

UNIVERSIDAD SALESIANA DE BOLIVIA

INGENIERÍA DE SISTEMAS

PROYECTO DE GRADO



**Para obtener el Grado Académico de
Licenciatura en Ingeniería de Sistemas**

SISTEMATIZACIÓN Y ORGANIZACIÓN DE LA DOCUMENTACIÓN DE LA ASOCIACIÓN TUKUY PACHA

Postulante: Danny Gonzales Medina

Docente Guía: Lic. Victor Hugo Sarzuri Flores

COCHABAMBA-BOLIVIA
2025

DEDICATORIA

Dedico este trabajo con profundo amor y gratitud a mi familia, quienes han sido mi pilar fundamental en cada etapa de mi vida. A mis padres, por su esfuerzo incansable, apoyo incondicional y por enseñarme el valor del compromiso y la perseverancia.

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios, por brindarme salud, fortaleza y sabiduría a lo largo de esta travesía académica. A la Universidad Salesiana de Bolivia y a cada uno de los docentes de la carrera de Ingeniería de Sistemas, por su dedicación y por compartir su conocimiento.

Mi más sincero agradecimiento al Lic. Víctor Hugo Sarzuri Flores, docente guía de este proyecto, por su orientación, compromiso y valiosas sugerencias. A la Asociación Tukuy Pacha, por brindarme la oportunidad de trabajar junto a una causa tan noble y por su apertura durante el desarrollo del sistema.

Resumen Ejecutivo

El presente proyecto de grado tiene como objetivo diseñar e implementar un sistema informático que permita optimizar la gestión documental de la Asociación Tukuy Pacha, una organización boliviana dedicada desde 2006 al desarrollo inclusivo de personas con discapacidad. Actualmente, la asociación enfrenta serias limitaciones en la gestión de su información, debido al uso exclusivo de archivos físicos, lo que dificulta el acceso ágil a los datos, incrementa los riesgos de pérdida o deterioro de la documentación y compromete la seguridad y confidencialidad de información sensible.

El sistema propuesto se desarrolla mediante la metodología orientada a prototipos, que permite una constante retroalimentación por parte del usuario y un refinamiento progresivo de los requerimientos. La solución tecnológica se basa en un enfoque de tres capas: una capa de presentación implementada con Vue.js, una capa de aplicación desarrollada en Node.js con Express, y una capa de datos respaldada por MySQL. Se ha implementado una estrategia de seguridad robusta centrada en la autenticación segura con JWT, la protección activa contra vulnerabilidades comunes como la inyección SQL, y una validación integral a través de pruebas de integración para asegurar la estabilidad del sistema.

A lo largo del proceso, se emplearon diversas técnicas de recolección de información, incluyendo entrevistas, encuestas y observación directa, para entender en profundidad los procesos actuales y las necesidades específicas de Tukuy Pacha. El sistema incluye funcionalidades como registro y gestión de participantes, carga y categorización de documentos digitales, control de acceso por roles.

Entre los principales beneficios esperados se destacan la reducción de tiempos de búsqueda y consulta de documentos, el fortalecimiento de la seguridad de la información, la optimización del espacio físico, y la mejora en la eficiencia

operativa y toma de decisiones. Todo ello permitirá a Tukuy Pacha ampliar el impacto de sus programas sociales, mejorando la atención a personas con discapacidad y fortaleciendo su capacidad institucional.

En conclusión, este proyecto aporta una solución tecnológica adecuada a las necesidades y contexto de una organización comprometida con la inclusión y el desarrollo comunitario, constituyendo un paso fundamental hacia su modernización digital.

ÍNDICE

CAPÍTULO I MARCO REFERENCIAL	1
1.1. Introducción.....	2
1.2. Antecedentes	2
1.3. Identificación del Problema	4
1.3.1. Planteamiento del Problema.....	5
1.3.2. Árbol de Problemas.....	5
1.3.3. Formulación del Problema	6
1.4. Objetivos	7
1.4.1. Objetivo General	7
1.4.2. Objetivos Específicos	7
1.5. Justificación.....	7
1.5.1. Justificación Técnica	7
1.5.2. Justificación Económica.....	8
1.5.3. Justificación Social	8
1.6. Alcance.....	8
1.6.1. Temático.....	8
1.6.2. Geográfico	9
1.6.3. Temporal.....	10
CAPÍTULO II MARCO TEÓRICO	11
2.1. Gestión Documental	12
2.2. Seguridad de Información	12
2.3. Metodología de Desarrollo	13
2.3.1. Metodología Orientada a Prototipos	13
2.3.1.1. Ciclo de Vida de un Sistema basado en Prototipo	15
2.4. Técnicas de Recolección de Información	17
2.5. Plataforma de Desarrollo	18
2.5.1. Vue.js	18
2.5.2. Node.js con Express	19
2.6. Administrador de Base de Datos.....	20
2.6.1. MySQL.....	20

2.7. Herramientas.....	22
2.7.1. Visual Studio Code.....	22
2.7.2. Git y Github.....	22
2.7.3. Bootstrap	23
2.7.4. Laragon.....	24
CAPÍTULO III ÁREA DE APLICACIÓN	25
3.1. Antecedentes Generales	26
3.1.1. Mandatos Institucionales: Visión, Misión y Valores	27
3.1.1.1. Visión	27
3.1.1.2. Misión	27
3.1.1.3. Valores	27
3.1.2. Marco de Referencia Legal Política.....	29
3.2. Definiciones.....	30
3.3. Procesos	31
3.4. Documentación e Información.....	34
CAPÍTULO IV DESARROLLO DEL PROYECTO	36
4.1. Análisis del Sistema.....	37
4.1.1. Arquitectura de software	37
4.1.1.1. Capa de Presentación-Vue.js.....	37
4.1.1.2. Capa de Aplicación – Node.js con Express	38
4.1.1.3. Capa de datos - MySql	39
4.1.2. Diagrama de Actividades	40
4.1.3. Diagrama de Secuencia.....	41
4.2. Análisis de Requerimientos	42
4.2.1. Encuesta.....	42
4.2.2. Requerimientos funcionales	49
4.2.3. Requerimientos no funcionales	50
4.2.4. Casos de Uso UML	51
4.2.4.1. Diagrama de Caso de Uso General	51
4.2.4.2. Diagrama de Caso de Uso Ingreso al Sistema.....	52

4.2.4.3. Diagrama de Caso de Uso Administración de Personal Técnico	52
4.2.4.4. Diagrama de Caso de Uso Registro de Participante.....	53
4.2.5. Modelo Entidad-Relación.....	54
4.2.5.1. Diccionario de Datos	54
4.3. Identificación de Riesgos	57
4.4. Interfaz.....	58
4.5. Pruebas	64
4.5.1. Pruebas de Seguridad	64
4.5.1.1. Prueba de Integración.....	64
4.5.1.2. Prueba de seguridad contra inyecciones SQL	84
4.6. Planificación	91
4.6.1. Fases del Desarrollo del Proyecto.....	92
4.6.1.1. Fase Inicial: Escuchar al Cliente y Recolección de Requisitos	92
4.6.1.2. Fase Final: Desarrollo Completo e Implementación.....	95
Conclusiones	96
Recomendaciones	98
Bibliografía.....	100
ANEXOS.....	102

ÍNDICE FIGURAS

Figura 1 Árbol de problemas	6
Figura 2 Ubicación de la oficina sur de Tukuy Pacha	9
Figura 3 Diagrama de Gantt.....	10
Figura 4 Etapas del Modelo de Prototipo	15
Figura 5 Relación entre View, ViewModel y Model en Vue	19
Figura 6 Arquitectura MVC (Model-View-Controller)	20
Figura 7 Arquitectura de una Aplicación Web con MySQL.....	22
Figura 8 Control de Versiones con Git y GitHub	23
Figura 9 Arquitectura de Software General	37
Figura 10 Capa de Presentación	38
Figura 11 Capa de Aplicación	39
Figura 12 Capa de Datos	40
Figura 13 Diagrama de Actividades.....	41
Figura 14 Diagrama de Secuencia.....	42
Figura 15 Frecuencia de Acceso a Archivos de Participantes.....	43
Figura 16 Información Personal Necesaria para Almacenar sobre los Participantes	43
Figura 17 Importancia de las Secciones en el Archivo de Cada Participante..	44
Figura 18 Gestión Actual de los Archivos de los Participantes	44
Figura 19 Métodos Actuales de Acceso a los Archivos	45
Figura 20 Asociación de un Patrocinador a Múltiples Participantes.....	45
Figura 21 Información del Patrocinador que Debería Almacenarse.....	46
Figura 22 Quiénes Deberían Tener Acceso a la Información del Sistema	46
Figura 23 Importancia de Diferentes Niveles de Acceso al Sistema	47
Figura 24 Medidas de Seguridad Necesarias para Proteger la Información	47
Figura 25 Aspectos Importantes para Mejorar la Gestión Documental en la Asociación Tukuy Pacha	48
Figura 26 Diagrama de Caso de Uso General	51
Figura 27 Diagrama de Caso de Uso Inicio de Sesión	52
Figura 28 Diagrama de Caso de Uso Administración de Personal Técnico	53

Figura 29 Diagrama de Caso de Uso Registro de Participante	53
Figura 30 Modelo Entidad-Relación.....	54
Figura 31 Pantalla de Inicio de Sesión	59
Figura 32 Panel de Administración del Sistema.....	60
Figura 33 Formulario para Registrar Nuevo Usuario	61
Figura 34 Lista de Usuarios Registrados.....	61
Figura 35 Formulario para Registrar Nuevo Participante.....	62
Figura 36 Detalles del Participante	63
Figura 37 Panel del Personal del Sistema.....	63
Figura 38 Prueba de Integración de App.js	66
Figura 39 Prueba de Integración de Auth.....	67
Figura 40 Prueba de Integración de Documentos.....	68
Figura 41 Prueba de Integración de Participantes	70
Figura 42 Prueba de Integración de Patrocinadores.....	72
Figura 43 Prueba de Integración de Usuarios.....	73
Figura 44 Prueba de Integración de Routes del Fronted	75
Figura 45 Prueba de Integración del Login.....	76
Figura 46 Prueba de Integración del Panel del Administrador	77
Figura 47 Prueba de Integración del Panel del Personal.....	78
Figura 48 Prueba de Integración al Registrar Participante	80
Figura 49 Prueba de Integración al Editar Participante	81
Figura 50 Prueba de Integración al Registrar Nuevo Usuario	82
Figura 51 Prueba de Integración al Registrar Nuevo Patrocinador	83
Figura 52 Progreso del escaneo de seguridad realizado con ZAP by Checkmarx	85

ÍNDICE TABLAS

Tabla 1 <i>Valores</i>	28
Tabla 2 <i>Normativas Referidas a Las Poblaciones Participantes</i>	29
Tabla 3 <i>Componentes de la Matriz RBC.....</i>	32
Tabla 4 <i>Etapas del Sistema de Planificación, Monitoreo y Evaluación Institucional.....</i>	34
Tabla 5 <i>Requerimientos Funcionales.....</i>	49
Tabla 6 <i>Requerimientos No Funcionales</i>	50
Tabla 7 <i>Identificación de Riesgos.....</i>	58
Tabla 8 <i>Tipos de ataques evaluados durante la prueba de seguridad con OWASP ZAPataques evaluados</i>	86
Tabla 9 <i>Pruebas de seguridad.....</i>	89
Tabla 10 <i>Esquema Basado en la Metodología de Prototipos</i>	92

CAPÍTULO I

MARCO

REFERENCIAL

1.1. Introducción

En la actualidad, la correcta gestión de la información se ha convertido en un elemento fundamental para el desarrollo eficiente de organizaciones que brindan servicios a poblaciones vulnerables. En este contexto, la Asociación Tukuy Pacha, con una trayectoria de más de una década en Bolivia, se dedica al desarrollo inclusivo de personas con discapacidad, ofreciendo apoyo integral que incluye datos personales sensibles e información sobre sus discapacidades y los profesionales que les brindan asistencia.

Sin embargo, la gestión actual de estos datos se realiza principalmente a través de archivos físicos, lo que presenta desafíos significativos. La búsqueda y recuperación de información es lenta y laboriosa, dificultando la toma de decisiones oportunas y estratégicas. Además, la confidencialidad de la información sensible se ve comprometida por la falta de un sistema de seguridad adecuado, y existe el riesgo de pérdida o daño de los archivos físicos, lo que afecta la continuidad del trabajo de la asociación.

Se espera que la implementación de este sistema tenga un impacto significativo en la calidad de la atención brindada y en la eficiencia operativa de la asociación, contribuyendo así al cumplimiento de su misión de desarrollo inclusivo y mejorando su capacidad de respuesta ante las necesidades de sus participantes.

1.2. Antecedentes

En la Universidad de Granma (Cuba) se desarrolló una aplicación web para la gestión de información de pacientes con discapacidad, utilizando una metodología de desarrollo de software XP. El diseño del sistema se basó en tecnologías como Python 2.7, HTML, CSS, JavaScript, Apache 2.0, SQLite 3.3 y Django 1.9 como framework. La muestra incluyó la gestión integral de datos de pacientes, programación de consultas y generación de informes sobre discapacidades, así como reportes mensuales y trimestrales exportables a Excel. Como conclusión, se evidenció que la aplicación optimiza el registro y procesamiento de información, mejorando la consistencia, seguridad y eficiencia

en el control y análisis organizativo, lo que la convierte en una herramienta valiosa para el sector salud (Blanco, 2018).

En el ámbito general, la Muy Ilustre Municipalidad de Guayaquil gestiona centros de apoyo para personas con capacidades especiales, donde se identificó la necesidad de optimizar procesos como matriculación, registro de notas e información médica, actualmente realizados de forma manual, lo que generaba problemas como pérdida de datos e ineficiencia en la búsqueda de información. El tipo de diseño propuesto fue una aplicación web desarrollada con el framework ASP.NET MVC, utilizando la estructura modelo-vista-controlador (MVC), junto con herramientas como Visual Studio, SQL Server para la base de datos, y Bootstrap y JavaScript para el diseño y funcionalidad. La muestra incluyó la automatización de procesos clave, como la matriculación, registro de notas y generación de reportes, dirigida al personal administrativo de los centros de apoyo. Como conclusión, la implementación de esta herramienta permitió mejorar la eficiencia en la gestión de información, reducir errores y optimizar el tiempo de acceso a datos, contribuyendo a un mejor control y organización de los procesos en los centros municipales (Altamirano Bajaña & Palma Murillo, 2018).

La Asociación Tukuy Pacha es una organización activa en Bolivia desde el año 2006. Su misión es contribuir al desarrollo inclusivo, con énfasis en poblaciones de atención prioritaria, mediante acciones estratégicas de base comunitaria. Tukuy Pacha incide en la capacidad de gestión del Estado y en la sociedad civil buscando mejorar las condiciones de vida de los grupos en situación de vulnerabilidad. Por medio de relaciones estratégicas con instituciones como la Fundación Don Carlo Gnocchi ONLUS de Italia, Tukuy Pacha desarrolla y ejecuta propuestas de Rehabilitación Basada en la Comunidad (RBC) en diferentes municipios del Departamento de Cochabamba (Vinto, Arani, Punata, entre otros).

