas, por ser mi fuente constante de inspiración y alegría. Gracias por comprender mis ausencias y celebrar mis logros con tanto entusiasmo.

A mi amigo Alexander, quien siempre ha estado como un hermano para ayudarme y aconsejarme en los momentos de alegría y en las encrucijadas de la adversidad. Su amistad ha sido un pilar fundamental que ha hecho más llevadera esta travesía académica.

Paulo Alexis Martinez Altamirano

AGRADECIMIENTO

Quiero expresar mis sinceros agradecimientos a mis padres por haberme ayudado alcanzar en meta de mi vida, por siempre estar ahí aconsejándome y guiándome por el sendero correcto.

Agradezco a los profesores por su dedicación y por compartir sus conocimientos de esta maravillosa carrera. Su influencia positiva ha sido una fuente constante de inspiración y motivación.

Un agradecimiento especial a mi tutor Ing. Fernando Ibarra por haberme permitido formar parte de grupo de tutorados, además recalcar su apoyo y paciencia durante el desarrollo de este proyecto.

No puedo pasar por alto el apoyo invaluable de mis amigos y compañeros. Juntos enfrentamos desafíos y celebramos triunfos, creando una experiencia universitaria inolvidable.

Finalmente, agradezco a todas aquellas personas que, de alguna manera, contribuyeron al desarrollo de esta investigación. Cada aporte, ya sea grande o pequeño, ha dejado una marca indeleble en este trabajo.

Paulo Alexis Martínez Altamirano

INDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE DE ANEXOS	xx
RESUMEN EJECUTIVO	xxi
ABSTRACT	xxii
CAPÍTULO I. MARCO TEÓRICO	1
1.1. Tema de investigación.....	1
1.1.1. Planteamiento del problema.....	1
1.2. Antecedentes investigativos	2
1.3. Fundamentación teórica	4
1.3.1. Página Web	5
1.3.2. Ventajas de una página Web	6

1.3.3.	Desventajas de una página Web	6
1.3.4.	Servicios Web.....	7
1.3.5.	Concepto de interfaz de usuario	8
1.3.6.	Diseño de interfaz de usuario	8
1.3.7.	Frameworks web.....	9
1.3.8.	IDE'S de desarrollo	9
1.3.9.	Tecnologías de desarrollo web y móvil.....	10
1.3.10.	Framework de Diseño.....	13
1.3.11.	Base de datos.....	14
1.3.12.	Diseño de base de datos	15
1.3.13.	SQL Server	15
1.3.14.	MySQL	16
1.3.15.	Concepto de metodología ágil.....	16
1.3.16.	Extreme Programming (XP)	16
	Ciclo de XP.....	17
1.3.17.	Rapid Application Development (RAD).....	17
	Ciclo de RAD	18
1.3.18.	Scrum.....	18
	Ciclo de Scrum	19

1.4.	Objetivos	19
1.4.1.	Objetivo general.....	19
1.4.2.	Objetivos específicos	20
CAPÍTULO II. METODOLOGÍA.....		21
2.1.	Materiales	21
2.1.1.	Diagrama de procesos de la gestión de la carpeta estudiantil	21
2.1.2.	Modalidad de Investigación	22
2.1.3.	Población y muestra.....	23
2.1.4.	Recolección de Información	23
2.1.5.	Procesamiento y análisis de datos.....	26
CAPÍTULO III. RESULTADOS Y DISCUSIÓN		44
3.1.	Análisis y discusión de resultados	44
3.1.1.	Diagrama de actividades del sistema como usuario Admin	44
3.1.2.	Diagrama de actividades del sistema como usuario secretario	39
3.1.3.	Análisis de Metodología de Desarrollo.....	41
3.1.4.	Análisis de los Frameworks de Desarrollo	42
3.1.5.	Análisis de los Frameworks de CSS	44
3.2.	Desarrollo de la propuesta.....	45
3.2.1.	Fase I: Recopilación de requisitos	46

3.2.2.	Fase II: Prototipo y retroalimentación por parte del cliente	55
3.2.3.	Fase III: Construcción de la aplicación	61
3.2.4.	Fase IV: Pruebas	92
3.2.5.	Fase V: Implementación	100
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES		107
4.1.	Conclusiones	107
REFERENCIAS BIBLIOGRÁFICAS		108
ANEXOS		114

INDICE DE TABLAS

Tabla 1: Población.....	23
Tabla 2: Tabla de frecuencias - Pregunta 1	26
Tabla 3: Tabla de frecuencias - Pregunta 2	28
Tabla 4: Tabla de frecuencias - Pregunta 3	29
Tabla 5: Tabla de frecuencias - Pregunta 4	31
Tabla 6: Tabla de frecuencias - Pregunta 5	32
Tabla 7: Tabla de frecuencias - Pregunta 6	34
Tabla 8: Tabla de frecuencias - Pregunta 7	36
Tabla 9: Tabla de frecuencias - Pregunta 8.....	37
Tabla 10: Tabla de frecuencias - Pregunta 9.....	39
Tabla 11: Tabla de frecuencias - Pregunta 10.....	40
Tabla 12: Tabla de frecuencias - Pregunta 11	42
Tabla 13: Cuadro comparativo de las metodologías de desarrollo XP, RAD, Scrum	41
Tabla 14: Cuadro comparativo entre los Frameworks de desarrollo, Flutter, Vue.js, React	42
Tabla 15: Cuadro comparativo entre los Frameworks de diseño Vuetify, Bootstrap, Tailwind	44
Tabla 16: Requerimientos del sistema	46

Tabla 17: Tabla de requerimientos funcionales	47
Tabla 18: Tabla de requerimientos no funcionales	48
Tabla 19: Evaluación Heurística	70
Tabla 20: Prueba de aplicación - Requerimiento R001	92
Tabla 21: Prueba de aplicación - Requerimiento R002	92
Tabla 22: Prueba de aplicación - R003	93
Tabla 23: Prueba de aplicación - R004	93
Tabla 24: Pruebas de aplicación - R005.....	94
Tabla 25: Pruebas de aplicación - R006.....	94
Tabla 26: Pruebas de aplicación - R007.....	95
Tabla 27: Pruebas de aplicación - R008.....	96
Tabla 28: Pruebas de aplicación - R009.....	96
Tabla 29: Prueba de aplicación - R010	97
Tabla 30: Pruebas de interfaz de usuario	97
Tabla 31: Pruebas de aceptación - Iteración 1	98
Tabla 32: Pruebas de aceptación - Iteración 2	99
Tabla 33: Pruebas de aceptación - Iteración 3	99
Tabla 34 Pruebas de aceptación - Iteración 4.....	100
Tabla 35: Cronograma de implementación	100

Tabla 36: Análisis de tiempos estimados de procesos	102
Tabla 37: Cuestionario de evaluación del modelo TAM	103
Tabla 38: Tabla de frecuencias – Modelo TAM – Utilidad percibida	104
Tabla 39: Tabla de frecuencias - modelo TAM - Facilidad de uso.....	105

INDICE DE FIGURAS

Figura 1 Diagrama de procesos de la gestión de la carpeta estudiantil.....	22
Figura 2: Resultados de la encuesta - Pregunta 1.....	27
Figura 3: Resultados de la encuesta - Pregunta 2.....	28
Figura 4: Resultados de la encuesta - Pregunta 3.....	30
Figura 5: Resultados de la encuesta - Pregunta 4.....	31
Figura 6: Resultados de la encuesta - Pregunta 5.....	33
Figura 7: Resultados de la encuesta - Pregunta 6.....	34
Figura 8: Resultados de la encuesta - Pregunta 7.....	36
Figura 9: Resultados de la encuesta - Pregunta 8.....	38
Figura 10: Resultados de la encuesta - Pregunta 2.....	39
Figura 11: Resultados de la encuesta - Pregunta 10.....	41
Figura 12: Resultados de la encuesta - Pregunta 11.....	43
Figura 13: Diagrama general de actividades como usuario admin	38
Figura 14: Diagrama general de actividades como usuario secretaria	40
Figura 15: Arquitectura del sistema	50
Figura 16: Diagrama entidad-relación de la Base de datos.....	54
Figura 17: Prototipo de la interfaz del ingreso a la aplicación.....	55

Figura 18: Prototipo de interfaz principal del sistema	55
Figura 19: Prototipo de interfaz Usuarios	56
Figura 20: Prototipo de interfaz para crear y editar un usuario.....	57
Figura 21: Prototipo interfaz carreras de cada usuario.....	57
Figura 22: Prototipo de interfaz Estudiantes	58
Figura 23: Prototipo de interfaz para crear y editar un estudiante	58
Figura 24: Prototipo de interfaz para crear carpetas	59
Figura 25: Prototipo de interfaz para cada ítem de la tabla Estudiantes	59
Figura 26: Prototipo de interfaz para ítem de un directorio	60
Figura 27: Prototipo de interfaz Plantilla	60
Figura 28: Prototipo de interfaz para crear Plantillas.....	61
Figura 29: Comando para instalar vue.js.....	62
Figura 30: Comando para crear un proyecto en vue.js.....	62
Figura 31: Archivos de la aplicación luego de crear la misma	63
Figura 32: Comando para añadir Vuetify.....	63
Figura 33: Datos para la conexión con el servidor.....	64
Figura 34: Código .php para conexión con el servidor	64
Figura 35: Comando para instalar phpseclib.....	64
Figura 36: Login.....	65

Figura 37: Interfaz donde se visualiza los estudiantes por carrera.....	65
Figura 38: Interfaz donde se crean las subcarpetas de los estudiantes.....	66
Figura 39: Creación de una nueva subcarpeta.....	66
Figura 40: interfaz archivos	67
Figura 41: interfaz usuario	67
Figura 42: ventana para crear un usuario	68
Figura 43: Ventana para visualizar permisos	68
Figura 44: interfaz plantillas	69
Figura 45: ventana para crear una plantilla.....	69
Figura 46: Pagina Login.....	71
Figura 47: Código .js para API Login.....	71
Figura 48: Código .js para crear un token de sesión	72
Figura 49: Página Principal, se muestran las carreras que tiene el usuario asignadas	72
Figura 50: Código .js para solicitar los datos a la API.....	73
Figura 51: Código vue.js para verificar la autenticación y cargar carreras.....	73
Figura 52: Pagina de estudiantes.....	74
Figura 53: Código .js para consumo de API cargar permisos usuarios	74
Figura 54: Método para cargar estudiantes	75
Figura 55: Ventana modal para crear un estudiante.....	75

Figura 56: Código .js con métodos para crear un estudiante	76
Figura 57: Código .js para crear subcarpetas	76
Figura 58: Código .js para API crear estudiante	77
Figura 59: Opción para importar datos desde un archivo Excel	78
Figura 60: Método para crear subcarpetas al importar datos	78
Figura 61: Pagina para mostrar subcarpetas de cada usuario.....	79
Figura 62: Método para obtener los permisos del usuario	80
Figura 63: Código .js para API cargar subcarpetas.....	80
Figura 64: Código .js para descargar una carpeta	80
Figura 65: Ventana para crear carpetas.....	81
Figura 66: Pagina para mostrar los archivos	81
Figura 67: Método para obtener las rutas para crear la subcarpeta.....	81
Figura 68: Código para API crear archivos.....	82
Figura 69: Método para obtener la ruta.....	82
Figura 70: Código para API descargar archivos	83
Figura 71: Pagina Usuarios, accedida solo por un usuario admin	84
Figura 72: Código .js de consumo del API para cargar usuarios	84
Figura 73: Ventana Modal para crear y editar un usuario.....	85
Figura 74: Código .js con métodos para crear un usuario.....	85

Figura 75: Ventana para ver y asignar permisos a un usuario	86
Figura 76: Código .js para consumo de API eliminar permisos	86
Figura 77: Código .js para API agregar permisos	87
Figura 78: Código .js para agregar permisos	87
Figura 79: Pagina para poder visualizar y crear plantillas	88
Figura 80: Código .js para API cargar plantillas.....	88
Figura 81: Código .js para API cargar plantillas.....	89
Figura 82: Ventana modal para crear plantillas	89
Figura 83: Código .js con métodos para crear una plantilla.....	90
Figura 84: Código .js para generar una plantilla y editar una carpeta.....	90
Figura 85: Código para API crear plantilla	91
Figura 86: Código .js para API crear subcarpetas.....	91
Figura 87: Código .js para API crear relación entre plantillas subcarpetas	92
Figura 88: Resultado del modelo TAM- Utilidad percibida	104
Figura 89: Resultado del modelo TAM- Facilidad de uso.....	106

ÍNDICE DE ANEXOS

Anexo A	114
Anexo B	118
Anexo C	120

RESUMEN EJECUTIVO

La gestión de la carpeta académica actualmente se realiza de manera manual por las secretarías de la FISEI, encargadas del proceso de titulación. Este método presenta diversas problemáticas, como la pérdida de documentos, demoras significativas y dificultades en el acceso a la información necesaria para llevar a cabo dicho proceso. El propósito fundamental de la presente investigación ha sido proponer un sistema que pueda gestionar la carpeta académica con el fin de simplificar este procedimiento para las secretarías, quienes han desempeñado esta labor de manera manual durante años. En este contexto, se utilizaron técnicas de recopilación de datos, como entrevistas y encuestas, para identificar las problemáticas y posibles soluciones que se pueden aplicar al proceso identificado. De igual manera, las encuestas propuestas ayudaron a recopilar los requerimientos que serán los pilares sobre los cuales se implementará este sistema. En el proceso de desarrollo, se utilizó la metodología Rapid Application Development (RAD), la cual se centra en el desarrollo rápido, otorgando mayor énfasis a las tareas de desarrollo que a la planificación en sí misma. Se enfoca en desarrollar software en corto tiempo a través de prototipos iterativos que son revisados y refinados en colaboración con los usuarios finales. Para el proceso de desarrollo de software, se utilizó MySQL como gestor de bases de datos y vue.js para el desarrollo de la aplicación web. Mediante el modelo TAM se determinó que el 82% de los participantes se ven satisfechos en base a las características de usabilidad y facilidad de uso del sistema.

Palabras clave: Rapid Application Development (RAD), vue.js, carpeta académica, TAM.

ABSTRACT

The management of the academic portfolio is currently done manually by the secretaries of the School of Systems, Electronics and Industrial Engineering, who are in charge of the degree process. This method presents several problems, such as the loss of documents, significant delays and difficulties in accessing the information necessary to carry out this process. The main purpose of this research has been to propose a system that can manage the academic portfolio in order to simplify this procedure for the secretaries, who have been performing this task manually for years. In this context, data collection techniques, such as interviews and surveys, were used to identify the problems and possible solutions that can be applied to the identified process. Similarly, the proposed surveys helped to gather the requirements that will be the pillars on which this system will be implemented. In the development process, the Rapid Application Development (RAD) methodology was used, which focuses on rapid development, giving more emphasis to the development tasks than to the planning itself. It focuses on developing software in a short time through iterative prototypes that are reviewed and refined in collaboration with end users. For the software development process, MySQL was used as database manager and vue.js for the development of the web application. Using the TAM model, it was determined that 82% of the participants were satisfied with the usability and ease of use of the system.

Keywords: Rapid Application Development (RAD), vue.js, academic portfolio, TAM.

CAPÍTULO I. MARCO TEÓRICO

1.1. Tema de investigación

1.1.1. Planteamiento del problema

La gestión de la carpeta académica de los estudiantes de la FISEI (FISEI) de la Universidad Técnica de Ambato, ha sido una problemática recurrente debido a la falta de control y al uso de métodos manuales y poco confiables. Actualmente, los encargados se ven obligados a llevar a cabo tareas administrativas de forma manual o en archivos de Excel, lo que genera retrasos en los trámites y pérdida de tiempo tanto para los estudiantes como para las secretarías encargadas.

La FISEI carece de una aplicación informática específica para gestionar las carpetas académicas estudiantiles. Como resultado, se producen retrasos significativos en el proceso de búsqueda y obtención de los documentos necesarios para la creación de la carpeta académica, lo que afecta directamente la titulación de los estudiantes mencionan las secretarías encargadas. Por otro lado, esta problemática también repercute en las secretarías encargadas de llevar a cabo estas tareas, ya que la búsqueda manual de los documentos necesarios para cada estudiante se vuelve complicada y existe el riesgo de extravío de estos, lo que genera un cuello de botella en el proceso.

Las secretarías encargadas de la gestión de la carpeta académica han expresado su frustración por el uso constante de métodos ambiguos en este tipo de procesos, así como para la comunidad estudiantil de la FISEI en su conjunto. Estos métodos generan repercusiones negativas a corto y largo plazo, ya que los estudiantes se ven afectados emocionalmente al no lograr completar el proceso en las fechas asignadas, lo que genera estrés y frustración.

El presente proyecto de investigación se justifica con base en el requerimiento realizado por la Abogada Daniela Montenegro, Secretaria de Facultad, quien a partir del diagnóstico de necesidades de las distintas secretarías para la constitución y almacenamiento electrónico de la carpeta académica estudiantil, concluye que la elaboración de un sistema integrado permitirá concentrar toda la documentación estudiantil de manera adecuada, ordenada y segura, facilitará la participación de las distintas áreas y optimizará el tiempo y los recursos que se emplean para una adecuada gestión documental, determinando previamente el flujo del proceso.

1.2. Antecedentes investigativos

Después de una rigurosa búsqueda en diferentes bases de datos de tesis relacionadas con nuestro tema, se encontró lo siguiente:

Según Ana Yajaira Gómez Méndez [1] en su trabajo de titulación titulado "Aplicación Web para el control de procesos de la Unidad de Titulación del Instituto Tecnológico Superior José Chiriboga Grijalva de la ciudad de Ibarra, provincia de Imbabura", realizado en la Universidad Tecnológica Israel en el año 2019, se concluye lo siguiente:

- En el Instituto Tecnológico Superior José Chiriboga Grijalva de la ciudad de Ibarra, provincia de Imbabura, se reconoce la importancia de optimizar y agilizar el proceso de titulación. Actualmente, la falta de una aplicación Web de control de procesos dificulta el seguimiento y la gestión eficiente de las actividades llevadas a cabo por los estudiantes en su camino hacia la titulación.
- La metodología utilizada en el proyecto de tesis fue Extreme Programming (XP). Esta metodología se basa en un enfoque ágil de desarrollo de software que se centra en la colaboración, la adaptabilidad y la entrega continua de software funcional. Con XP, se planificaron, diseñaron, codificaron y probaron las diferentes etapas del proyecto, lo que permitió optimizar el desarrollo de los procesos implementados y documentados.

- La aplicación web sirve como una alternativa de gestión y comunicación entre tutores, estudiantes y personal de apoyo, facilitando la coordinación y seguimiento de las actividades relacionadas con el proceso de titulación.

Según Jorge Enrique Mendoza Rivilla [2], en su proyecto técnico titulado 'Implementación de un sistema Web para la gestión y control de los procesos de la Unidad de Titulación de la carrera de Ingeniería en Sistemas de la Universidad Politécnica Salesiana, Sede Guayaquil', realizado en el año 2017, se concluye que:

- La Universidad Politécnica Salesiana ha tenido que adaptarse al nuevo proceso de titulación planteado por el Consejo de Educación Superior (CES). Este nuevo proceso ha generado la necesidad de controlar y dar seguimiento a los estudiantes durante su proceso de titulación. Actualmente, la gestión del proceso se lleva a cabo de manera poco automática y manual, lo que genera dificultades para el coordinador de la unidad y los docentes de apoyo, así como para los propios estudiantes al momento de proponer y registrar su opción de titulación.
- La Universidad Politécnica Salesiana, se ha identificado la necesidad de contar con un sistema de información que facilite la gestión y control de los procesos de titulación de los estudiantes.

Aunque no se encontró información específica sobre la gestión y creación de la carpeta académica en los textos proporcionados, se reconoce la importancia de contar con un sistema que permita organizar y almacenar de manera eficiente la documentación y los avances de los estudiantes en su proceso de titulación.

Según Medina Picuasi Maribel Jacqueline [3] en su artículo científico 'Desarrollo de una aplicación Web para la generación de resoluciones HCD del proceso de titulación FICA–UTN con herramientas libres', trabajo realizado como artículo científico de la Universidad Técnica del Norte en el año 2017, se concluye que:

- La implementación de un sistema automatizado para agilizar y mejorar el proceso de titulación en una universidad. El estudio reveló que la implementación de un sistema basado en tecnología permitió reducir significativamente los tiempos de espera, simplificar los trámites administrativos y mejorar la comunicación entre los diferentes actores involucrados en el proceso.
- Se examinaron los factores que contribuyen al éxito en el proceso de titulación de los estudiantes. El estudio identificó la importancia de contar con un sistema de apoyo académico y administrativo sólido, así como la motivación y el compromiso por parte de los estudiantes para completar exitosamente el proceso de titulación.

Según Alvarado Reyes Ángelo Jamil [4] en su trabajo de titulación titulado: "Desarrollo del módulo de aplicación Web para la gestión de sustentación del proceso de titulación en la carrera de Sistemas de Información de la Facultad de Ingeniería Industrial", realizado como trabajo de titulación en la Universidad de Guayaquil en el año 2018, se concluye que:

- Mucha de la información se registra en libros de Excel, lo cual lleva mucho tiempo al ser revisada. Por otro lado, puede generar información inapropiada para quien la requiera, aumentando el desconcierto y la falta de transparencia en cierta información y/o reportes.
- El proceso de sustentación tiene otras características que lo hacen problemático, como su variabilidad, ya que los requisitos pueden cambiar de acuerdo con las disposiciones y reformas académicas de la LOES (Ley Orgánica de Educación Superior) y RRA (Reglamento de Régimen Académico) aprobadas por la UG (Universidad de Guayaquil).
- Al contar con un sistema que permita agilizar las actividades, se ahorraría tiempo, ya que con el uso de las herramientas tecnológicas es mucho más fácil realizar cualquier actividad.

1.3. Fundamentación teórica

1.3.1. Página Web

Una página Web es un sistema electrónico que almacena información multimedia, como videos, textos, audios e imágenes, de manera orgánica e integrada a la red mundial de internet, siguiendo las normas de la World Wide Web. Se puede acceder a ella mediante un ordenador conectado a internet y un navegador Web, y su desarrollo se basa en un sistema de protocolos de comunicación como el lenguaje HTTP (Hypertext Transfer Protocol) o HTTPS (HyperText Transfer Protocol Secure).[5]

La información se almacena en archivos informáticos distribuidos en todo el mundo y conectados entre sí mediante una red de comunicación física. Las páginas Web pueden ser utilizadas como un punto de encuentro para los usuarios, donde pueden encontrar información sobre empresas, organizaciones, servicios, universidades y entidades gubernamentales, así como realizar gestiones o trámites en línea [6].

Características

- Una página web esta creada a través de lenguajes de marcado como HTML, PHP, ASP, JSP, mismo que son interpretados por los navegadores y permite incluir información como textos, sonidos, animaciones, videos. [5]
- Su diseño tiene diferentes funcionalidades, además su diseño tiene que ser amigable y atractivo para el usuario. [5]
- Un diseño responsivo es la parte esencial de una página web, garantiza un aspecto amigable y que funcione correctamente en todos los dispositivos. [7]
- Un buen sitio web debe hacer hincapié en la seguridad para ciberataques como la inyección SQL y el cross-site-scripting, que pueda comprometer la integridad del sitio web. [7]
- Un sitio web debe mantenerse actualizado para garantizar una experiencia de usuario positiva. [7]

- La consistencia asegura que los datos que se proporcionan desde el servidor se presenten de manera armoniosa y coherente en el Front-End para el cliente. La personalización combina la información biográfica del cliente y sus patrones de navegación en el sitio Web para crear un perfil de sus intereses y necesidades, y así ofrecer un servicio personalizado.[5]

1.3.2. Ventajas de una página Web

La creación de una página Web presenta diversas ventajas, entre las que se incluyen:

- Fácil acceso a la organización, empresa y/o información, ya que no se necesita una plataforma física sino virtual, lo que resulta beneficioso para personas con dificultades físicas o de movilidad.
- Mantener un portal Web actualizado y en funcionamiento constante brinda una imagen positiva y progresiva de la empresa o institución.
- La creación y el mantenimiento de una página Web son opciones de bajo costo, lo que permite la creación de tiendas virtuales y otros servicios.
- Disponibilidad las 24 horas del día y los 365 días del año, lo que facilita el acceso de los clientes potenciales en cualquier momento.
- Trabaja a menudo con sistemas automatizados, lo que ahorra tiempo, energía y costos en la interacción con los usuarios. Permite la simultaneidad de dinámicas con clientes y usuarios, lo que hace posible la ejecución de diversas transacciones y procesos al mismo tiempo sin mayores costos.
- Las herramientas para la creación de una página Web están ampliamente disponibles y pueden ser dirigidas por una única persona. [8]

1.3.3. Desventajas de una página Web

- Para utilizarlos se necesita acceso a Internet.

- Requieren dispositivos electrónicos con condiciones adecuadas tanto de software como de hardware.
- Crear un portal Web es complejo desde el punto de vista informático y requiere cuidados en diseño y estructura.
- Necesitan una actualización continua y su respectivo mantenimiento.
- Es necesario cargar constantemente nuevos contenidos para mantener la accesibilidad y vigencia del portal.[5]
- Si hay un error en la programación, el sistema entero puede caer y el acceso al portal Web sería inhabilitado.
- Debido a que están orientados a temáticas o contenidos específicos, pueden carecer de enlaces a otros sitios Web de la red.
- Las transacciones comerciales en línea suelen ser hipotéticas o tentativas hasta que el cliente las completa físicamente.
- Al estar disponible todo el tiempo, es posible que el soporte técnico no esté disponible inmediatamente, sino durante las horas de servicio institucional o corporativo.[8]

1.3.4. Servicios Web

Los servicios Web se refieren a aplicaciones automáticas que se ejecutan, definen, describen, localizan y publican mediante una red digital. Estos servicios proporcionan soporte para la inserción, aplicación y desenlace de diversas actividades, contenidos y procesos que no están integrados directamente en el sitio Web. En otras palabras, los servicios Web mejoran la capacidad de los procesos al integrarse con otras aplicaciones para permitir la comunicación. [9]

Para que los servicios Web funcionen correctamente, se utilizan tres componentes principales: WSDL (Web Services Description Language), que describe los sitios Web al enlazarlos con el servicio; SOAP (Simple Object Access Protocol), un protocolo basado en XML (Extensible Markup Language) que solicita el servicio; y UDDI (Universal

Description, Discovery and Integration), un registro que actúa y sostiene el servicio como si fuera una base de contactos intermediaria; [8] REST (Representational State Transfer) una interfaz para conectar varios sistemas basados en el protocolo HTTP, utilizado para obtener y generar datos, devolviendo esos datos en formato muy específicos, como XML y JSON. Uno de los formatos más utilizados en la actualidad es JSON, siendo este más ligero y legible en comparación con XML, REST se apoya en los métodos básico de HTTP, como son POST, GET, PUT, PATCH, DELETE. [10]

En conjunto, estos componentes garantizan una correcta ejecución de los servicios Web y su integración con otras aplicaciones.

1.3.5. Concepto de interfaz de usuario

Las interfaces de usuario han pasado por al menos tres ciclos durante los últimos diez años: desde la interfaz de texto, pasando por la interfaz gráfica de usuario, hasta llegar a la interfaz Web, y esta tendencia continúa avanzado rápidamente. La definición de una interfaz gráfica de usuario (GUI) se ha ampliado considerablemente. En primer momento, eran menús e iconos en el ordenador, algo puramente funcional. En la actualidad una GUI es donde coinciden el diseño de la interacción y el de la interfaz.[11]

1.3.6. Diseño de interfaz de usuario

El diseño de la user interface (UI) se refiere al proceso de diseñar la apariencia visual y la interacción de un sitio Web o aplicación, con el objetivo de maximizar la facilidad de uso y la eficiencia para el usuario. Esto implica diseñar elementos como botones, menús, campos de entrada, esquemas de color, fuentes, iconos y otros elementos visuales para que se vean atractivos y sean fáciles de usar para el usuario final. El diseño de la UI tiene en cuenta la experiencia del usuario, las necesidades y preferencias del usuario, y el contexto en el que se utilizará la interfaz. Un buen diseño de UI es esencial para crear una

experiencia de usuario satisfactoria y aumentar la eficiencia y la efectividad del usuario en la utilización del sistema Web o aplicación. [12]

1.3.7. Frameworks web

Los Frameworks web son un conjunto de herramientas, estilos y librerías dispuestas a través de una estructura o esqueleto base, para el desarrollo de aplicaciones web más escalables y sencillas de mantener. [13] Se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas. [14] Los Frameworks web permiten a los desarrolladores acelerar el trabajo, reducir los errores, fomentar la colaboración y obtener un resultado de más calidad. [13]

Un Framework Web, por tanto, podemos definirlo como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. [13]

Algunas de las ventajas que nos ofrecen estos Frameworks web son:

- Amplia cantidad de documentación con una gran comunidad detrás, respondiendo preguntas y desarrollando nuevas funcionalidades.
- Modularidad de su código y la capacidad para poder hacer múltiples proyectos con el mismo código, cambiando simplemente los textos.
- Cuentan con grandes cantidades de plantillas y componentes desarrollados por su extensa comunidad.
- Cuentan con medidas de seguridad para proteger nuestros datos y los de nuestros clientes.
- Ofrecen el posicionamiento web SEO on page es muy importante si queremos lograr aparecer en las primeras posiciones de buscadores como Google. [13]

1.3.8. IDE'S de desarrollo

Un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Aumenta la productividad de los desarrolladores al combinar capacidades como editar, crear, probar y empaquetar software en una aplicación fácil de usar. [15] Generalmente, un IDE cuenta con las siguientes características:

- Editor de texto que ayuda a escribir el código de software con funciones como el resaltado de la sintaxis con indicaciones visuales.
- Herramientas que automatizan las tareas sencillas y repetitivas, como la compilación del código fuente de la computadora en código binario, el empaquetado de ese código y la ejecución de pruebas automatizadas.
- Depurador, que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica. [16]

1.3.9. Tecnologías de desarrollo web y móvil

a. Flutter

Flutter es un marco de código abierto desarrollado y compatible con Google. Los desarrolladores de Front-End y pila completa utilizan Flutter para crear una interfaz de usuario de aplicación para varias plataformas con un único código base, Flutter es compatible con el desarrollo de aplicaciones en seis plataformas: iOS, Android, web, Windows, MacOS y Linux. Se basa en el desarrollo rápido, El Hot Reload (Recarga Rápida) permite usar un completo set de widgets (componentes) personalizables para crear interfaces nativas de forma veloz y eficiente, así como corregir bugs más rápido. [17]

Flutter consta de dos funcionalidades importantes:

- SDK (Software Development Kit): se trata de una colección de herramientas que permite desarrollar aplicaciones. Esto incluye elementos para compilar código para iOS y Android. [17]
- Framework (Biblioteca de interfaz de usuario basada en widgets): una colección de elementos de interfaz de usuario reutilizables (por ejemplo: botones, entradas de texto, controles deslizantes, entre otros) que pueden personalizarse según lo que requiera el proyecto. [17]

Para desarrollar con Flutter, se utiliza un lenguaje de programación llamado Dart (lenguaje que comparte ciertas similitudes con JavaScript), se enfoca en el desarrollo Front-End y puede usarse para crear aplicaciones móviles y web. [18]

Ventajas

- Flutter permite desarrollar aplicaciones para múltiples plataformas, incluyendo Android, iOS, web y escritorio, desde un solo código base. [17]
- Se puede construir interfaces de usuario a medida utilizando widgets para adaptarse a las necesidades específicas de tu aplicación. [18]
- Se integra bien con IDE'S populares como Visual Studio Code y Android Studio. [18]
- Es de código abierto y está respaldado por Google. [17]
- Tiene una amplia documentación todo está muy detallado con ejemplos sencillos para casos de uso básicos. [18]

b. Vue js

Vue.js es un Framework progresivo creado por Evan You en 2013 mientras trabajaba en Google y lanzado en 2014 tomando las mejores características de Angular y React, gana popularidad casi instantáneamente y se convirtió en el favorito de muchos desarrolladores cuando se trata de crear aplicaciones de una sola página.[19]