En la actualidad, Tukuy Pacha busca ampliar y profundizar los resultados de su marco de intervención, por medio de proyectos como “Inclusión de Personas con Discapacidad - Generando procesos familiares y comunitarios” y el proyecto

“INCREIBLES – Inclusión y Resiliencia de Personas con Discapacidad”, orientado al desarrollo de habilidades, competencias y emprendimientos para la generación de ingresos y la inclusión laboral. En la zona sur del municipio de Cochabamba, el asocio entre Tukuy Pacha y Save the Children permite realizar iniciativas conjuntas de promoción del desarrollo integral de la infancia, como el proyecto “Inclusión de niñas y niños con discapacidad y retraso en el desarrollo en unidades educativas de nivel inicial y primario” (Tukuy Pacha; Save the Children, 2024).

El crecimiento y diversificación de los programas de Tukuy Pacha han generado una acumulación significativa de documentación, lo que hace cada vez más urgente la implementación de un sistema de gestión documental eficiente. La necesidad de garantizar la seguridad, integridad y accesibilidad de la información ha llevado a la asociación a considerar la digitalización como un paso clave en su evolución. Un sistema informático permitirá mejorar la organización de los datos, reducir los tiempos de búsqueda y consulta, y fortalecer la toma de decisiones basada en información precisa y actualizada. Con esta modernización, Tukuy Pacha podrá optimizar sus recursos y potenciar el impacto de sus programas en beneficio de las poblaciones vulnerables a las que atiende.

1.3. Identificación del Problema

La Asociación Tukuy Pacha ha acumulado una gran cantidad de documentos físicos, lo que ha generado dificultades en la gestión de la información. El almacenamiento de archivos en formato físico ocupa un espacio considerable y dificulta un acceso ágil y eficiente a los datos. La falta de digitalización limita la capacidad de la asociación para responder rápidamente a las necesidades de sus participantes, retrasando procesos administrativos y obstaculizando la toma de decisiones fundamentadas en información actualizada y precisa.

Además, el manejo manual de los documentos expone la información a riesgos como el extravío, deterioro y acceso no autorizado. La falta de protocolos estructurados de seguridad compromete la confidencialidad de los datos

sensibles de los participantes y del personal involucrado en los programas de la asociación. Esta vulnerabilidad no solo representa un problema interno, sino que también puede afectar la credibilidad de la institución ante sus colaboradores y entidades financieradoras.

La recopilación manual de datos requiere un esfuerzo significativo, ralentiza los procesos administrativos y restringe la capacidad de planificación estratégica de la organización. Implementar un sistema informático permitirá optimizar la organización y acceso a la información, fortalecer la seguridad de los datos y mejorar la toma de decisiones dentro de la asociación, consolidando así su labor en favor de las personas con discapacidad.

1.3.1. Planteamiento del Problema

El uso exclusivo de archivos físicos también representa riesgos considerables en términos de seguridad y conservación de la información. La documentación está expuesta a deterioro, extravío y acceso no autorizado, lo que compromete la confidencialidad de los datos de los participantes y del personal. La ausencia de un sistema digital que respalde la información impide establecer mecanismos eficientes de seguridad y control de acceso.

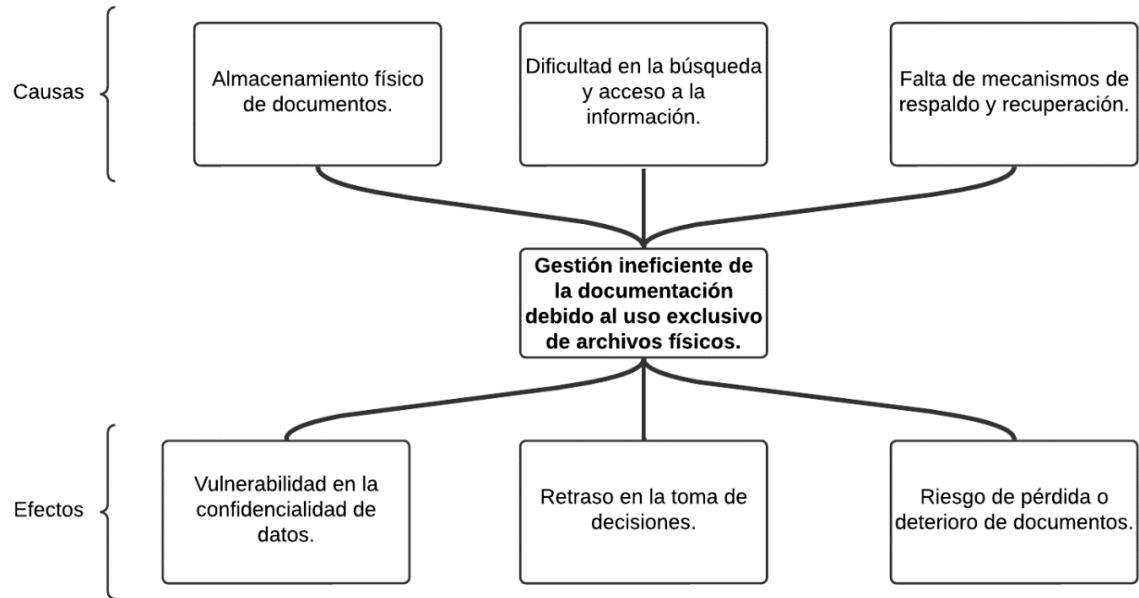
Frente a estos desafíos, se hace evidente la necesidad de implementar un sistema informático que permita la digitalización y organización de la información. Un sistema de gestión documental facilitará la consulta rápida de datos, reducirá los riesgos de pérdida de información y optimizará los procesos administrativos dentro de la asociación. Con esta solución, Tukuy Pacha podrá mejorar su eficiencia operativa y fortalecer su impacto en la comunidad a la que brinda apoyo.

1.3.2. Árbol de Problemas

La Figura 1 muestra el árbol de problemas asociado a la gestión documental en la Asociación Tukuy Pacha. En esta representación gráfica se exponen las principales causas que generan dificultades en el manejo de documentos físicos, como el almacenamiento inadecuado y la falta de mecanismos de respaldo.

Asimismo, se detallan los efectos derivados, entre los que destacan la vulnerabilidad de la información, el retraso en la toma de decisiones y el riesgo de pérdida de documentos. Este análisis visual permite comprender de manera integral la problemática y sirve como base para identificar soluciones eficaces.

Figura 1
Árbol de problemas



Nota, propuesta por el postulante

1.3.3. Formulación del Problema

¿Cómo puede la Asociación Tukuy Pacha mejorar la gestión y seguridad de su documentación mediante la implementación de un sistema informático, optimizando el acceso a la información y fortaleciendo su capacidad de atención a las personas con discapacidad?

Para abordar los desafíos en la gestión documental de la Asociación Tukuy Pacha, es necesario desarrollar un sistema informático que permita digitalizar y organizar eficientemente la información. Este sistema facilitará el acceso rápido y seguro a los datos, reduciendo los tiempos de búsqueda y mejorando la toma de decisiones dentro de la organización. Además, garantizará la confidencialidad

de la información sensible, minimizando los riesgos de extravío, deterioro o acceso no autorizado.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar e implementar un sistema informático que optimice la gestión documental de la Asociación Tukuy Pacha, garantizando la seguridad, accesibilidad y confidencialidad de la información, para mejorar la eficiencia operativa y la calidad del servicio brindado a las personas con discapacidad.

1.4.2. Objetivos Específicos

Diseñar un sistema de gestión documental que permita la digitalización, organización y almacenamiento seguro de los archivos de la Asociación Tukuy Pacha.

Implementar un marco de seguridad robusto que incluya protocolos de autenticación y control de acceso, para proteger la integridad, confidencialidad y disponibilidad de la información sensible de los participantes y del personal.

Desarrollar funcionalidades que permitan una búsqueda y recuperación rápida y eficiente de la información, reduciendo los tiempos de respuesta.

Mejorar la eficiencia del flujo de trabajo documental, optimizando los tiempos y costos que derivan de la gestión manual de los archivos.

1.5. Justificación

1.5.1. Justificación Técnica

El sistema informático garantizará una gestión eficiente de la información, utilizando tecnologías modernas que permitan un almacenamiento seguro, consultas rápidas. Además, se implementarán medidas de seguridad informática para proteger la confidencialidad de los datos.

1.5.2. Justificación Económica

La digitalización reducirá costos operativos a largo plazo, minimizando el uso de papel y el espacio destinado a almacenamiento. También optimizará el tiempo del personal, permitiéndoles enfocarse en tareas estratégicas en lugar de la gestión manual de documentos.

1.5.3. Justificación Social

Este proyecto beneficiará a las personas con discapacidad al mejorar la eficiencia de los servicios de Tukuy Pacha. La digitalización permitirá una mejor planificación y seguimiento de los programas, asegurando un impacto positivo en la calidad de vida de los participantes.

1.6. Alcance

1.6.1. Temático

El alcance temático del proyecto se enfoca en el desarrollo e implementación de un sistema de gestión documental digital para la Asociación Tukuy Pacha. Este sistema abarca procesos clave como:

- Registro y administración de participantes con discapacidad.
- Gestión de documentos personales y técnicos (fichas clínicas, informes, fotografías, actas, etc.).
- Control de acceso basado en roles (administrador, personal técnico, responsables).
- Gestión de usuarios sobre las actividades del sistema.
- Seguridad de la información mediante autenticación y control de sesiones.
- Digitalización de documentos físicos para su almacenamiento, búsqueda y visualización en línea.

Estos procesos contribuyen directamente a mejorar la organización interna de la institución, reduciendo tiempos operativos, garantizando la seguridad de la información y facilitando la toma de decisiones basada en datos.

1.6.2. Geográfico

La implementación del sistema se centrará en las oficinas de la Asociación Tukuy Pacha en las oficinas sur de Cochabamba.

Figura 2

Ubicación de la oficina sur de Tukuy Pacha



1.6.3. Temporal

El desarrollo e implementación del sistema se llevará a cabo en un periodo de 6 a 8 meses, incluyendo desarrollo, y pruebas.

Figura 3

Diagrama de Gantt

	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8
Análisis y Diseño								
Desarrollo Inicial								
Validacion de Prototipo								
Desarrollo Avanzado								
Pruebas y Seguridad								
Capacitación e Implementación								

Nota, propuesta por el postulante

CAPÍTULO II

MARCO TEÓRICO

2.1. Gestión Documental

La gestión documental se define como el conjunto de procesos y prácticas orientadas a la administración eficiente de documentos, con el objetivo de garantizar su disponibilidad, integridad y seguridad a lo largo de su ciclo de vida. Esta disciplina abarca tanto la organización como la digitalización de información, permitiendo mejorar la accesibilidad y reducir el tiempo requerido para la búsqueda y recuperación de datos. Según (Sommerville, 2017) una adecuada gestión documental incrementa la eficiencia organizativa y minimiza los errores asociados al manejo manual de datos.

2.2. Seguridad de Información

En cuanto a la seguridad de la información, esta se enfoca en la protección de los datos contra accesos no autorizados, pérdida o alteraciones. Los protocolos de seguridad, como los establecidos por (Pfleeger & Pfleeger, 2002), incluyen mecanismos de autenticación, autorización y auditoría, que permiten regular y monitorear el acceso a los datos sensibles en los sistemas de gestión documental. Además, garantiza que la información solo sea legible por las partes autorizadas. (Widenius, Axmark, & Arno, 2002) destacan la importancia de algoritmos robustos como AES y RSA en la protección de información crítica, enfatizando su aplicación en entornos empresariales.

El control de acceso es un componente esencial dentro de la seguridad de la información, ya que implica la implementación de políticas que regulan quién tiene permiso para visualizar, modificar o eliminar datos según su rol dentro de la organización. Para garantizar un acceso seguro, se recomienda el uso de autenticación basada en tokens, como los JSON Web Tokens (JWT), los cuales permiten la transmisión segura de información entre partes y contribuyen significativamente a la protección de datos en sistemas de gestión documental. Asimismo, la autenticación multifactor (MFA) agrega una capa adicional de seguridad al requerir múltiples formas de verificación de identidad, reduciendo el riesgo de acceso no autorizado y robo de credenciales. Estas tecnologías son fundamentales en los entornos modernos para asegurar la integridad de los

sistemas y prevenir vulnerabilidades asociadas al acceso indebido (Pfleeger & Pfleeger, 2002).

Los estándares y normativas en torno a la gestión y seguridad de la información establecen un marco de trabajo que asegura el cumplimiento de buenas prácticas en el desarrollo de sistemas de gestión documental. (Sommerville, 2017) menciona que normativas como ISO 27001 y GDPR proporcionan directrices esenciales para garantizar un entorno controlado que cumpla con los requerimientos legales y operativos de cualquier organización.

2.3. Metodología de Desarrollo

Los sistemas de software son abstractos e intangibles. No están limitados por las propiedades de los materiales ni gobernados por leyes físicas o procesos de fabricación, lo que les otorga un alto potencial de desarrollo. Sin embargo, esta flexibilidad también conlleva una complejidad considerable, ya que los sistemas pueden volverse difíciles de entender y modificar. Existen múltiples tipos de software, desde sistemas embebidos hasta plataformas de información a gran escala, cada uno requiriendo enfoques y metodologías específicas. Por ello, el desarrollo de un sistema de gestión documental demanda el uso de prácticas especializadas en ingeniería de software para garantizar su eficiencia y adaptabilidad. (Sommerville, 2017)

2.3.1. Metodología Orientada a Prototipos

Según Pfleeger (2002), el modelo de prototipos facilita la construcción rápida de todo el sistema o de algunas de sus partes, con el objetivo de comprender y aclarar aspectos clave. Esto asegura que el desarrollador, el usuario y el cliente estén alineados en cuanto a las necesidades y la solución propuesta, reduciendo así el riesgo y la incertidumbre durante el desarrollo. Este enfoque se centra en crear diseños que puedan ser analizados y modificados a medida que se incorporan nuevas especificaciones. Es especialmente útil para evaluar el alcance del producto, aunque no garantiza su implementación final.

Este modelo se aplica principalmente cuando el cliente tiene objetivos generales para el software, pero no ha definido detalladamente los requisitos de entrada, procesamiento y salida. Es ideal en situaciones donde no hay certeza sobre la eficacia de un algoritmo, la adaptabilidad del sistema o la interacción entre el usuario y la máquina. El modelo de prototipos ayuda tanto al ingeniero de sistemas como al cliente a visualizar y comprender mejor el resultado final una vez que se hayan satisfecho todos los requisitos. (Pfleeger, 2002)

De acuerdo a (Pfleeger S. L., 2002): El paradigma de construcción de prototipos se estructura en tres etapas clave:

Escuchar al cliente y recopilar requisitos: En esta fase, se identifican y definen los objetivos generales del proyecto, se reconocen los requisitos conocidos y se determinan las áreas que requieren una mayor especificación. Este paso es fundamental para establecer una base clara y comprender las necesidades del cliente.