Está diseñado para ser flexible y completamente adaptable a las necesidades del desarrollador, a pesar de que su propósito inicial es trabajar el lado Front-End de una aplicación o sitio web, “Es por eso por lo que se llama a Vue "El Framework Progresivo": un marco que puede crecer contigo y adaptarse a tus necesidades” señalan en su página web oficial.[20]

Ventajas

- Una de las principales ventajas de Vue.js es su tamaño, ya que pesa solo 18 KB lo que lo convierte en un Framework ideal para principiantes en el desarrollo de aplicaciones y sitios web, quienes es posible que no cuenten con una gran máquina.
- En una página web con VueJS, cada parte de esta se divide en componentes individuales. Estos componentes pueden estar escritos en HTML, CCS y JavaScript, siendo esta otra ventaja.
- Otra característica ideal para principiantes. Vue.js es fácil de aprender y de dominar en poco tiempo.[19][20]

c. React

ReactJS es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web. Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.[21] Toda aplicación web React se compone de componentes reutilizables que conforman partes de la interfaz de usuario. React también es una aplicación de una sola página. Por tanto, en lugar de enviar una petición al servidor cada vez que hay que renderizar una nueva página, el contenido de la página se carga directamente desde los componentes de React. Esto conduce a una renderización más rápida sin recargas de la página. [22]

Regularmente dentro de la programación web se manejan tres carpetas: la de la estructura de la página web (HTML), la de la presentación de la interfaz (CSS) y la del

comportamiento de esta (JS). Por su lado, React decidió juntar todas estas en un mismo lugar, de manera dinámica, para dar lugar a los componentes.[23]

Ventajas

- React es fácil de aprender y comprender siempre que se conozcan bien los requisitos previos, como es saber JavaScript.
- Tiene una sólida documentación y muchos recursos gratuitos online. creados por otros desarrolladores a través de la muy activa comunidad de React.
- Cada componente en React tiene su propia lógica que puede reutilizarse en cualquier parte de la aplicación.
- La renderización de las páginas puede hacerse más rápidamente. Al utilizar una biblioteca de enrutamiento como React Router.
- Aumenta las posibilidades de conseguir un trabajo. [22][23]

1.3.10. Framework de Diseño

a. Vuetify

Es una biblioteca de interfaz de usuario de Vue que ofrece componentes de Material Design Framework destinada a la creación de interfaces de usuario que implementa componentes de usabilidad listos para usar.[24]

Utiliza un sistema de cuadrícula para organizar diseños de página que pueden adaptarse a una gran cantidad de resoluciones de pantalla. Tiene una enorme biblioteca de elementos, desde botones, sliders, diferentes tipos de entradas hasta elementos más avanzados como mapas, lo que nos permitirá hacer toda una gama de elementos interactivos con el usuario.[25]

El objetivo primordial de este Framework es proporcionar a los desarrolladores las herramientas que necesitan para crear experiencias de usuario enriquecedoras y atractivas.[25]

b. Bootstrap

Bootstrap es una biblioteca de herramientas de código abierto optimizada para diseñar sitios web y aplicaciones. La plataforma está basada en lenguajes HTML y CSS e incluye una amplia gama de elementos de diseño como formularios, botones y menús que se adaptan a diferentes formatos de navegación. [26]

Su código está optimizado para dispositivos con pantallas pequeñas e interfaces interactivas sensibles al tacto. A partir de aquí, Bootstrap se aplica a otros formatos, como los de pantallas de escritorio. Por lo tanto, construir con Bootstrap asegurará que su sitio se cargue correctamente y le permitirá crear interfaces verdaderamente interactivas.[26]

c. Tailwind

Tailwind es un Framework CSS que, a diferencia de los Frameworks tradicionales como Bootstrap, no usa una sola clase para crear componentes, sino que tiene algo llamado Utility Classes, que son clases especiales para cada cosa. Por ejemplo, una clase para texto, otra para residuos, una para colores, entre otros. El objetivo es permitirle personalizar completamente. [27]

La ventaja de usar Tailwind entre otros Framework de diseño es que puedes personalizar todo a tu gusto y no al gusto del Framework, como ocurre con Bootstrap y otros similares. puedes personalizar todo a tu gusto y no al gusto del Framework, como ocurre con Bootstrap y otros similares.[27]

1.3.11. Base de datos

Una base de datos se define como una colección de archivos interrelacionados, permite almacenar información de diversas entidades. Estos archivos se pueden considerar como una colección de registros, siendo cada registro una combinación de campos que almacenan atributos de entidades del mundo real. En el contexto de una base de datos, la

colección de archivos se refiere a las tablas, donde las filas representan los registros de dichas entidades y las columnas corresponden a los campos asociados a ellas. [28]

También se ve como un almacén de datos relacionados con diferentes modos de organización. Una base de datos representa algunos aspectos del mundo real, aquellos que le interesan al usuario, y almacena datos con un propósito específico. Un dato hace referencia a la unidad mínima indivisible de información no procesada, donde un conjunto de datos forma información y un conjunto de información forma conocimiento. [29]

1.3.12. Diseño de base de datos

La Gestión de Bases de Datos (Database Management) es el conjunto de técnicas y tecnologías que se utilizan para administrar, organizar y mantener una colección de datos, que se almacenan en una base de datos. Esto incluye la definición de la estructura de la base de datos, la inserción, actualización y eliminación de datos, la seguridad y el acceso a los datos, el monitoreo y la optimización del rendimiento, entre otras actividades. La gestión de bases de datos es esencial en aplicaciones informáticas que manejan grandes cantidades de información, como sistemas de gestión de inventarios, sistemas de información de recursos humanos, sistemas de información geográfica, entre otros.[30]

1.3.13. SQL Server

SQL Server es un sistema de gestión de datos, que permite, diseñar y administrar Bases de Datos Relacionales.[31] Es el servidor de bases de datos de Microsoft, seguro, robusto y con las más avanzadas prestaciones: transacciones, procedimientos almacenados, trigger. [32]

Tiene como principal función la administración de la información de una determinada plataforma. De tal forma que dicha información esté disponible para el servidor y para los clientes. Microsoft SQL Server se estructura principalmente alrededor de tablas basadas en filas que conectan todos los elementos de los datos relacionados en diferentes tablas

entre sí. Con ello se evita el almacenamiento de datos de forma redundante en varios lugares dentro de una misma base de datos. [33]

1.3.14. MySQL

MySQL es un sistema de gestión de bases de datos relacionales de código abierto que se utiliza para almacenar, organizar y administrar grandes cantidades de datos. Es una herramienta esencial para la programación de sistemas y se utiliza en una amplia variedad de aplicaciones, desde sitios Web hasta aplicaciones empresariales. MySQL es altamente escalable y se puede utilizar en sistemas operativos Windows, Linux y Mac OS X. Además, es compatible con una amplia variedad de lenguajes de programación, incluyendo PHP, Java, Python y C++. Si está trabajando en una tesis relacionada con la gestión de bases de datos, MySQL es una excelente opción para considerar.[34]

1.3.15. Concepto de metodología ágil

"Agile" es mucho más que una metodología para el desarrollo de proyectos que requieren rapidez y flexibilidad; es una filosofía que implica una forma diferente de trabajar y organizarse. En esta metodología, cada proyecto se divide en pequeñas partes que deben completarse y entregarse en pocas semanas. El objetivo es desarrollar productos y servicios de calidad que se ajusten a las necesidades de los clientes, cuyas prioridades cambian a un ritmo cada vez más rápido.[35]

1.3.16. Extreme Programming (XP)

La metodología XP dio impulso al movimiento actual de las metodologías ágiles. Entre los principios más importantes de esta metodología, se pueden mencionar los siguientes: en cada iteración se determina el alcance de la próxima iteración, combinando prioridades del negocio y estimaciones técnicas, definiendo la estrategia de planificación durante el proceso de desarrollo. Las entregas son frecuentes y continuas. Cada versión liberada debe

ponerse en producción rápidamente. El desarrollo es guiado a través de historias de usuario simples. El cliente o un representante del cliente se integra al equipo de desarrollo, estando disponible en todo momento para responder preguntas del equipo relacionadas con las reglas de negocio. El cliente es responsable de escribir los casos de prueba, demostrando que la funcionalidad está finalizada. Se recomienda que el desarrollo de las funciones del producto sea realizado por dos personas en el mismo puesto, practicando la programación en pares. Una vez finalizada la implementación de una funcionalidad, se deben corregir todos los defectos encontrados. [36]

Ciclo de XP

- **Planificación:** Según la identificación de las historias de usuario, se priorizan y se descomponen en mini-versiones.
- **Diseño:** Se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Además, para el diseño del software orientado a objetos, se crearán tarjetas CRC (Clase-Responsabilidad-Colaboración).
- **Codificación:** La programación aquí se hace «a dos manos», en parejas en frente del mismo ordenador. Incluso, a veces se intercambian las parejas. De esta forma, nos aseguramos de que se realice un código más universal, con el que cualquier otro programador podría trabajar y entender.
- **Pruebas:** Se deben realizar pruebas automáticas continuamente. El propio cliente puede hacer pruebas, proponer nuevas pruebas e ir validando las mini-versiones.
- **Lanzamiento:** Tenemos un software útil y podemos incorporarlo en el producto.

[36]

1.3.17. Rapid Application Development (RAD)

RAD es una metodología que se centra en desarrollar aplicaciones rápidamente por medio de iteraciones frecuentes y aprobaciones con comentarios continuos de los clientes. Al priorizar los lanzamientos de prototipos ágiles y rápidos, RAD incide en la usabilidad del

software, los comentarios de los usuarios y la entrega rápida, gracias a su rapidez y agilidad, la popularidad de RAD va en aumento. [37]

Ciclo de RAD

- **Recopilación de requisitos:** se debe utilizar herramientas como entrevistas y encuestas para recolectar requisitos fundamentales para el sistema.
- **Prototipado:** con los requisitos identificados se crea un prototipo del sistema. Este prototipo sirve para probar los requisitos y obtener una retroalimentación por parte del cliente.
- **Construcción:** si el prototipo es aprobado se procede a construir el software de manera iterativa (construyendo y probando) en bloques pequeños.
- **Pruebas:** en la penúltima etapa se prueba el sistema verificando que cumpla con los requisitos.
- **Implementación:** si el sistema cumple todos los requisitos procede a implementar en producción. [38]

1.3.18. Scrum

La metodología Scrum permite abordar proyectos complejos desarrollados en entornos dinámicos y cambiantes de un modo flexible. Está basada en entregas parciales llamadas sprint (mini-proyecto con una duración no mayor a un mes que se interconecta con otros mini-proyectos) y regulares del producto final en base al valor que ofrecen a los clientes. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. [39]

El tamaño óptimo de un Equipo Scrum es generalmente de seis a diez personas. Esto asegurará que el Equipo Scrum sea suficientemente grande como para poseer las

habilidades necesarias para entregar el proyecto y suficientemente pequeño como para colaborar. [40]

Ciclo de Scrum

- **Inicio:** Se encarga de estudiar y analizar el proyecto identificando las necesidades básicas del sprint, Las preguntas a hacer en la fase de inicio son:
 - ¿Qué quiero?
 - ¿Cómo lo quiero?
 - ¿Cuándo lo quiero?
- **Planificación y estimación:** Esta fase incluye crear, estimar y comprometer historias de usuario, Identificar y estimar tareas, Crear el sprint backlog o iteración de tareas, esto ayudará a establecer metas fijas y a cumplir con los plazos.
- **Implementación:** Se cumplen los siguientes procesos: crear entregables, realizar daily stand-up, Refinanciamiento del backlog priorizado del producto; En la fase de implementación no deberían hacerse cambios innecesarios de última hora, con excepción de si es un movimiento que será clave para el éxito del sprint, se lo deberá discutir con el grupo y los stakeholders.
- **Revisión y retrospectiva:** En esta fase se deberá hacer la revisión del proceso, que no es más que la autocrítica o evaluación interna del grupo respecto a su propio trabajo. Entre los pasos más importantes para realizar en esta fase tenemos: Demostrar y validar el sprint y Retrospectiva del sprint.
- **Lanzamiento:** Desenlace del proyecto y entrega del producto, donde se deberá cumplir con 2 únicas tareas que son: enviar entregables y enviar retrospectiva del proyecto. [41]

1.4. Objetivos

1.4.1. Objetivo general

Implantar una aplicación Web para la gestión de la carpeta académica de los estudiantes en la FISEI de la Universidad Técnica de Ambato.

1.4.2. Objetivos específicos

- Analizar el proceso de gestión de la carpeta estudiantil en las diferentes coordinaciones de la FISEI de la Universidad Técnica de Ambato.
- Desarrollar un prototipo de aplicación Web que gestione la carpeta estudiantil de los estudiantes de la FISEI.
- Desarrollar una aplicación Web multiplataforma para gestión de la carpeta académica de los estudiantes de la FISEI.

CAPÍTULO II. METODOLOGÍA.

2.1. Materiales

Para el presente trabajo de investigación se realizará una entrevista a la secretaria general de la FISEI, para conocer los procesos respectivos que se realizan actualmente en la creación de la carpeta académica de los estudiantes, en caso de ejecutarse de manera errónea la carpeta de un estudiante quedaría incompleta impidiéndole el proceso de titulación. Por otra parte, se aplicará a una encuesta a las demás secretarías encargadas del proceso de titulación de cada carrera para conocer la efectividad y opinión acerca del sistema planteado.

El cuestionario realizado para la encuesta y la guía de la entrevista se encuentran en los Anexo A y Anexo B, respectivamente.

2.1.1. Diagrama de procesos de la gestión de la carpeta estudiantil

Proceso de gestión la carpeta académica.

- 1) La secretaria encargada crea una carpeta con el nombre del estudiante para poder almacenar los documentos desde su inicio en la universidad.
- 2) Registra los documentos (Información personal del estudiante) pdf en la carpeta, donde:
 - a. Si no tiene proceso titulación sube la información básica (todavía no está en el último nivel de la carrera).
 - b. Si tiene proceso de titulación se notificará a titulación.
- 3) La secretaria de titulación se encarga sube los documentos de requisitos habilitantes.
- 4) La secretaria de titulación registra la documentación de prácticas y vinculación, donde:
 - a. Si se graduó, la secretaria general de carrera registrara el Acta y el título.

b. Si no se graduó la secretaria de carrera registra el acta y las novedades.

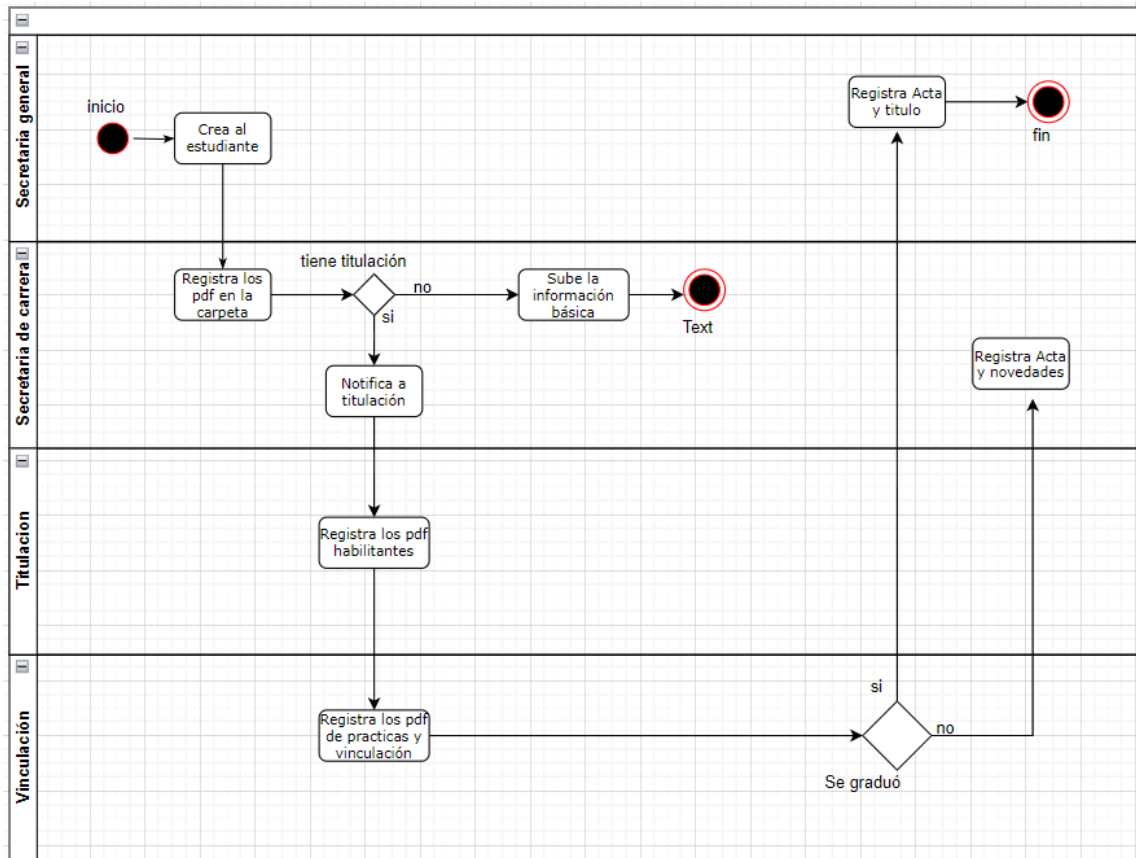


Figura 1 Diagrama de procesos de la gestión de la carpeta estudiantil

2.1.2. Modalidad de Investigación

La investigación se realizará con base en un estudio de campo y bibliográfico, como se detalla a continuación:

a. Investigación de campo

Se ha optado por esta modalidad debido a que se llevará a cabo una recopilación de información directamente con las secretarias encargadas del proceso de titulación de los estudiantes de la FISEI, a través de la realización de entrevistas.

b. Investigación bibliográfica

La presente investigación adopta un enfoque bibliográfico, ya que resulta imperativo recopilar información proveniente de fuentes como artículos académicos, tesis y libros. Estos recursos fungirán como pilares de apoyo, para contextualizar adecuadamente la propuesta que se pretende desarrollar.

c. Investigación aplicada

La naturaleza de la investigación es aplicada, ya que se utilizarán los conocimientos adquiridos a lo largo de la carrera en el desarrollo de la propuesta.

2.1.3. Población y muestra

La población para estudiar en este caso serán las secretarias encargadas de gestionar la información académica de los estudiantes de la FISEI. Se identificó un total de 8 personas. En este trabajo se tomará en cuenta la población completa.

Tabla 1: Población

Grupo	Población
Secretarias	8
Total	8

2.1.4. Recolección de Información

Resultados de la entrevista aplicada a la secretaria general de la FISEI.

<p>Estimada secretaria general</p> <p>La información recolectada se usará con fines académicos. Instrucciones: Responder las preguntas con total honestidad.</p> <p>Entrevistador: Paulo Martinez</p>			
N°	Pregunta	Respuesta	Observación
1	¿Cuál es el proceso actual de gestión de la carpeta académica de los estudiantes en la FISIE?	Todo el proceso es manual, se crean carpetas físicas para cada documento como cédula, notas de inglés, entre otras, donde se va almacenando la documentación de cada estudiante referente a la carpeta creada.	El sistema deberá permitir crear plantillas que tendrá las subcarpetas como: cédula, inglés, que llevará los estudiantes.
2	¿Cuáles son las tareas específicas que realizan las secretarias en relación con la carpeta académica de los estudiantes?	<p>Crear las carpetas y subcarpetas que tendrá cada usuario para el registro de su documentación.</p> <p>De acuerdo con la unidad académica o al puesto deben subir la documentación que se requiera para cada carpeta.</p>	
3	¿Qué tipo de información se almacena en la carpeta académica de los estudiantes?	Se almacenan documentos en formato PDF.	El sistema deberá permitir subir cualquier tipo de documento.
4	¿Cómo se manejan los documentos físicos en el proceso de gestión de la carpeta académica?	Se lleva de dos maneras, la primera de manera física y la segunda se escanea y se sube a una carpeta en la nube.	

<p>Estimada secretaria general</p> <p>La información recolectada se usará con fines académicos. Instrucciones: Responder las preguntas con total honestidad.</p> <p>Entrevistador: Paulo Martinez</p>			
N°	Pregunta	Respuesta	Observación
5	<p>¿Qué características o funcionalidades consideran esenciales para una aplicación web de gestión de la carpeta académica?</p>	<ul style="list-style-type: none"> • Que sea fácil de usar, que permita migrar información de estudiantes antiguos ya que las carreras tienen ese inconveniente. • Que permita manejar por dependencia o puesto de trabajo una misma carpeta. • Que permita crear dinámicamente las carpetas para los estudiantes. • Que permita crear una plantilla a ser utilizada. Teniendo en cuenta que la plantilla es un formato ya establecido de carpetas que se podría usar o aplicar a un determinado estudiante. 	<p>La plantilla será utilizada para crear las subcarpetas del estudiante dependiendo de tipo de estudiante, sea antiguo o nuevo.</p>
6	<p>¿Requieren la capacidad de importar datos desde archivos externos?</p>	<p>Si ya que se requerirá la migración de datos masivos de todos los estudiantes.</p>	
7	<p>¿Considera necesario un sistema de gestión de permisos y roles para los usuarios de la aplicación?</p>	<p>Si se considera necesario el manejo de roles debido a que no todos los usuarios deben tener acceso a todas las funcionalidades del sistema y deberá existir una persona que otorgue los debidos permisos que los demás usuarios pueden hacer en el sistema.</p>	<p>Se plantea manejar con roles de un usuario administrador y usuarios secretarias.</p>

<p>Estimada secretaria general</p> <p>La información recolectada se usará con fines académicos. Instrucciones: Responder las preguntas con total honestidad.</p> <p>Entrevistador: Paulo Martinez</p>			
N°	Pregunta	Respuesta	Observación
8	¿Desean una función de auditoría para rastrear cambios y accesos a los registros académicos?	Si, es importante para poder verificar quién modificó un determinado documento en caso de existir un altercado.	
9	¿Qué tipo documentos debe manejar aplicación?	PDF, imágenes, archivos de Word, y dejar habilitado para ser necesario subir incluso otro tipo de archivos.	
<p>Conclusión: Al finalizar la entrevista, se ha recopilado información invaluable que será fundamental para identificar los requerimientos necesarios para la implementación de un sistema destinado a gestionar la carpeta académica.</p>			

2.1.5. Procesamiento y análisis de datos

En esta sección, se realiza un análisis detallado de los resultados obtenidos a partir de las encuestas realizadas a las secretarías de FISEI, responsables del proceso de titulación.

1) ¿Ha tenido la oportunidad de utilizar una aplicación web para la gestión de la carpeta académica?

Tabla 2: Tabla de frecuencias - Pregunta 1

Indicador	Frecuencia	Porcentaje
SI	8	88,9%
No	1	11,1%
Suma Total	9	100%

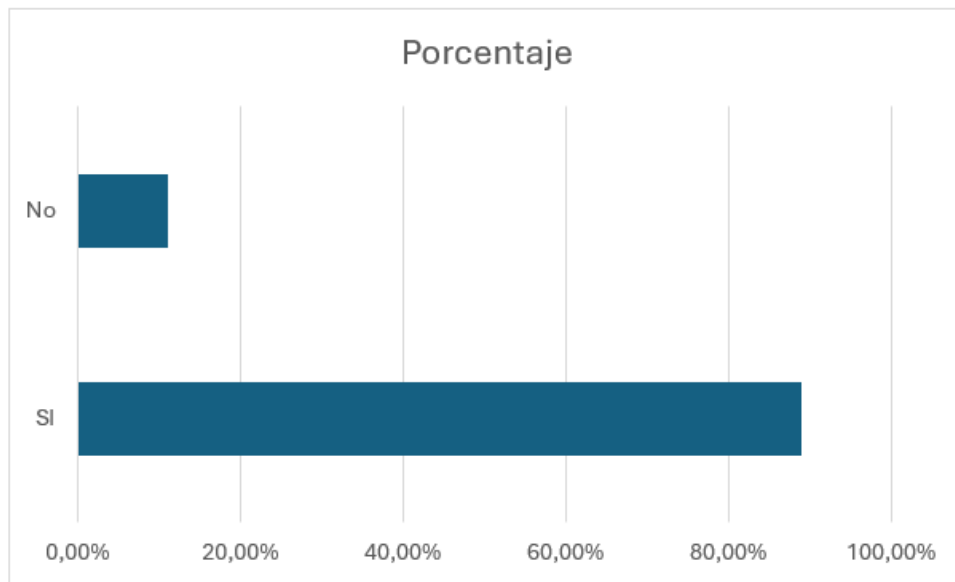


Figura 2: Resultados de la encuesta - Pregunta 1

Análisis e interpretación de datos

Analysis:

- Se recopilaron un total de 9 respuestas para el indicador dado.
- De estas respuestas, 8 personas (88.9%) respondieron "Sí" y 1 persona (11.1%) respondió "No".

Interpretación:

- La mayoría de los encuestados, específicamente el 88.9%, respondieron afirmativamente al indicador.
- Solo el 11.1% de los encuestados respondieron negativamente al indicador.
- Esto sugiere que la mayoría de los encuestados están en acuerdo con lo que representa el indicador, mientras que una minoría significativa no lo está.

2) ¿Qué tipo de herramienta utilizas actualmente para gestión de la carpeta académica?

Tabla 3: Tabla de frecuencias - Pregunta 2

Indicador	Frecuencia	Porcentaje
Hojas de cálculo (como Excel o Google Sheets)	9	100%
Plataforma de gestión académica de la universidad	0	0%
Otros	0	0%
Suma Total	9	100%

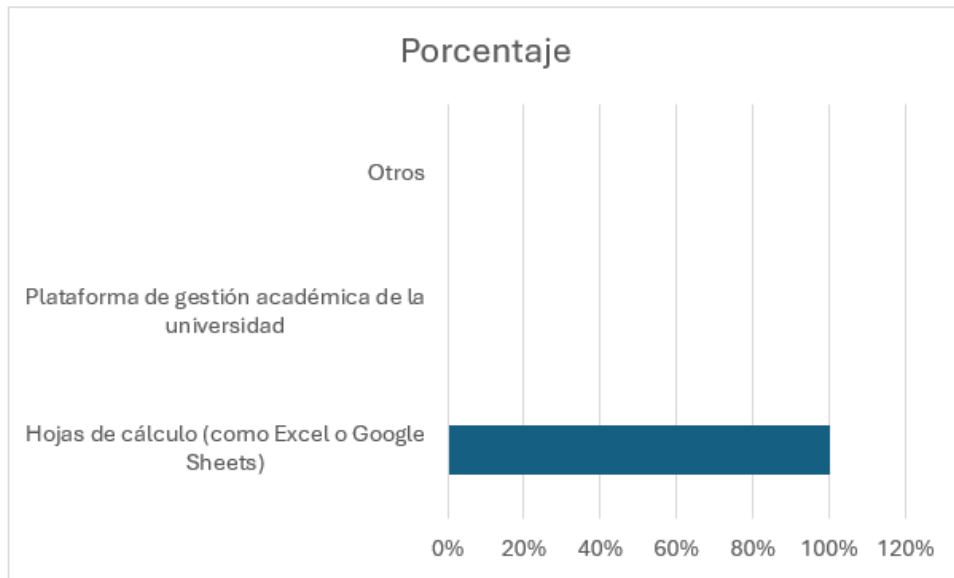


Figura 3: Resultados de la encuesta - Pregunta 2

Análisis e interpretación de datos

Análisis:

La Figura 2 indica que el 100% de los encuestados usan actualmente hojas de cálculo para gestionar su carpeta académica, en contraste con el 0% de plataformas de gestión académica de la universidad, seguido por 0% de otras herramientas.

Interpretación:

- La totalidad de los encuestados depende principalmente de hojas de cálculo, como Excel o Google Sheets, para gestionar sus carpetas académicas. Esto puede deberse a la familiaridad, facilidad de acceso o funcionalidades que estas herramientas ofrecen.
- La falta de uso de la plataforma de gestión académica de la universidad podría indicar una posible oportunidad de mejora en la promoción o la funcionalidad de esta herramienta por parte de la institución.
- La ausencia de otras herramientas alternativas sugiere que los encuestados pueden no estar al tanto de opciones adicionales o que las herramientas alternativas no son adecuadas para sus necesidades.

3) ¿Cómo calificarías la efectividad de la herramienta actual en la gestión de tu carpeta académica?

Tabla 4: Tabla de frecuencias - Pregunta 3

Indicador	Frecuencia	Porcentaje
Muy efectiva	0	0%
Efectiva	0	0%
Neutral	0	0%
Poco efectiva	9	100%
No efectiva	0	0%
Suma Total	9	100%

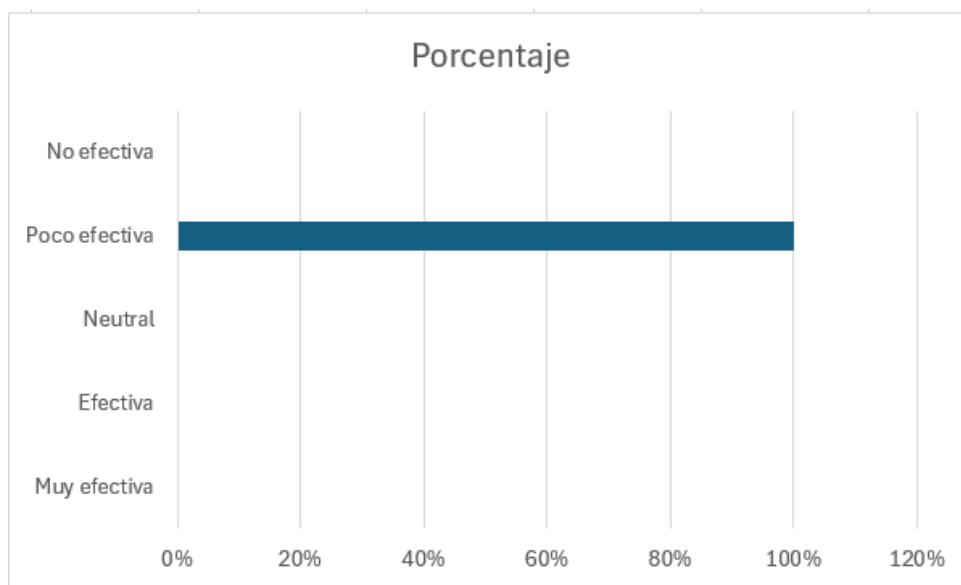


Figura 4: Resultados de la encuesta - Pregunta 3

Análisis e interpretación de datos

Análisis:

La Figura 3 indica que el 100% de los encuestados consideran que las herramientas actuales de la gestión de carpeta académica son poco efectivas en este proceso.

Interpretación:

- La totalidad de los encuestados considera que la herramienta actual para la gestión de la carpeta académica es poco efectiva. Esto sugiere que existe un consenso generalizado entre los encuestados de que la herramienta actual no cumple adecuadamente con sus necesidades o expectativas en términos de eficacia.
- La ausencia de calificaciones en las categorías de "Muy efectiva", "Efectiva", "Neutral" o "No efectiva" indica una falta de diversidad en las percepciones de los encuestados sobre la efectividad de la herramienta. Todos los encuestados parecen estar de acuerdo en que la herramienta actual es poco efectiva.

4) ¿Está usted de acuerdo en la implementación de una aplicación web para la gestión de la carpeta académica en términos de mejorar la eficiencia de las tareas administrativas en la Facultad?

Tabla 5: Tabla de frecuencias - Pregunta 4

Indicador	Frecuencia	Porcentaje
Si	9	100%
No	0	0%
No estoy seguro	0	0%
Suma Total	9	100%

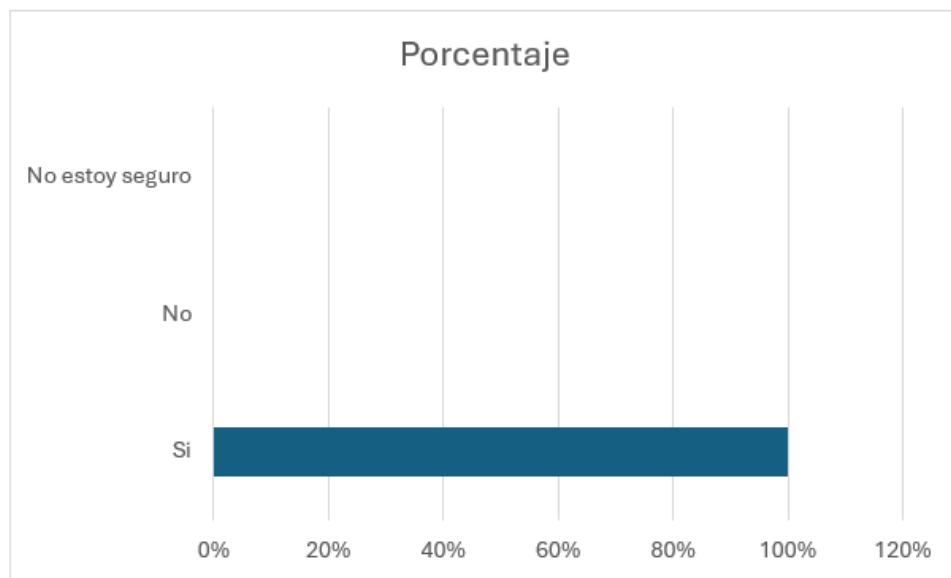


Figura 5: Resultados de la encuesta - Pregunta 4

Análisis e interpretación de datos

Análisis:

La Figura 4 indica que el 100% de los encuestados consideran que la implementación de una aplicación web para la gestión de la carpeta académica mejoraría significativamente este proceso.

Interpretación:

- Los resultados muestran un fuerte apoyo entre los encuestados hacia la implementación de una aplicación web para la gestión de la carpeta académica. Este alto nivel de acuerdo puede indicar una percepción generalizada de que esta tecnología sería beneficiosa para mejorar la eficiencia en las tareas administrativas en la Facultad.
- La falta de respuestas negativas o de incertidumbre sugiere un consenso claro entre los encuestados sobre el potencial beneficio de una aplicación web para la gestión académica.

5) ¿Cuáles son los principales desafíos o dificultades que encuentras al gestionar las carpetas académicas de los estudiantes?

Tabla 6: Tabla de frecuencias - Pregunta 5

Indicador	Frecuencia	Porcentaje
Pérdida de documentos	8	88,9%
Falta de seguimiento y actualización en tiempo real	5	55,6%
Dificultades para acceder a la información	6	66,7%
Falta de automatización	3	33,3%
Otros	0	0%
Suma Total	22	244,5%

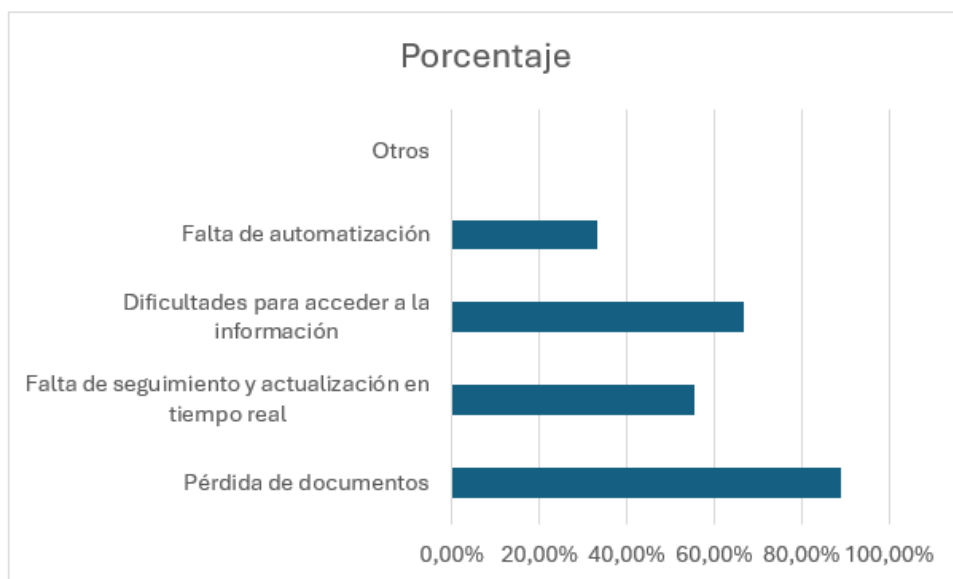


Figura 6: Resultados de la encuesta - Pregunta 5

Análisis e interpretación de datos

Análisis:

La Figura 5 revela que un significativo 88.9% de los encuestados identifican la pérdida de documentos como una de las principales complicaciones al gestionar la carpeta académica. En segundo lugar, el 55.36% señala la falta de seguimiento y actualización en tiempo real como otro desafío destacado, seguido por el 66.7% que enfrenta dificultades para acceder a la información y el 33.3% que percibe la falta de automatización como un obstáculo. Estos resultados subrayan la importancia de abordar estos problemas para mejorar eficazmente la gestión de la carpeta académica.

Interpretación:

- Los resultados subrayan la importancia de abordar la pérdida de documentos como el principal desafío en la gestión de carpetas académicas, seguido de cerca por la dificultad para acceder a la información.
- Estos hallazgos resaltan la necesidad de implementar soluciones que mejoren la organización, la accesibilidad y la actualización en tiempo real de la información

académica, así como la automatización de procesos para optimizar la eficiencia en la gestión de carpetas académicas.

6) ¿Qué características o funcionalidades te gustaría ver en una aplicación web de gestión académica?

Tabla 7: Tabla de frecuencias - Pregunta 6

Indicador	Frecuencia	Porcentaje
Creación de directorios para cada usuario	9	100%
Plantillas para la creación de carpetas académicas	0	0%
Notificaciones en caso de falta de documentos en alguna carpeta académica	5	55,6%
Importación de grandes cantidades de datos al sistema	4	44,4%
Subida de diferentes archivos al sistema	3	33,3%
Otro	5	55,6%
Suma Total	26	288,9%

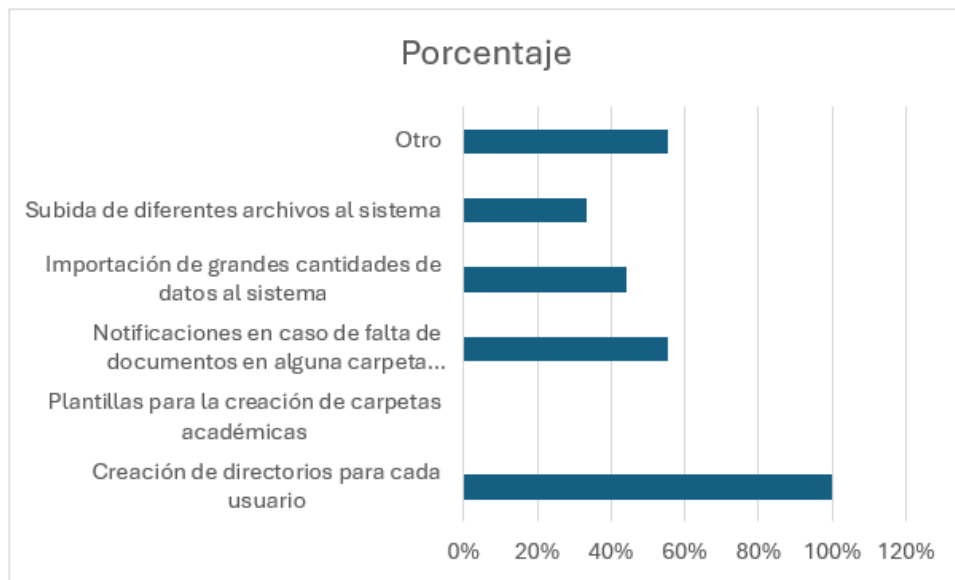


Figura 7: Resultados de la encuesta - Pregunta 6

Análisis e interpretación de datos

Análisis:

El análisis de la Figura 6 destaca áreas clave en la gestión de carpetas académicas. La creación de directorios para cada usuario es universal (100%), mientras que la falta de plantillas es una oportunidad de mejora (0%). Notificaciones por falta de documentos en carpetas académicas son esenciales (55.6%). Importar grandes cantidades de datos (44.4%) y subir diferentes archivos (33.3%) son desafíos. El indicador "Otro" (55.6%) sugiere problemáticas específicas. Estos hallazgos ofrecen una visión detallada para identificar soluciones y mejoras.

Interpretación:

- La creación de directorios personalizados es una característica altamente deseada por todos los encuestados, lo que subraya la importancia de la organización individualizada en la gestión académica.
- Las notificaciones sobre la falta de documentos y otras características relacionadas con la gestión documental son aspectos valorados por más de la mitad de los encuestados, lo que indica una preocupación por la accesibilidad y disponibilidad de la información.
- La importación eficiente de datos es una necesidad percibida por una proporción significativa de encuestados, lo que destaca la importancia de la facilidad de uso y la eficiencia en la interacción con la aplicación.

7) ¿Qué ventajas cree que podría tener una aplicación web multiplataforma para la carpeta académica en comparación con las herramientas actuales que maneja para este proceso?

Tabla 8: Tabla de frecuencias - Pregunta 7

Indicador	Frecuencia	Porcentaje
Optimización de tiempo en la creación de la carpeta académica	9	100%
Acceso desde cualquier lugar	2	22,2%
Mejor organización de los documentos y archivos	5	55,6%
Seguridad y control de acceso	1	11,1%
Suma Total	27	188,9%

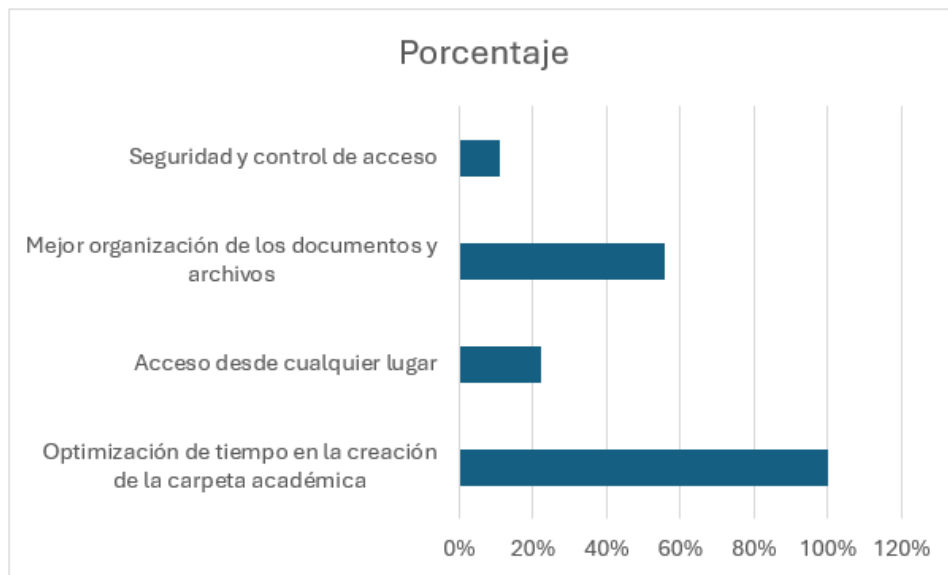


Figura 8: Resultados de la encuesta - Pregunta 7

Análisis e interpretación de datos

Análisis:

El análisis de la Figura 7 revela que la optimización del tiempo en la creación de la carpeta académica es una prioridad universal, con el 100% de los participantes destacándolo. La mejora en la organización de documentos es también crucial, con un 55.6% de frecuencia. Aunque el acceso desde cualquier lugar tiene una importancia menor con un 22.2%, sigue siendo un factor significativo. La seguridad y el control de acceso, aunque menos

destacados, preocupan al 11.1% de los participantes. En conjunto, estos resultados resaltan la necesidad de abordar eficientemente la creación de carpetas y la organización de documentos para mejorar la gestión académica.

Interpretación:

- La optimización del tiempo en la creación de carpetas académicas es la característica más valorada por los encuestados, lo que sugiere una necesidad clara de eficiencia en la gestión académica.
- La mejor organización de documentos y archivos también es altamente valorada, lo que indica una preocupación por la estructura y la accesibilidad de la información.
- Aunque el acceso desde cualquier lugar y la seguridad/control de acceso reciben menos énfasis, aún son considerados relevantes por una parte de los encuestados, lo que sugiere una diversidad de necesidades y prioridades en cuanto a funcionalidades de la aplicación web.

8) ¿Qué desafíos consideraría al usar una nueva aplicación web para gestionar la carpeta académica?

Tabla 9: Tabla de frecuencias - Pregunta 8

Indicador	Frecuencia	Porcentaje
Dificultad de usabilidad del sistema	9	100%
Robo de información	3	33,3%
Sistema no cumpla con los requerimientos buscados	4	44,4%
Incompatibilidad con navegadores	1	11,1%
Otra	0	0%
Suma Total	17	188,8%

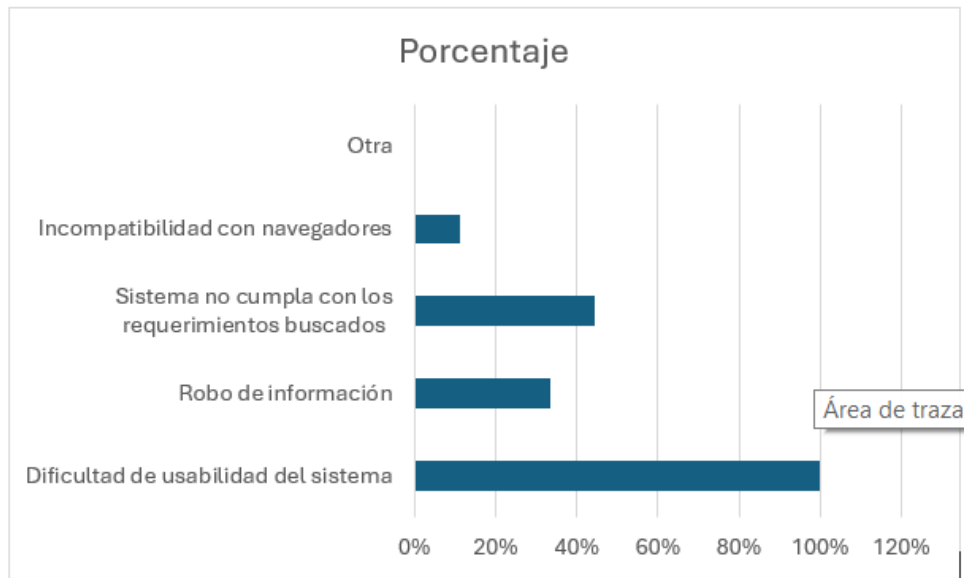


Figura 9: Resultados de la encuesta - Pregunta 8

Análisis e interpretación de datos

Análisis:

El análisis de la Figura 8 proporciona una visión detallada de las percepciones de los usuarios sobre la dificultad de usabilidad del sistema. El 100% de los participantes expresaron enfrentar este desafío, lo que indica una preocupación universal en cuanto a la experiencia de usuario. Además, el 33.3% reporta preocupaciones por el robo de información, mientras que el 44.4% señala que el sistema no cumple con los requerimientos buscados. La incompatibilidad con navegadores afecta al 11.1% de los usuarios encuestados.

Interpretación:

- Los resultados indican que la usabilidad del sistema es un área crítica que requiere atención inmediata para mejorar la experiencia del usuario.
- Las preocupaciones sobre la seguridad de los datos y la compatibilidad con los navegadores también deben abordarse para garantizar la confianza y la funcionalidad del sistema.

- La discrepancia entre los requerimientos de los usuarios y las características ofrecidas por el sistema destaca la importancia de involucrar a los usuarios en el proceso de diseño y desarrollo para satisfacer mejor sus necesidades y expectativas.

9) ¿Qué tipo de dispositivo electrónico utiliza con mayor frecuencia para sus actividades laborales?

Tabla 10: Tabla de frecuencias - Pregunta 9

Indicador	Frecuencia	Porcentaje
Computadoras de escritorio	9	100%
Computadora portátil	0	0%
Tableta	0	0%
Teléfono móvil	0	0%
Suma Total	9	100%

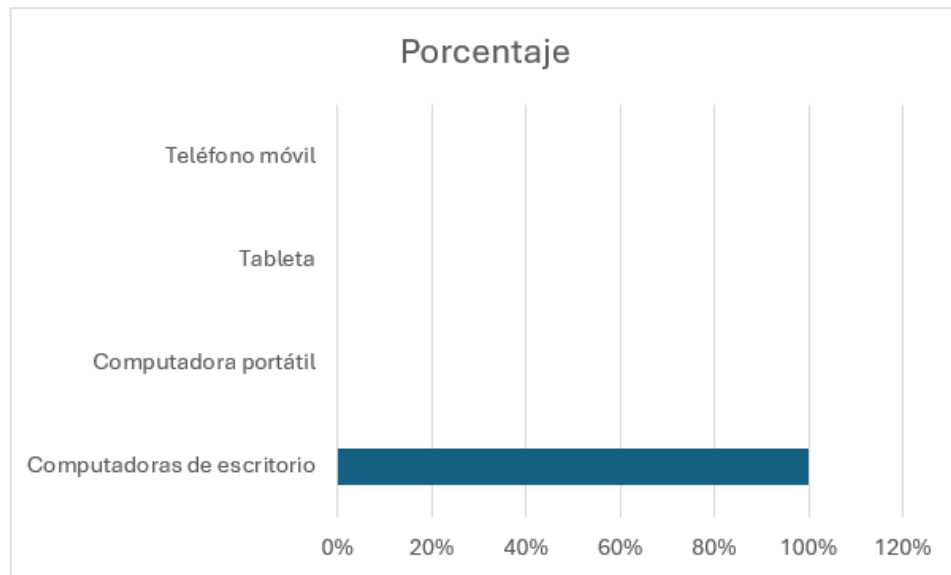


Figura 10: Resultados de la encuesta - Pregunta 2

Análisis e interpretación de datos

Análisis:

La Figura 9 muestra que el 100% de los encuestados utilizaran las computadoras de escritorio para acceder a la carpeta académica, mientras que no se reportó el uso de computadoras portátiles, tabletas o teléfonos móviles. Esta información es crucial al considerar mejoras en la accesibilidad y la interfaz de usuario de la carpeta académica, ya que sugiere que las soluciones deben centrarse en la experiencia de usuario de las computadoras de escritorio.

Interpretación:

- El hecho de que todos los encuestados utilicen computadoras de escritorio como su dispositivo principal para actividades laborales sugiere una preferencia generalizada por este tipo de dispositivo en este contexto.
- La ausencia de uso de computadoras portátiles, tabletas o teléfonos móviles como dispositivos principales puede indicar que estos dispositivos no son tan adecuados o necesarios para las tareas laborales de los encuestados en esta muestra particular.

10) ¿Cuál considera que sería el impacto más importante de esta aplicación en la gestión académica?

Tabla 11: Tabla de frecuencias - Pregunta 10

Indicador	Frecuencia	Porcentaje
Reducción de errores y duplicación de datos	8	88,9%
Mejora en la comunicación entre estudiantes, profesores y personal administrativo	0	0%
Mayor acceso a información relevante	1	11,1%
Mayor capacidad de programación y organización	0	0%
Suma Total	9	100%

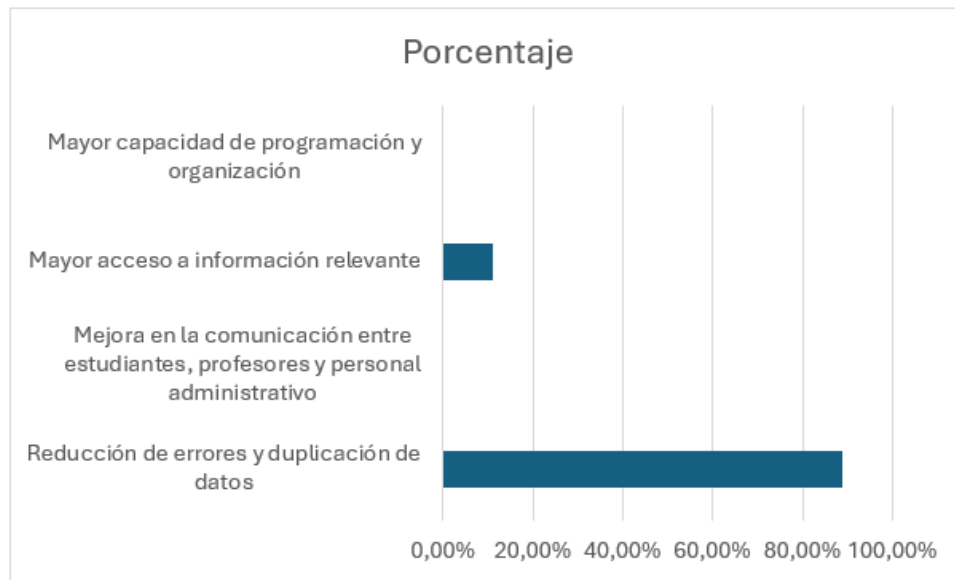


Figura 11: Resultados de la encuesta - Pregunta 10

Análisis e interpretación de datos

Análisis:

La Figura 10 muestra que el 88.9% de los participantes identifica la "Reducción de errores y duplicación de datos" como el principal beneficio esperado de la gestión de la carpeta académica. Aunque ninguna respuesta destaca la "Mejora en la comunicación entre estudiantes, profesores y personal administrativo", un 11.1% valora el "Mayor acceso a información relevante". Sorprendentemente, ninguna respuesta indica la "Mayor capacidad de programación y organización". Este análisis sugiere que la prioridad principal se centra en la eficiencia y precisión de los datos, mientras que el potencial de mejora en la comunicación y la organización podría requerir una mayor atención para optimizar la gestión de la carpeta académica.

Interpretación:

- La mayoría de los encuestados perciben que la reducción de errores y duplicación de datos sería el impacto más significativo de la aplicación en la gestión académica,

lo que subraya la importancia de mejorar la precisión y la eficiencia en la gestión de la información académica.

- Aunque una minoría menciona el mayor acceso a información relevante como un impacto importante, sigue siendo una consideración valiosa que resalta la importancia de la accesibilidad y disponibilidad de la información en la gestión académica.

11) ¿Qué factores considera usted que podrían afectar el tiempo necesario para encontrar información en comparación de los métodos actuales, mediante una aplicación web?

Tabla 12: Tabla de frecuencias - Pregunta 11

Indicador	Frecuencia	Porcentaje
Reduciría moderadamente el tiempo	9	100%
No tendría un impacto significativo	0	0%
Incrementaría moderadamente el tiempo	0	0%
Incrementaría significativamente el tiempo	0	0%
Suma Total	9	100%

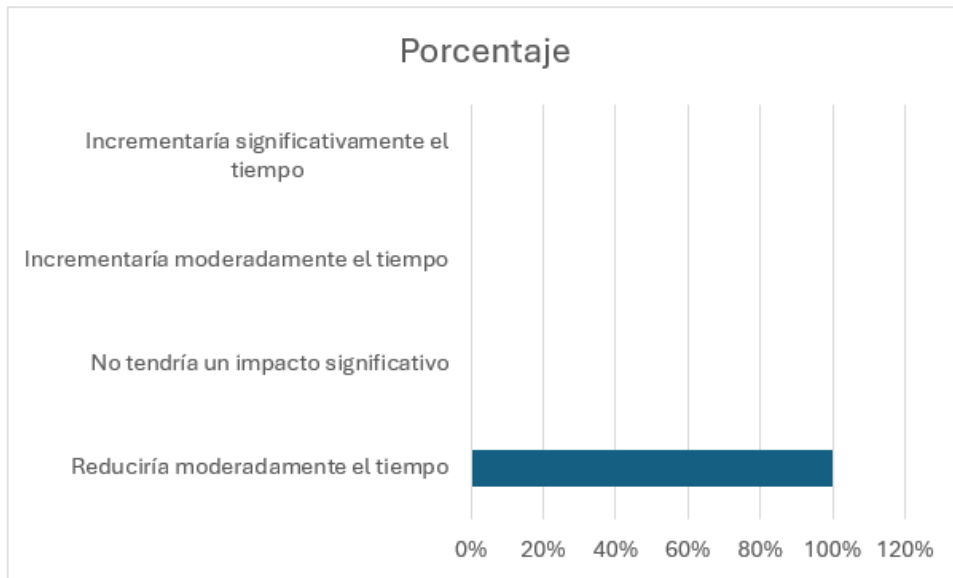


Figura 12: Resultados de la encuesta - Pregunta 11

Análisis e interpretación de datos

Análisis:

El análisis de la Figura 11 revela que el 100% de los participantes indican que la implementación reduciría moderadamente el tiempo, Estos resultados destacan la viabilidad y aceptación general de la medida propuesta para reducir el tiempo, sugiriendo que podría ser una solución eficiente y bien recibida por la comunidad académica.

Interpretación:

- La alta proporción de encuestados que esperan una reducción moderada en el tiempo necesario para completar tareas académicas con una aplicación web sugiere un nivel de optimismo y confianza en los beneficios potenciales de esta tecnología.
- Esta percepción positiva puede ser un indicador alentador para la implementación de una aplicación web para la gestión académica, ya que sugiere que los usuarios están receptivos a la adopción de nuevas herramientas que puedan mejorar la eficiencia y la productividad en el ámbito académico.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de resultados

En esta sección se presenta los diagramas de actividades que puede realizar dentro del sistema como un usuario admin así, como un usuario secretaria, por otro lado, se presenta también el diseño de la base de datos. Además, se presenta un análisis y selección de las herramientas de desarrollo que se utilizaron para la creación de este proyecto.

3.1.1. Diagrama de actividades del sistema como usuario Admin

- 1)El usuario ingresa sus credenciales y se verifica su rol
- 2)Si el usuario existe accede al sistema caso contrario regresa al login
- 3)Dentro del sistema tendrá acceso al siguiente menú (Usuarios, Estudiantes, Plantillas)
- 4)En el menú estudiantes podrá visualizar todos usuario que existen en el sistema, así mismo podrá crear nuevos usuarios, eliminar y editar a los mismos.
- 5)En el menú Estudiantes tendrá una vista de todas las carreras asignadas (dentro de cada carrera podrá visualizar todos los estudiantes existentes)
 - 5.1) Por cada carrera podrá crear un estudiante dependiendo del tipo de estudiante (Estudiantes nuevos, Estudiantes Antiguos); editar y eliminar un estudiante.
 - 5.2) De ser el caso podrá importar datos(estudiantes) desde un archivo Excel.
 - 5.2) Por cada estudiante este podrá generar directorios y subir archivos.
 - 5.3) En caso de un directorio no contenga un archivo, se enviará notificaciones para qué es el usuario pueda subir el archivo necesario.

6)En el menú Plantilla el usuario podrá visualizar las plantillas existentes (Estudiantes nuevos, Estudiantes Antiguos), así mismo podrá generar nuevas plantillas, editar o eliminar las mismas.

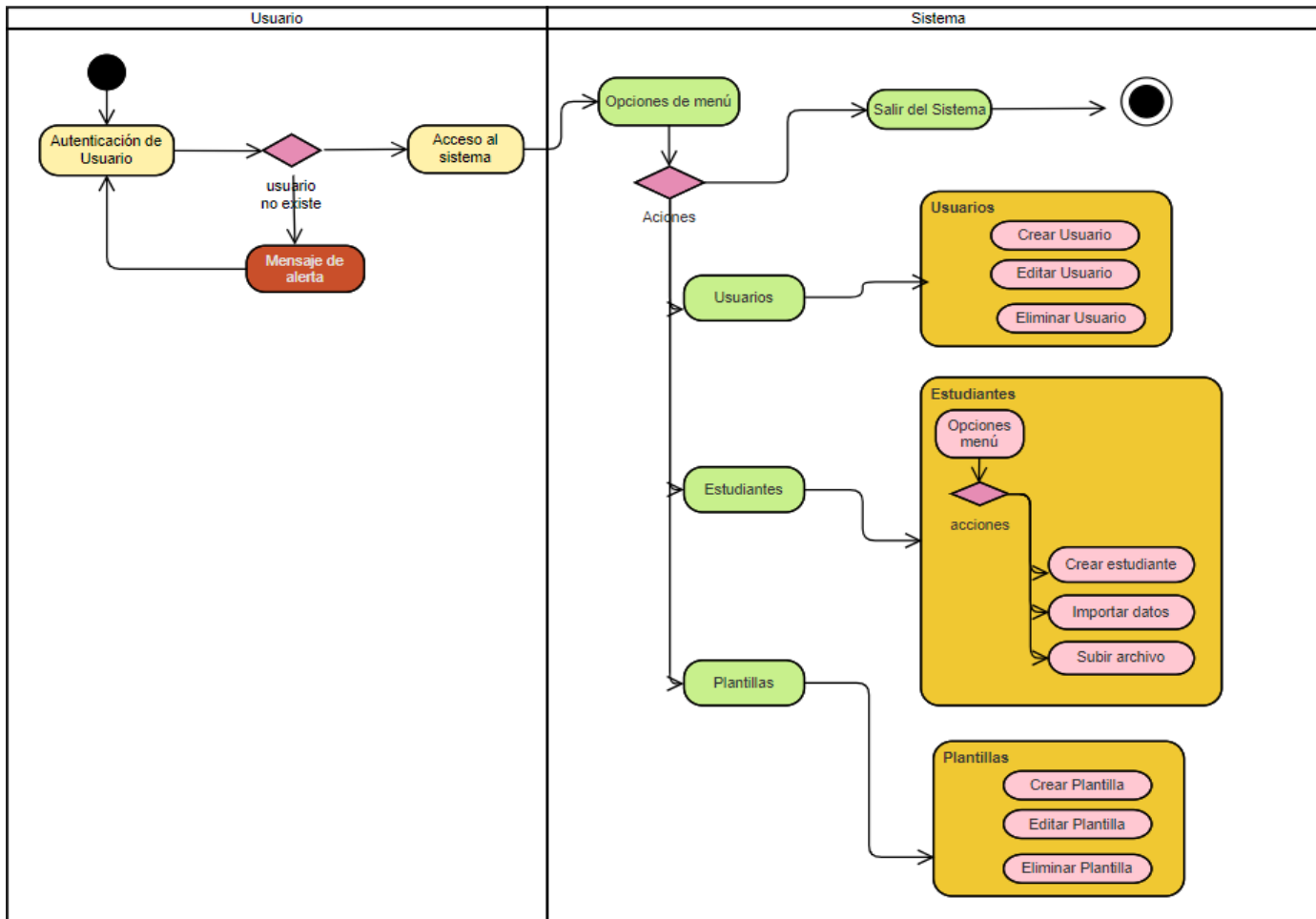


Figura 13: Diagrama general de actividades como usuario admin

3.1.2. Diagrama de actividades del sistema como usuario secretario

- 1) El usuario ingresa sus credenciales y se verifica su rol
- 2) Si el usuario existe accede al sistema caso contrario regresa al login
- 3) Dentro del sistema tendrá acceso al siguiente menú (Estudiantes, Plantillas)
- 4) En el menú Estudiantes tendrá una vista de todas las carreras asignadas (dentro de cada carrera podrá visualizar todos los estudiantes existentes)
 - 4.1) Por cada carrera podrá crear un estudiante dependiendo del tipo de estudiante (Estudiantes nuevos, Estudiantes Antiguos); editar y eliminar un estudiante.
 - 4.2) De ser el caso podrá importar datos (estudiantes) desde un archivo Excel.
 - 4.3) Por cada estudiante este podrá subir archivos.
 - 4.4) En caso de un directorio no contenga un archivo, se enviará notificaciones para que es el usuario pueda subir el archivo necesario.