Construir y revisar el prototipo: Con base en la información recopilada, se desarrolla una maqueta o prototipo que refleja las funcionalidades principales del sistema. Este prototipo es una versión preliminar que permite visualizar y evaluar aspectos clave del software.

Prueba y refinamiento por parte del cliente: El cliente utiliza y evalúa el prototipo, lo que le permite proporcionar retroalimentación y refinar los requisitos del software. Esta iteración ayuda a ajustar el sistema según las expectativas y necesidades reales del usuario.

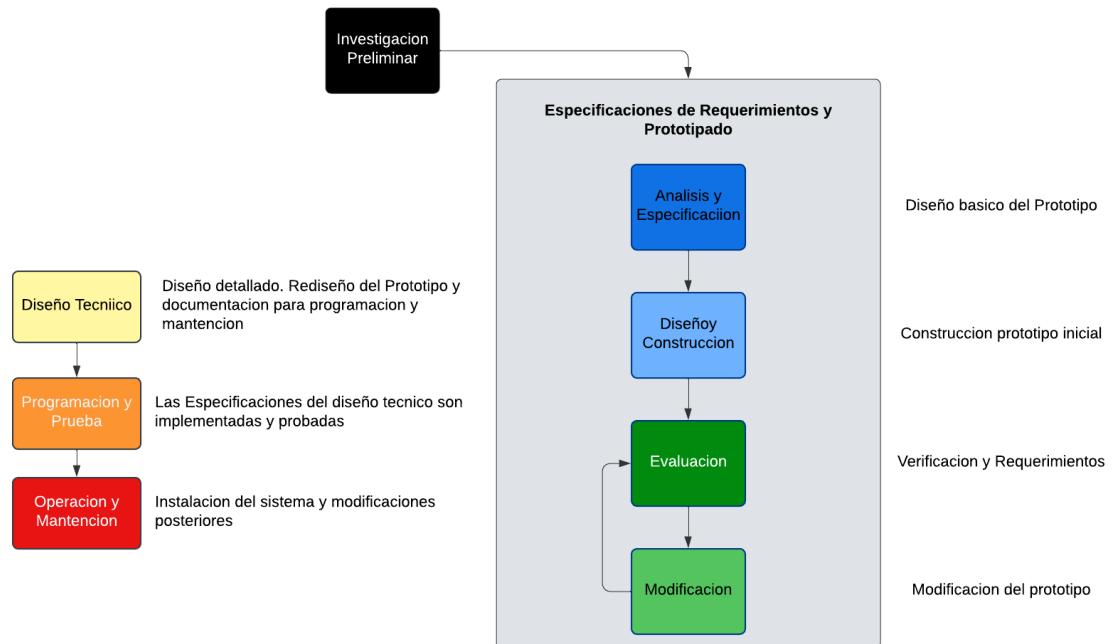
Este modelo es especialmente útil en situaciones donde:

- El cliente no tiene claros los requisitos detallados desde el inicio.
- El equipo de desarrollo tiene incertidumbre sobre la eficiencia de un algoritmo, la compatibilidad con el sistema operativo o la efectividad de la interfaz hombre-máquina.

La figura 4 muestra visualmente las principales etapas del modelo de prototipo

Figura 4

Etapas del Modelo de Prototipo



Nota, etapas del Modelo de Prototipo, (Pfleeger S. L., 2002).

2.3.1.1. Ciclo de Vida de un Sistema basado en Prototipo

Según (Pfleeger S. L., 2002), el paradigma de construcción de prototipos es un enfoque iterativo que facilita la creación rápida de sistemas para abordar requisitos iniciales poco definidos. Este modelo se estructura en etapas clave que aseguran una colaboración efectiva entre desarrolladores y clientes, fomentando la refinación constante de los objetivos del proyecto. El ciclo de vida basado en prototipos permite gestionar la incertidumbre y adaptar el sistema progresivamente a las necesidades reales del usuario, minimizando riesgos durante el desarrollo. A continuación, se detallan las etapas del ciclo de vida de un sistema basado en prototipos, siguiendo el marco conceptual propuesto por Pfleeger.

Identificación de Requisitos Iniciales

En esta fase, el equipo de desarrollo y el cliente colaboran para identificar los objetivos generales del sistema y los requisitos básicos.

Se enfocan en entender las necesidades del usuario, aunque no se definen todos los detalles, ya que estos se irán refinando en iteraciones posteriores.

Diseño Rápido

Con base en los requisitos iniciales, se desarrolla un prototipo rápido y funcional que cubre las características principales del sistema.

Este prototipo no es el producto final, sino una versión simplificada que permite visualizar cómo funcionará el sistema.

Se prioriza la funcionalidad sobre la calidad o el rendimiento, ya que el objetivo es obtener retroalimentación rápidamente.

Evaluación del Prototipo por Parte del Cliente

El cliente prueba el prototipo y lo utiliza para evaluar si cumple con sus expectativas y necesidades.

En esta etapa, el cliente proporciona retroalimentación sobre lo que funciona, lo que no y qué aspectos deben mejorarse o agregarse.

Esta interacción es crucial para refinar los requisitos y asegurar que el sistema se alinee con las expectativas del usuario.

Refinamiento y Mejora del Prototipo

Con la retroalimentación del cliente, el equipo de desarrollo realiza ajustes y mejoras al prototipo.

Se añaden nuevas funcionalidades, se corrigen errores y se optimizan aspectos que no cumplían con las expectativas.

Este proceso es iterativo, lo que significa que el prototipo puede pasar por múltiples revisiones hasta que se considere satisfactorio.

Implementación del Sistema Final

Una vez que el prototipo ha sido refinado y aprobado por el cliente, se procede a desarrollar el sistema completo basado en el prototipo final.

En esta etapa, se optimiza el código, se mejora el rendimiento y se asegura que el sistema cumpla con los estándares de calidad requeridos.

El sistema final debe ser robusto, escalable y listo para su implementación en un entorno real.

Pruebas y Validación

El sistema final se somete a pruebas exhaustivas para garantizar que cumple con todos los requisitos y funciona correctamente.

Se valida que el sistema sea usable, eficiente y libre de errores críticos.

El cliente participa en esta fase para confirmar que el producto final cumple con sus expectativas.

Mantenimiento y Actualización

Una vez implementado, el sistema puede requerir ajustes o actualizaciones para adaptarse a nuevas necesidades o cambios en el entorno.

El mantenimiento puede incluir la corrección de errores, la mejora de funcionalidades o la adición de nuevas características.

2.4. Técnicas de Recolección de Información

Para obtener información relevante en un proceso de análisis y diseño, se aplican diversas técnicas de recolección de datos que permiten comprender necesidades, identificar problemas y definir soluciones adecuadas.

Entrevistas: Consisten en conversaciones estructuradas o semiestructuradas con personas clave para recopilar información detallada sobre un tema específico. Permiten obtener perspectivas directas, aclarar dudas y profundizar en aspectos relevantes a partir de la experiencia de los entrevistados.

Observación Directa: Implica examinar de manera sistemática un entorno, proceso o comportamiento sin intervención directa. A través de esta técnica, se pueden identificar patrones, detectar problemas y validar información obtenida mediante otros métodos.

Revisión Documental: Consiste en el análisis de documentos, normativas, informes o registros existentes con el fin de obtener antecedentes, verificar información y establecer lineamientos basados en marcos teóricos o normativos.

Encuestas: Se basan en la aplicación de cuestionarios a un grupo de personas con el objetivo de recolectar datos cuantitativos o cualitativos sobre sus opiniones, conocimientos o experiencias. Suelen ser útiles para identificar tendencias, evaluar percepciones y obtener información estadística.

2.5. Plataforma de Desarrollo

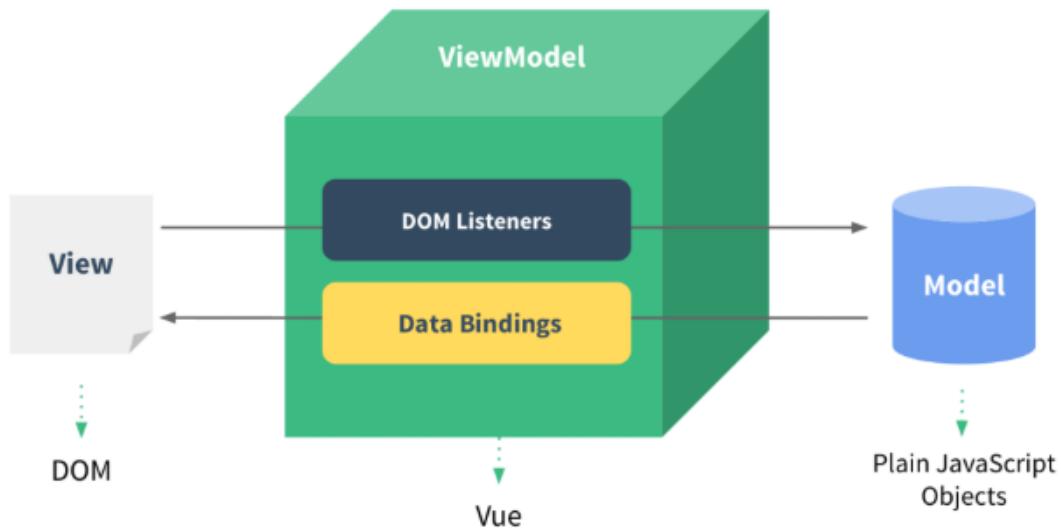
2.5.1. Vue.js

Es un framework progresivo de JavaScript que facilita el desarrollo de interfaces de usuario dinámicas y reactivas. Vue.js permite una gestión eficiente del estado, reutilización de componentes y una integración sencilla con otras herramientas, proporcionando una experiencia de usuario fluida e intuitiva. Su estructura basada en componentes facilita la modularización del código, lo que permite un desarrollo más ordenado y mantenible.

Además, Vue.js ofrece un ecosistema robusto con herramientas como Vue Router para la gestión de rutas y Vuex o Pinia para la gestión del estado global de la aplicación. Estas características permiten crear una interfaz de usuario altamente interactiva y optimizada para un rendimiento eficiente. Gracias a su compatibilidad con librerías y frameworks de diseño como Bootstrap, se garantiza una interfaz moderna y responsive que mejora la accesibilidad y usabilidad del sistema (You, s.f.).

Figura 5

Relación entre View, ViewModel y Model en Vue



Nota, separa la interfaz de usuario (Vista) de la lógica de negocio (Modelo) mediante un ViewModel que gestiona la presentación y la interacción de datos.

Tomado de (Palacios, 2016).

2.5.2. Node.js con Express

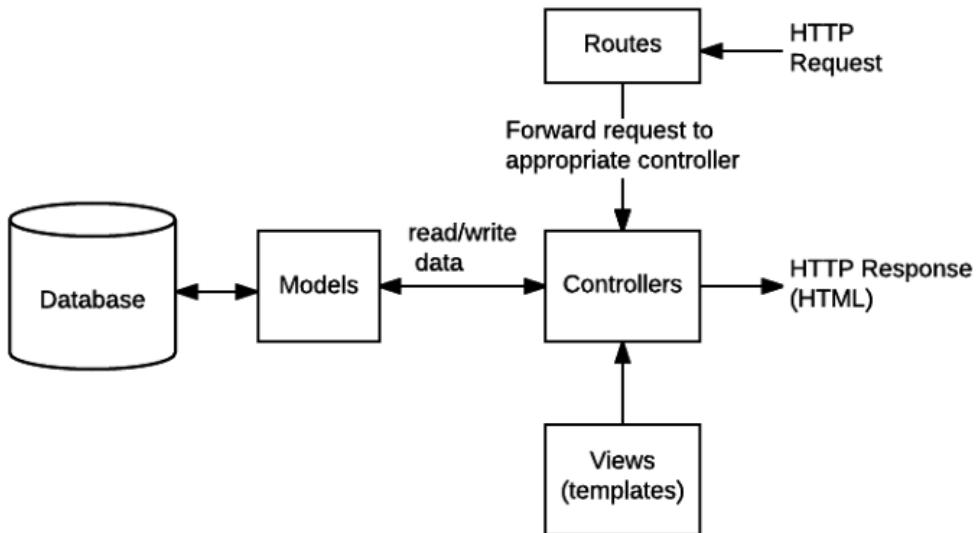
Es un entorno de ejecución basado en JavaScript que permite manejar múltiples solicitudes de manera eficiente. Node.js es una plataforma basada en eventos y asincronía, lo que la hace ideal para aplicaciones web que requieren alta escalabilidad y rendimiento en tiempo real. “Express, siendo un framework minimalista y flexible, permite la creación de APIs RESTful de manera sencilla y eficiente, lo que facilita la comunicación entre el cliente y el servidor” (Holowaychuk, n.d.).

Además, Node.js cuenta con una amplia variedad de paquetes y módulos disponibles a través de NPM (Node Package Manager), lo que permite integrar funcionalidades adicionales sin necesidad de desarrollar todo desde cero. Para garantizar la seguridad en la transmisión de datos, se implementará autenticación con JSON Web Tokens (JWT), y protección contra ataques comunes como inyección SQL y Cross-Site Scripting (XSS). Con estas herramientas, el backend

del sistema ofrecerá un rendimiento estable y seguro para la gestión de la información (Dahl, s.f.).

Figura 6

Arquitectura MVC (*Model-View-Controller*)



Nota, este diagrama ilustra la arquitectura MVC (Modelo-Vista-Controlador), donde las **Rutas** dirigen las solicitudes HTTP a los **Controladores**, los cuales interactúan con los **Modelos** para acceder a la **Base de Datos**, usan las **Vistas** (plantillas) para generar la respuesta HTML y devuelven una **HTTP Response**. Tomado de (Mozilla, s.f.)