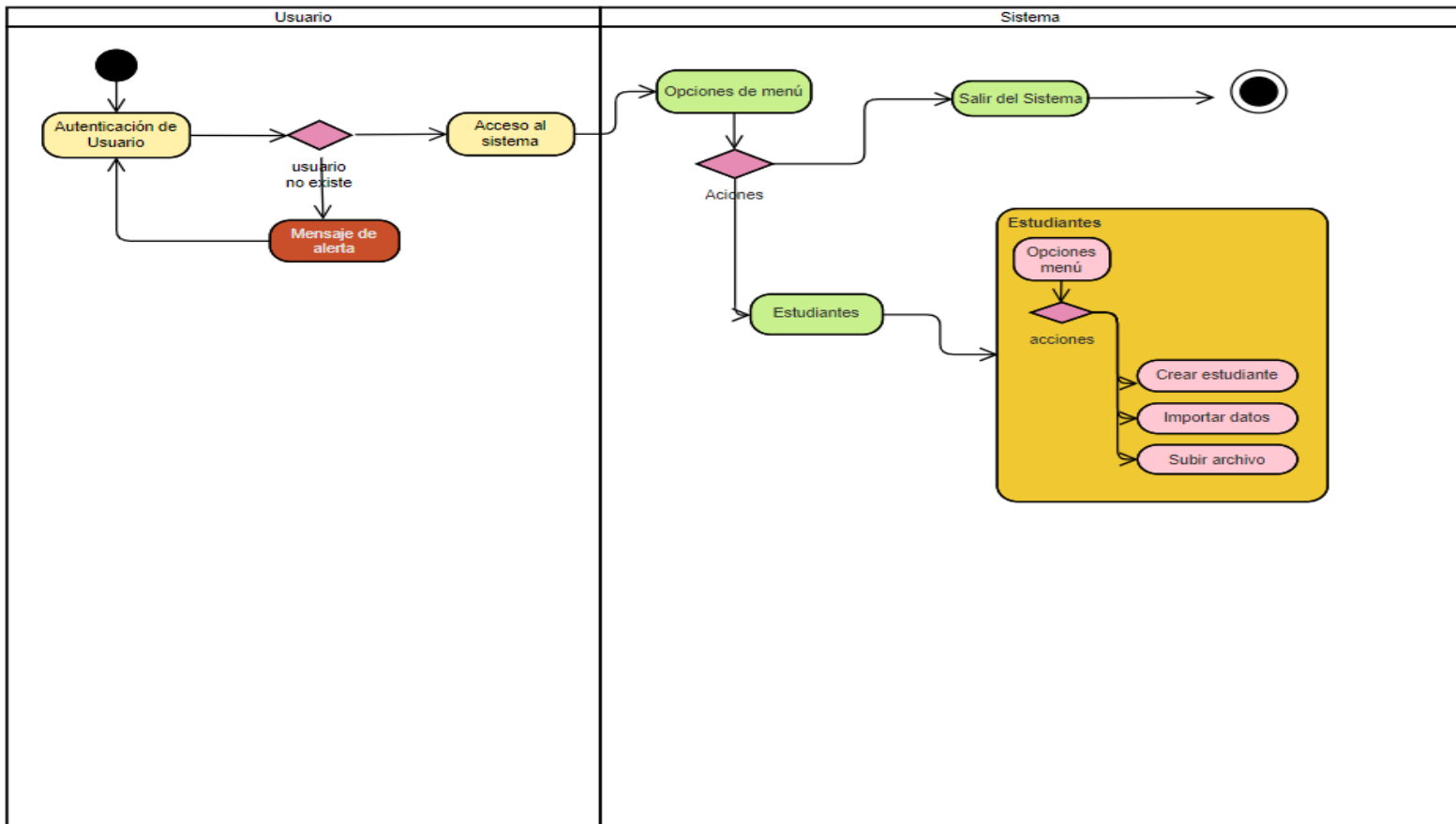


Figura 14: Diagrama general de actividades como usuario secretaria

3.1.3. Análisis de Metodología de Desarrollo

Dentro del análisis de las metodologías de desarrollo, se realiza un cuadro comparativo, como se muestra en la Tabla 13:

Tabla 13: Cuadro comparativo de las metodologías de desarrollo XP, RAD, Scrum

Criterio	XP	RAD	SCRUM
Enfoque Principal	Planificación continua [36]	Prototipado rápido y entrega rápida de software final. [37].	Gestión de Proyectos centrada en la iteración y la colaboración. [39]
Ciclo de Vida	Iterativo e Incremental	Iterativo e Incremental	Iterativo e Incremental
Entrega de software	Frecuente y regular	Entrega rápida	Entrega en ciclos cortos y regulares llamados sprints
Planificación	Flexible y adaptativa. [36]	Desarrollo rápido, orientada a prototipos [37]	Planificación por iteraciones. [39]
Clientes Involucrados	Implicación activa y colaboración cercana [36]	Retroalimentación constante a lo largo del mismo por parte del cliente. [37]	Retroalimentación activa por parte del cliente en cada iteración entregada. [39]
Tamaño de equipo	3-5 personas	2-6 personas	6 -10 personas
Tamaño de proyecto	Se puede aplicar a proyectos de diversos tamaños.[36]	Pequeños y medianos con requisitos cambiantes	Medianos y grandes [40]

Después de analizar las características de los métodos enumerados se puede determinar que RAD nos permite trabajar con un número menor de integrantes en equipo además la entrega de prototipos en constante interacción con el cliente nos permite un desarrollo más rápido. Además, dado a las restricciones que presenta la universidad con utilizar metodologías ya conocidas por el estudiante se excluye la metodología XP y SCRUM, en

conclusión, se ha optado por utilizar la metodología de Desarrollo Rápido de Aplicaciones (RAD). Esta elección se basa en su capacidad para proporcionar soluciones rápidas y prototipos funcionales a nuestros clientes, lo cual tendrá un impacto significativo en el desarrollo de este proyecto debido a las restricciones de tiempo para su entrega. Además, el enfoque RAD, con su capacidad de iteración y escalabilidad, nos permite obtener resultados tangibles en un plazo de tiempo relativamente breve.

3.1.4. Análisis de los Frameworks de Desarrollo

Para el desarrollo del sistema se opta por tres tecnologías de desarrollo, de las cuales, vue.js y react.js están basadas en JavaScript lo que facilita el uso de estas tecnologías para aquellos que tienen conocimiento en JavaScript. Por último, esta Flutter, es un Frameworks utilizado para desarrollar sistemas multiplataforma con un mismo código base; para poder seleccionar una de estas tecnologías se realiza una tabla comparativa como se muestra en la Tabla 14:

Tabla 14: Cuadro comparativo entre los Frameworks de desarrollo, Flutter, Vue.js, React

Criterio	Flutter	Vue.js	React
Lenguaje de Programación	Dart	JavaScript	JavaScript
Tipo de Tecnología	Framework de desarrollo basado en el lenguaje dart.[18]	Framework de JavaScript. [20]	Librería de JavaScript. [21]
Tipo de Aplicaciones	Aplicaciones móviles y web	Aplicaciones web y móviles	Aplicaciones web y móviles
Arquitectura	Basado en el diseño de widgets personalizados	Basado en componentes y el sistema de MVVM	Basado en componentes y el sistema de Component-Based Architecture

Criterio	Flutter	Vue.js	React
Componentización	Componentes reutilizables	Componentes reutilizables	Componentes reutilizables
Comunidad y Popularidad	Crece en popularidad rápidamente, comunidad activa.[17]	Crece en popularidad rápidamente, comunidad activa. [20]	Ampliamente utilizado y establecido, gran comunidad. [22]
Flexibilidad	Mayormente adecuado para aplicaciones móviles y de una sola página. [18]	Altamente adaptable y adecuado para aplicaciones web y de una sola página. [20]	Altamente adaptable y adecuado para aplicaciones web y de una sola página. [23]
Aprendizaje	Requiere aprender Dart (lenguaje de Dart).[18]	Requiere aprender JavaScript y Vue.js. [19]	Requiere aprender JavaScript y React. [22]
Rendimiento	Buen rendimiento y velocidad en aplicaciones móviles	Buen rendimiento y velocidad en aplicaciones web	Buen rendimiento y velocidad en aplicaciones web y móviles
Comunidad de desarrollo	Creciendo rápidamente. [18]	Activa y en crecimiento. [20]	Amplia y establecida. [22]
Nivel de experiencia con la tecnología de desarrollo por el autor	Bajo	Medio	Bajo

Habiendo evaluado todos los atributos de los marcos enumerados en la tabla, se concluye que vue.js es el más apropiado para este proyecto, dado que este proyecto está centrado en el desarrollo web, Flutter no es una opción, así mismo la activa comunidad de vue.js hace que sea más fácil el desarrollo y la solución errores. Además, uno de los factores clave es el conocimiento del autor con el Framework vue.js y JavaScript, lo que simplificará el tiempo de implementación de desarrollo ya no requerirá aprender una nueva tecnología de desarrollo. De igual manera, la arquitectura basada en componentes y el sistema de arquitectura basada en componentes de Vue.js proporciona un enfoque de desarrollo modular y bien estructurado.

La capacidad de Vue.js para adaptarse tanto a aplicaciones web como a aplicaciones de una sola página es otro factor que vale la pena considerar.

3.1.5. Análisis de los Frameworks de CSS

Para el desarrollo de interfaces del sistema se opta por 3 Frameworks de diseño basados en CSS, se realiza una tabla comparativa como se muestra en la Tabla 15:

Tabla 15: Cuadro comparativo entre los Frameworks de diseño Vuetify, Bootstrap, Tailwind

Criterio	Vuetify	Bootstrap	Tailwind
Interfaz de usuario	Componentes de Material Design Framework con componentes de usabilidad listos para usar.	Elementos de diseño adaptativos optimizados para sitios web y aplicaciones.	Utility Classes para personalización completa de estilos y diseños.
Sistema de cuadrícula	Sí (para organizar diseños de página adaptable a diferentes resoluciones de pantalla).	Sí (con enfoque en diseño adaptable a pantallas pequeñas y dispositivos táctiles).	Sí (para organizar diseños de página adaptable a diferentes resoluciones de pantalla).
Biblioteca de elementos	Amplia biblioteca de elementos desde botones, sliders, entradas hasta elementos avanzados como mapas.	Amplia gama de elementos de diseño, como formularios, botones, menús, adaptados a diferentes formatos de navegación.	Personalización completa de estilos y diseños mediante Utility Classes.
Personalización	Ofrece cierta personalización dentro de su marco de trabajo.	Personalización posible, pero en menor medida que Tailwind.	Altamente personalizable mediante Utility Classes para cada aspecto.

Criterio	Vuetify	Bootstrap	Tailwind
Enfoque	Enfocado en proporcionar componentes listos para usar.	Enfocado en diseño adaptable y funcionalidad interactiva.	Enfocado en personalización completa de componentes y flexibilidad

Después de analizar cuidadosamente las características de los Frameworks presentados en la tabla, he decidido optar por Vuetify. Esta elección se basa principalmente en la sólida compatibilidad que este Framework tiene con Vue.js, ya que está diseñado específicamente para integrarse de manera óptima con este entorno. Ofrece una experiencia de desarrollo ágil gracias a sus componentes predefinidos y cuenta con una amplia biblioteca de elementos, como barras deslizantes, diseños de navegación, botones, entre otros.

Si bien las otras dos opciones son posibles, en esta situación en particular he descartado Bootstrap porque no es compatible con Vue3, que es la versión que tenemos previsto utilizar para este proyecto. En cuanto a Tailwind, aunque comparte algunas características con Bootstrap al ofrecer elementos de interfaz predefinidos, el hecho de tener que pagar por ciertos elementos hace que no sea la opción más beneficiosa para este proyecto.

En resumen, la elección de Vuetify se basa en su estrecha integración con Vue.js, su enfoque en el desarrollo rápido y su extensa colección de componentes, lo que lo posiciona como la solución más adecuada para nuestros objetivos y requisitos específicos.

3.2. Desarrollo de la propuesta

En sección del documento se presentan cada una de las fases de la metodología RAD, a continuación, se detalla cada una de las fases de desarrollo.

Fase I: En esta fase se presenta los requerimientos funcionales que obtuvieron a través de las entrevistas y encuesta realizadas a las secretarías de la FISEI. Además, se presenta un plan de entrega donde se observa las funciones que se implementaron en cada iteración. Como último se puede visualizar la arquitectura con que se está construido el sistema y diseño de la base de datos.

Fase II: En esta fase se presenta un prototipo del sistema con las diferentes pantallas implementadas y la descripción de cada una.

Fase III: En esta están presente las tecnologías de desarrollo que se utilizó para la implementación del sistema, así como el código fuente de ciertas partes del desarrollo del sistema permitiendo conocer un poco de su implantación con el Framework vue.js.

Fase IV: En esta fase están presentes las pruebas realizadas al sistema tales como: pruebas de aplicación, pruebas de interfaz de usuario, Pruebas de aceptación.

3.2.1. Fase I: Recopilación de requisitos

a. Definición de roles

Tabla 16: Requerimientos del sistema

Responsable	Rol	Función
Paulo Martínez	Desarrollador	Desarrollo del sistema acorde a los requisitos.
Ing. Fernando Ibarra	Verificador	Monitoreo y seguimiento del desarrollo del sistema.
Secretaria General	Cliente	Definición de requisitos del sistema y retroalimentación de las entregas del producto.

b. Listado de requerimientos Funcionales

Tabla 17: Tabla de requerimientos funcionales

ID	Requerimiento	Descripción	Prioridad	Riesgo
R001	Autenticación de usuario	Control de acceso al sistema	Alto	Alto
R002	Gestión de usuarios	Creación admin debo poder crear, editar, visualizar y eliminar usuarios.	Alto	Alto
R003	Manejo de roles	Deberán manejar roles para las secretarias y un usuario admin	Alto	Alto
R004	Control de permisos a para usuarios	Como admin debo controlar las acciones que cada usuario podrá hacer en los archivos que están en el sistema ya sea eliminar, ver o editar	Alto	Alto
R005	Gestión de directorios	Como Usuario debo poder crear, editar, visualizar y eliminar un directorio para cada estudiante que sea registrado en el sistema	Alto	Alto
R006	Gestión de estudiantes	Como Usuario debo poder crear, editar, visualizar y eliminar un estudiante que sea registrado en el sistema	Alto	Alto
R007	Importación de archivos Excel	Como usuario debo poder importar grandes cantidades de datos al sistema y estos se creen como directorios	Alto	Alto
R008	Gestión de plantillas	Como usuario quiero poder crear una plantilla para la creación de modelos para estudiantes nuevos y viejos para poder reutilizar en otros estudiantes que registre en el sistema	Alto	Alto

ID	Requerimiento	Descripción	Prioridad	Riesgo
R009	Crear una observación por cada archivo creado	Como usuario quiero poder registrar una observación cada vez que suba un archivo al sistema	Alto	Alto
R010	Gestión de archivos	Como usuario quiero crear, borrar, editar y visualizar los archivos subidos en el sistema	Alto	Alto
R011	Envío de notificaciones	Como usuario quiero poder recibir notificaciones en caso de un directorio de un estudiante no cuente con la documentación necesaria	Bajo	Bajo

c. Requerimientos no Funcionales

Tabla 18: Tabla de requerimientos no funcionales

ID	Requerimiento	Descripción	Prioridad	Riesgo
R001	Rendimiento	El sistema deberá proporcionar tiempos de respuesta rápidos para garantizar una experiencia de usuario eficiente, especialmente en las operaciones de CRUD y en la autenticación.	Alto	Alto
R002	Usabilidad	La interfaz de usuario debe ser intuitiva y fácil de usar, especialmente para las funciones de creación, edición y eliminación de usuarios, directorios y archivos.	Alto	Bajo
R003	Escalabilidad	El sistema debe ser capaz de manejar un aumento significativo en el número de usuarios, archivos y directorios sin degradar el rendimiento.	Alto	Alto
R004	Disponibilidad	Se debe garantizar la disponibilidad del sistema durante el horario de trabajo.	Alto	Bajo
R005	Seguridad	Todas las transmisiones de datos, especialmente durante la autenticación, deben estar cifradas para garantizar la confidencialidad de la información.	Alto	Alto

ID	Requerimiento	Descripción	Prioridad	Riesgo
R006	Compatibilidad	El sistema deberá ser compatible con los navegadores web más comunes.	Alto	Bajo
R007	Mantenimiento	El sistema debe ser diseñado de manera que las actualizaciones y parches puedan ser implementados sin Interrumpir el funcionamiento normal del sistema.	Alto	Bajo

d. Plan de entrega

Con los requerimientos ya recopilados y definidos, se elaboró el siguiente plan de entrega para controlar y organizar el desarrollo del proyecto.

Iteración	Código	Requerimientos	Tiempo Estimado	
			Horas	Días
Iteración 1	R001	Autenticación de usuario	5	1
	R002	Gestión de usuarios	12	2
	R003	Manejo de roles	5	1
Iteración 2	R004	Control de permisos a para usuarios	6	1
	R005	Gestión de directorios	12	2
	R006	Gestión de estudiantes	12	3
Iteración 3	R007	Importación de archivos Excel	8	2
	R008	Gestión de plantillas para los estudiantes	5	1
	R009	Subida de archivos con fecha y observaciones	6	1
Iteración 4	R010	Gestión de archivos	12	2
	R011	Envío de notificaciones	6	1

e. Arquitectura del sistema

Es desarrollo del sistema adoptara una arquitectura cliente-servidor, la cual se describe detalladamente a continuación:

Cliente: el cliente se compone del sitio web, donde se gestiona de manera completa la funcionalidad del sistema. En esta situación, se envían peticiones al servidor y este responde con las respuestas correspondientes.

Servidor: es la interacción entre la base de datos donde se almacena toda información del sistema el cual recibirá las peticiones del cliente y enviará respuestas al este.

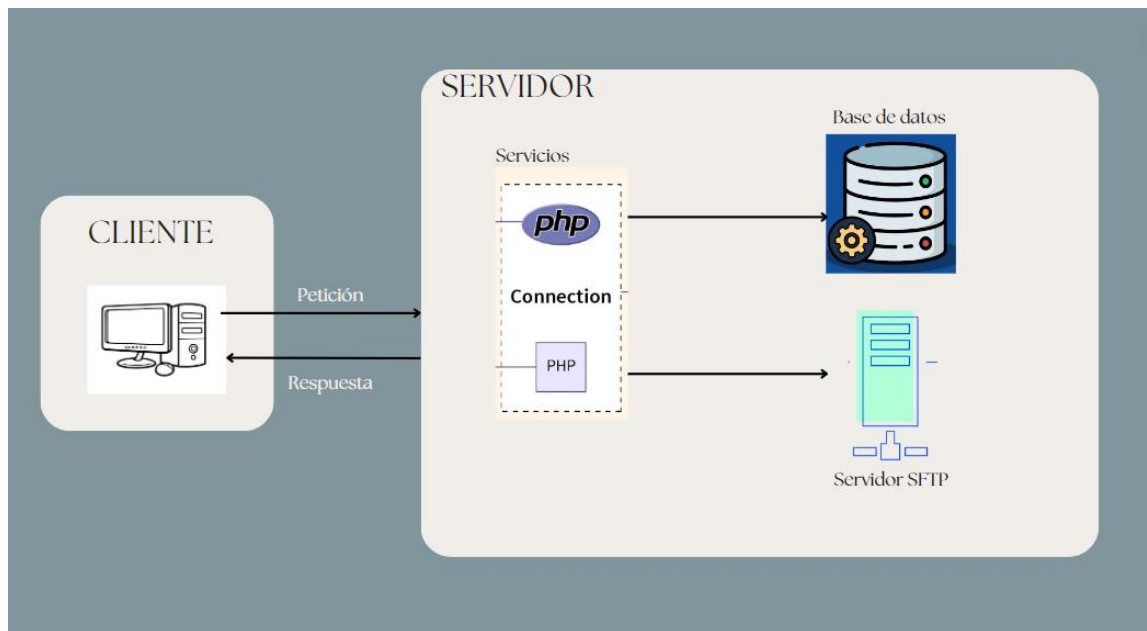


Figura 15: Arquitectura del sistema

f. Modelado de la Base de Datos

Para este proyecto se trabajó con MySQL dado a la solicitud de la universidad, ya que es el gestor de base de datos que ellos utilizan.

Con el objetivo de facilitar un acceso más eficiente a la información, se ha desarrollado un modelo relacional para la base de datos, con el fin de mantener una estructura

organizada y garantizar la integridad de los datos. La implementación de este modelo se realizó en el sistema MySQL para la gestión de bases de datos. Durante este proceso, se crea un modelo entidad-relación, como se muestra en la Figura 15, para mejorar la comprensión del mencionado modelo.

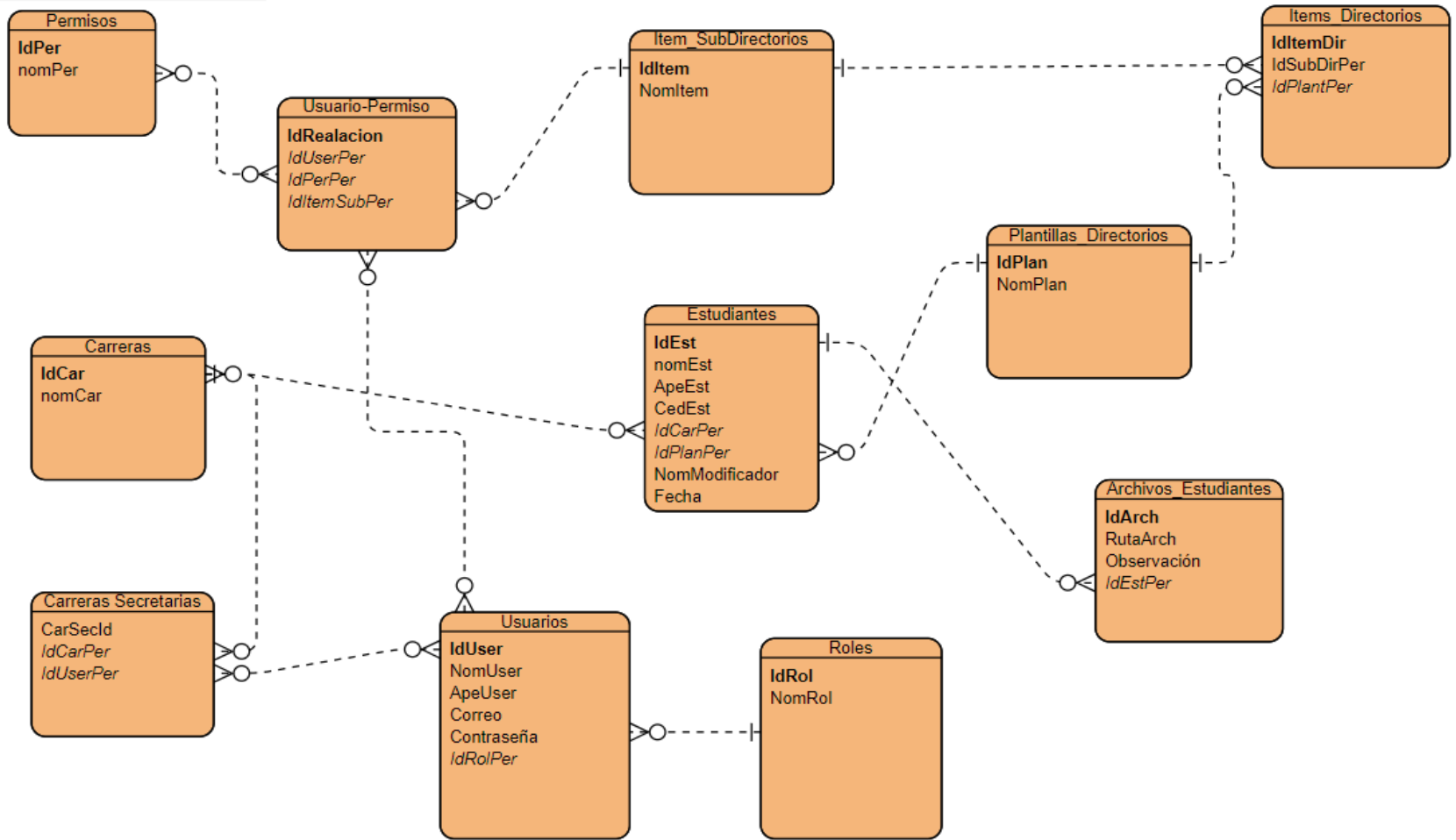


Figura 16: Diagrama entidad-relación de la Base de datos

3.2.2. Fase II: Prototipo y retroalimentación por parte del cliente

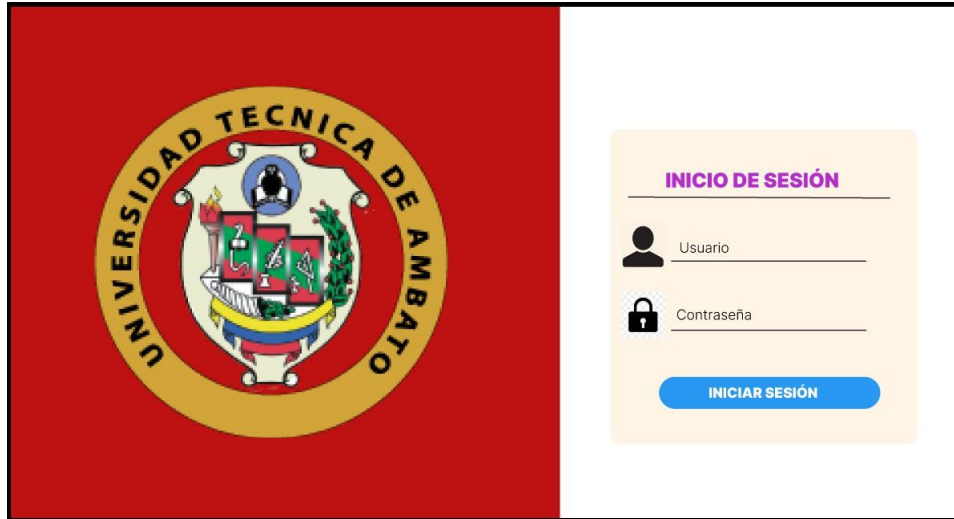


Figura 17: Prototipo de la interfaz del ingreso a la aplicación

Autenticación del usuario

En la Figura 16 se muestra la forma en la que usuario deberá ingresar sus credenciales para poder acceder al sistema, donde se deberá ingresa un correo y su contraseña.

Prototipo para la interfaz Dashboard



Figura 18: Prototipo de interfaz principal del sistema

En la Figura 17 se muestra la interfaz principal del sistema desde el usuario podrá navegar hacia las demás interfaces como son usuarios, estudiantes, plantillas; además desde la misma interfaz el usuario podrá cerrar sesión.

Prototipo para la interfaz Usuarios



Figura 19: Prototipo de interfaz Usuarios

En la Figura 18 se muestra la interfaz usuarios a la solo tendrá acceso un usuario administrador, desde esta interfaz el usuario podrá ver los usuarios existentes, así como crear, editar, eliminar y actualizar usuarios.

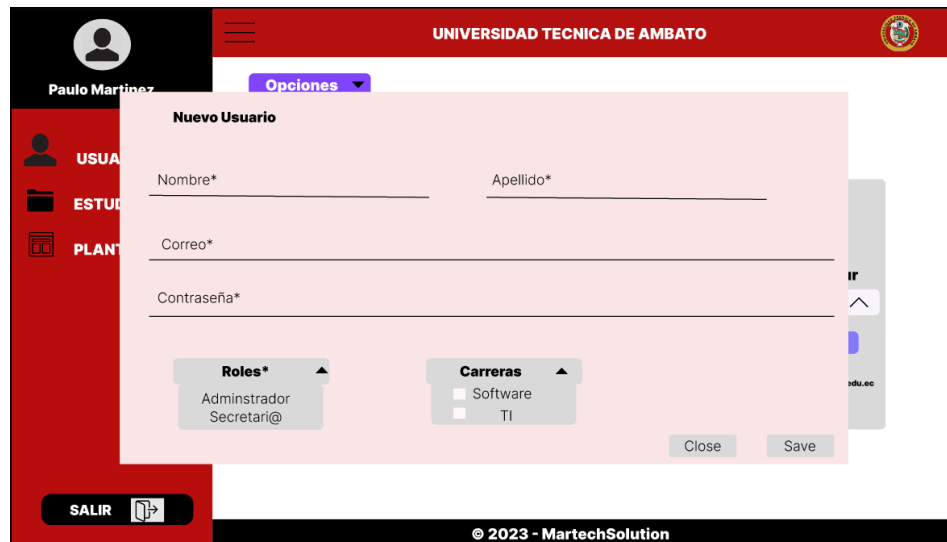


Figura 20: Prototipo de interfaz para crear y editar un usuario

En la Figura 19 se muestra la interfaz que permitirá al usuario tanto crear como editar un usuario del sistema al presionar en el botón “Nuevo Usuario” de las “Opciones”. En la Figura 20 se muestra la interfaz desde el usuario podrá visualizar los permisos asignados a un determinado usuario, así mismo podrá agregar o eliminar permisos al mismo.

Prototipo para la interfaz Estudiantes



Figura 21: Prototipo interfaz carreras de cada usuario

La Figura 21 se puede visualizar la manera en la que el usuario podrá observar los datos de las carreras que tenga a su cargo, cada una de estas filas será un directorio.



Figura 22: Prototipo de interfaz Estudiantes

En la Figura 22 se observa la manera en la que el usuario visualizará los datos de los estudiantes que tenga a su cargo, cada una de estas filas será un directorio.



Figura 23: Prototipo de interfaz para crear y editar un estudiante

En la Figura 23 se muestra la interfaz que permitirá al usuario tanto crear como editar un estudiante del sistema al presionar en el botón “Nuevo Carpeta” de las “Opciones”.

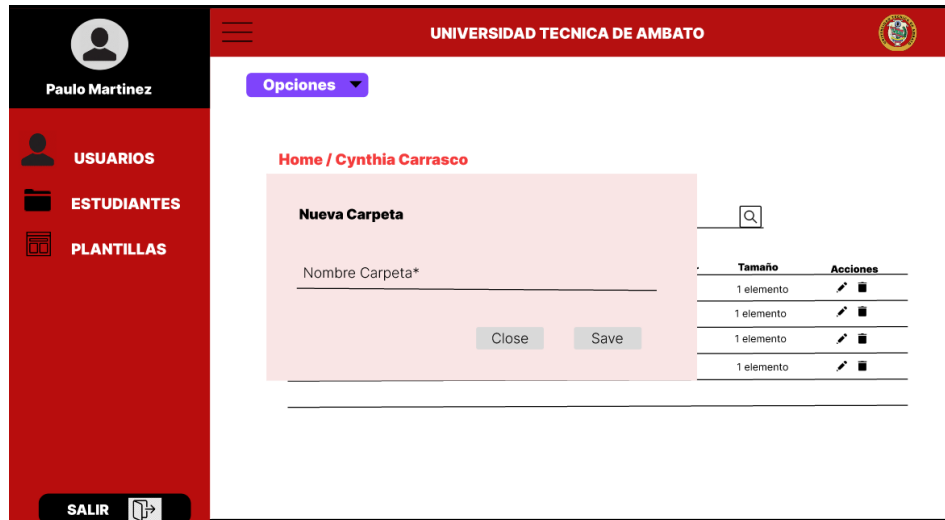


Figura 24: Prototipo de interfaz para crear carpetas

Prototipo de interfaz para cada ítem de la tabla Estudiantes

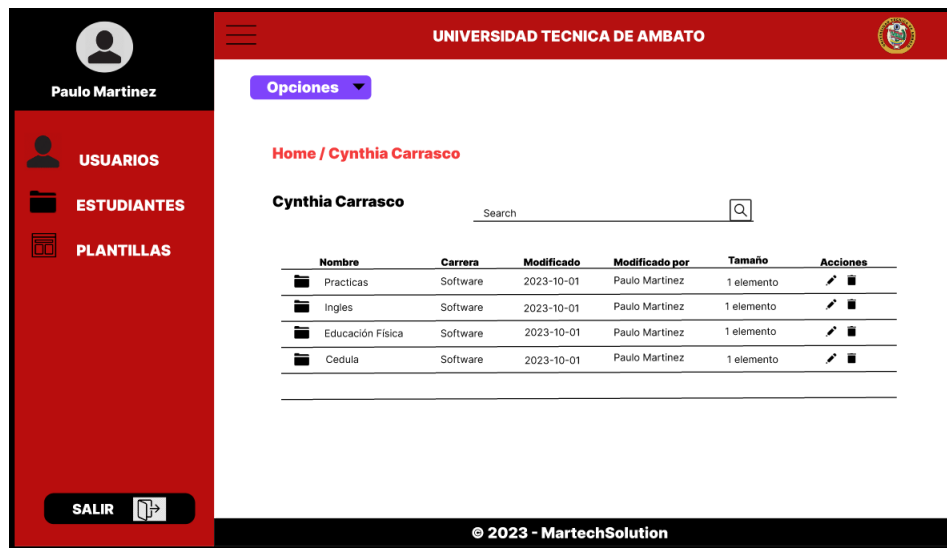


Figura 25: Prototipo de interfaz para cada ítem de la tabla Estudiantes

Al seleccionar una fila de la Figura 22 esta mostrara una nueva tabla con todos los directorios que ese estudiante tiene como se muestra en la Figura 24.