2.6. Administrador de Base de Datos

2.6.1. MySQL

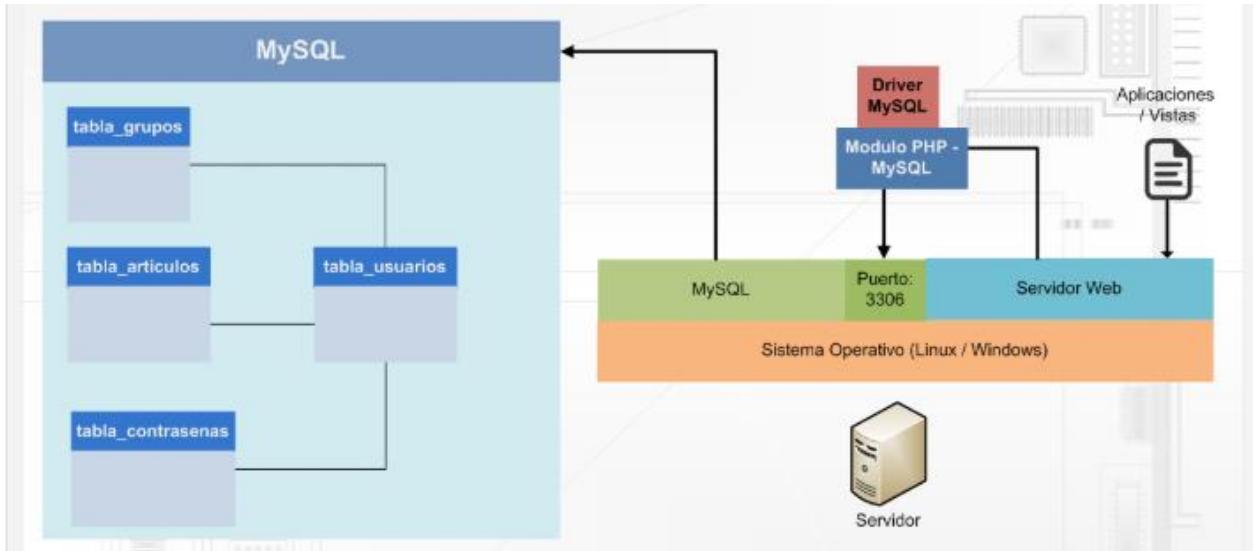
MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que permite almacenar, organizar y gestionar grandes volúmenes de información de manera eficiente. Utiliza el lenguaje SQL (Structured Query Language) para realizar consultas, insertar, actualizar y eliminar datos, asegurando la integridad y seguridad de la información. Su arquitectura permite manejar múltiples usuarios simultáneamente sin comprometer el rendimiento, y es ampliamente utilizado en aplicaciones web debido a su compatibilidad con diferentes lenguajes de programación, su estabilidad y su facilidad de integración con herramientas como Node.js. (Axmark, Larsson, & Widenius, n.d.)

MySQL garantiza la seguridad, integridad y eficiencia en la manipulación de datos a través de diversas características avanzadas. Su sistema de permisos basado en privilegios restringe el acceso no autorizado, asegurando que solo los usuarios con permisos específicos puedan realizar operaciones sobre la base de datos. Además, su compatibilidad con mecanismos de cifrado protege la información tanto en reposo como en tránsito, reforzando la confidencialidad de los datos. En términos de integridad, MySQL emplea el modelo relacional con esquemas estructurados que evitan redundancias y mantienen la consistencia mediante restricciones de clave primaria y foránea. Para optimizar la eficiencia, MySQL implementa índices, almacenamiento en caché y diferentes motores de almacenamiento como InnoDB, que permite transacciones ACID, asegurando un procesamiento confiable de las operaciones. Estas características lo convierten en una solución robusta y confiable para la gestión de datos (Axmark, Larsson, & Widenius, n.d.).

El uso de una base de datos relacional como MySQL en lugar de una no relacional (NoSQL) se debe a varios factores clave. Primero, los sistemas que requieren manejar datos estructurados con relaciones bien definidas entre entidades, como usuarios, permisos y auditorías, se benefician de la integridad referencial que proporciona un modelo relacional. MySQL permite realizar consultas complejas de manera eficiente sin comprometer la consistencia de los datos. Además, su adopción a nivel global y el soporte activo de la comunidad garantizan un mantenimiento confiable y continuo, facilitando la administración y escalabilidad del sistema. Mientras que las bases de datos NoSQL son más adecuadas para datos no estructurados o en constante cambio, MySQL es preferido en entornos donde la coherencia y las relaciones entre datos son esenciales para la integridad del sistema (Axmark, Larsson, & Widenius, n.d.).

Figura 7

Arquitectura de una Aplicación Web con MySQL



Nota, el diagrama ilustra cómo las aplicaciones web interactúan con una base de datos MySQL a través de un servidor web, el módulo PHP-MySQL y un driver MySQL, para gestionar y presentar datos. (Miguel, 2015).

2.7. Herramientas

2.7.1. Visual Studio Code

Es un editor de código ligero pero potente, desarrollado por Microsoft, que se ha convertido en una de las herramientas más populares para el desarrollo de software. Ofrece soporte para una amplia variedad de lenguajes de programación, incluyendo JavaScript, Python, entre otros. Con características como resaltado de sintaxis, autocompletado inteligente, depuración integrada y una extensa biblioteca de extensiones, VS Code facilita la escritura, edición y depuración de código de manera eficiente. Además, su integración con Git y otras herramientas de desarrollo lo convierte en una opción ideal para proyectos colaborativos y de gran escala (Microsoft, n.d.).

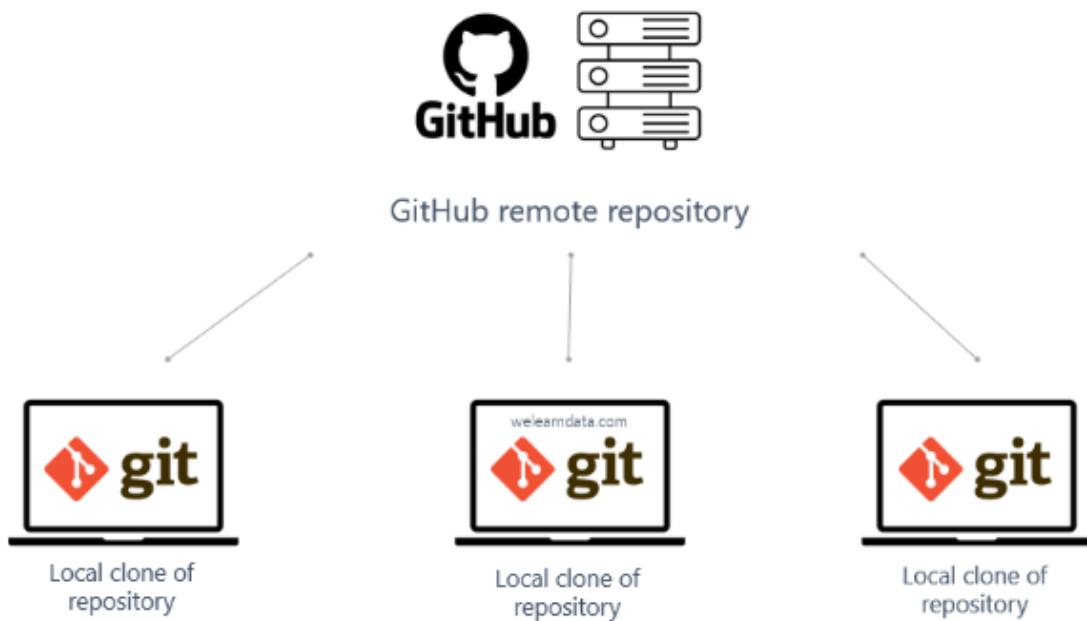
2.7.2. Git y Github

Git es un sistema de control de versiones distribuido que permite a los desarrolladores gestionar cambios en el código fuente de manera eficiente

(Torvalds, n.d.). GitHub, por su parte, es una plataforma basada en Git que facilita la colaboración entre equipos de desarrollo. Juntos, Git y GitHub permiten realizar un seguimiento de las modificaciones, revertir cambios no deseados y trabajar en paralelo en diferentes características del proyecto mediante ramas. Además, GitHub ofrece herramientas adicionales como la gestión de issues, pull requests y integración continua, lo que lo convierte en un entorno ideal para el desarrollo colaborativo y la revisión de código (Preston-Werner, Wanstrath, & Hyett, n.d.).

Figura 8

Control de Versiones con Git y GitHub



Nota, este diagrama ilustra cómo múltiples clones locales de un repositorio Git se sincronizan con un repositorio remoto centralizado en GitHub para la colaboración y gestión de versiones del código. (welearndata.com, s.f.)

2.7.3. Bootstrap

Bootstrap es un framework de diseño front-end que facilita la creación de interfaces de usuario modernas, responsivas y compatibles con diferentes dispositivos. Proporciona una amplia variedad de componentes predefinidos, como botones, formularios, barras de navegación y cuadrículas, que pueden

personalizarse fácilmente. Bootstrap también incluye utilidades de CSS y JavaScript para mejorar la interactividad y el diseño de las aplicaciones web. Su enfoque en la responsividad y la consistencia visual lo convierte en una herramienta esencial para mejorar la experiencia del usuario en aplicaciones web (Otto & Thornton, n.d.).

2.7.4. Laragon

Laragon es un entorno de desarrollo local que simplifica la configuración y gestión de servidores web, bases de datos y otras herramientas necesarias para el desarrollo de aplicaciones. Es especialmente útil para proyectos que utilizan tecnologías como PHP, MySQL, y Apache, ya que ofrece una configuración rápida y sencilla. Laragon incluye características como la creación de hosts virtuales, la gestión de bases de datos y la integración con herramientas como Git y Composer. Su facilidad de uso y portabilidad lo hacen ideal para desarrolladores que necesitan un entorno de desarrollo local eficiente y configurable (Khoa, n.d.).

CAPÍTULO III

ÁREA DE APLICACIÓN

3.1. Antecedentes Generales

Tukuy Pacha es una organización que trabaja en el desarrollo e inclusión de personas en situación de vulnerabilidad. Su enfoque principal está en la rehabilitación basada en la comunidad, promoviendo el acceso a oportunidades y derechos de personas con discapacidad, adultos mayores, niños, niñas y adolescentes. La institución opera bajo un modelo de trabajo colaborativo con la sociedad civil, el Estado y organismos internacionales, con el fin de generar un impacto positivo y sostenible.

La organización mantiene alianzas estratégicas con entidades nacionales e internacionales que contribuyen a fortalecer su accionar. Estas alianzas permiten ampliar la cobertura de sus proyectos y asegurar el financiamiento necesario para su sostenibilidad. Además, se enfoca en la capacitación y formación de su equipo de trabajo, garantizando la calidad y eficiencia de los servicios que brinda a la comunidad.

Actualmente, Tukuy Pacha implementa programas en áreas como salud, educación, inclusión social y desarrollo económico. Su estructura organizativa se basa en la participación activa de distintos actores, incluyendo participantes, personal técnico y autoridades gubernamentales. La institución se mantiene en constante evaluación y adaptación para responder a las necesidades cambiantes de la población a la que atiende.

Sus objetivos institucionales son:

- Impulsar la inclusión social de las personas en desventaja y en riesgo por medio de un trabajo multidisciplinario e inclusivo.
- Contribuir al desarrollo inclusivo con énfasis en poblaciones de atención prioritaria mediante acciones estratégicas de base comunitaria.
- Empoderar y fortalecer las capacidades de gestión del Estado y la sociedad civil para asegurar que las políticas y programas sean inclusivos y accesibles.

- Implementar la estrategia de Rehabilitación Basada en la Comunidad (RBC), enfocándose en componentes como la salud y el empoderamiento.
- Realizar talleres de capacitación para funcionarios y autoridades públicas, así como producir y difundir material didáctico sobre discapacidad.
- Promover la prevención y rehabilitación, apoyar la rehabilitación e independencia en la vida cotidiana de personas con discapacidad a través de seguimiento directo y visitas domiciliarias.

Nota, Objetivos sacados de (Gonzales, 2025)

3.1.1. Mandatos Institucionales: Visión, Misión y Valores

Los mandatos institucionales, como la Visión, Misión, Valores, Objetivos y Estructura Organizacional, son fundamentales para guiar el rumbo de una organización. La Visión representa los sueños y aspiraciones futuras que dan sentido al trabajo diario, como el deseo de mejorar la situación de las personas con discapacidad. Estos elementos ayudan a definir cómo se alcanzarán las metas y qué valores guiarán las acciones, permitiendo a la organización proyectar su camino con claridad.

3.1.1.1. Visión

Comunidades inclusivas y protagonistas de su desarrollo, con corresponsabilidad, equidad de género y acceso a oportunidades para cada persona.

3.1.1.2. Misión

Promover el desarrollo de capacidades, la participación y la plena inclusión de niñez, adolescencia, personas con discapacidad, personas adultas mayores y otras poblaciones de atención prioritaria en Bolivia.

3.1.1.3. Valores

Conceptualización de valor: son aquellas virtudes o cualidades que caracterizan positivamente a una organización. Son guías que ayudan el correcto

comportamiento del personal de la organización. Tukuy Pacha define como sus valores (Tabla 1), los siguientes:

Tabla 1

Valores

TRANSPARENCIA:	Es informar, rendir cuentas y poner a disposición de los ciudadanos toda la documentación de las gestiones, actividades e inversiones económicas realizadas. Basada en la democracia
COMPROBAMIENTO:	Es la obligación contraída por una persona que se compromete o es comprometida a algo. Es un convenio, por ende, un acuerdo que puede considerarse como un contrato no escrito en el cual las partes asumen ciertas obligaciones o, adquieren responsabilidades.
CONFIANZA:	Es la seguridad o esperanza firme que alguien tiene de otro individuo o de algo
INNOVACION:	Creación, modificación cambio de un producto, y su introducción en el trabajo. Aplica nuevas ideas, productos, conceptos, servicios y prácticas a una determinada cuestión, actividad, con la intención de ser eficaces, eficientes y efectivos
SOLIDARIDAD:	Es apoyar, colaborar, ayudar incondicionalmente en causas o intereses ajenos, especialmente en situaciones comprometidas o difíciles, sin recibir nada a cambio.
RESPONSABILIDAD:	Es el cumplimiento de las obligaciones o el cuidado al hacer o decidir algo, también es asumir todas las consecuencias de un acto que fue realizado de manera consciente e intencionada.
ASERTIVIDAD:	Habilidad personal que nos permite expresar sentimientos, opiniones y pensamientos, en el momento oportuno, de la forma adecuada y sin negar ni desconsiderar los derechos de los demás
EMPATIA:	Es la capacidad que tiene una persona de comprender e identificarse con las emociones y los sentimientos de los demás, basada en el reconocimiento del otro.
PROACTIVIDAD:	Capacidad de tomar la iniciativa y la responsabilidad para llevar a cabo acciones que conduzcan a resultados positivos, en lugar de simplemente reaccionar ante situaciones o esperar a que otros tomen la delantera. Asumir responsabilidad y

	adaptarse a nuevas situaciones, donde los fallos son oportunidades de crecimiento evitando posibles problemas que puedan surgir en el futuro.
--	-----------------------------------------------------------------------------------------------------------------------------------------------

Nota, valores sacados de (Gonzales, 2025)

3.1.2. Marco de Referencia Legal Política

Con el propósito de fundamentar el proyecto en un respaldo legal, se mencionan, de manera no exhaustiva, normativas tanto a nivel internacional como nacional que están relacionadas con las poblaciones participantes. Estas normativas son esenciales para garantizar el cumplimiento de los derechos y deberes correspondientes. A continuación, en la Tabla 2, se presentan algunas de las principales disposiciones legales aplicables.

Tabla 2

Normativas Referidas a Las Poblaciones Participantes

POBLACIÓN PARTICIPANTE	NORMATIVAS
Personas con Discapacidad	<ul style="list-style-type: none"> • Convención sobre los derechos de las Personas con Discapacidad • Agenda 2030 para el Desarrollo Sostenible en América Latina y el Caribe. • Constitución Política del Estado Plurinacional • Ley N° 223 – Ley general para personas con discapacidad
Adultos/as mayores	<ul style="list-style-type: none"> • Plan de Acción Internacional de Madrid sobre el envejecimiento. • Convención Interamericana sobre la Protección de los Derechos Humanos de las Personas Mayores (ratificada por Bolivia en 2016). • Constitución Política del Estado: Art. 67, 68, 69, Art. 300 (I.30) y Art. 302 (I.39). • Ley 369 (2013) Ley General de las Personas Adultas Mayores. • D.S. 1807 (27/11/2013), reglamento de la Ley 369 • Plan Multisectorial de desarrollo integral para vivir bien de las personas adultas mayores 2021-25 () • Ley Departamental N.958 (2020) de las personas adultos mayores”, Cochabamba

Infancia temprana	<ul style="list-style-type: none"> • Convención sobre los Derechos del Niño • Recomendaciones del Comité de los Derechos del Niño • Marco Normativo Nacional: <ul style="list-style-type: none"> - Constitución Política del Estado Plurinacional de Bolivia. - Ley N° 548, de 17 de julio de 2014 (Código Niña, Niño y Adolescentes) - Política Pública Plurinacional de la Primera Infancia. • Ley municipal de la primera infancia Ley 097 GAMC
Niñas, niños y adolescentes	<ul style="list-style-type: none"> • Convención Internacional sobre los Derechos de los niños/as. • Constitución Política del Estado Plurinacional de Bolivia. • Código NNA, Ley 548

Nota, normativas realizado de (Gonzales, 2025)

3.2. Definiciones

Tukuy Pacha maneja conceptos clave que guían su labor y definen su enfoque estratégico. La **inclusión social** es uno de los pilares fundamentales de la organización, promoviendo la participación activa de personas con discapacidad y otros grupos vulnerables en la sociedad. La organización trabaja para eliminar barreras físicas y actitudinales que limitan el acceso a oportunidades en distintas áreas como educación, empleo y salud.

Otro concepto clave es el **fortalecimiento organizacional**, el cual permite optimizar los procesos administrativos y operativos de la institución. Tukuy Pacha busca mejorar su estructura interna, desarrollar políticas de gestión eficientes y garantizar la sostenibilidad de sus programas a largo plazo. Este fortalecimiento se logra mediante la capacitación del personal, la implementación de herramientas tecnológicas y la mejora de sus estrategias de planificación.

La **sostenibilidad** es otro eje central en el trabajo de Tukuy Pacha. La organización desarrolla estrategias para diversificar sus fuentes de financiamiento y asegurar la continuidad de sus programas. Además, fomenta la

autosuficiencia de sus participantes a través de iniciativas de capacitación y generación de ingresos, promoviendo su autonomía y participación activa en la comunidad.

3.3. Procesos

Tukuy Pacha es una institución reconocida y valorada por autoridades y organizaciones debido a su contribución social. Esto le ha permitido establecer alianzas estratégicas con instituciones estatales y privadas, las cuales han sido fundamentales para el desarrollo de sus actividades y el logro de sus objetivos. Estos aliados son una fuente clave de apoyo, ya que sin ellos no habría sido posible alcanzar los resultados obtenidos en beneficio de los participantes.

En el proceso de elaboración de su Plan Estratégico Institucional (PEI), Tukuy Pacha ha identificado a sus aliados estratégicos, recomendándose mantener actualizada la matriz de actores y recurrir a ellos cuando sea necesario.

Las poblaciones beneficiarias de Tukuy Pacha incluyen:

- Personas con discapacidad, sus familiares, tutores, organizaciones relacionadas e instituciones públicas del sector.
- Niños y niñas de 0 a 5 años con retraso en el desarrollo, sus familiares, facilitadores de centros infantiles e instituciones públicas vinculadas. También se atiende a niños, adolescentes y jóvenes en condición de desventaja social y económica en zonas periurbanas y rurales.
- Personas adultas mayores en desventaja social y económica, sus familiares, organizaciones e instituciones públicas relacionadas.

El equipo de trabajo ha identificado acciones programáticas agrupadas en ámbitos de intervención (Tabla 3) basados en la matriz de Rehabilitación Basada en la Comunidad (RBC): salud, educación, sustento, inclusión social, empoderamiento, soporte a la gestión pública, datos abiertos y enfoque de género. Estas acciones buscan abordar las necesidades de las poblaciones beneficiarias de manera integral.

Tabla 3*Componentes de la Matriz RBC*

ÁMBITO	ACCIONES PROGRAMÁTICAS
SALUD	<ul style="list-style-type: none"> - Profundizar desarrollo de intervenciones orientadas a la terapia ocupacional y a las actividades de la vida diaria - Fisioterapia, estimulación psicomotora, impulso al desarrollo de habilidades - Provisión y adaptación de dispositivos de ayudas técnicas, incluyendo tecnologías digitales - Telerehabilitación - Capacitación de personal de salud en diferentes temas, incluyendo: <ul style="list-style-type: none"> • detección temprana de retraso y rezago en el desarrollo y de la discapacidad • orientación a madres y padres en la estimulación oportuna de niñas y niños - Impulso y fortalecimiento de mecanismos de participación social en salud (ley SAFCI) - Soporte a la implementación de políticas públicas de desarrollo infantil temprano - Transversalización de la discapacidad en los programas públicos de salud existentes (DIT, Juana Azurduy, otros) - Iniciativas relacionadas con la promoción de la salud mental
EDUCACIÓN	<ul style="list-style-type: none"> - Promover la crianza positiva, la comunicación asertiva y sin violencia en la familia, la generación de espacios seguros y enriquecedores para niñas y niños. - Fortalecer las capacidades de madres, padres, cuidadoras y cuidadores para promover el desarrollo integral de las niñas y niños. Incluye la elaboración de materiales y juegos de estimulación. - Promover el acceso y organizar cursos de tecnicificación en diferentes rubros - Soporte a procesos de inclusión educativa de niñas y niños con discapacidad y rezago en el desarrollo - Formación docente - Concientización de autoridades y comunidades educativas (dirección y plantel administrativo, docentes, estudiantes, madres y padres de familia) - Promoción de publicaciones y espacios de debate sobre la educación inclusiva

	<ul style="list-style-type: none"> - - Promoción de la accesibilidad de los espacios educativos
SUSTENTO	<ul style="list-style-type: none"> - Impulso de microemprendimientos personales, familiares y grupales - Fortalecer las habilidades blandas y de emprendimiento - Orientación vocacional de adolescentes y jóvenes - Desarrollo de proyectos de vida de adolescentes y jóvenes
INCLUSIÓN SOCIAL	<ul style="list-style-type: none"> - Sensibilización a madres, padres, cuidadores y cuidadoras sobre crianza sin violencia - Concientización sobre los derechos de las personas adultas mayores
EMPODERAMIENTO	<ul style="list-style-type: none"> - Promoción de grupos de auto y mutua ayuda - Fortalecimiento organizacional y de la agencia ciudadana de las OPCD - Desarrollo de proyectos conjuntos con OPCD - Articulación con OPCD de ámbito nacional
SOPORTE A LA GESTIÓN PÚBLICA Y DATOS ABIERTOS	<ul style="list-style-type: none"> - Elaboración y difusión de estudios e investigaciones temáticas - Desarrollo de portales de datos - Incidencia para la publicación de información pública y datos abiertos - Transferencia a instancias públicas de las metodologías implementadas - Desarrollo de soluciones informáticas para la gestión de datos poblacionales - Transversalización de la estrategia de la Rehabilitación Basada en la Comunidad - Desarrollo Inclusivo
ENFOQUE DE GENERO	<ul style="list-style-type: none"> - Transversalizar el enfoque de género en los programas y proyectos desarrollados - Concientizar sobre la corresponsabilidad de género en el trabajo de cuidados - Fortalecimiento del empoderamiento - Micro emprendimientos

Nota, Matriz RBC sacado de (Gonzales, 2025)

3.4. Documentación e Información

Tukuy Pacha gestiona su información de manera estructurada y transparente, asegurando que los datos recopilados sean utilizados para mejorar la toma de decisiones. La organización mantiene registros detallados sobre:

- Participantes: Incluyen datos demográficos, características específicas y necesidades identificadas.
- Proyectos: Información sobre objetivos, actividades, avances, recursos asignados y resultados obtenidos.
- Actividades: Detalles operativos y de ejecución para garantizar un monitoreo constante del impacto de los programas.

Este enfoque permite mantener la consistencia y calidad en la rendición de cuentas, dirigiéndose a financiadores, aliados estratégicos y la comunidad, para fortalecer la confianza y la colaboración.

La planificación, monitoreo y evaluación son realizados por el equipo técnico y administrativo de Tukuy Pacha en colaboración con los coordinadores y responsables de cada proyecto. El proceso se desarrolla de la siguiente manera (Tabla 4):

Tabla 4

Etapas del Sistema de Planificación, Monitoreo y Evaluación Institucional

Etapa del Proceso	Descripción
Planificación	Definición de objetivos estratégicos, actividades y cronogramas, junto con la asignación de los recursos necesarios para cumplirlos.
Monitoreo	Evaluación continua del progreso de las actividades utilizando indicadores establecidos, incluyendo revisiones periódicas y seguimiento.
Evaluación	Ánálisis profundo al final del ciclo para identificar aprendizajes, ajustar estrategias y asegurar la efectividad de las intervenciones.

Nota, sacado de (Gonzales, 2025)

Además, Tukuy Pacha promueve la accesibilidad de la información a todos los actores involucrados en su labor. La organización elabora materiales educativos, guías y manuales dirigidos a participantes, autoridades y público en general. También utiliza plataformas digitales y redes sociales para difundir información relevante y sensibilizar sobre la importancia de la inclusión social y el respeto a los derechos humanos.

CAPÍTULO IV

DESARROLLO

DEL PROYECTO

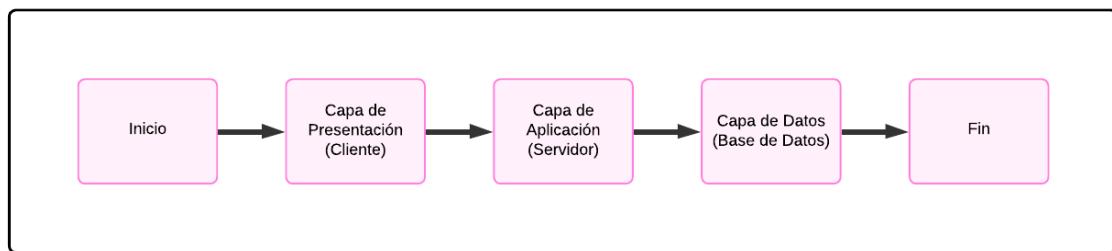
4.1. Análisis del Sistema

4.1.1. Arquitectura de software

La arquitectura de software general del sistema se organiza en **tres capas principales: presentación, aplicación y datos**, siguiendo el modelo **cliente-servidor**. Esta estructura facilita la separación de responsabilidades, la escalabilidad y el mantenimiento del sistema. El flujo inicia con la interacción del usuario en la interfaz gráfica (cliente), continúa en el servidor que procesa la lógica del negocio (aplicación), y culmina con la consulta o almacenamiento de información en la base de datos.

Figura 9

Arquitectura de Software General



Nota, propuesta por el postulante

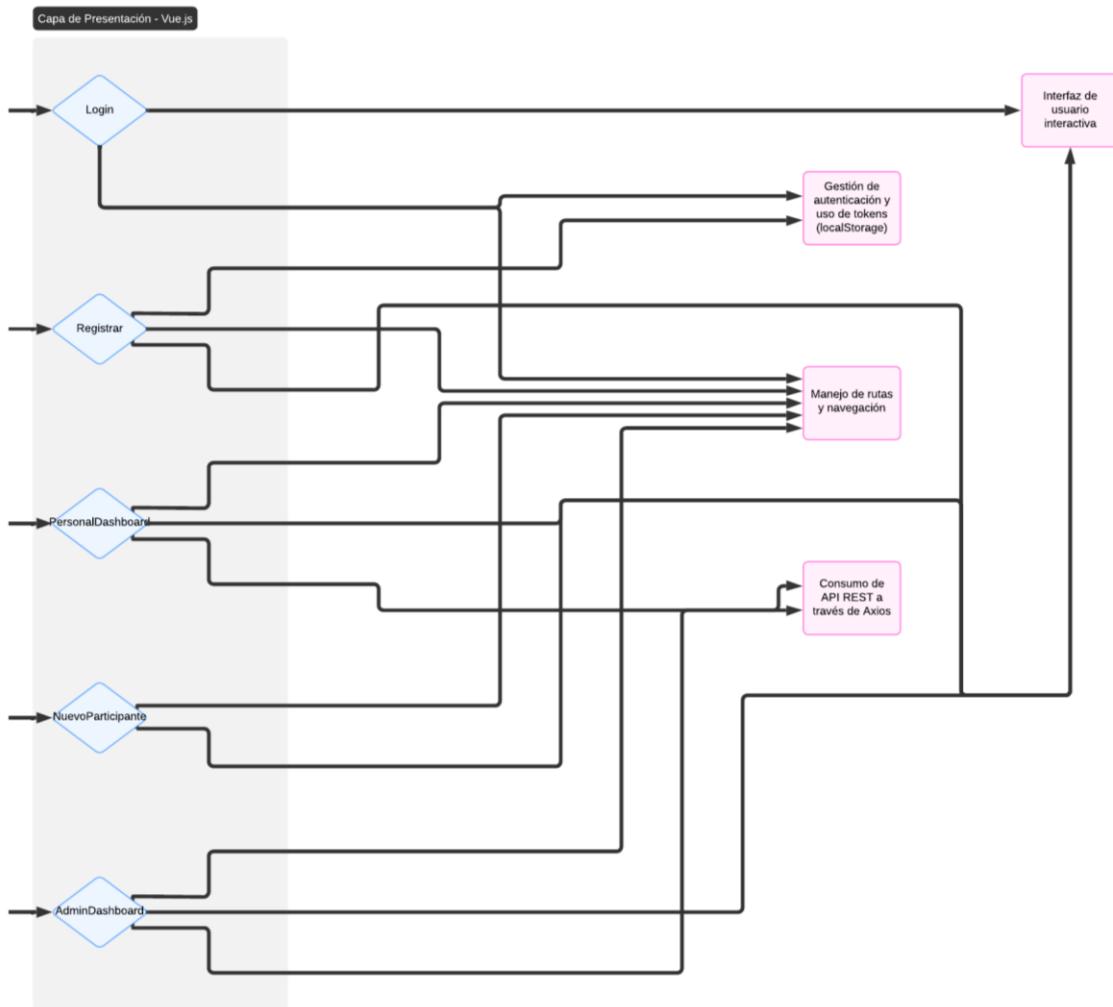
4.1.1.1. Capa de Presentación-Vue.js

Representa la estructura del cliente web. Incluye componentes como Login, Registrar, PersonalDashboard, AdminDashboard, que interactúan con:

- Autenticación y uso de tokens
- Rutas de navegación internas
- Consumo de APIs usando Axios

Todo esto se combina para ofrecer una **interfaz de usuario interactiva**.