Al estar en la sección de subcarpetas de cada usuario y el este desee crear una nueva carpeta se mostrará un nuevo modal diferente, como se muestra en la Figura 25.

Prototipo de interfaz para cada ítem de un directorio



Figura 26: Prototipo de interfaz para ítem de un directorio

Al seleccionar un ítem de la Figura 24 este mostrara una nueva tabla con los archivos que existan en ese directorio, como se muestra en la Figura 26.

Prototipo de interfaz Plantilla



Figura 27: Prototipo de interfaz Plantilla

En la Figura 27 se puede observar las diferentes plantillas que el usuario habrá creado con anterioridad, así tendrá las opciones de eliminar, actualizar y crear nuevas plantillas.



Figura 28: Prototipo de interfaz para crear Plantillas

3.2.3. Fase III: Construcción de la aplicación

a. Herramientas de desarrollo

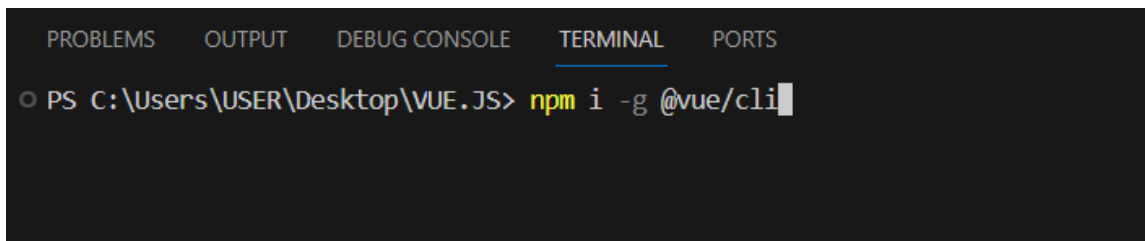
Para la implementación de este proyecto se usaron las siguientes herramientas:

- Visual Studio Code: Para la creación y edición de código destinado para Banck-End como Front-End.
- GitHub: Para almacenar el proyecto.
- Vue.js: Para creación del en Front-End.
- Vuetify: Para la creación de los componentes.
- Figma: Para la creación de los Muckups.
- PHP: Para la creación de las APIS que conectaran con la Base de Datos y el servidor.

- Máquina Virtual: Utilizada para montar un servidor en Linux en centOS7 perteneciente a la distribución de Red Hat.

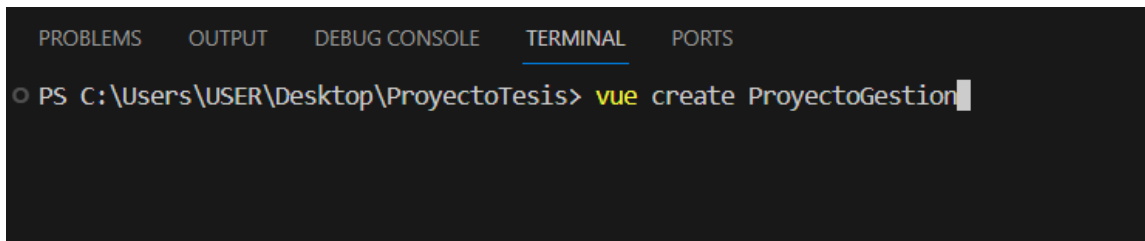
b. Creación del Proyecto

Para la creación del proyecto ejecutamos en Visual Studio Code el comando mostrado en la Figura 30, para crear un proyecto en vue.js una vez que ya hayamos instalado vue.js



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\USER\Desktop\VUE.JS> npm i -g @vue/cli
```

Figura 29: Comando para instalar vue.js



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\USER\Desktop\ProyectoTesis> vue create ProyectoGestion
```

Figura 30: Comando para crear un proyecto en vue.js

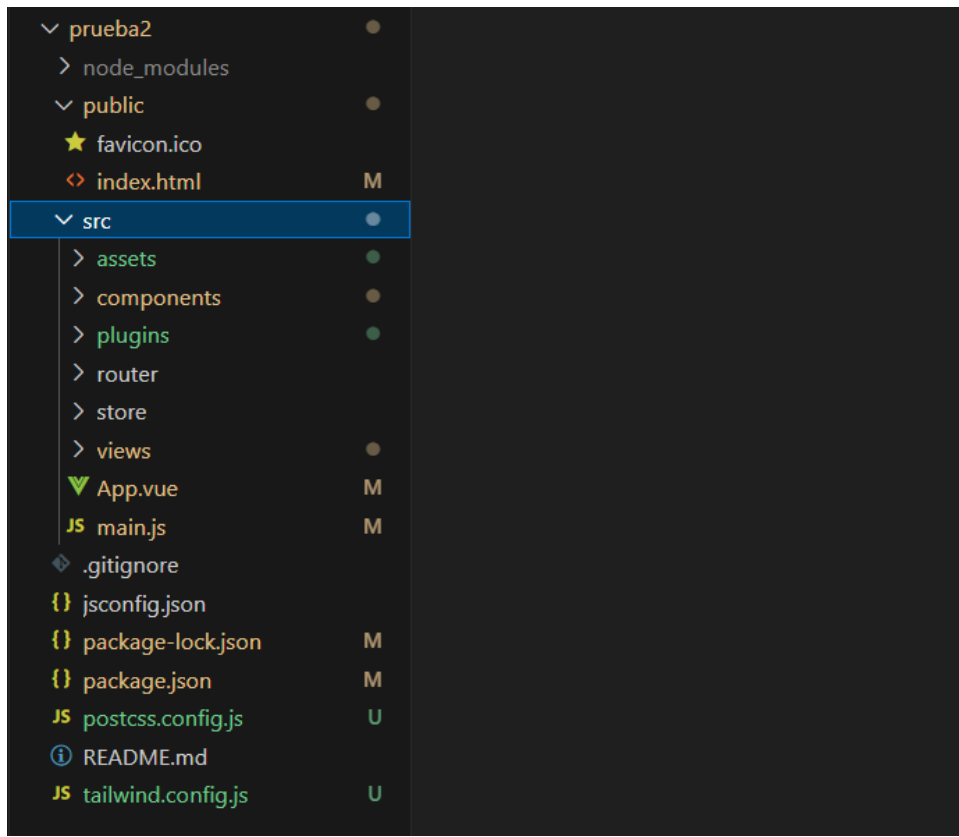


Figura 31: Archivos de la aplicación luego de crear la misma

c. Librerías utilizadas

- **Vuetify:** Es una biblioteca de interfaz de usuario de Vue que ofrece componentes de Material Design Framework destinada a la creación de interfaces de usuario que implementa componentes de usabilidad listos para usar.[17] Para agregar Vuetify a nuestro de vue.js ejecutamos el comando que se muestra en la Figura 32.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\USER\Desktop\VUE.JS> cd .\prueba2\
○ PS C:\Users\USER\Desktop\VUE.JS\prueba2> vue add vuetify
```

Figura 32: Comando para añadir Vuetify

- **Phpseclib:** Es una biblioteca de PHP que provee implementaciones puras de SSH2, SFTP, RSA, DSA, esta librería se ha usado para realizar la conexión con el servidor SFTP. Para poder instalar esta librería lo haremos a través del gestor de dependencias de PHP composer.

```
Microsoft Windows [Version 10.0.18363.1316]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\ForVideos\php\RebexTinySftpServerBin-1.0.8>composer require phpseclib/phpseclib
```

Figura 35: Comando para instalar phpseclib

```
<?php
//Datos para el servidor
$servidor='192.168.100.65';
$user='admin';
$pass='admin243';
$puerto='22';

?>
```

Figura 33: Datos para la conexión con el servidor

```
require_once('Servidor.php');
require_once('vendor/autoload.php');
use phpseclib3\Net\SFTP;

try {
    $sftp = new SFTP($servidor, $puerto);
    if (!$sftp->login($user, $pass)) {
        throw new Exception('No se pudo autentificar en el servidor SFTP');
    } else {
```

Figura 34: Código .php para conexión con el servidor

d. Construcción de un Prototipo funcional

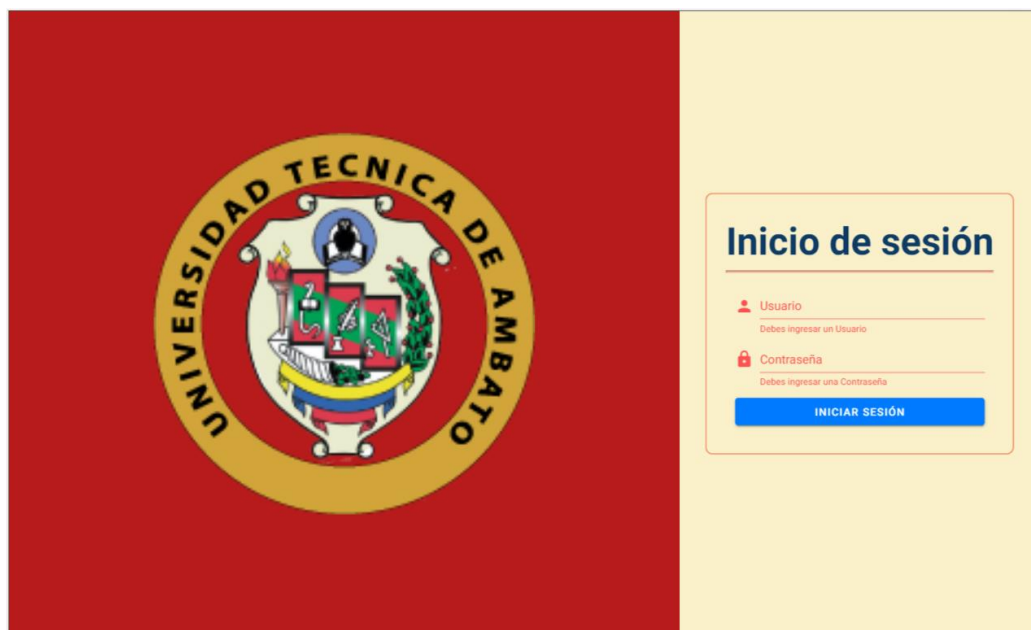


Figura 36: Login

Página principal del sistema, donde se ingresan los datos correspondientes para acceder al sistema.

Tipo	Nombre	Carrera	Modificado	Modificador por	ACCIONES
Folder	Carlos Ramirez	Software	2023-01-12	Paulo Martinez	✎ 🗑️ ⬇️
Folder	Paulo Martinez	Software	2024-01-13	David Vaca	✎ 🗑️ ⬇️
Folder	Cristina Silva	Software	2024-01-13	David Vaca	✎ 🗑️ ⬇️

Figura 37: Interfaz donde se visualiza los estudiantes por carrera

En esta interfaz se cargarán todos los estudiantes que estén en una determinada carrera.

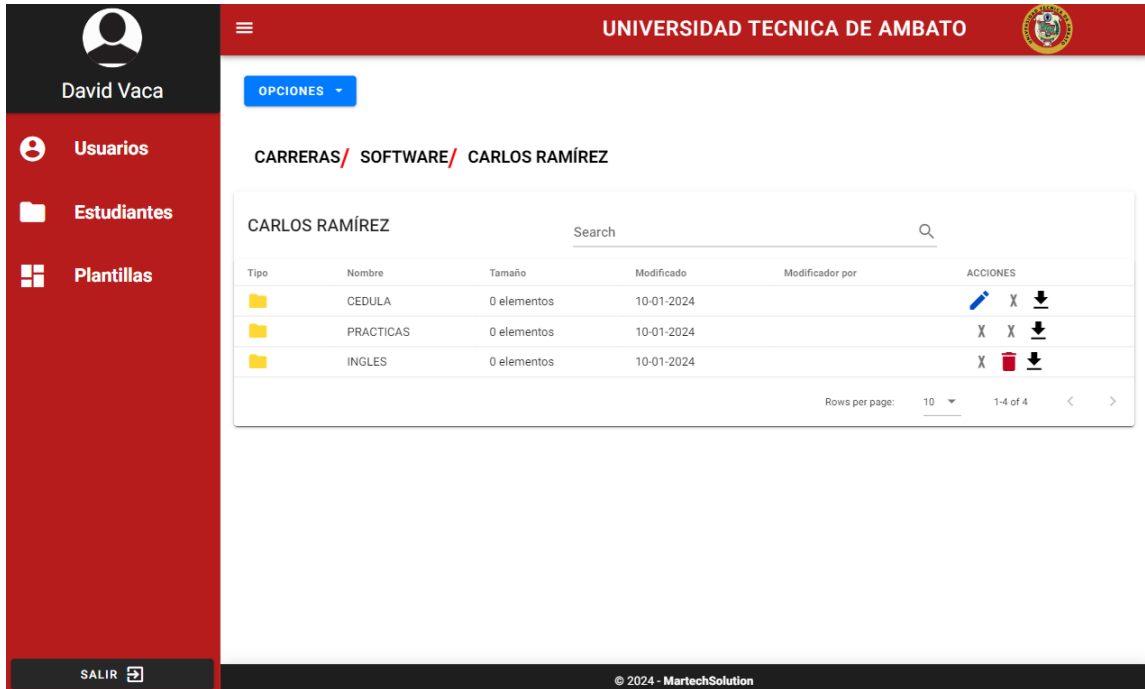


Figura 38: Interfaz donde se crean las subcarpetas de los estudiantes

En esta interfaz estarán presentes las carpetas que son necesarias para la titulación de un estudiante.

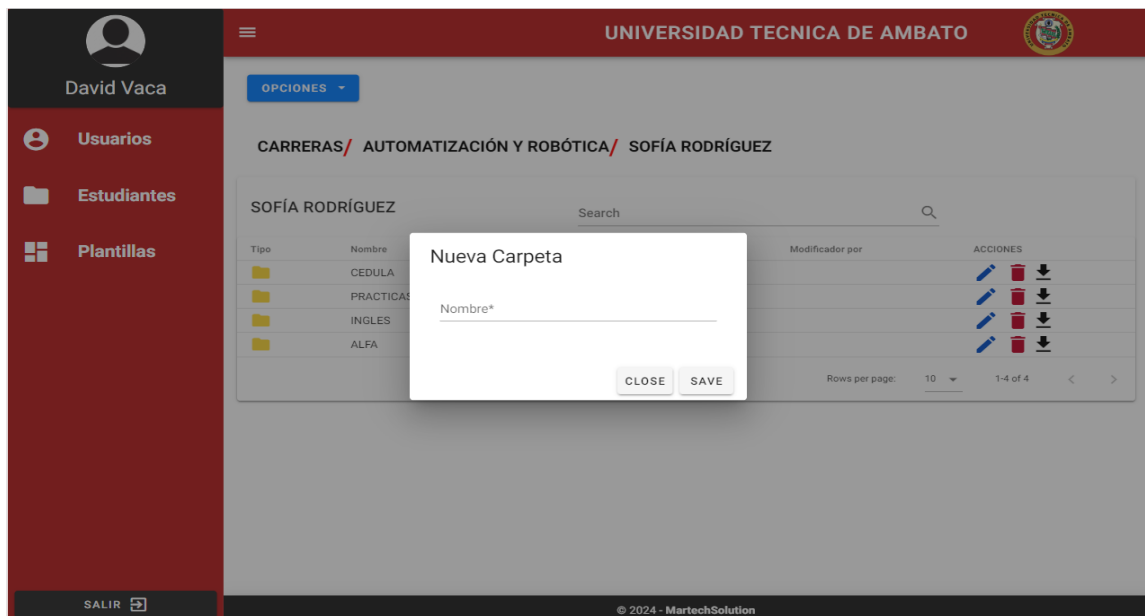


Figura 39: Creación de una nueva subcarpeta

Esta ventana de dialogo de la Figura 39 permite crear una nueva carpeta en caso de ser necesario.



Figura 40: interfaz archivos

Esta interfaz permite alojar los archivos referentes a cada carpeta

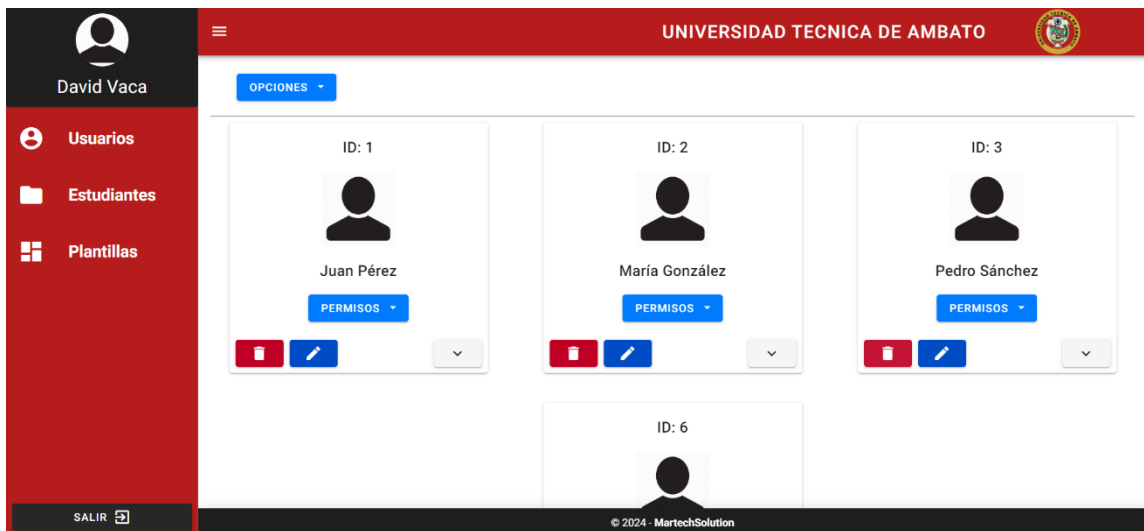


Figura 41: interfaz usuario

Esta interfaz permite visualizar todos los usuarios del sistema, desde esta podremos crear, eliminar o actualizar un usuario.

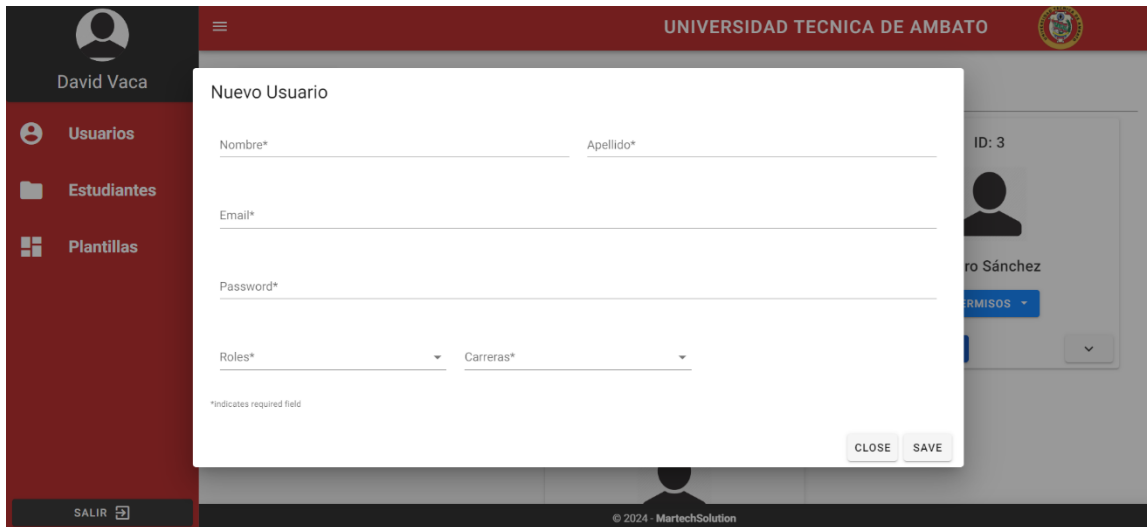


Figura 42: ventana para crear un usuario

Desde esta ventana modal crearemos un nuevo usuario, así como editarlo.

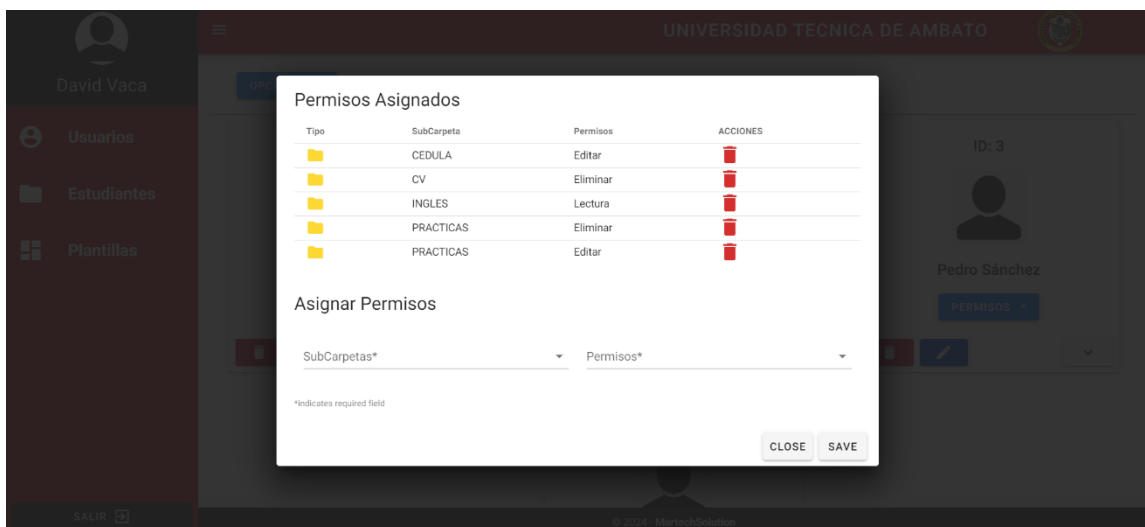


Figura 43: Ventana para visualizar permisos

En esta ventana se podrá visualizar los permisos que el usuario tienen ciertas carpetas del sistema, además, podremos otorgarle nuevos permisos.



Figura 44: interfaz plantillas

Esta interfaz permite visualizar las plantillas que existen en el sistema, además nos permite crear nuevas plantillas.

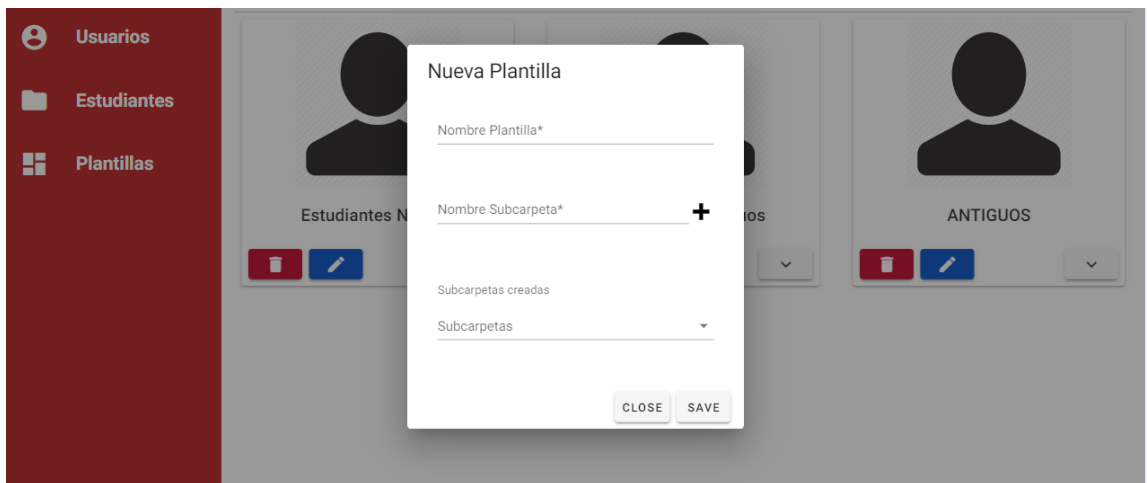


Figura 45: ventana para crear una plantilla

Desde esta ventana modal se podrá crear una nueva plantilla, así mismo se podrá editar la plantilla.

Evaluación Heurística del prototipo funcional

A continuación, se presenta las pruebas de usabilidad definidas para un prototipo funcional.

Tabla 19: Evaluación Heurística

N°	Principio de usabilidad	Evaluación	Nota
1	Visibilidad y estado del sistema	Si	El sistema le permite conocer al usuario las acciones que este realiza a través de alertas
2	Conexión entre el sistema y el mundo real	Si	
3	Control y libertad de usuario	Si	A través de mensajes se le informa al usuario las acciones que este haciendo erróneamente.
4	Consistencia y estándares	Si	
5	Prevención de errores	No	Existe el caso de no validar un ingreso repetido de un estudiante.
6	Reconocer mejor que recordar	Si	
7	Flexibilidad y eficiencia de uso	Si	El usuario tiene la opción de poder importar gran cantidad. Así mismo tiene navegación que le permite navegar entre directorio
8	Diseño estético y minimalista	Si	
9	Ayuda al usuario a reconocer y recuperarse de los errores	Si	Mensajes a través de alertas para que el usuario sepa donde se equivoco
10	Ayuda y documentación	Si	Cuenta con un manual de usuario

Al observar la tabla se puede observar rápidamente que se cumple con casi todos los principios de usabilidad a excepción del principio número cinco, el cual tendremos que resolver en la implementación de la aplicación.

e. Aplicación

- **Página de login**

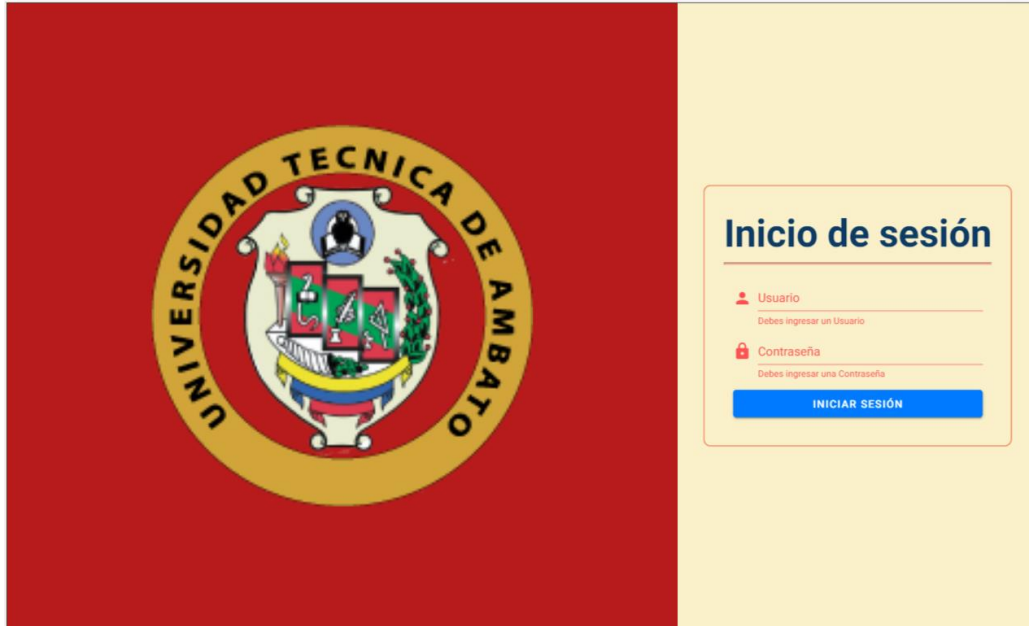


Figura 46: Pagina Login

```
async login({ commit, state }, { user, pass }) {
  try {
    const datosUser = new FormData();
    datosUser.append('user', user);
    datosUser.append('pass', pass);
    const setting = {
      method: 'POST',
      body: datosUser,
    }
    const url = "http://localhost/Api-UTA/Login.php";
    const data = await fetch(url, setting)
    if (data.ok) {
      const json = await data.json();
      const firstObject = json[0];
      console.log(firstObject);
      if (user.toLowerCase() === firstObject.Correo.toLowerCase() && pass === firstObject.Contraseña) {
        commit('setAuthentication', true);
        commit('setUser', firstObject);
        localStorage.setItem('Authentication', true);
        localStorage.setItem('user', JSON.stringify(firstObject));
      } else {
        console.log("No se encontro el usuario...");
      }
    } else {
      console.log(" Not found => Error 400 ");
    }
  } catch (error) {
    console.log("Error de Autenticacion:" + error);
  }
}
```

Figura 47: Código .js para API Login

Como se evidencia en la Figura 46, para ingresar al sistema se una solicitud hacia la base de datos para validar que el usuario exista, posteriormente se guarda los datos en un localStorage para poder mantener iniciada la sesión, así se cierre el navegador.

```
methodLogin: async function () {
  // Llama a la acción 'login' para autenticar al usuario.
  await this.login({ user: this.user, pass: this.pass });
  const autenticacion = localStorage.getItem('Authentication');
  //console.log(this.autenticacion);
  if (autenticacion) {
    this.$router.push("/Estudiantes");
    //this.obtenerPermisosDirectorios();
  }
},
```

Figura 48: Código .js para crear un token de sesión

En la Figura 48 se puede observar el método donde se utiliza al método login de la Figura 47, en este método recurrimos al usuario guardado en localStore.

- **Pagina Estudiantes/ carreras**

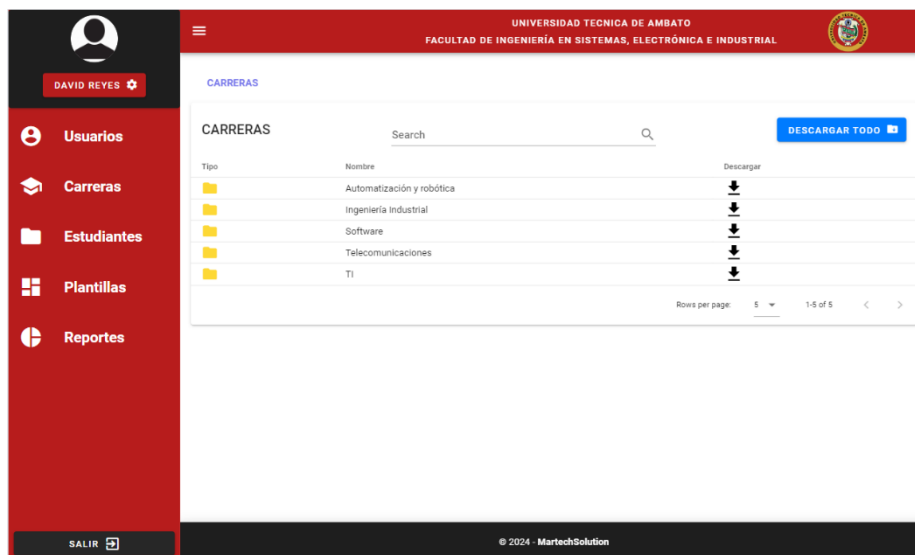


Figura 49: Página Principal, se muestran las carreras que tiene el usuario asignadas

```
cargarCarrerasUser:async function({commit}, idUser){
  try {
    const datosUser = new FormData();
    datosUser.append('idUser', idUser);
    //console.log(idUser);

    const setting ={
      method:'POST',
      body:datosUser,
    }
    const url = 'http://localhost/ApiS-UTA/carrerasUsuarios.php';
    const data = await fetch(url, setting);
    const json = await data.json();
    //console.log(json);
    commit('llenarCarrerasUser', json);
  } catch (error) {
    console.error('Error en la solicitud:', error);
  }
},
},
```

Figura 50: Código .js para solicitar los datos a la API

En la Figura 50 se presenta el método que se utiliza para hacer un llamado a la API que nos permite acceder a todas las carreras que un determinado usuario tiene asignadas.