Figura 10
Capa de Presentación



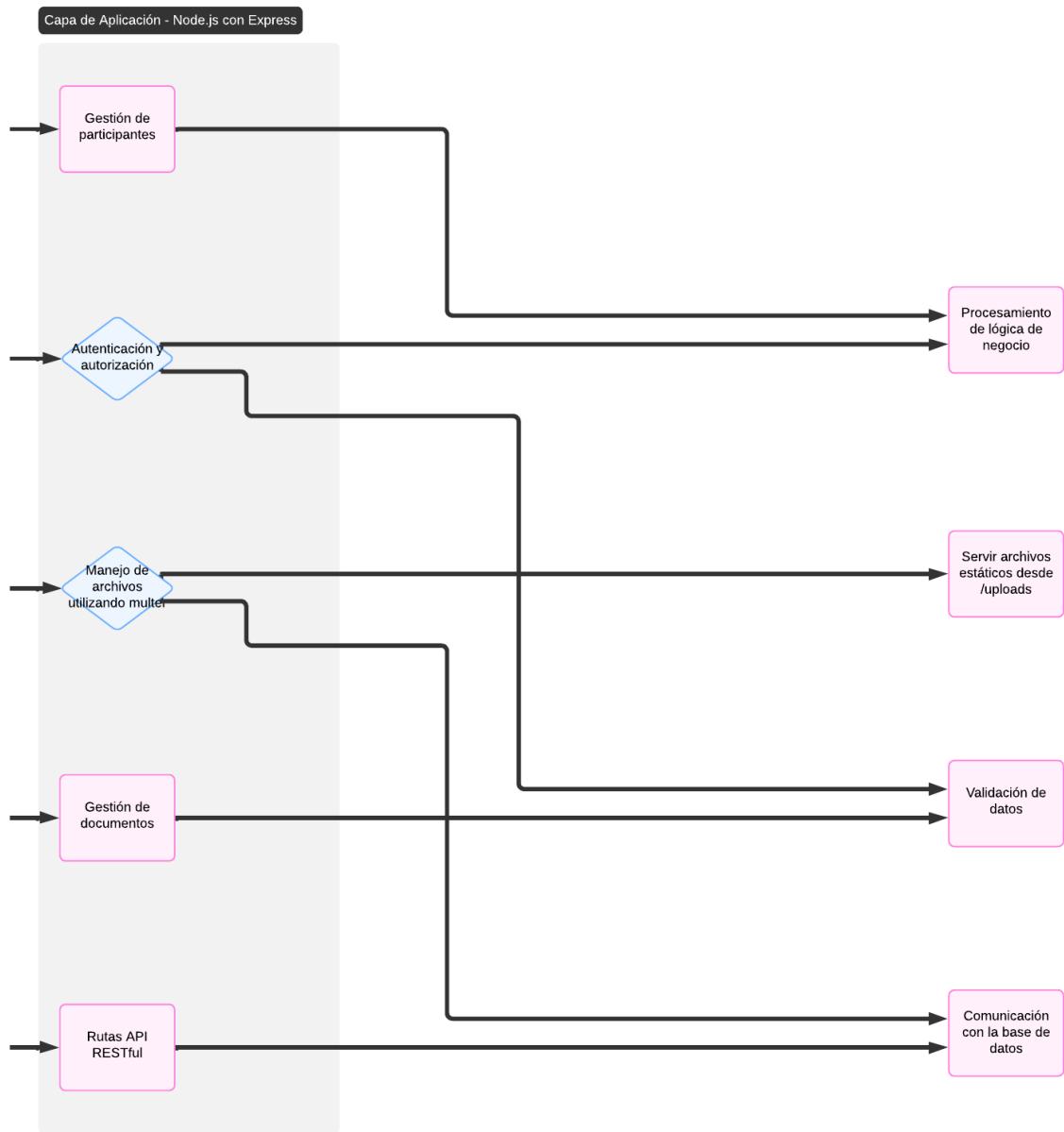
4.1.1.2. Capa de Aplicación – Node.js con Express

Modelo del servidor que expone rutas API y gestiona:

- Autenticación/autorización
- Manejo de archivos (multer)
- Validación de datos
- Lógica de negocio

Actúa como puente entre el cliente y la base de datos, asegurando el flujo correcto de la información.

Figura 11
Capa de Aplicación



4.1.1.3. Capa de datos - MySql

Esquema de base de datos con tablas como:

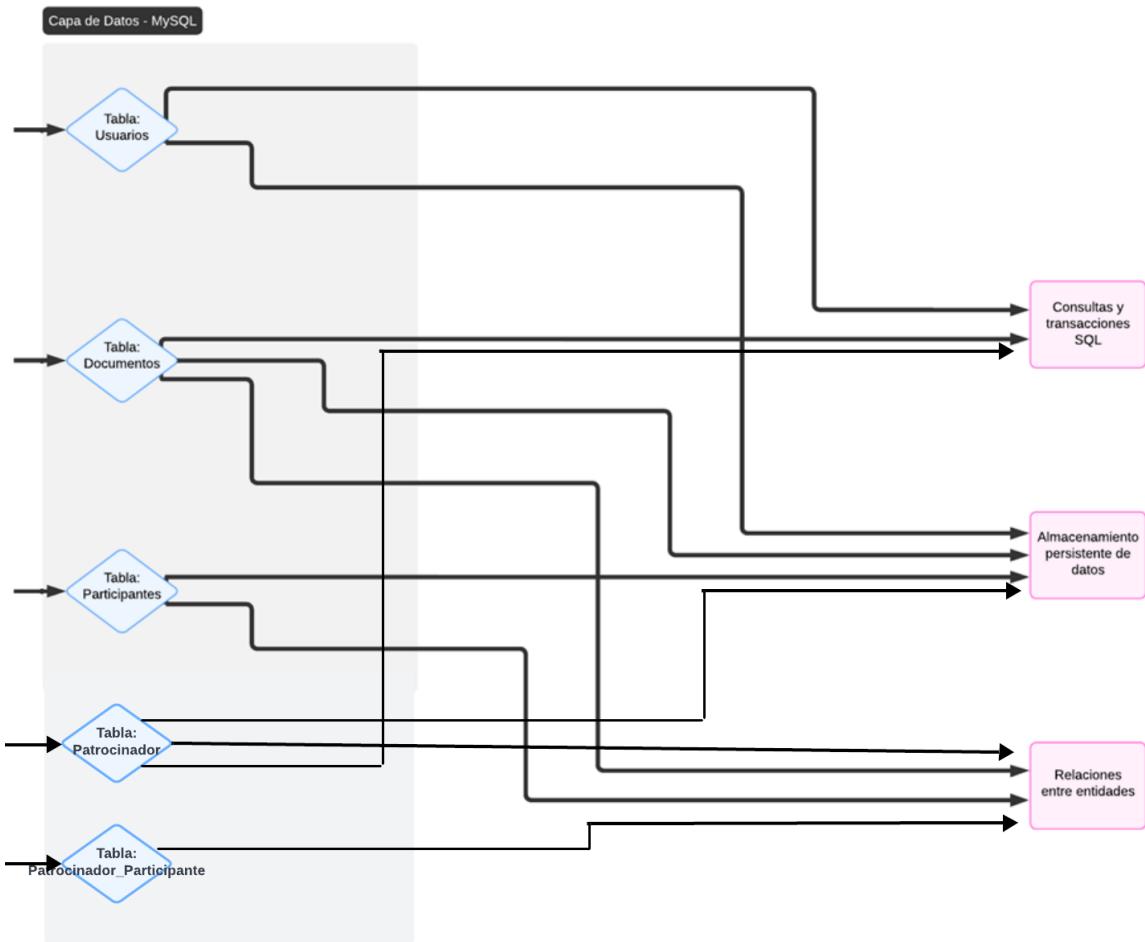
- Usuarios
- Documentos
- Participantes
- Patrocinador

- Patrocinador_Participante

Se muestra cómo estas tablas interactúan mediante relaciones, consultas SQL y almacenamiento persistente de datos para garantizar integridad y disponibilidad.

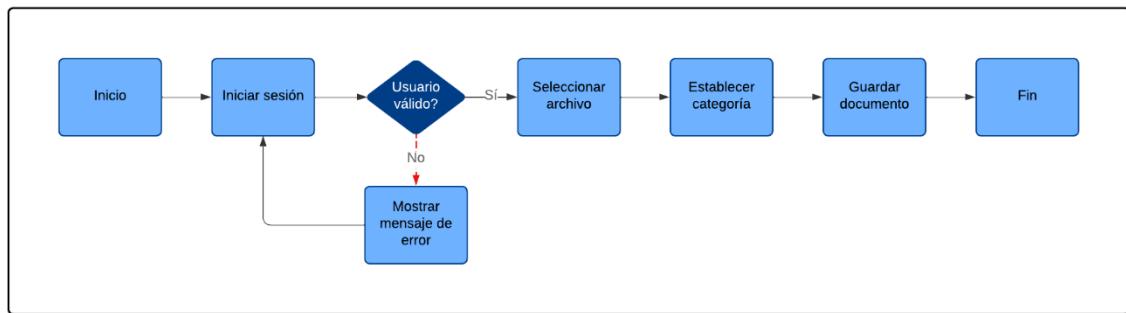
Figura 12

Capa de Datos



4.1.2. Diagrama de Actividades

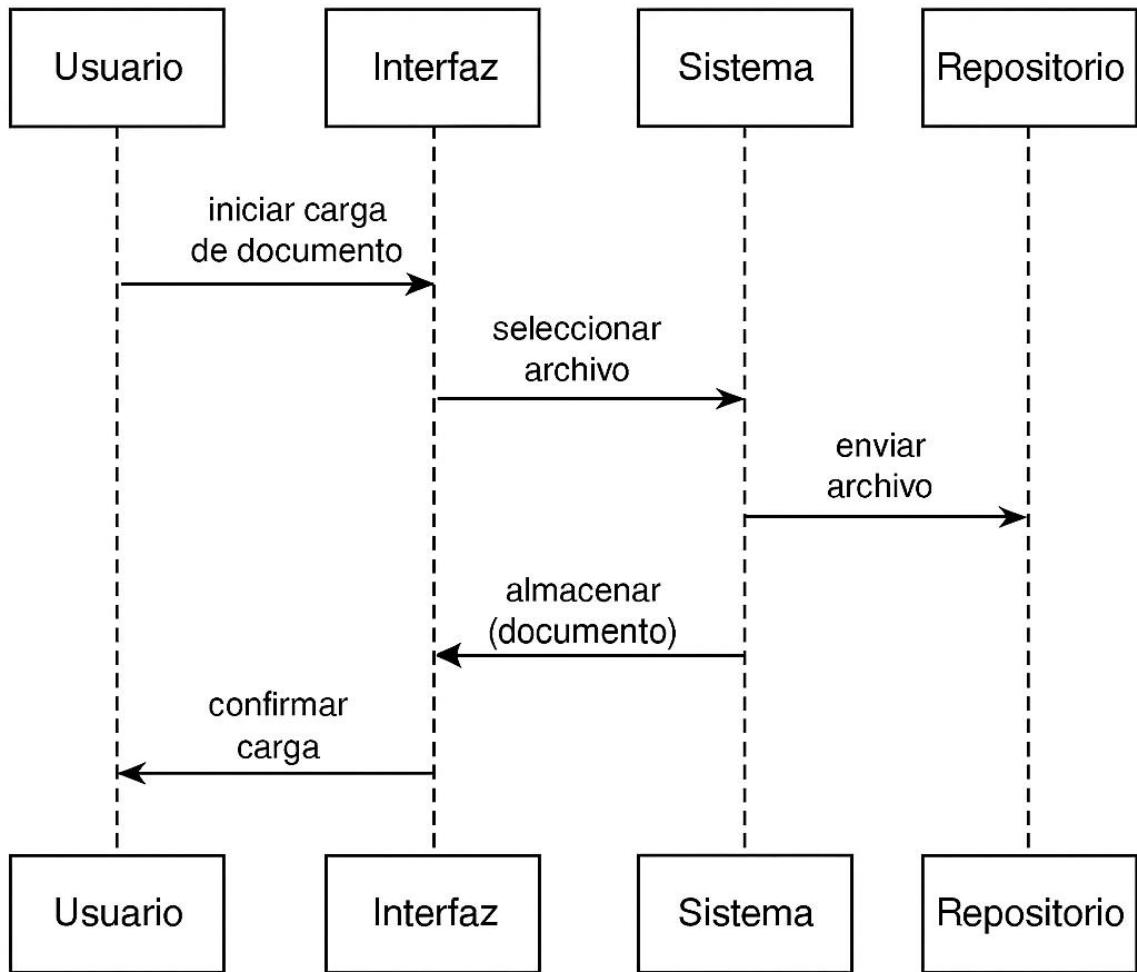
Este diagrama representa el flujo de procesos relacionados con la carga de documentos. Se identifican los pasos desde el inicio de sesión, validación del usuario, selección del archivo, categorización, almacenamiento, hasta la confirmación final. Ayuda a visualizar la lógica operativa y a detectar posibles cuellos de botella.

Figura 13*Diagrama de Actividades*

4.1.3. Diagrama de Secuencia

El diagrama de secuencia muestra la interacción temporal entre el usuario, la interfaz web, el servidor y la base de datos durante una acción como “consultar documentos”. Este permite identificar el orden y sincronización de los mensajes entre objetos.

Figura 14
Diagrama de Secuencia



4.2. Análisis de Requerimientos

Se realizó un proceso de levantamiento de requerimientos utilizando entrevistas, observación directa y revisión documental.

4.2.1. Encuesta

Encuesta realizada para recopilar información clave para diseñar un sistema informático eficiente que permita gestionar los archivos de los participantes de la Asociación Tukuy Pacha.

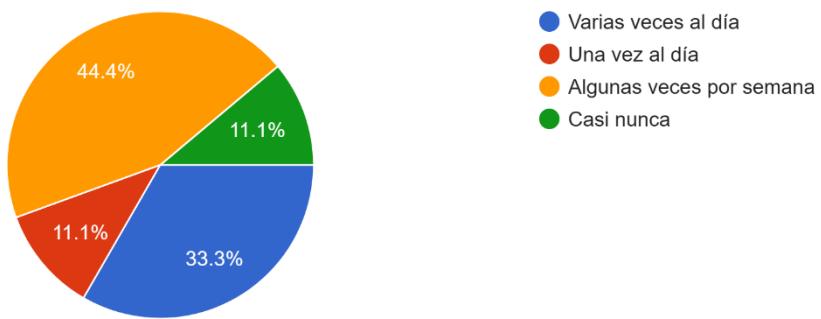
En la figura 15 presenta datos relacionados con la frecuencia de acceso a los archivos de los participantes. Muestra visualmente la frecuencia con la que se acceden a los archivos de los participantes dentro del sistema.