```
created() {
  const isAuthenticated = localStorage.getItem('Authentication') === 'true';
  if (!isAuthenticated) {
    this.$router.push("/");
  } else {
    const storedUser = JSON.parse(localStorage.getItem('user'));
    this.idUser = storedUser.IdUser;
    console.log(this.carreraSeleccionada);
    this.cargarCarrerasUser(this.idUser);
    //this.cargarSubCarpetas();
    this.obtenerPermisosDirectorios();
    this.acciones(this.itemsBread[0]);
  }
},
```

Figura 51: Código vue.js para verificar la autenticación y cargar carreras

En la Figura 51 se puede observar la llamada al método cargar carreras, donde recuperamos el “idUser” logeado para enviarlo en el método.

- **Pagina Estudiantes**

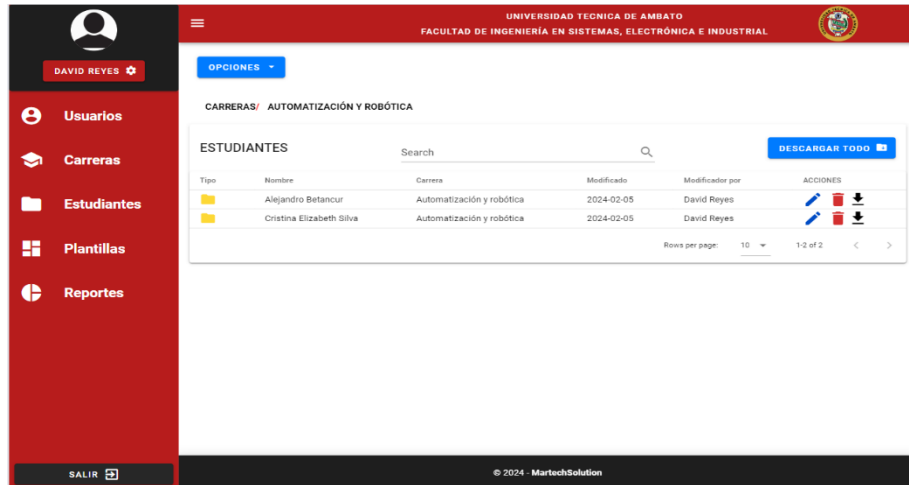


Figura 52: Pagina de estudiantes

En esta interfaz se mostrará los estudiantes por carrera como se puede evidenciar en la Figura 52, desde esta interfaz ya tendremos la opción para poder crear un estudiante nuevo.

```
cargarEstudiantes: async function ({ commit, rootState }, { idCar, idUser }) {
  try {
    const datosEST = new FormData();
    datosEST.append('idUser', idUser);
    datosEST.append('idCar', idCar);
    const setting = {
      method: 'POST',
      body: datosEST,
    };
    const url = 'http://localhost/Api-UTA/estudiantesSelect.php';
    const data = await fetch(url, setting);
    const json = await data.json();

    if (data.ok) {
      commit('llenarItems', json);
    } else {
      console.log('La respuesta no es un JSON válido:', await data.text());
    }
  } catch (error) {
    console.error('Error en la solicitud:', error);
  }
},
```

Figura 53: Código .js para consumo de API cargar permisos usuarios

En la Figura 53 se puede observar al método para cargar los estudiantes, dicho método requiere de dos parámetros como son el “idUser” y “IdCarrera”, con esos datos podremos hacer la llamada a la API.

```
abrirVentana(item) {
  const recuperarPermisos = localStorage.getItem('PermisosSubDirectorios');
  const permisosSubdirectorio = new Map(JSON.parse(recuperarPermisos));
  console.log("nuevos permiso" + permisosSubdirectorio.get("CEDULA"));
  this.obtenerPermisosDirectorios();

  this.setBreadcrumbs(item.nomCar.toUpperCase());
  this.obtnerIdCarrera();
  const storedUser = JSON.parse(localStorage.getItem('user'));
  this.cargarEstudiantes({ idCar: this.carrera, idUser: storedUser.IdUser });
  this.setCarreraSelect(item.nomCar);
  this.setVentanaCarreras(false);
  setTimeout(() => {
    this.setVentanaEst(true);
  }, 3000);
},
```

Figura 54: Método para cargar estudiantes

En el código .js de la Figura 54 se puede observar el método “abrirVentana” el mismo que se encargará de llamar al método “cargarEstudiantes” y algunas acciones más, este método recurre al “IdCarrera” como “idUser” que son los datos requeridos para el método.

- **Ventana para crear estudiantes**

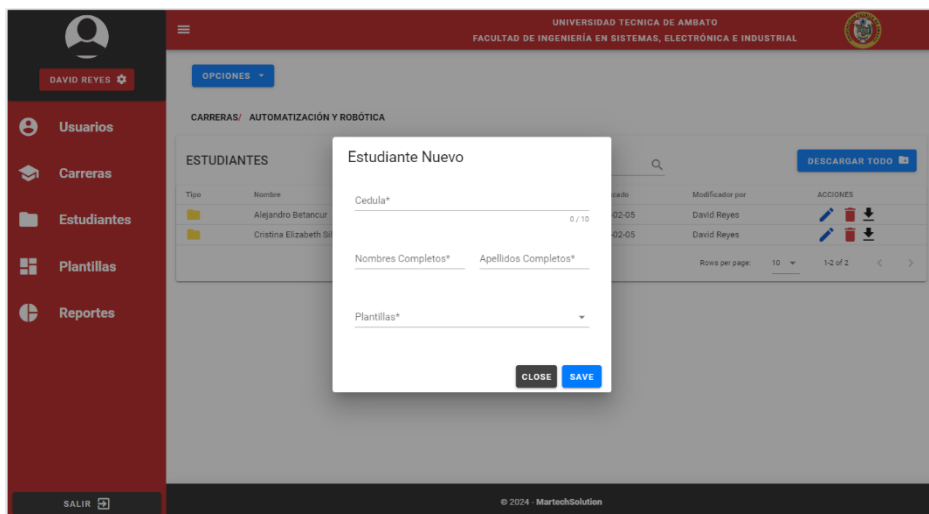


Figura 55: Ventana modal para crear un estudiante

En la Figura 55 se puede observar la ventana diseñada para poder generar un nuevo estudiante, misma que requiere de algunos datos sean ingresados, uno de los datos importantes es la selección de la plantilla, ya es usado para conocer las subcarpetas a crear.

```
agregar: async function () {
  try {
    //console.log(this.ItemEstudiante);
    this.rutaNueva();
    console.log(this.path);
    const storedUser = JSON.parse(localStorage.getItem('user'));
    this.idUser = storedUser.IdUser;
    //console.log(this.ItemEstudiante);
    if(this.itemsBread.length<3){
      await this.AgregarEstudiante(this.ItemEstudiante);
      await this.crearCarpeta({datos:this.ItemEstudiante, path:this.path, oldPath:this.rutaAnterior });
      if( this.ItemEstudiante.IdEst == 0 ){
        await this.crearSubDirectorios( this.ItemEstudiante, this.path );
      }
      await this.cargarEstudiantes({ idCar: this.idCarreraSelect, idUser: this.idUser});
    }else{
      await this.crearCarpeta({datos:this.ItemEstudiante, path:this.path, oldPath:this.rutaAnterior });
      await this.cargarEstudiantes({ idCar: this.idCarreraSelect, idUser: this.idUser });
    }
    this.$alertify.success(this.ItemEstudiante.IdEst == 0 ? "Estudiante creado" : "Estudiante Actualizado");
    this.cerrarDialog();
    this.path='';
  } catch (error) {
    console.error('Error al agregar estudiante:', error);
  }
},
```

Figura 56: Código .js con métodos para crear un estudiante

Para crear un estudiante no solo requiere insértalo en la base de datos, este proceso requiere de más métodos como generar estudiante en el servidor y crear las subcarpetas de la plantilla asignada y posteriormente actualizar el Front-End como se puede observar en la Figura 56.

```
crearSubDirectorios:async function(datosEst, ruta){
  await this.cargarItemsPlantillas(datosEst);
  const cadenaDeItems = this.getItemsPlantillas[0].Items;
  const auxItemsPlan = cadenaDeItems.split(',');
  for( let i=0; i< auxItemsPlan.length; i++ ){
    await this.crearSubCarpeta({datos:datosEst, path:ruta, nombre:auxItemsPlan[i] });
  }
},
```

Figura 57: Código .js para crear subcarpetas

En la Figura 57 se puede observar el método para generar subcarpetas, mismo que hace llamado a otro método y lo itera hasta el número de carpetas que existe en la plantilla.

```
AgregarEstudiante: async function ({ commit, dispatch }, datos) {
  console.log(datos);
  try {
    const datosUser = new FormData();
    datosUser.append('IdEst', datos.IdEst);
    datosUser.append('NomEst', datos.NomEst);
    datosUser.append('ApeEst', datos.ApeEst);
    datosUser.append('CedEst', datos.Cedula);
    datosUser.append('Fecha', datos.Fecha);
    datosUser.append('NomModificador', datos.NomModificador);
    datosUser.append('IdCarPer', datos.NomCar);
    datosUser.append('IdPlanPer', datos.idPlanPer);

    const setting = {
      method: 'POST',
      body: datosUser,
    }
    //console.log("datossss");
    //console.log(datos);
    var url = "";
    if (datos.IdEst == 0) {
      url = 'http://localhost/Api-UTA/insertarEstudiantes.php';
    } else {
      url = 'http://localhost/Api-UTA/actualizarEstudiante.php';
    }
    const data = await fetch(url, setting);
    const json = await data.text();
    if (json.startsWith('{')) {
      const jsonData = JSON.parse(json); // Analiza como JSON si parece válido
      //dispatch('cargarEstudiantes');
    } else {
      //dispatch('cargarEstudiantes');
    }
  }
}
```

Figura 58: Código .js para API crear estudiante

En la Figura 58 se muestra el método agregar estudiantes mismo que recibe un parámetro de tipo objeto el cual contendrá los datos del estudiante, este método cumple dos funciones como es insertar y actualizar dependiendo del “IdUser” se consumirá la API necesaria.

- **Ventana para importar datos**

En la Figura 49 se puede observar el botón opciones desplegado que cuenta con algunas funciones, una de ellas es importar datos, los datos deberán ser importados de un archivo Excel con un formato especificado, de igual manera se deberán generar carpetas y subcarpetas en el servidor.

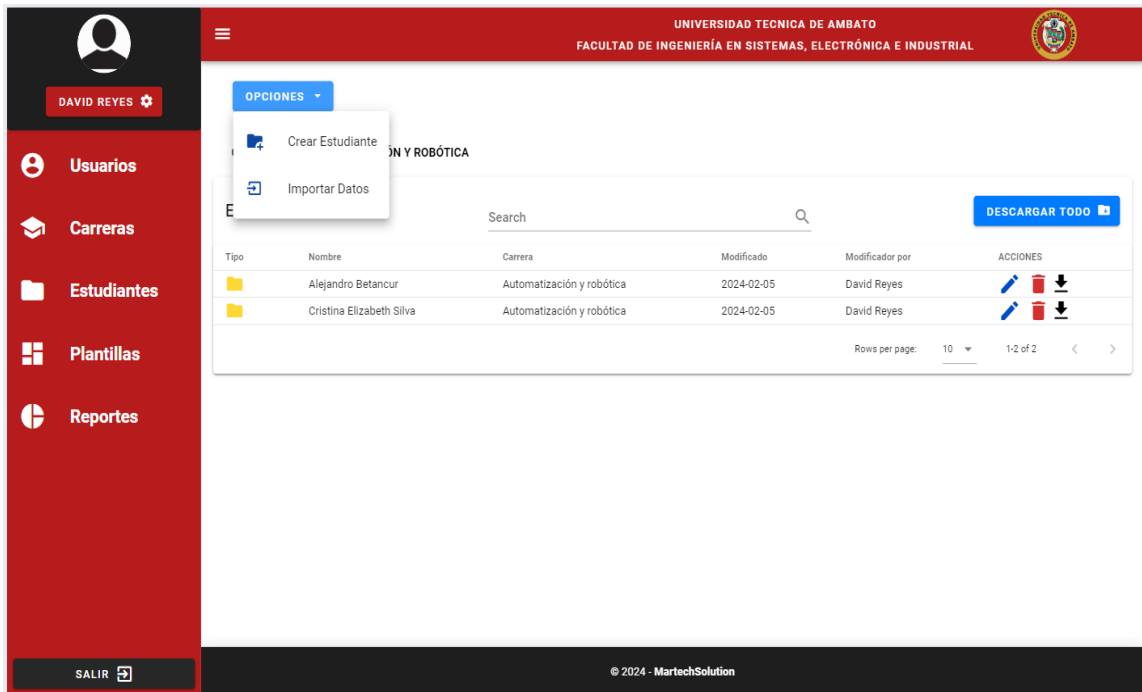


Figura 59: Opción para importar datos desde un archivo Excel

```

subirExcel() {
  this.rutaNueva();
  const storedUser = JSON.parse(localStorage.getItem('user'));
  this.idUser = storedUser.IdUser;
  this.nomUser = storedUser.NomUser + ' ' + storedUser.ApeUser;
  let input = document.getElementById("archivoExcel");
  readXlsFile(input.files[0]).then(async (rows) => {
    this.items = rows;
    this.fechaActual();
    for (let i = 1; i < this.items.length; i++) {
      const row = this.items[i];
      console.log(row);
      const datos = {
        IdEst: row[0],
        NomEst: row[1],
        ApeEst: row[2],
        Cedula: row[3],
        Fecha: this.fecha,
        NomModificador: this.nomUser,
        NomCar: this.idCarreraSelect,
        idPlanPer: row[4],
      };
      await this.AgregarEstudiante(datos);
      await this.crearCarpeta({ datos: datos, path: this.path.trim(), oldPath: this.rutaAnterior.trim() });
      //await this.crearSubDirectorios(datos, this.path);
    }
    await this.cargarEstudiantes({ idCar: this.idCarreraSelect, idUser: this.idUser });
    this.$alertify.success("Estudiantes Insertados");
    input.value = null;
  });
}

```

Figura 60: Método para crear subcarpetas al importar datos

El método de la Figura 60 se encarga de capturar el archivo Excel y recorrerlo fila por fila para inserta los estudiantes tanto en la base de datos como en el servidor, de igual forma genera las subcarpetas.

- **Pagina para mostrar las subcarpetas**

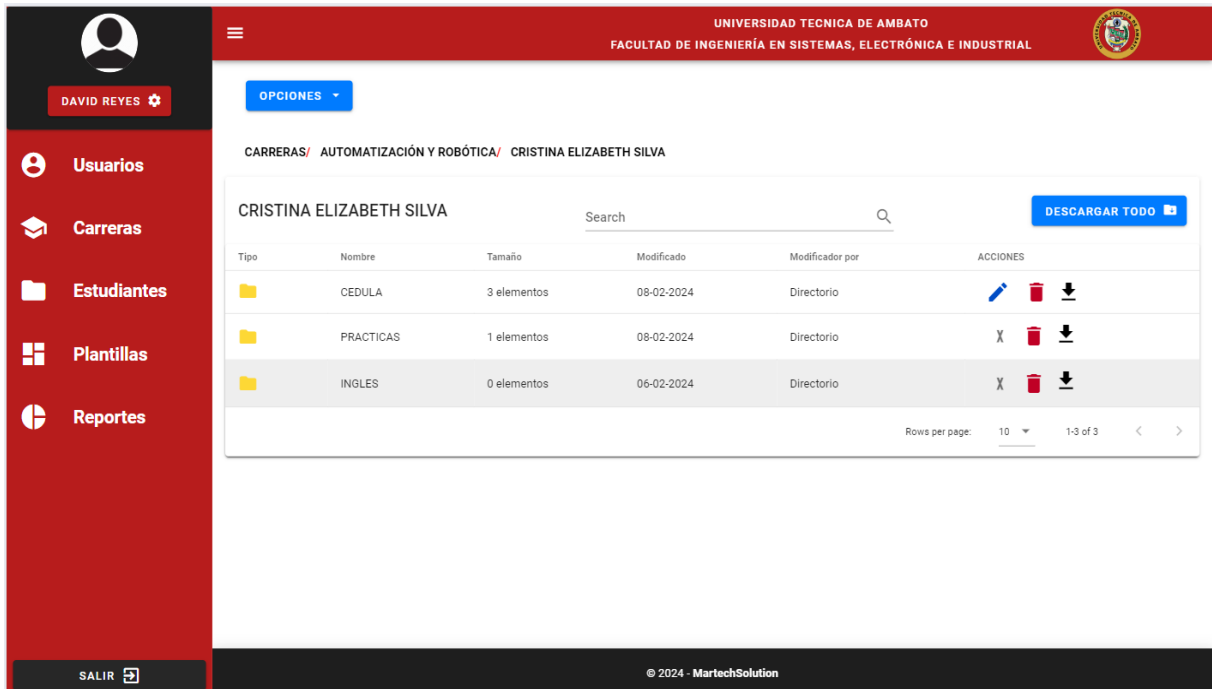


Figura 61: Pagina para mostrar subcarpetas de cada usuario

En la Figura 61 se puede observar las subcarperas que se generan a partir de la plantilla seleccionada, además estas carpetas tienen asignados ciertos permisos que el usuario puede realizar, mismos permisos serán heredados por sus subcarpetas o archivos.

Para poder obtener los permisos que el usuario tiene asignado a cada carpeta se hace uso de un método “obtenerPermisosDirectorios” mismo que a su vez hace uso de otro método “cargarPermisosDirectorio” el cual es que hace la petición a la base de datos, posteriormente se almacenan en el localStorage para su uso como se puede observar en la Figura 62.

```

obtenerPermisosDirectorios: async function () {
  console.log('Metodo tienePermisoEnCarpeta lanzado');
  const storedUser = JSON.parse(localStorage.getItem('user'));
  const idUser = storedUser.IdUser;
  for (const item of this.getSubCarpetas) {
    await this.cargarPermisosDirectorios({ idUser: idUser, nomItem: item.NomItem });
    this.permisosDirectorios.set(item.NomItem, this.getPermisosDirectorios[0].NomPer);
    console.dir(`Permisos ${item.NomItem} -> ${this.getPermisosDirectorios[0].NomPer}`);
  }
  const permission = JSON.stringify(Array.from(this.permisosDirectorios.entries()));
  localStorage.setItem('PermisosSubDirectorios', permission);
  console.log(this.permisosDirectorios);
},

```

Figura 62: Método para obtener los permisos del usuario

```

cargarSubCarpetas: async function({commit}){
  try {
    const setting = {
      methods: 'GET',
    }
    const url = 'http://localhost/Api-UTA/subCarpetasSelect.php';
    const data = await fetch(url, setting);
    const json = await data.json();
    commit('llenarSubCarpetas', json);
  } catch (error) {
    console.error('Error en la solicitud:', error);
  }
},

```

Figura 63: Código .js para API cargar subcarpetas

```

descargarItem: async function (item) {
  this.rutaNueva();
  console.log(item.nombre);
  console.log(this.path);
  await this.descargarCarpeta({ ruta: this.path, nombre: item.NomEst + " " + item.ApeEst });
  this.$alertify.success('Archivo Descargado');
  this.path = '';
},

```

Figura 64: Código .js para descargar una carpeta

En la Figura 64 se puede observar el método creado para descargar un archivo donde recibirá en el nombre el archivo para posteriormente crear una ruta que es la ruta donde se encuentra almacenado el archivo en el servidor.

- **Ventana para crear carpetas**

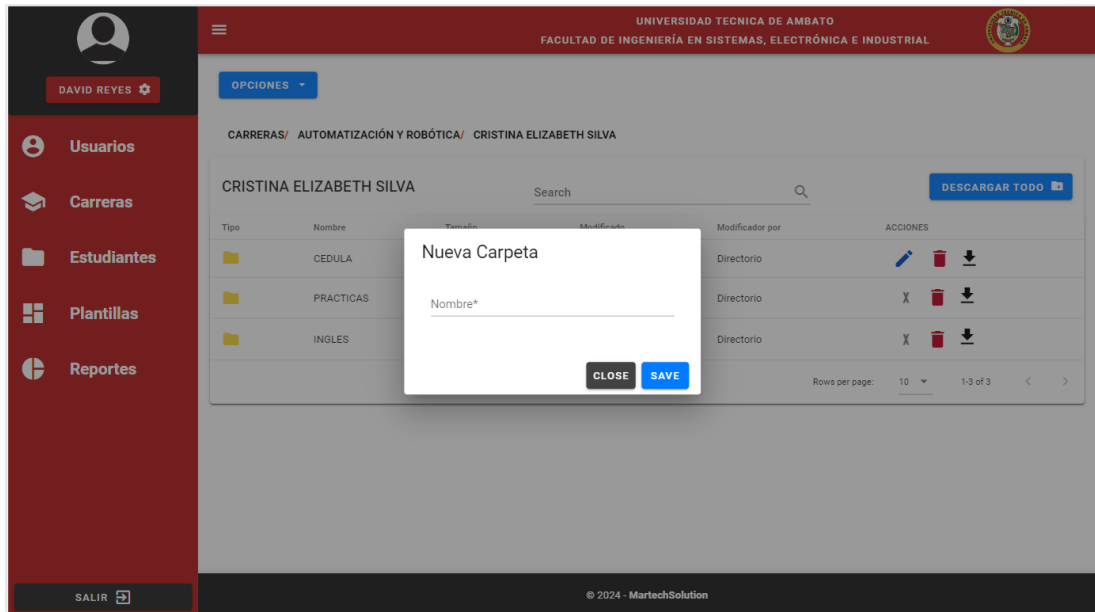


Figura 65: Ventana para crear carpetas

Esta interfaz es diferente a crear un estudiante, pero su opción se encuentra integrado en el mismo botón, cambiará de modal como se puede observar en la Figura 65 dependiendo de la tabla en la se encuentre, sea estudiante o carpetas, mismo modal hará uso de los métodos para crear un estudiante mostrado anteriormente.

- **Pagina para visualizar los archivos insertados**

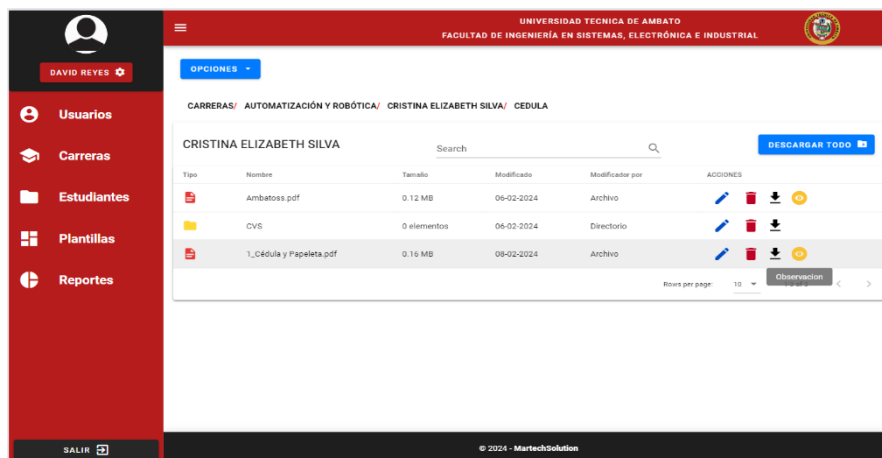


Figura 66: Pagina para mostrar los archivos

```
crearArchivos: async function ({ commit, dispatch }, { ruta, archivo }) {
  try {
    const datosArchivos = new FormData();
    datosArchivos.append('rutaRemota', ruta);
    datosArchivos.append('file', archivo);

    const setting = {
      method: 'POST',
      body: datosArchivos,
    }
    const url = "http://localhost/Apiis-UTA/crearArchivos.php";
    const data = await fetch(url, setting);
    const json = await data.text();
    if (json.startsWith('{')) {
      const jsonData = JSON.parse(json);
      //dispatch('cargarArchivos');
    } else {
      //dispatch('cargarArchivos');
    }
  } catch (error) {
    console.error('Error en la solicitud:', error);
  }
},
```

Figura 68: Código para API crear archivos

La Figura 67 cuenta con el menú de opciones desde donde podremos subir un archivo al servidor, para poder enviar el archivo al servidor se hace uso del método mostrado en la Figura 68, mismo que requiere como parámetros la ruta donde subir el archivo y nombre de este.

```
rutaNueva() {
  //metodo para obtener la ruta
  for (let i = 0; i < this.itemsBread.length; i++) {
    this.path += this.itemsBread[i] + "/";
  }
  //console.log(this.path);
},
```

Figura 69: Método para obtener la ruta

El método mostrado en la Figura 69 es el empleado para poder obtener la ruta a la cual enviar el archivo, este método trabaja con arreglo, se almacena los nombres donde el usuario hizo un click, ejemplo “Software/Emily Vinueza/Cedula”.

```
descargarArchivo: async function ({ commit }, {ruta, nombre}) {
  try {
    console.log('RUTA '+ruta);
    console.log('NOMBRE'+nombre);
    const datosArchivos = new FormData();
    datosArchivos.append('rutaRemota', ruta );

    const setting = {
      method: 'POST',
      body:datosArchivos,
    }
    const url = "http://localhost/Api-UTA/descargarDir_Arch.php";
    const response = await fetch(url, setting);
    //const json = await response.text();
    if (response.ok) {
      const blob = await response.blob();
      // Crear un enlace de descarga y hacer clic en él
      const url = window.URL.createObjectURL(blob);
      const a = document.createElement('a');
      a.href = url;
      a.download = nombre; // Nombre del archivo que se guardará localmente
      document.body.appendChild(a);
      a.click();
      window.URL.revokeObjectURL(url);
    }
  } catch (error) {
    console.error('Error en la solicitud:', error);
  }
},
```

Figura 70: Código para API descargar archivos

En la Figura 70 se puede evidenciar el método empleado para descargar un archivo, este conecta con el servidor a través del API y requiere como parámetros la ruta donde se encuentra el archivo donde se hace uso del método mostrado en la Figura 69 y el nombre del archivo con el cual se guardará este en nuestra máquina.

- **Página Usuarios**

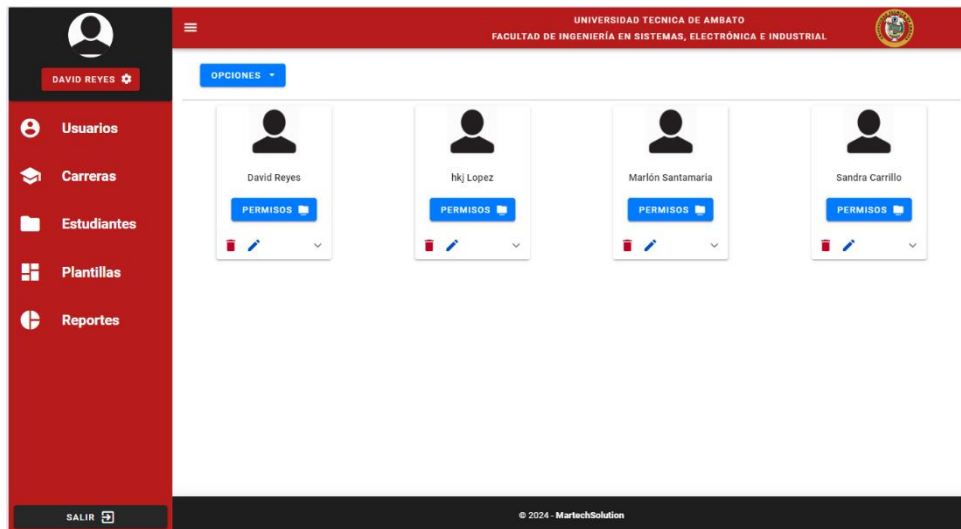


Figura 71: Pagina Usuarios, accedida solo por un usuario admin

La interfaz que se muestra en la Figura 71 es la interfaz que maneja el usuario admin desde donde podrá generar nuevos usuarios y asignarles permisos para los distintos directorios.

```
cargarUsuarios: async function ({ commit }) {  
  try {  
    const setting = {  
      methods: 'GET',  
    }  
    const url = 'http://localhost/Api-UTA/usuariosSelect.php';  
    const data = await fetch(url, setting);  
    const json = await data.json();  
    commit('llenarItems', json);  
  } catch (error) {  
  }  
},
```

Figura 72: Código .js de consumo del API para cargar usuarios

La Figura 72 muestra el método el cual conecta con el servidor a través del API para poder obtener los datos de todos los usuarios.

- Creación de Usuarios

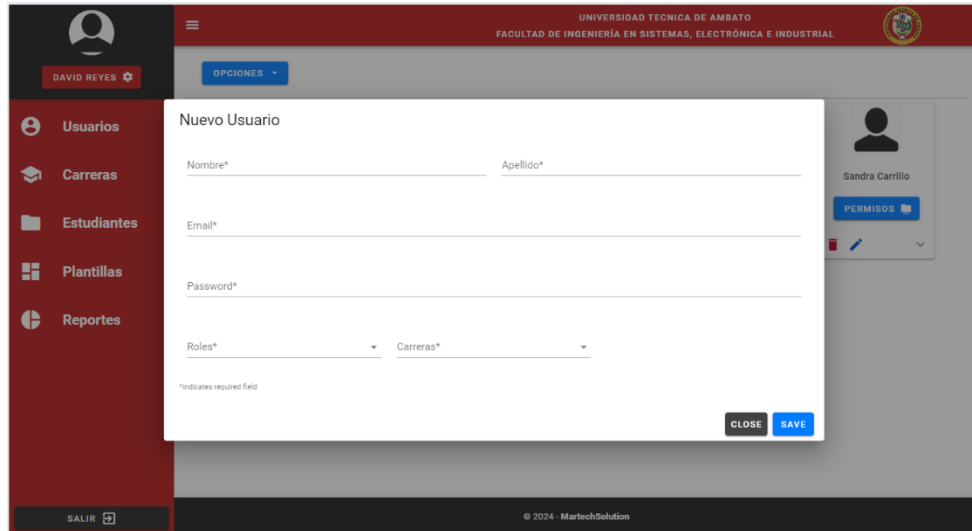


Figura 73: Ventana Modal para crear y editar un usuario

En la Figura 73 se puede observar los campos que serán necesarios para poder generar un nuevo usuario.

```
created() {
  this.cargarCarreras();
  this.cargarRoles();
  this.cargarPermisos();
},

methods: {
  ...mapActions('Carreras', ['cargarCarreras']),
  ...mapActions('Usuarios', ['AgregarUsuario', 'AgregarUsuarioCarreras']),
  ...mapActions('Roles', ['cargarRoles']),
  ...mapActions('Permisos', ['cargarPermisos']),
  ...mapMutations('Dialogo', ['setDialog']),

  agregar: async function () {
    try {
      await this.AgregarUsuario(this.ItemUsuario);
      await this.AgregarUsuarioCarreras(this.ItemUsuario);
      this.cerrarDialog();
      this.limpiarCampos();
      this.$alertify.success(this.ItemUsuario.id == 0 ? "Usuario Insertado" : "Usuario Actualizado");
    } catch (error) {
      console.error('Error al agregar usuario:', error);
    }
  },
}
```

Figura 74: Código .js con métodos para crear un usuario

Como se observa en la Figura 74, para poder crear un usuario se usó de dos métodos, para insertar el mismo usuario y para insertar las carreras a las cuales es asignado.

- **Ventana para asignar y visualizar permisos**

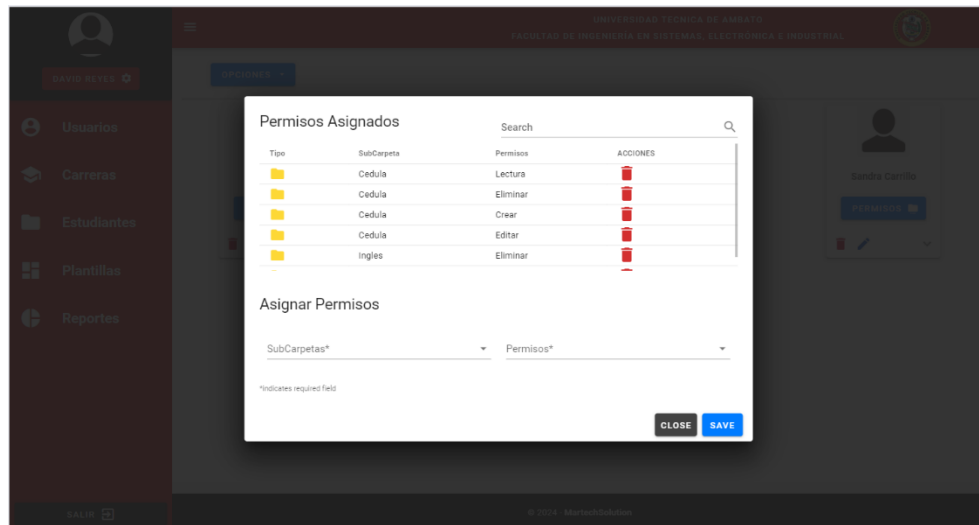


Figura 75: Ventana para ver y asignar permisos a un usuario

La Figura 75 muestra la venta modal que nos permite visualizar los permisos que tiene el usuario y además nos permite generar nuevos permisos.

```

eliminarPermisosSub_User: async function ({ commit, dispatch }, {idUser, idPer, idItem}) {
  try {
    const datosPermisos = new FormData();
    datosPermisos.append('userPer', idUser);
    datosPermisos.append('perPer', idPer);
    datosPermisos.append('itemSub', idItem);

    const setting = {
      method: 'POST',
      body: datosPermisos,
    };
    const url = "http://localhost/Api-UTA/eliminarPermisosSub_User.php";
    const data = await fetch(url, setting);
    const json = await data.text();
    if (json.startsWith('{')) {
      const jsonData = JSON.parse(json);
      dispatch('cargarPermisosSubDir_User', idUser);
    } else {
      dispatch('cargarPermisosSubDir_User', idUser);
    }
  } catch (error) {
    console.log("Error de eliminación " + error);
  }
}

```

Figura 76: Código .js para consumo de API eliminar permisos

Para poder eliminar un permiso a un usuario requerimos de tres parámetros como son “idUser, IdPermiso, Idsubcarpeta”, en la Figura 76 se puede observar cómo funciona este método.

```
AgregarUsuarioPermisos: async function ({ commit, dispatch }, datos) {
  var aux = 0;
  while (aux < datos.IdPerPer.length) {
    try {
      const datosPermisos = new FormData();
      datosPermisos.append('userPer', datos.IdUserPer);
      datosPermisos.append('perPer', datos.IdPerPer[aux]);
      datosPermisos.append('itemSub', datos.IdItemSubPer);

      const setting = {
        method: 'POST',
        body: datosPermisos,
      };

      var url = "";
      if (datos.IdRelacion == 0) {
        url = 'http://localhost/Api-UTA/insertarPermisosUser.php';
      } else {
        url = 'http://localhost/Api-UTA/actualizarPermisosUser.php';
      }

      const data = await fetch(url, setting);
      const json = await data.text();
      if (json.startsWith('{')) {
        const jsonData = JSON.parse(json); // Analiza como JSON si parece válido
        //dispatch( 'cargarPermisosSubDir_User', datos.IdUserPer );
      } else {
        //console.log('La respuesta no es JSON:', data);
        //dispatch( 'cargarPermisosSubDir_User', datos.IdUserPer );
      }
    }
  }
}
```

Figura 77: Código .js para API agregar permisos

```
agregar: async function () {
  if (this.PermisosUsuario.IdItemSubPer > 0 &&
    this.PermisosUsuario.IdPerPer > 0) {
    console.log('Gaurdado');
    console.log(this.PermisosUsuario);
    await this.AgregarUsuarioPermisos(this.PermisosUsuario);
    this.cerrarDialog();
    this.$alertify.success(this.PermisosUsuario.IdRelacion == 0 ? "Permisos Insertados" : "Permisos Actualizado");
  } else {
    this.$alertify.success("Complete todos campos para llevar acabo el proceso");
  }
},

cerrarDialog() {
  this.setDialogPermisos(false);
},
```

Figura 78: Código .js para agregar permisos

En las Figuras 77 y 78 se pueden observar los métodos empleados para poder generar un nuevo permiso, mismo que requiero de un parámetro de tipo objeto que tendrá la información requerida de los permisos para insertar.

- **Pagina Plantillas**

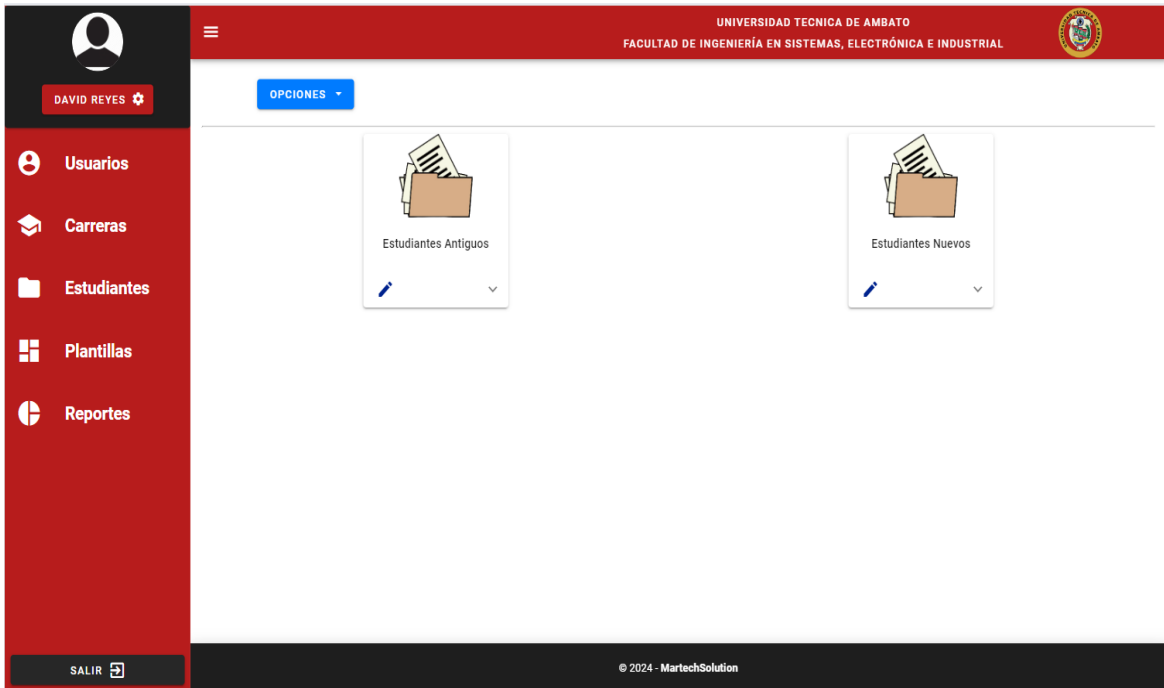


Figura 79: Pagina para poder visualizar y crear plantillas

En la Figura 79 se muestra una interfaz similar a la del usuario, esta interfaz nos permitirá visualizar las plantillas y sus carpetas que existen.

```
created() {
  const isAuthenticated = localStorage.getItem('Authentication') === 'true';
  if (!isAuthenticated) {
    this.$router.push("/");
  } else {
    this.cargarPlantillas();
  }
},
```

Figura 80: Código .js para API cargar plantillas

```
cargarPlantillas: async function ({ commit }) {
  try {
    const setting = {
      method: 'GET',
    };
    const url = 'http://localhost/Apis-UTA/plantillasSelect.php';
    const data = await fetch(url, setting);
    const json = await data.json();

    if (data.ok) {
      commit('llenarPlantillas', json);
    } else {
      console.log('La respuesta no es un JSON válido:', await data.text());
    }
  } catch (error) {
    console.error('Error en la solicitud:', error);
  }
},
```

Figura 81: Código .js para API cargar plantillas

En la Figura 81 se puede visualizar el método empleado para la carga de estos datos, este método muestra una llamada a la base de datos a través de una API. Mientras que en la Figura 70 se puede observar la llamada al método, antes de mostrar datos se verifica que existe un usuario autenticado.

- **Ventana crear Plantillas**

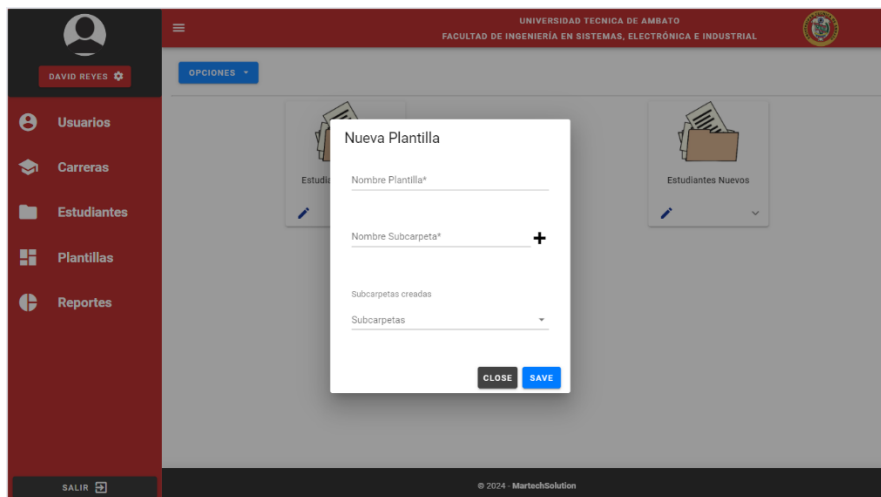


Figura 82: Ventana modal para crear plantillas

En la Figura 82 se puede observar la ventana modal desde la cual podremos generar y editar una plantilla, el icono de (+) tiene dos funcionalidades, la primera es agregar la carpeta a una lista cuando se crea una plantilla y a la segunda funcionalidad es editar una carpeta de la plantilla cuando se edita, dicha funcionalidad se muestra en la Figura 84.

```
agregar: async function() {
  console.log(this.ItemPlantilla);
  await this.AgregarPlantilla( this.ItemPlantilla );
  await this.AgregarItemsSubDirectorios( this.ItemPlantilla );
  await this.AgregarItemsDirectorios( {datos:this.ItemPlantilla, idPlan:'' } );
  this.$alertify.success(this.ItemPlantilla.idPlan == 0 ? "Plantilla creada" : "Plantilla Actualizada");
  this.cerrarDialog();
},
```

Figura 83: Código .js con métodos para crear una plantilla

Al crear una plantilla también necesitamos crear sus carpetas y la relación entre esta, la Figura 83 muestra los métodos empleados para este proceso.

```
agregarItem: async function () {
  // Comprobar si el valor en el campo de texto ha sido modificado
  if (this.aux !== this.item) {
    if (this.ItemPlantilla.idPlan !== 0) {
      console.log("actualizar valor en base de datos");
      await this.actualizarSubCapeta({ nomItem: this.aux, nuevoItem: this.item });
    }
    // Buscar y eliminar el valor antiguo en la lista de items
    const foundItemIndex = this.ItemPlantilla.items.indexOf(this.aux);
    if (foundItemIndex !== -1) {
      this.ItemPlantilla.items.splice(foundItemIndex, 1);
    }
    // Agregar el valor modificado a la lista de items
    this.ItemPlantilla.items.push(this.item);
  }
  console.log(this.ItemPlantilla.items);

  this.item = ''; // Limpiar el campo de texto después de agregar
},
```

Figura 84: Código .js para generar una plantilla y editar una carpeta

```

AgregarPlantilla: async function ({ commit, dispatch }, datos) {
  try {
    console.log(datos.nomPlan.toUpperCase());
    const datosPlantilla = new FormData();
    datosPlantilla.append('NomPlantilla', datos.nomPlan.toUpperCase());

    const setting = {
      method: 'POST',
      body: datosPlantilla,
    }
    var url = "";
    if (datos.idPlan == 0) {
      url = 'http://localhost/Api-UTA/insetarPlantilla.php';
    } else {
      url = 'http://localhost/Api-UTA/actualizarEstudiante.php';
    }
    const data = await fetch(url, setting);
    const json = await data.text();
    if (json.startsWith('{')) {
      const jsonData = JSON.parse(json); // Analiza como JSON si parece válido
      //dispatch('cargarPlantillas');
    } else {
      //dispatch('cargarPlantillas');
    }
  }
}

```

Figura 85: Código para API crear plantilla

```

AgregarItemsSubDirectorios: async function ({ commit, dispatch }, datos) {
  var aux = 0;
  while (aux < datos.items.length) {
    try {
      const datosItem = new FormData();
      datosItem.append('NomSubDirectorio', datos.items[aux].toUpperCase());
      const setting = {
        method: 'POST',
        body: datosItem,
      }
      const url = 'http://localhost/Api-UTA/insertarItemsSubdirectorios.php';
      const data = await fetch(url, setting);
      const json = await data.text();
      if (json.startsWith('{')) {
        const jsonData = JSON.parse(json); // Analiza como JSON si parece válido
        dispatch('cargarPlantillas');
      } else {
        dispatch('cargarPlantillas');
      }
    } catch (error) {
      console.error('Error en la solicitud:', error);
    }
    aux++;
  }
}

```

Figura 86: Código .js para API crear subcarpetas