Figura 15

Frecuencia de Acceso a Archivos de Participantes

¿Con qué frecuencia necesitas acceder a los archivos de los participantes?

9 respuestas



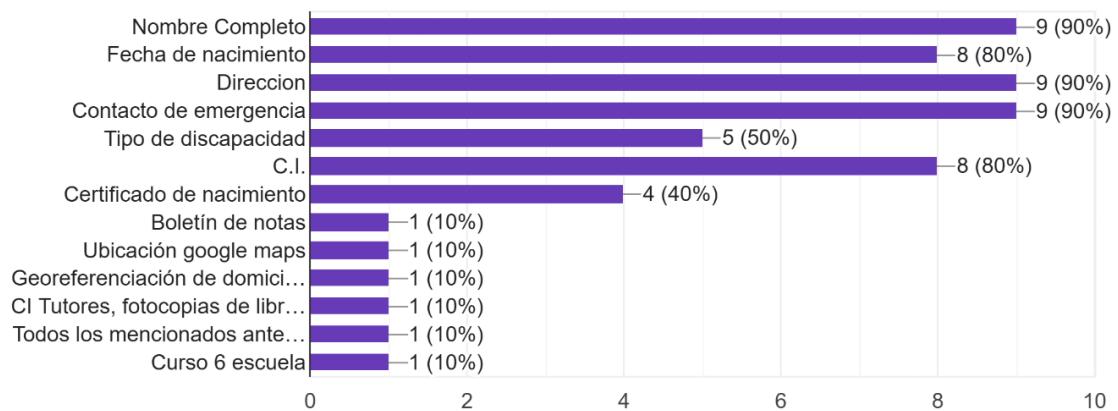
En la figura 16 muestra la información personal que se requiere almacenar de los participantes dentro del sistema. Describe los campos de datos específicos que el sistema recopila y retiene.

Figura 16

Información Personal Necesaria para Almacenar sobre los Participantes

¿Qué información personal debe almacenarse sobre los participantes?

10 respuestas



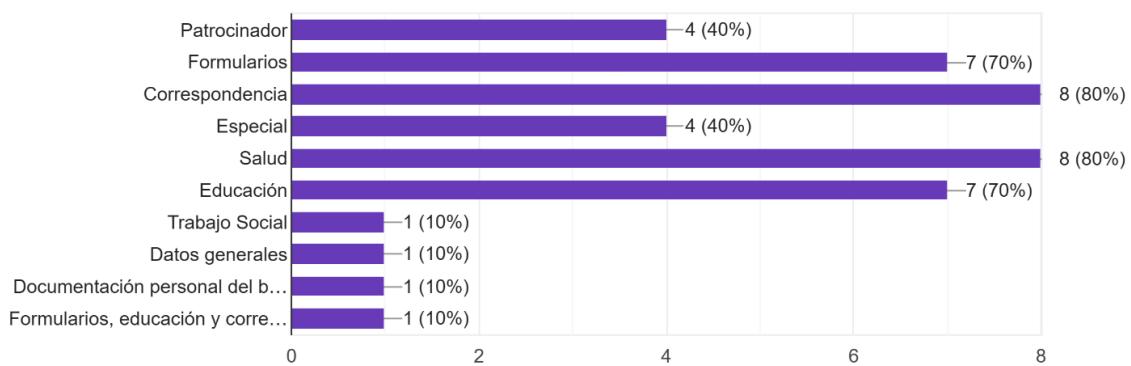
La figura 17 ilustra la importancia de las diferentes secciones dentro del archivo de cada participante. Representa visualmente la importancia relativa o la prioridad de las diversas categorías de datos.

Figura 17

Importancia de las Secciones en el Archivo de Cada Participante

Cada participante tiene su propio archivo con diferentes secciones. ¿Cuáles consideras más importantes?

10 respuestas



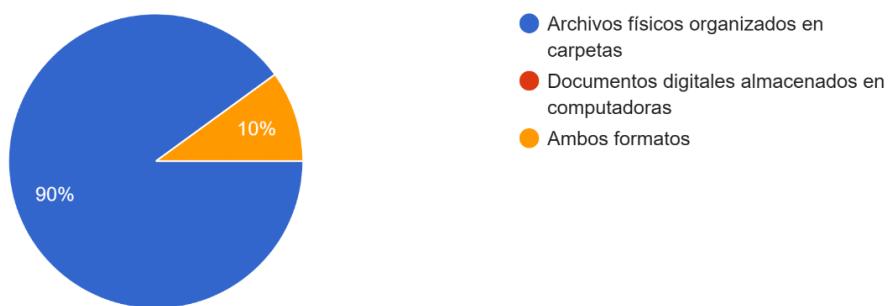
En la figura 18 representa el proceso actual para la gestión de los archivos de los participantes. Muestra el flujo de trabajo o los pasos involucrados en el sistema actual de gestión de archivos.

Figura 18

Gestión Actual de los Archivos de los Participantes

¿Cómo se gestionan actualmente los archivos de los participantes?

10 respuestas



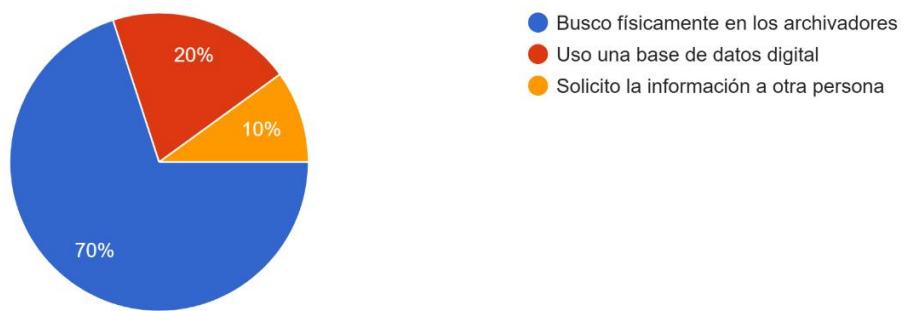
En la figura 19 muestra los métodos utilizados actualmente para acceder a los archivos de los participantes. Muestra las formas en que los usuarios actualmente recuperan o localizan la información de los participantes.

Figura 19

Métodos Actuales de Acceso a los Archivos

¿Cómo accedes a estos archivos actualmente?

10 respuestas



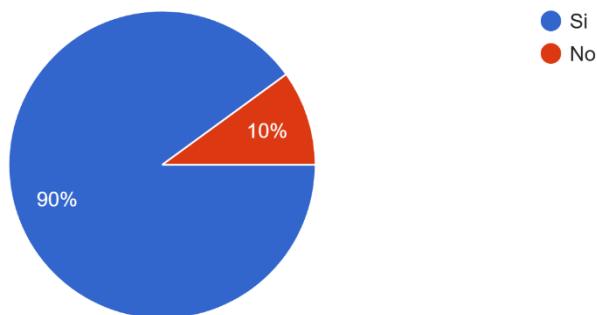
Esta figura 20 ilustra la asociación o relación entre un patrocinador y múltiples participantes. Representa visualmente cómo los patrocinadores están vinculados a los participantes que apoyan.

Figura 20

Asociación de un Patrocinador a Múltiples Participantes

¿Un patrocinador puede estar asociado a más de un participante?

10 respuestas



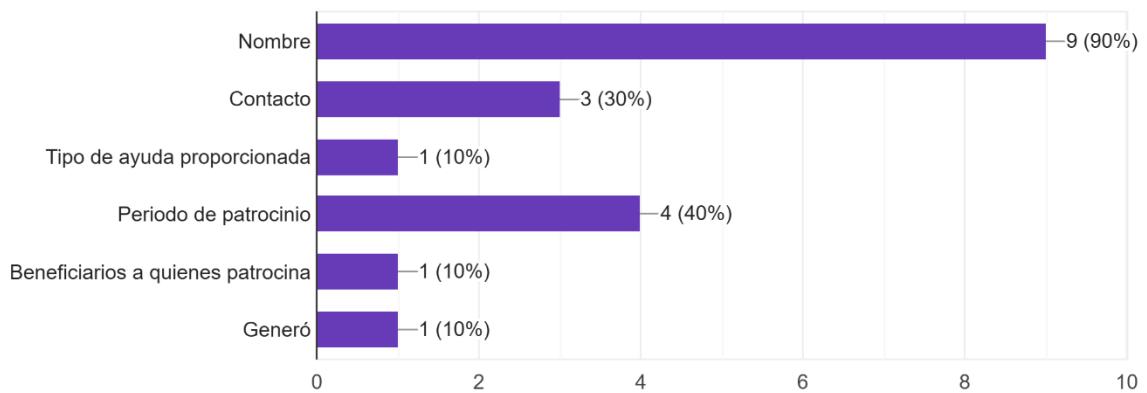
Esta figura 21 muestra la información sobre un patrocinador que se almacena dentro del sistema. Describe los campos de datos recopilados y retenidos para los registros de los patrocinadores.

Figura 21

Información del Patrocinador que Debería Almacenarse

¿Qué información del patrocinador debería almacenarse?

10 respuestas



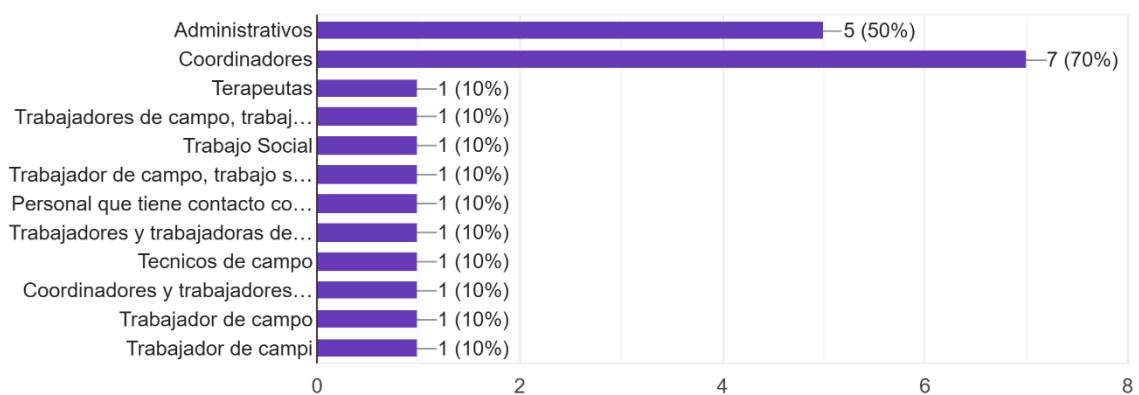
Esta figura 22 muestra quién tiene acceso a la información dentro del sistema. Representa los niveles de acceso de usuario o los permisos del sistema.

Figura 22

Quiénes Deberían Tener Acceso a la Información del Sistema

¿Quiénes deberían tener acceso a la información del sistema?

10 respuestas

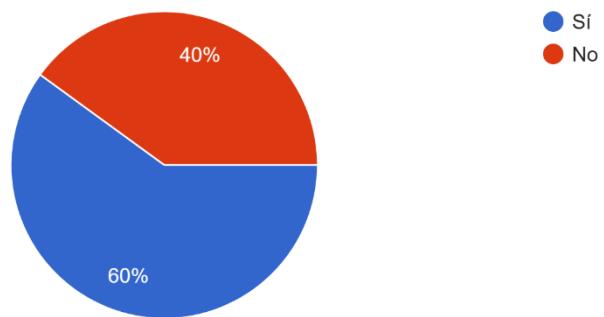


Esta figura 23 ilustra la importancia de tener diferentes niveles de acceso dentro del sistema. Demuestra visualmente la justificación o la necesidad del control de acceso.

Figura 23

Importancia de Diferentes Niveles de Acceso al Sistema

¿Consideras importante que el sistema tenga diferentes niveles de acceso?
10 respuestas

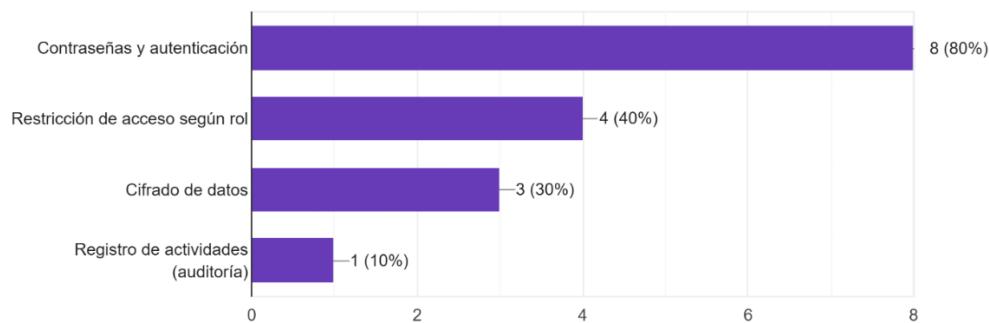


Esta figura 24 muestra las medidas de seguridad necesarias para proteger la información dentro del sistema. Describe las salvaguardias implementadas o propuestas para garantizar la seguridad de los datos.

Figura 24

Medidas de Seguridad Necesarias para Proteger la Información

¿Qué medidas de seguridad consideras necesarias para proteger la información?
10 respuestas



Esta figura 25 resume los aspectos importantes para mejorar la gestión documental dentro de la Asociación Tukuy Pacha. Representa visualmente las recomendaciones clave o las áreas de mejora.

Figura 25

Aspectos Importantes para Mejorar la Gestión Documental en la Asociación Tukuy Pacha

¿Qué otros aspectos consideras importantes para mejorar la gestión documental en la Asociación Tukuy Pacha?