```

AgregarItemsDirectorios: async function ({ commit, dispatch }, {datos, idPlan }) {
  var aux = 0;
  while (aux < datos.items.length) {
    try {
      const datosItem = new FormData();
      datosItem.append('NomItem', datos.items[aux].toUpperCase());
      datosItem.append('IdPlan', idPlan);
      const setting = {
        method: 'POST',
        body: datosItem,
      }
      var url = "";
      if (datos.idPlan == 0) {
        url = 'http://localhost/Api-UTA/insertarItemDirectorio.php';
      } else {
        url = 'http://localhost/Api-UTA/actualizarCarreerasSecre.php';
      }
      const data = await fetch(url, setting);
      const json = await data.text();
      if (json.startsWith('{')) {
        const jsonData = JSON.parse(json); // Analiza como JSON si parece válido
        dispatch('cargarPlantillas');
      }
    }
  }
}

```

Figura 87: Código .js para API crear relación entre plantillas subcarpetas

En las Figuras 85, 86 y 87 se puede observar los métodos requeridos para crear una plantilla.

3.2.4. Fase IV: Pruebas

a. Pruebas de aplicación

Tabla 20: Prueba de aplicación - Requerimiento R001

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R001	Autenticación de usuario	Controlar quienes pueden tener acceso al sistema	El usuario deberá ingresar su correo y su contraseña en los campos señalados	Ingreso a la página principal del sistema dependiendo del rol	Aprobado

Tabla 21: Prueba de aplicación - Requerimiento R002

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R002	Gestión de usuarios	Creación de usuarios	La interfaz estará compuesta por un formulario con los	Si todos los datos son ingresados correctamente, se creará	Aprobado

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
			siguientes campos: Nombre, Apellido, Correo, Contraseña, Rol de usuario y las carreras a asignar	un nuevo usuario y le mostrará un mensaje de éxito, por el contrario, se le mostrará un mensaje de error.	
		Editar Usuarios	La pantalla cargará los datos del usuario a modificar, esto serán los siguientes campos: Nombre, Apellido, Correo, Contraseña, Rol de usuario y las carreras a asignar	Si todos los datos son ingresados correctamente, se modificará el usuario y le mostrará un mensaje de éxito, por el contrario, se le mostrará un mensaje de error	Aprobado
		Visualización de Usuarios	No necesita ingresar datos	El sistema le mostrará todos los usuarios que existen hasta el momento	Aprobado
		Eliminación de usuario	El usuario deberá seleccionar el usuario a eliminar	Se eliminará el usuario y los datos se recargarán automáticamente	Aprobado

Tabla 22: Prueba de aplicación - R003

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R003	Manejo de roles	Deberán manejar roles para las secretarías y un usuario admin	El usuario deberá ingresar su correo y su contraseña en los campos señalados	El usuario tendrá acceso a ciertas funcionalidades dependiendo de su rol	Aprobado

Tabla 23: Prueba de aplicación - R004

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R004	Control de permisos a para usuarios	Como admin debo controlar las acciones que cada usuario podrá hacer en los archivos que están en el sistema,	Seleccionar un usuario al cual desee dar permisos, se le mostrará al usuario los permisos que existen y las carpetas a las cuales podrá permitir	Se le mostrará en una tabla con los permisos asignados en ese momento,	Aprobado

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
		ya sea eliminar, ver o editar	acceder al usuario seleccionado	así como los que ya tenía	

Tabla 24: Pruebas de aplicación - R005

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R005	Gestión de directorios	Visualizar directorios	No necesita ingresar datos	El sistema le mostrará todos los directorios que existen hasta el momento en el servidor	Aprobado
		Creación de carpetas	Se le mostrará una interfaz donde deberá ingresar el nombre del directorio a crear	Si los datos son correctos se creará la carpeta con sus respectivas subcarpetas dependiendo de la plantilla que tenga el usuario asignado	Aprobado
		Eliminación de directorios	Se deberá seleccionar el directorio a eliminar	Se eliminará el directorio y los datos se recargarán automáticamente, además se le enviará un mensaje de éxito	Aprobado
		Edición de directorios	La pantalla cargará los datos del directorio a modificar	Se modificará el nombre del directorio y se recargarán los datos	Aprobado

Tabla 25: Pruebas de aplicación - R006

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R006	Gestión de estudiantes	Visualizar estudiantes	No necesita ingresar datos	El sistema le mostrará todos los estudiantes que existen y están asignados a ese usuario	Aprobado
		Creación de estudiantes	Se le mostrará una interfaz	Si los datos son correctos, se creará el	Aprobado

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
			donde deberá ingresar los siguientes campos: Cédula, Nombre y La plantilla a cuál pertenece (nuevo o antiguo) estudiante	estudiante y se le enviará un mensaje de éxito	
		Eliminación de estudiantes	Se deberá seleccionar el estudiante a eliminar	Se eliminará el estudiante y los datos se recargarán automáticamente, además se le enviará un mensaje de éxito	Aprobado
		Edición de estudiante	Se los datos del estudiante seleccionado donde podrá editar los siguientes campos: Cédula, Nombre y La plantilla a cuál pertenece (nuevo o antiguo) estudiante	Si los datos son correctos, se modificará el estudiante y se le enviará un mensaje de éxito	Aprobado

Tabla 26: Pruebas de aplicación - R007

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R007	Importación de archivos Excel	Como usuario debo poder importar grandes cantidades de datos al sistema y estos se creen como directorios	La interfaz le mostrará al usuario el explorador de archivos donde podrá seleccionar un archivo Excel para importar sus datos	Los datos del archivo se cargarán en la Base y se mostrarán en la interfaz, además por cada dato se creará las subcarpetas dependiendo del tipo de usuario	Aprobado

Tabla 27: Pruebas de aplicación - R008

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R008	Gestión de plantillas	Visualización de plantillas	No necesita ingresar datos	La interfaz le mostrará al usuario todas las plantillas que existen hasta el momento con sus respectivos directorios	Aprobado
		Crear una plantilla	Se le mostrará una interfaz donde deberá ingresar el nombre de la plantilla, así como los directorios que tendrá la misma	Si los datos están completos se creará la plantilla y se le mostrará un mensaje de éxito caso contrario, se le mostrará un mensaje de no poder crear la plantilla	Aprobado
		Edita una plantilla	Se le cargaran los datos de la plantilla seleccionada y podrá modificar tanto sus directorios como el nombre de la plantilla	Si los datos son correctos se modificará la plantilla o los directorios y se le enviará un mensaje de éxito	Aprobado
		Eliminación de plantilla	No necesita ingresar datos	No podrá eliminar una plantilla ya que cada plantilla tiene relación con los estudiantes	Aprobado

Tabla 28: Pruebas de aplicación - R009

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R009	Crear una observación por cada archivo creado	Como usuario quiero poder registrar una observación cada	La interfaz le mostrará un campo donde deberá ingresar la observación	La observación se generará y la podrá visualizar desde el sistema	Aprobado

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
		vez que suba un archivo al sistema	cuando suba un archivo		

Tabla 29: Prueba de aplicación - R010

ID	Requerimiento	Descripción	Entrada	Resultado	Estado
R010	Gestión de archivos	Visualizar archivos	No necesita ingresar datos	El sistema le mostrará al usuario los archivos que el estudiante tiene en una determinada carpeta	Aprobado
		Subir un archivo	La interfaz le mostrará el explorador de archivos donde seleccionará el archivo deseado	El archivo se subirá al servidor y seguido se mostrará al usuario	Aprobado
		Editar un archivo	La interfaz cargará el nombre del archivo el cual podrá modificar	Si el nombre del archivo fue cambiado se modificará	Aprobado
		Eliminar un archivo	Deberá seleccionar el archivo a eliminar	El archivo se eliminará del servidor	Aprobado

b. Pruebas de interfaz de usuario

Tabla 30: Pruebas de interfaz de usuario

N	Prueba	Descripción	Observación	Cumple
1	Login	Controlar el acceso al sistema a través de credenciales	Ninguna	si
2	Navegación	Navegación entre las diferentes pantallas del sistema	Ninguna	si
3	Registro	Registrar nuevos usuarios o datos en el sistema.	Ninguna	si
4	Edición	Modificar la información existente en el sistema.	Ninguna	si

N	Prueba	Descripción	Observación	Cumple
5	Eliminación	Eliminar registros o datos previamente ingresados en el sistema.	Ninguna	si
6	Visualización	En situaciones donde se muestren datos, estos deben presentarse de manera coherente y sin inconvenientes	Ninguna	si
7	Responsive	El aspecto del sistema se adapta a diferentes tamaños de pantalla	Ninguna	si
8	Mensajes	Las acciones realizadas en el sistema deben ser notificadas a través de mensajes	Ninguna	si
9	Elementos de interfaces	Los deben estar claramente ubicados y etiquetados	Ninguna	si

c. Pruebas de aceptación

- Iteración 1**

Tabla 31: Pruebas de aceptación - Iteración 1

Prueba de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar el ingreso al sistema a través de credenciales como correo y contraseña.	Aceptado	Usuario
2	Comprobar que se puedan generar, modificar, eliminar usuarios.	Aceptado	Usuario
3	Comprobar el acceso al sistema a través de roles (admin y secretaria)	Aceptado	Usuario
4	Verificar que los datos se almacenen en la base de datos.	Aceptado	Programador
5	Validar la correcta carga de datos en el sistema.	Aceptado	Usuario
6	Verificar campos de entrada.	Aceptado	Usuario

- Iteración 2**

Tabla 32: Pruebas de aceptación - Iteración 2

Prueba de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar el poder asignar permisos a los usuarios.	Aceptado	Usuario
2	Comprobar la validación de los permisos.	Aceptado	Usuario
3	Comprobar que se puedan generar, modificar, eliminar directorios.	Aceptado	Usuario
4	Comprobar que se puedan generar, modificar, eliminar estudiantes.	Aceptado	Usuario
5	Verificar que los datos se almacenen en la base de datos.	Aceptado	Programador
6	Validar la correcta carga de datos en el sistema.	Aceptado	Usuario
7	Verificar campos de entrada.	Aceptado	Usuario

- **Iteración 3**

Tabla 33: Pruebas de aceptación - Iteración 3

Prueba de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar que se puedan importar datos desde un archivo Excel.	Aceptado	Usuario
2	Comprobar que se puedan crear, modificar, eliminar plantillas.	Aceptado	Usuario
3	Comprobar que los archivos se suban al servidor	Aceptado	Programador
4	Verificar que los datos se almacenen en la base de datos.	Aceptado	Programador
5	Validar la correcta carga de datos en el sistema.	Aceptado	Usuario
6	Verificar campos de entrada.	Aceptado	Usuario

- **Iteración 4**

Tabla 34 Pruebas de aceptación - Iteración 4

Prueba de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar que se puedan generar, modificar, eliminar archivos.	Aceptado	Usuario
2	Comprobar que se puedan enviar notificaciones.	Aceptado	Usuario
3	Comprobar que los archivos se suban al servidor.	Aceptado	Programador
4	Validar la correcta carga de datos en el sistema.	Aceptado	Usuario
5	Verificar campos de entrada.	Aceptado	Usuario

3.2.5. Fase V: Implementación

Para la implementación se llevó a cabo un cronograma donde se detalla la cantidad de días requeridos para la última fase de la metodología RAD.

Tabla 35: Cronograma de implementación

N	Actividad	Días									
		1	2	3	4	5	6	7	8	9	10
1	Preparación de datos	■									
2	Verificación de infraestructura	■	■								
3	Instalación de librerías	■	■	■							
4	Despliegue	■	■	■	■						
5	Pruebas de ejecución	■	■	■	■						
6	Entrega del producto	■	■	■	■	■					

Preparación de datos: Se trabajo juntamente con la secretaria general para recopilar los datos que se almacenaran en la base de datos acordes a las necesidades.

Verificación de infraestructura: Se llevó a cabo una revisión exhaustiva de los equipos con el fin de asegurar su correcto funcionamiento y evitar posibles interferencias que puedan afectar el sistema.

Instalación de librerías: Se verificó la presencia de las librerías necesarias en el servidor para habilitar la conexión a la base de datos MySQL en los archivos PHP que contienen los servicios encargados de establecer conexión con el sistema.

Entrega del producto: Finalmente, se entregó el producto terminado, se adjunta la documentación para el usuario en el Anexo C.

3.3. Resultados

Tabla 36: Análisis de tiempos estimados de procesos

N°	Proceso	Tiempo estimado del proceso sin automatizar	Tiempo estimado del proceso automatizado	Porcentaje de mejoramiento
1	Creación de carpeta	60 min	1 min	98.33%
2	Búsqueda, descarga de documentación	15 min	2 min	86.67%
3	Archivar documentación en carpeta estudiantil	10 min	3 min	70%
4	Revisión de la carpeta estudiantil para entregar a secretaria general	15 min	5 min	66.67%

Análisis de tiempos estimados de procesos

Creación de carpeta:

- Se observa un tiempo estimado de 60 minutos para el proceso sin automatizar, que se reduce drásticamente a solo 1 minuto con la automatización.
- El porcentaje de mejora es del 98.33%, lo que indica un cambio significativo en la eficiencia del proceso.

Búsqueda, descarga de documentación:

- El tiempo estimado se reduce de 15 minutos a 2 minutos con la automatización.
- El porcentaje de mejora es del 86.67%, lo que demuestra una mejora notable en la eficiencia del proceso.

Archivar documentación en carpeta estudiantil:

- El tiempo estimado disminuye de 10 minutos a 3 minutos con la automatización.
- El porcentaje de mejora es del 70%, lo que indica una mejora significativa en la eficiencia, aunque ligeramente menor que en los procesos anteriores.

Revisión de la carpeta estudiantil para entregar a secretaría general:

- Se observa una reducción del tiempo estimado de 15 minutos a 5 minutos con la automatización.
- El porcentaje de mejora es del 66.67%, mostrando una mejora sustancial en la eficiencia del proceso, aunque ligeramente menor que en algunos de los otros procesos.

Resultados del modelo TAM

Finalizados las diferentes etapas del ciclo de vida de la metodología RAD y obteniendo un resultado positivo en las pruebas de aceptación, se procede a aplicar el Modelo de Aceptación Tecnológica (TAM) para medir estadísticamente los resultados del presente proyecto.

Tabla 37: Cuestionario de evaluación del modelo TAM

Cuestionario de evaluación	
UP	UTILIDAD PERCIBIDA
UP1	La aplicación le permite gestionar las carpetas académicas de manera más eficiente.
UP2	El acceso y búsqueda a la información es mucho rápido y en tiempos cortos.
UP3	La aplicación se adapta a las necesidades establecidas por la entidad.
UP4	Cómo evalúa la eficacia de los tiempos de respuesta del sistema.
FUP	FACILIDAD DE USO PERCIBIDA
FUP1	El sistema para la gestión de la carpeta académica es sencillo de utilizar
FUP2	El proceso de creación de las carpetas de los estudiantes es fácil e intuitivo.
FUP3	El proceso de almacenamiento de archivos no requiere mucho esfuerzo

Cuestionario de evaluación	
UP	UTILIDAD PERCIBIDA
FUP4	El proceso de importación de datos es fácil de usar y efectivo.

Resultados del cuestionario

Utilidad percibida

Tabla 38: Tabla de frecuencias – Modelo TAM – Utilidad percibida

Utilidad percibida					
UP	Muy en desacuerdo	En desacuerdo	Neutral	De acuerdo	Muy de acuerdo
UP1	0%	0%	10%	60%	40%
UP2	0%	0%	30%	45%	30%
UP3	0%	0%	30%	33%	30%
UP4	0%	0%	16%	25%	28%
MEDIA	0%	0%	22%	41%	32%

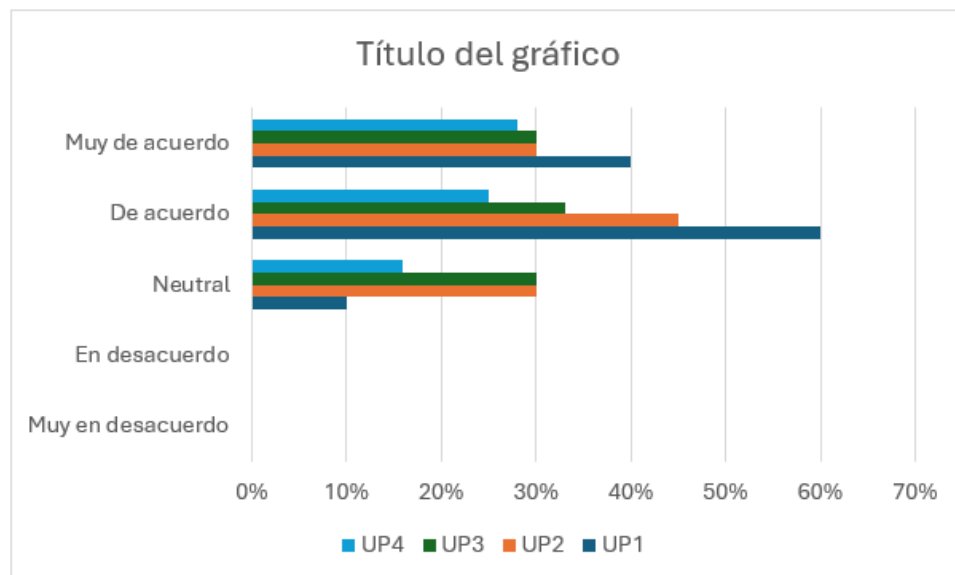


Figura 88: Resultado del modelo TAM- Utilidad percibida

Análisis:

- En general, la mayoría de los encuestados están de acuerdo o muy de acuerdo con las afirmaciones presentadas en las preguntas UP1 a UP4.
- La pregunta UP1 tiene el mayor porcentaje de respuestas "De acuerdo" y "Muy de acuerdo", lo que indica que la afirmación relacionada con UP1 tiene una alta percepción de utilidad por parte de los encuestados.
- Las preguntas UP2, UP3 y UP4 también muestran una tendencia similar, con la mayoría de las respuestas en las categorías "De acuerdo" y "Muy de acuerdo", aunque con algunos porcentajes menores en las categorías "Neutral" y "En desacuerdo".
- El cálculo de la media muestra que, en promedio, el 73% de los encuestados están de acuerdo o muy de acuerdo con las afirmaciones relacionadas con la utilidad percibida.

Facilidad de uso

Tabla 39: Tabla de frecuencias - modelo TAM - Facilidad de uso

Facilidad de uso					
UP	Muy en desacuerdo	En desacuerdo	Neutral	De acuerdo	Muy de acuerdo
UP1	0%	0%	5%	30%	36%
UP2	0%	0%	5%	45%	55%
UP3	0%	0%	10%	20%	50%
UP4	0%	0%	10%	35%	58%
MEDIA	0%	0%	8%	33%	50%

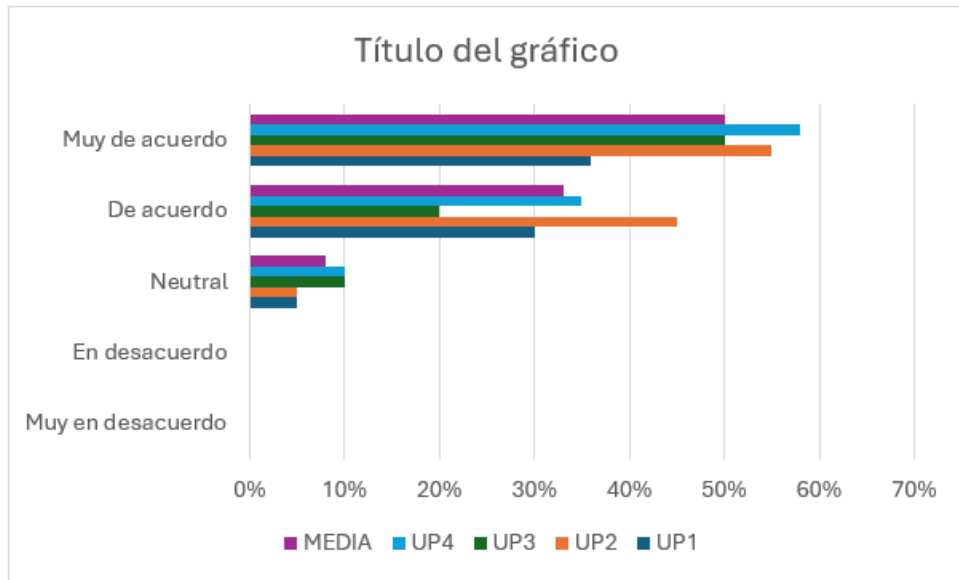


Figura 89: Resultado del modelo TAM- Facilidad de uso

Análisis:

- En general, hay una tendencia positiva en las respuestas para todas las preguntas UP1 a UP4.
- La mayoría de los encuestados están de acuerdo o muy de acuerdo con las afirmaciones presentadas en las preguntas UP1 a UP4.
- La pregunta UP4 tiene el mayor porcentaje de respuestas "De acuerdo" y "Muy de acuerdo", lo que indica que la afirmación relacionada con UP4 tiene la percepción más alta de utilidad por parte de los encuestados.
- El cálculo de la media muestra que, en promedio, el 83% de los encuestados están de acuerdo o muy de acuerdo con las afirmaciones relacionadas con la facilidad de uso.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

En el marco de esta investigación, se propuso como objetivo principal la creación e implementación de una aplicación web multiplataforma destinada a gestionar la carpeta académica de los estudiantes pertenecientes a la FISEI de la Universidad Técnica de Ambato, con un enfoque específico en los departamentos de titulación. Como resultado de este estudio, se derivan las siguientes conclusiones:

- Luego de un profundo análisis minucioso del proceso de gestión de la carpeta estudiantil en las diversas coordinaciones de la FISEI de la Universidad Técnica de Ambato, se emplearon encuestas específicas dirigidas a las secretarías responsables de este procedimiento. Además, se llevó a cabo una entrevista con la secretaria general para obtener información detallada sobre las necesidades y expectativas de los usuarios finales. Este enfoque permitió recopilar de manera exhaustiva los datos necesarios para la definición precisa de los requisitos que conformarán el sistema mencionado anteriormente.
- Luego de la fase de desarrollo centrada en el objetivo de crear un prototipo de aplicación web para gestionar la carpeta estudiantil de la FISEI, concluyo que el prototipo representa un avance significativo en la modernización y eficiencia en la administración de la información académica. El sistema se esfuerza por cumplir las especificaciones y necesidades identificadas en la etapa inicial del proyecto, proporcionando una plataforma funcional y accesible que puede beneficiar positivamente al personal administrativo. Este prototipo representa un paso clave en el camino hacia la implementación de una solución integral para la gestión de carpetas estudiantiles en la mencionada Facultad.
- Después de llevar a cabo la implementación de la aplicación web para la gestión de la carpeta académica de los estudiantes en la FISEI de la Universidad Técnica de Ambato,

se constata que la herramienta puede optimizar eficazmente los procesos vinculados al seguimiento y administración de la información académica. La aplicación se erige como un recurso valioso al facilitar la administración de documentación crucial para el proceso de titulación, cumpliendo así con los requisitos establecidos y mejorando significativamente la eficiencia y accesibilidad. En consecuencia, esta solución se presenta como una contribución positiva y necesaria para la comunidad estudiantil y el cuerpo académico involucrado.

4.2. Recomendaciones

- Se recomienda ampliar la implementación de este sistema para las demás Facultades de Universidad Técnica de Ambato, Actualmente el enfoque se centra exclusivamente en la FISEI.
- Se sugiere documentar detalladamente los procedimientos realizados dentro de la Facultad para facilitar la rápida obtención de requisitos en caso de que se considere una nueva implementación destinada a mejorar algún proceso interno.
- Se sugiere la implementar la enseñanza de nuevas de la metodología de desarrollo como RAD en los estudiantes de la Facultad. Este enfoque nos permite la creación de software en tiempos cortos a través de prototipos funcionales y con interacción directa con el cliente.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Y. Gómez Méndez, “Aplicación Web para el control de procesos de la Unidad de Titulación del Instituto Tecnológico Superior José Chiriboga Grijalva de la ciudad de Ibarra, provincia de Imbabura,” Trabajo de Titulación, Ingeniería en Sistemas Informáticos, Universidad Tecnológica Isael, Quito, 2019.
- [2] J. E. Mendoza Rivilla, “Implementación del sistema Web para la gestión y control de los procesos de la unidad de titulación de la carrera de ingeniería en sistemas de la Universidad Salesiana, sede Guayaquil”, Proyecto Técnico, Ingeniería de sistemas, Universidad Politécnica Salesiana Sede Guayaquil, Guayaquil, 2017.
- [3] M. J. Medina Picuasi, “Desarrollo de una aplicación Web de generación de resoluciones HCD del proceso de titulación FICA-UTN con herramientas libres,” 2017.
- [4] A. J. Alvarado Reyes, “Desarrollo del módulo de aplicación Web para la gestión de sustentación del proceso de titulación en la carrera de Sistemas de Información de la Facultad de Ingeniería Industrial,” Trabajo de Titulación, Desarrollo de software, Universidad de Guayaquil, Guayaquil, 2018.
- [5] "Página Web", Humanidades.com, [Online]. Available: <https://humanidades.com/pagina-web/>
- [6] I. García, M. Polo, F. Ruiz, and M. Piattini, “Servicios Web,” Informe Técnico, Departamento de Informática, Universidad de Castilla-La Mancha, España, 2005.
- [7] "Características de una Página Web", Hostinger, [Online]. Available: https://www.hostinger.es/tutoriales/caracteristicas-de-una-pagina-web?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=1005380&gad_source=1&gclid=Cj0KCQiAzoouBhDqARIsAMdH14FHi-en0fBhuPb9WbHEKcHEAy-k-nlDkv4mQEydod2kM1owsDBhkXhQaAlpiEALw_wcB#Caracteristicas

- [8] D. Crovi Druetta, D. Aguirre, J. Apodaca, and O. Camacho, "Página Web. Una propuesta para su análisis," *Rev Mex Cienc Polit Soc*, vol. 45, no. 185, Apr. 2015, doi: 10.22201/fcpys.2448492xe.2002.185.48362.
- [9] M. C. Gallegos Valera "Introducción a los Servicios Web Índice," 2008. [Online]. Available: <chrome-extension://efaidnbnmnibpcajpcglclefindmkaj/http://repositorio.utn.edu.ec/bitstream/123456789/624/3/CAPÍTULO%20IV.pdf>
- [10] J. M. Rosa Moncayo, "Qué es REST: conoce su potencia", OpenWebinars, [Online]. Available: <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>
- [11] A. Ullate, J. María, S. Oliván, J. Antonio, and P. Clave, "Interfaces de usuario: diseño de la visualización de la información como medio para mejorar la gestión del conocimiento y los resultados obtenidos por el usuario."
- [12] L. Sebastián and M. Gómez, "Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación."
- [13] Illusion Studio, "¿Qué es un Framework Web?", Illusion Studio, 2022. [Online]. Available: <https://www.illusionstudio.es/que-es-un-framework-web>
- [14] "Frameworks en el Desarrollo Web: las Mejores Prácticas para tu Negocio Online", We Are Marketing, [Online]. Available: <https://www.wearemarketing.com/es/blog/frameworks-en-el-desarrollo-web-las-mejores-practicas-para-tu-negocio-online.html>
- [15] "What is IDE?", Amazon Web Services, [Online]. Available: <https://aws.amazon.com/es/what-is/ide/>
- [16] "What is an IDE?", Red Hat, [Online]. Available: <https://www.redhat.com/es/topics/middleware/what-is-ide>

- [17] Talently, "¿Qué es Flutter?" Talently Blog. [Online]. Available: <https://talently.tech/blog/que-es-flutter/>
- [18] "Qué es Flutter y por qué deberías aprenderlo en 2020", freeCodeCamp, [Online]. Available: <https://www.freecodecamp.org/espanol/news/que-es-flutter-y-porque-deberias-aprenderlo-en-2020/>
- [19] Vue.js, "Guía de Vue.js", [Online]. Available: <https://es.vuejs.org/v2/guide/>
- [20] "Qué es Vue.js y para qué sirve", Dongee, [Online]. Available: <https://www.dongee.com/tutoriales/que-es-vue-js-para-que-sirve/#:~:text=Vue%20JS%20es%20un%20framework,app%20y%20los%20sitios%20web.>
- [21] Hostinger, "¿Qué es React? Tutorial React JS," Hostinger, 2022. [Online]. Available: https://www.hostinger.es/tutoriales/que-es-react#Componentes_y_Props
- [22] Kinsta, "¿Qué es React.js?", Kinsta Base de Conocimiento, [Online]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-react-js/#qu-es-react>
- [23] HubSpot, "¿Qué es React? La Guía Completa," HubSpot Blog, 2022. [Online]. Available: <https://blog.hubspot.es/website/que-es-react>
- [24] Microsoft, "Vue.js: una introducción," Microsoft, [Online]. Available: <https://learn.microsoft.com/es-es/windows/dev-environment/javascript/vue-overview>
- [25] OpenWebinars, "VueJS + Vuetify: crea interfaces de usuario reutilizables," OpenWebinars Blog, [Online]. Available: <https://openwebinars.net/blog/vuejs-vuetify-crea-interfaces-de-usuario-reutilizables/#qué-es-vuetify>

- [26] HubSpot, "Qué es Bootstrap y cómo puede ayudar a tu equipo de marketing," HubSpot, 2022. [Online]. Available: <https://blog.hubspot.es/website/que-es-bootstrap>
- [27] EDteam, "¿Qué es Tailwind CSS y por qué usarlo?," EDteam Blog, 2022. [Online]. Available: <https://ed.team/blog/que-es-tailwind-css-y-por-que-usarlo>
- [28] M. A. Cruz-Chávez, "Bases de Datos, Conceptos y sus Características," Centro de Investigación en Ingeniería y Ciencias Aplicadas, Universidad autónoma del Estado de Morelos, Cuernavaca, Morelos, 2011.
- [29] M.C. Alejandro Gutiérrez Díaz, "Bases de Datos MIS 308", Academia.edu, [Online]. Disponible: https://www.academia.edu/6515020/BASES_DE_DATOS_MIS_308
- [30] J. Sanchez. "Diseño Conceptual de Bases de Datos." [Online]. Available: www.jorgesanchez.net
- [31] J. O. Portillo Peña, "Tutorial de prácticas del gestor de Base de Datos SQL Server," Trabajo de graduación, Escuela de Ingeniería de Sistemas, Universidad del Azuay, Cuenca, 2010.
- [32] Arsys, "MS SQL Server", Arsys Blog, [Online]. Available: <https://www.arsys.es/blog/ms-sql-server>
- [33] "SQL Server", WE Educación, [Online]. Available: <https://we-educacion.com/sql-server>.
- [34] R.A. Harb Hoecker "Mysql My struct query lenguaje," Departamento de Electrónica, Universidad Técnica Federico Santa María.
- [35] M. Tena. (2022, oct 03). "¿Qué es la metodología 'agile'? ¿Revolución de las formas de trabajo?". [Online]. Available:

<https://www.bbva.com/es/innovacion/metodologia-agile-la-revolucion-las-formas-trabajo/>

- [36] M.A. Fonseca Vargas, E.I. Obregón Hernández y L.D. Espinoza Jaén, “La Programación Extrema aplicada al desarrollo del sistema de información Web para la gestión de presupuestos en CEM Comunicaciones S.A.”, Departamento de Computación, UNAN – Managua, Nicaragua.
- [37] Microsoft, "Rapid Application Development (RAD)", Microsoft Power Apps, [Online]. Available: <https://powerapps.microsoft.com/es-es/rapid-application-development-rad/>
- [38] "Metodología RAD: ¿Qué es y sus ventajas?", HostingPlus, [Online]. Available: <https://www.hostingplus.pe/blog/metodologia-rad-que-es-y-sus-ventajas/>
- [39] "Qué es Scrum?", Proyectos Ágiles, [Online]. Available: <https://proyectosagiles.org/que-es-scrum/>
- [40] "Características esenciales del Equipo Scrum", TenStep, [Online]. Available: <https://www.tenstep.ec/portal/articulos-boletin-tenstep/41-scrum/427-caracteristicas-esenciales-del-equipo-scrum#:~:text=Seg%C3%BAn%20se%20define%20en%20el,suficientemente%20peque%C3%B1o%20como%20para%20colaborar>
- [41] "Fases de la Metodología Scrum", ComparaSoftware Blog, [Online]. Available: <https://blog.comparasoftware.com/fases-metodologia-scrum/>

ANEXOS

Anexo A

Cuestionario realizado para la encuesta

Objetivo: Determinar el beneficio y opinión sobre la implementación de un sistema para la creación de la carpeta académica de los estudiantes.

- 1) **¿Ha tenido la oportunidad de utilizar una aplicación web para la gestión de la carpeta académica?**
 - a. Sí
 - b. No

- 2) **¿Qué tipo de herramienta utilizas actualmente para la gestión de la carpeta académica? (Selecciona todas las que correspondan)**
 - a. Hojas de cálculo (como Excel o Google Sheets)
 - b. Plataforma de gestión académica de la universidad
 - c. Otro _____

- 3) **¿Cómo calificarías la efectividad de la herramienta actual en la gestión de tu carpeta académica?**
 - a. Muy efectiva
 - b. Efectiva

- c. Neutral
- d. Poco efectiva
- e. No efectiva

4) ¿Cuál es su opinión sobre la posible implementación de una aplicación web para la gestión de la carpeta académica en términos de mejorar la eficiencia de las tareas administrativas en la Facultad?

- a. Sí
- b. No
- c. No estoy seguro

5) ¿Cuáles son los principales desafíos o dificultades que encuentras al gestionar las carpetas académicas de los estudiantes? (Selecciona todas las que correspondan)

- a. Pérdida de documentos
- b. Falta de seguimiento y actualización en tiempo real
- c. Dificultades para acceder a la información
- d. Falta de automatización
- e. Otra _____

6) ¿Qué características o funcionalidades te gustaría ver en una aplicación web de gestión académica? (Selección múltiple)

- a. Creación de directorios para cada usuario
- b. Plantillas para la creación de carpetas académicas
- c. Notificaciones en caso de falta de documentos en alguna carpeta académica
- d. Importación de grandes cantidades de datos al sistema
- e. Subida de diferentes archivos al sistema
- f. Otra

7) ¿Qué ventajas cree que podría tener una aplicación web multiplataforma para la carpeta académica en comparación con las herramientas actuales que maneja para este proceso? (Selección múltiple)

- a. Optimización de tiempo en la creación de la carpeta académica
- b. Acceso desde cualquier lugar
- c. Mejor organización de los documentos y archivos
- d. Seguridad y control de acceso

8) ¿Qué desafíos consideraría al usar una nueva aplicación web para gestionar la carpeta académica? (Selección múltiple)

- a. Dificultad de usabilidad del sistema
- b. Robo de información
- c. Sistema no cumpla con los requerimientos buscados
- d. Incompatibilidad con navegadores
- e. Otra _____

9) ¿Qué tipo de dispositivo electrónico utiliza con mayor frecuencia para sus actividades laborales

- a. Computadora de escritorio
- b. Computadora portátil
- c. Tableta
- d. Teléfono móvil

10) ¿Cuál considera que sería el impacto más importante de esta aplicación en la gestión académica?

- a. Reducción de errores y duplicación de datos.
- b. Mejora en la comunicación entre estudiantes, profesores y personal administrativo.

- c. Mayor acceso a información relevante.
- d. Mayor capacidad de programación y organización.

11) ¿Cuál es su opinión sobre cómo una aplicación web para la gestión de la carpeta académica podría afectar el tiempo necesario para encontrar información en comparación con los métodos actuales?

- a. Reduciría significativamente el tiempo.
- b. Reduciría moderadamente el tiempo.
- c. No tendría un impacto significativo.
- d. Incrementaría moderadamente el tiempo.
- e. Incrementaría significativamente el tiempo.

Anexo B

Preguntas para entrevista

Guía de entrevista aplicada a la secretaria general de la FISEI.

N°	Pregunta	Respuesta	Observación
1	¿Cuál es el proceso actual de gestión de la carpeta académica de los estudiantes en la FISIE?		
2	¿Cuáles son las tareas específicas que realizan las secretarías en relación con la carpeta académica de los estudiantes?		
3	¿Qué tipo de información se almacena en la carpeta académica de los estudiantes?		
4	¿Cómo se manejan los documentos físicos en el proceso de gestión de la carpeta académica?		
5	¿Qué características o funcionalidades consideran esenciales para una aplicación web de gestión de la carpeta académica?		
6	¿Requieren la capacidad de importar datos desde archivos externos?		

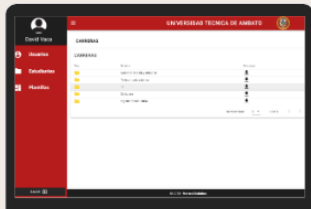
N°	Pregunta	Respuesta	Observación
7	¿Considera necesario un sistema de gestión de permisos y roles para los usuarios de la aplicación?		
8	¿Desean una función de auditoría para rastrear cambios y accesos a los registros académicos?		
9	¿Qué tipo documentos debe manejar aplicación?		

Anexo C

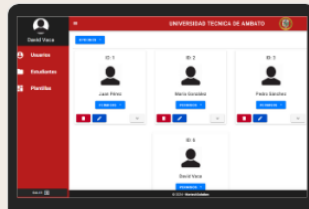
Manual de Usuario



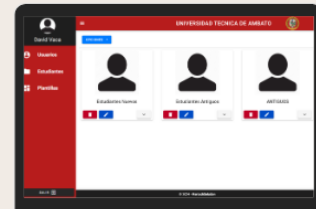
Pantallas Principales de la aplicación



**1) Pantalla
Estudinatess**



**2) Pantalla
Usuarios**



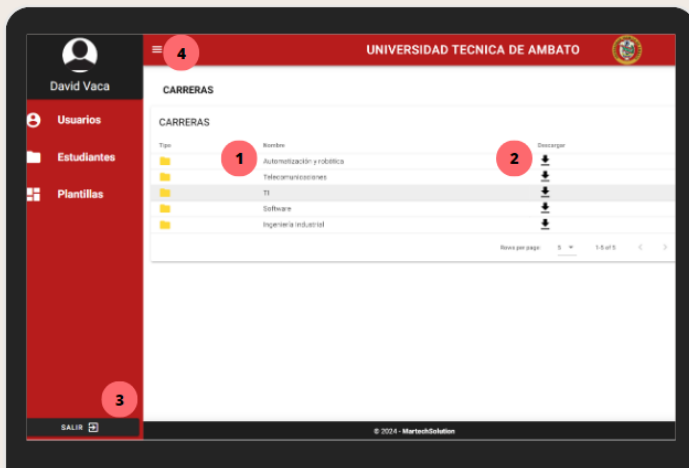
**3) Pantalla
Plantillas**

Pantalla Estudiantes



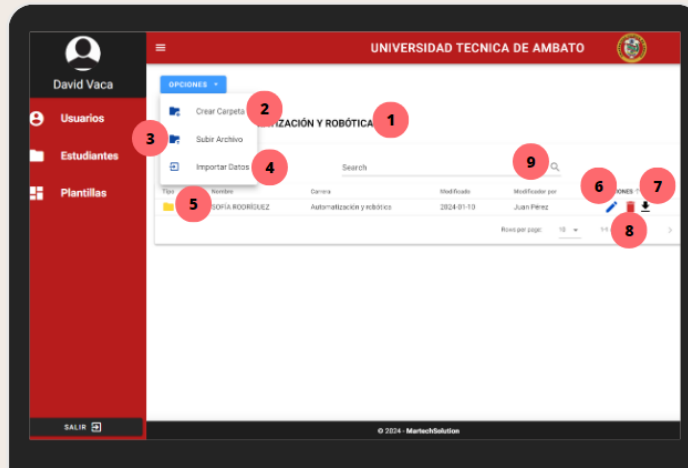
- 1) Sección Login
- 2) Ingresar correo
- 3) Ingresar contraseña
- 4) Sección de botón 'iniciar sesión'

Pantalla Estudiantes



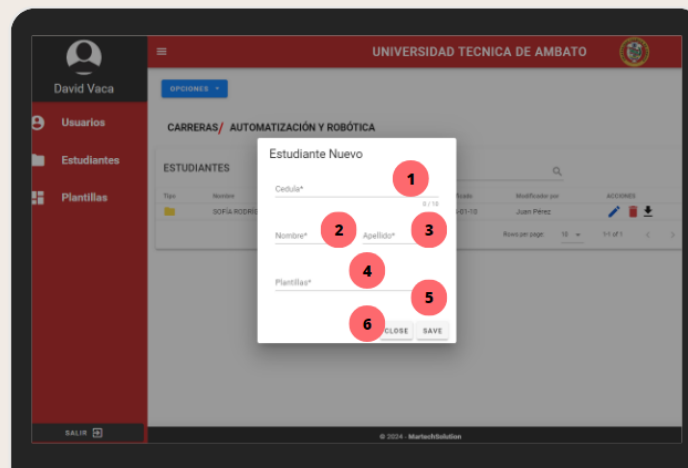
- 1) Ver estudiantes
- 2) Descargar
- 3) Cerrar sesión
- 4) Ocultar menú

Pantalla Estudiantes



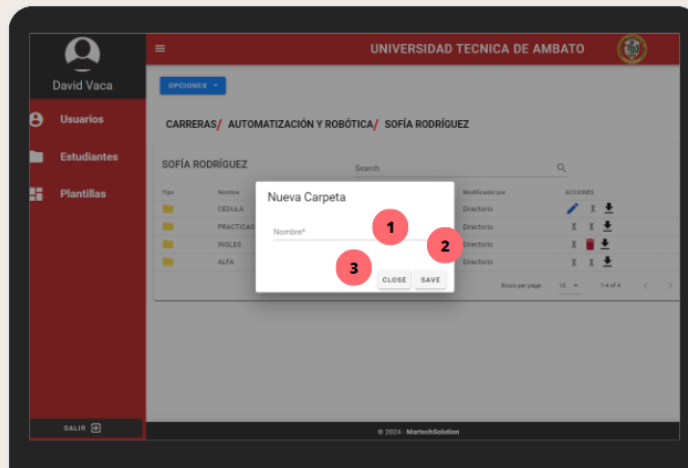
- 1) Navegación
- 2) Crear Estudiantes o carpeta
- 3) Subir un archivo
- 4) Importar datos de Excel
- 5) Ver subcarpetas del estudiante
- 6) Botón editar
- 7) Botón descargar
- 8) Botón borrar
- 9) Botón buscar

Pantalla Estudiantes



- 1) Ingresar cédula
- 2) Ingresar nombre
- 3) Ingresar apellido
- 4) Seleccionar una plantilla
- 5) Botón guardar
- 6) Botón cerrar ventana

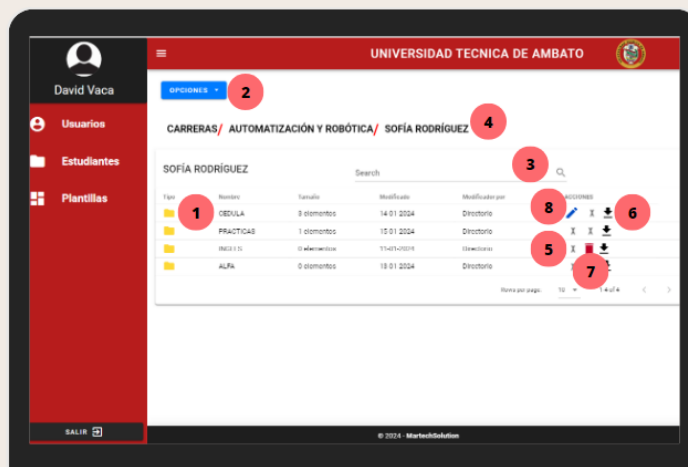
Pantalla Estudiantes



Nota: Cuando estos dentro de la carpeta de un estudiante la ventana modal crear carpeta cambia.

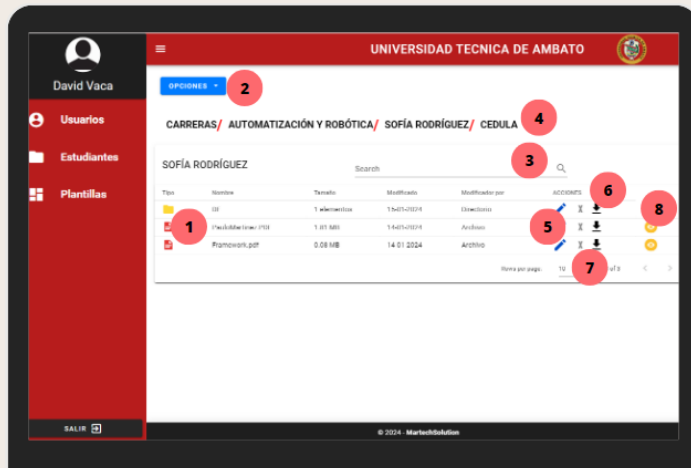
- 1) Ingresar nombre carpeta
- 2) Botón buscar
- 3) Botón guardar
- 4) Botón cerrar ventana

Pantalla Estudiantes



- 1) Ver archivos dentro de la carpeta
- 2) Botón opciones
- 3) Botón buscar
- 4) Navegación
- 5) No tiene permisos
- 6) Botón descargar
- 7) Botón borrar
- 8) Botón Editar

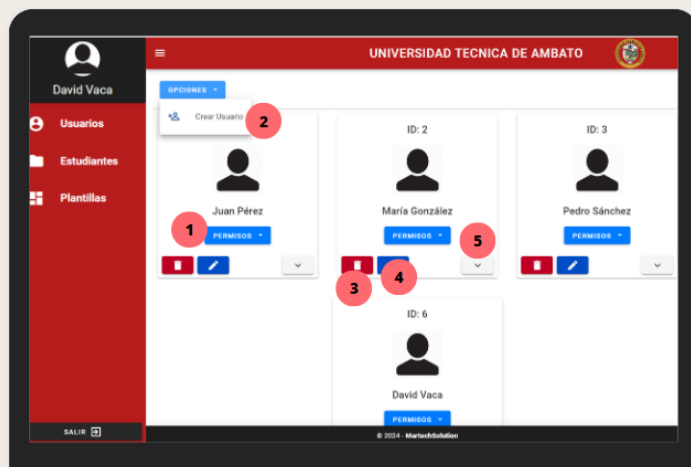
Pantalla Estudiantes



Nota: Las subcarpetas heredan los permisos de las carpetas padres.

- 1) Muestra el archivo
- 2) Botón opciones
- 3) Botón buscar
- 4) Navegación
- 5) Botón editar
- 6) Botón descargar
- 7) Botón borrar
- 8) Ver observación del archivo

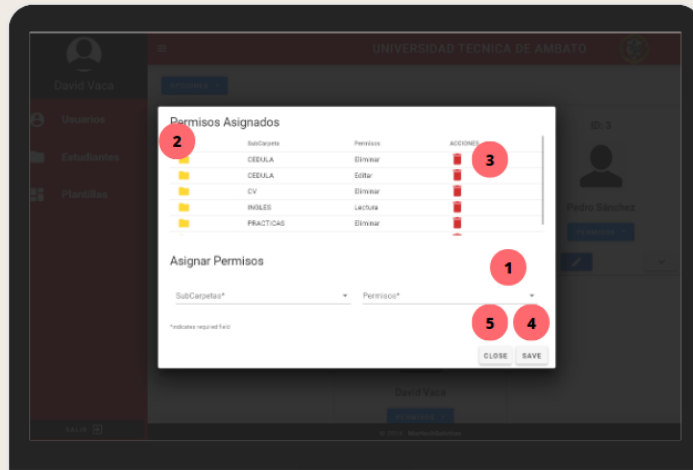
Pantalla Usuarios



Nota: Esta pantalla tendrá acceso solo un usuario admin

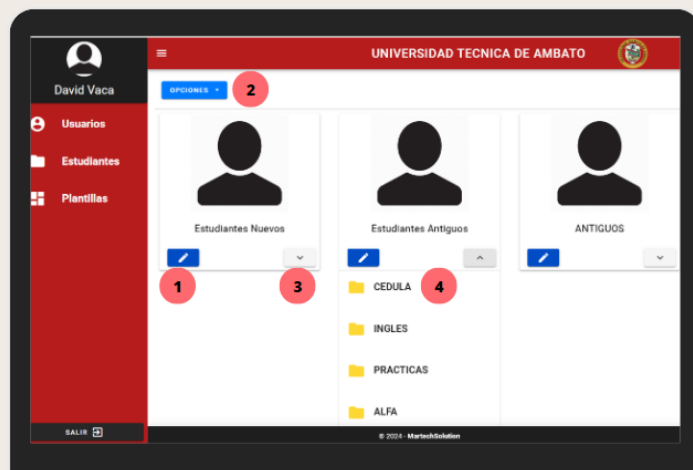
- 1) Visualizar y asignar permisos
- 2) Crear un nuevo usuario
- 3) Botón borrar
- 4) Botón Editar
- 5) Ver detalles del usuario

Pantalla Usuario



- 1) Sección crear nuevos permisos
- 2) Ver permisos
- 3) Botón borrar
- 4) Botón guardar
- 5) Botón cerrar ventana

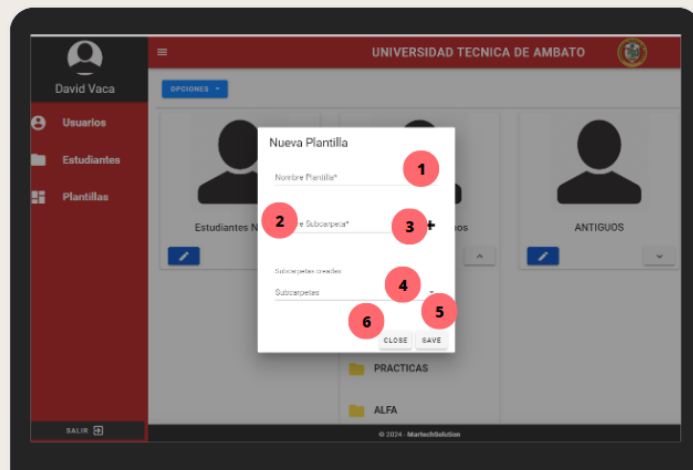
Pantalla Plantillas



Nota: Esta pantalla tendrá acceso solo un usuario admin

- 1) Botón editar
- 2) Botón opciones
- 3) Ver carpetas
- 4) Sección carpetas

Pantalla Plantillas



- 1) Nombre de la plantilla
- 2) Nombre de la carpeta
- 3) Agregar Carpeta a la lista/actualiza la carpeta cuando se edita una plantilla
- 4) Mostrara las carpetas agregadas
- 5) Botón guardar
- 6) Botón cerrar ventana