8 respuestas

- Que se puedan insertar los datos desde el celular
- Digitalización de la información
- Sistema intuitivo que re dirija al beneficiario y un resumen de su historial
- Informes sociales
- Uniformidad de la base de datos de todos los sectores de patrocinio de Tukuy Pacha
- Fotografía del patrocinado.
- Que se pueda ingresar desde celular
- Colores por sector

Los requerimientos se clasificaron en:

4.2.2. Requerimientos funcionales

Tabla 5

Requerimientos Funcionales

ROL	DESCRIPCIÓN	FUNCIÓN	PRIORIDAD	CRITERIO DE ACEPTACIÓN
RF-01	Registro, consulta, modificación y eliminación de participantes	Gestionar toda la información de los participantes	Alta	El sistema permite registrar, consultar, actualizar y eliminar participantes correctamente.
RF-02	Carga y gestión de documentos digitales	Subir, organizar, actualizar y eliminar documentos digitales	Alta	Se pueden cargar documentos y asociarlos correctamente a los participantes.
RF-03	Búsqueda avanzada de información	Realizar búsquedas por filtros y criterios específicos	Alta	El sistema devuelve resultados correctos aplicando filtros de búsqueda.
RF-04	Control de acceso según perfiles de usuario	Definir y restringir funcionalidades según el rol asignado	Alta	Solo los usuarios autorizados acceden a módulos según su perfil

4.2.3. Requerimientos no funcionales

Tabla 6

Requerimientos No Funcionales

ROL	DESCRIPCIÓN	FUNCIÓN	PRIORIDAD	CRITERIO DE ACEPTACIÓN
RNF-01	Seguridad: autenticación mediante JWT	Garantizar la protección de la información	Alta	Se utiliza JWT en cada sesión.
RNF-02	Usabilidad: interfaz intuitiva y accesible	Facilitar el uso por parte de todos los usuarios	Alta	Los usuarios completan tareas comunes sin asistencia adicional.
RNF-03	Rendimiento: tiempo de respuesta rápido y optimización de consultas	Minimizar los tiempos de respuesta en las operaciones	Alta	Las consultas y cargas responden en menos de 3 segundos.
RNF-04	Escalabilidad: capacidad de adaptación a nuevas funcionalidades	Permitir ampliaciones y mejoras sin reestructurar el sistema	Media	Se pueden agregar módulos o funciones sin afectar el funcionamiento.
RNF-05	Cumplimiento legal: alineamiento con normativas de protección de datos (Ley N° 223, GDPR, etc.)	Asegurar el cumplimiento de normativas legales vigentes	Alta	El sistema implementa políticas de privacidad y consentimiento informado.

4.2.4. Casos de Uso UML

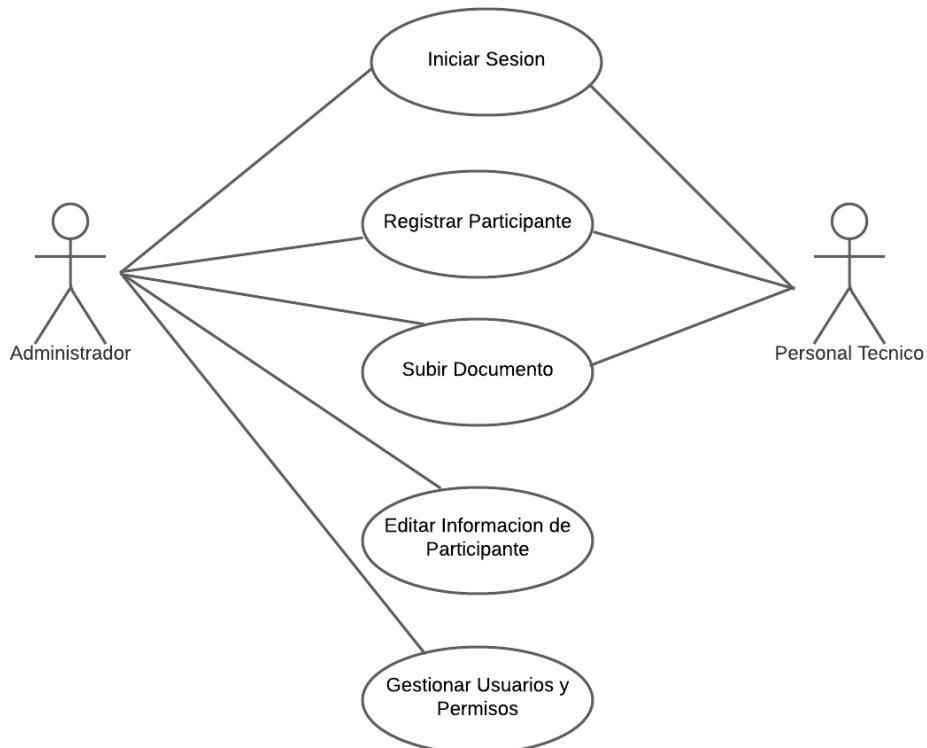
El diagrama de casos de uso ilustra las interacciones entre los distintos tipos de usuarios del sistema (administrador, terapeuta y responsable administrativo) y las funcionalidades clave que proporciona el sistema. Estas funcionalidades incluyen el registro de participantes, la gestión de documentos, la generación de reportes y la administración de usuarios. Este diagrama permite comprender de forma global cómo se relacionan los actores con el sistema, facilitando la identificación de requerimientos funcionales.

4.2.4.1. Diagrama de Caso de Uso General

Este diagrama figura 26 presenta una visión global del sistema, mostrando los diferentes actores involucrados (como el administrador, personal técnico y participantes) y los principales casos de uso que cada uno puede ejecutar. Proporciona una perspectiva integral de las funcionalidades clave ofrecidas por la plataforma.

Figura 26

Diagrama de Caso de Uso General

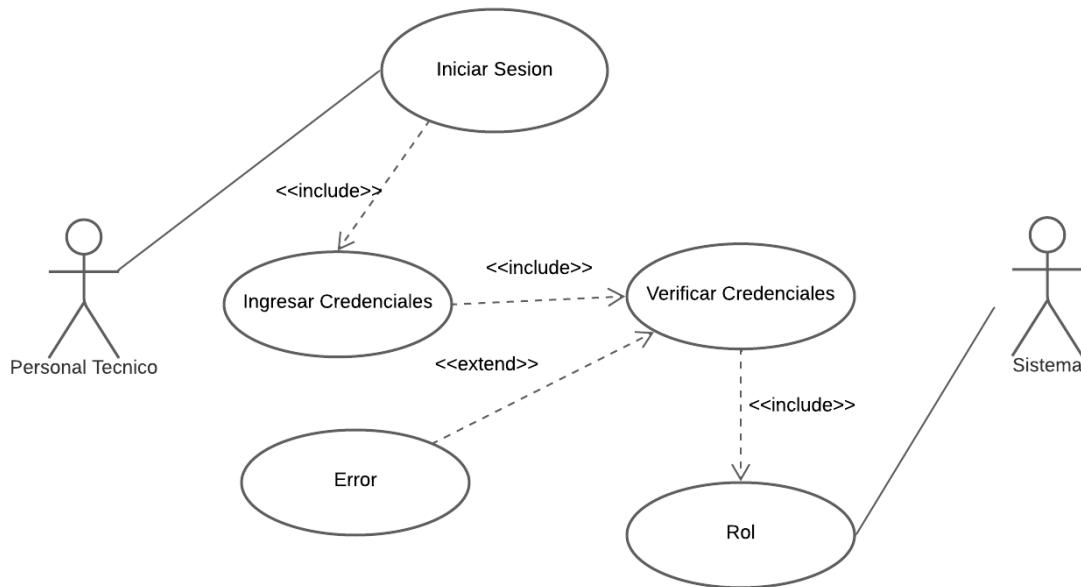


4.2.4.2. Diagrama de Caso de Uso Ingreso al Sistema

Representa el flujo del caso de uso relacionado con el inicio de sesión. Incluye al actor “usuario” y muestra cómo este interactúa con el sistema para autenticarse, detallando pasos como la introducción de credenciales y la validación de acceso.

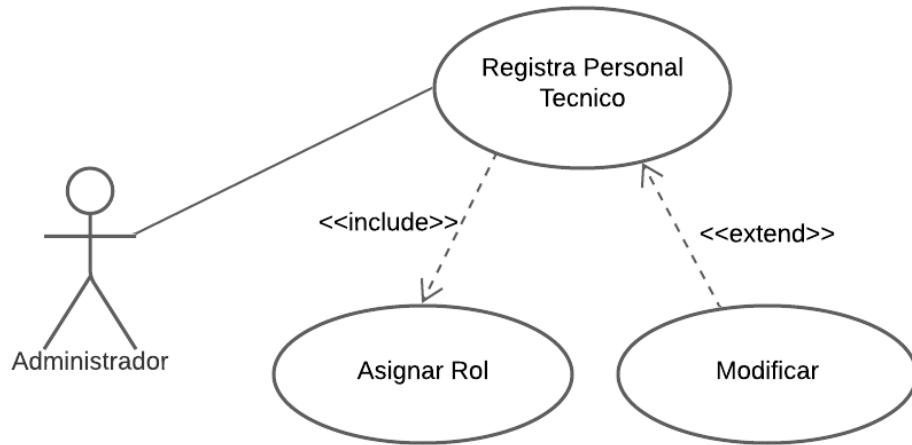
Figura 27

Diagrama de Caso de Uso Inicio de Sesión

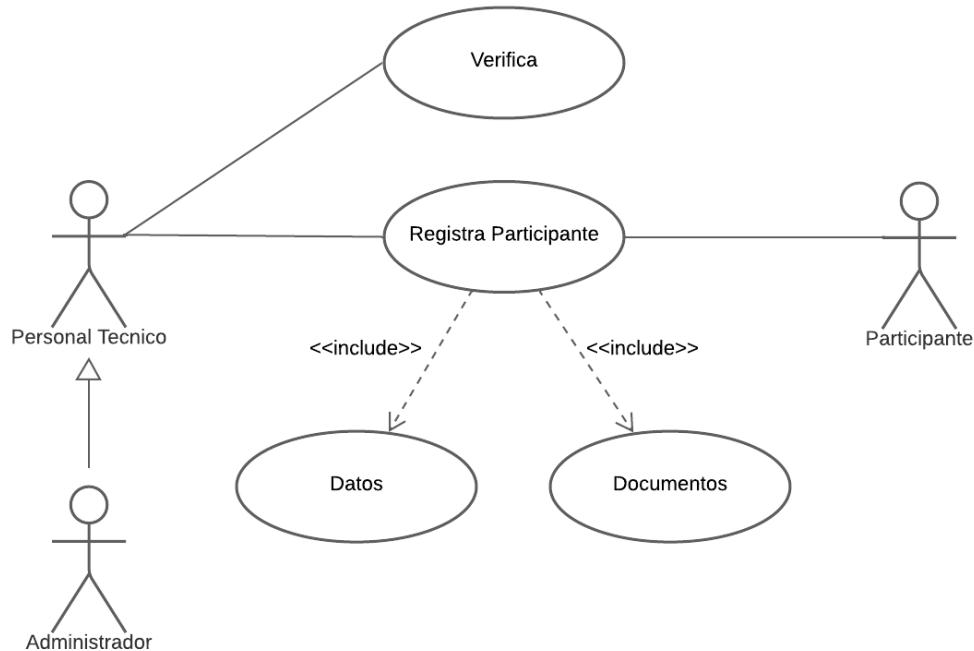


4.2.4.3. Diagrama de Caso de Uso Administración de Personal Técnico

Describe las interacciones entre el administrador del sistema y las funciones relacionadas con la gestión del personal técnico. Esto incluye registrar, actualizar, eliminar o consultar información de los técnicos encargados del mantenimiento o gestión de la plataforma.

Figura 28*Diagrama de Caso de Uso Administración de Personal Técnico***4.2.4.4. Diagrama de Caso de Uso Registro de Participante**

Muestra cómo el administrador o responsable puede registrar a un nuevo participante en el sistema. El diagrama especifica las acciones implicadas en este proceso, como ingresar datos personales, asignar roles o áreas y guardar la información en la base de datos.

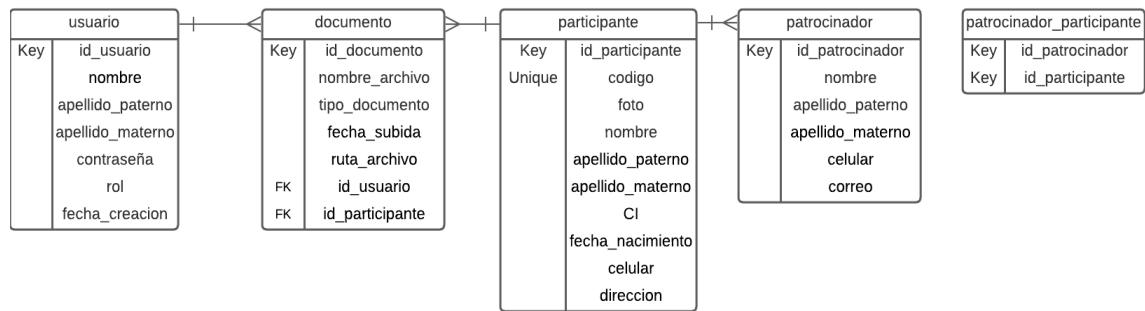
Figura 29*Diagrama de Caso de Uso Registro de Participante*

4.2.5. Modelo Entidad-Relación

El modelo Entidad-Relación representa la estructura de la base de datos del sistema de gestión documental. Incluye entidades clave como Participante, Documento, Usuario y Rol, junto con sus respectivas relaciones. Por ejemplo, un Usuario puede tener un Rol específico que define su nivel de acceso, y puede registrar múltiples Documentos vinculados a un Participante. Este modelo garantiza una organización clara y coherente de la información, permitiendo consultas eficientes y seguras.

Figura 30

Modelo Entidad-Relación



4.2.5.1. Diccionario de Datos

Entidad	Atributo	Tipo de Dato	Tamaño	Descripción
Usuario	id_usuario	INT	10	Identificador único del usuario
	nombre	VARCHAR	50	Nombre del usuario
	apellido_paterno	VARCHAR	50	Apellido paterno del usuario
	apellido_matern o	VARCHAR	50	Apellido Materno del usuario

	correo	VARCHAR	255	Correo electrónico del usuario
	contraseña	CHAR	60	Contraseña cifrada
	rol	ENUM	'Administrador', 'Personal'	Identificador del usuario con el rol de Administrador o Personal
	fecha_creacion	DATETIM E	-	Fecha y hora en el que se registra un nuevo usuario
Documento	id_documento	INT	10	Identificador único del documento
	nombre_archivo	VARCHAR	100	Título del documento
	tipo_documento	VARCHAR	50	Tipo de documento
	fecha_subida	DATETIM E	-	Fecha y hora en el que se subió el documento
	ruta_archivo	VARCHAR	255	Ruta donde se almacena el archivo digital
	id_usuario	INT	10	Llave foránea que referencia al usuario que subió el documento

	Id_participante	INT	10	Llave foránea que referencia al participante
Participante	id_participante	INT	10	Identificador único del participante
	código	VARCHAR	20	Para realizar la búsqueda
	foto	VARCHAR	255	Foto del participante
	nombre	VARCHAR	50	Nombre del participante
	apellido_paterno	VARCHAR	50	Apellido paterno del participante
	apellido_matern o	VARCHAR	50	Apellido materno del participante
	CI	int	10	Numero de documento de identidad
	fecha_nacimient o	DATE	-	Fecha de nacimiento del participante
	celular	VARCHAR	10	Celular del participante
	dirección	VARCHAR	150	Dirección del participante

Patrocinador	id_patrocinador	int	10	Identificador único del patrocinador
	nombre	VARCHAR	50	Nombre del patrocinador
	apellido_paterno	VARCHAR	50	Apellido paterno del patrocinador
	apellido_materno	VARCHAR	50	Apellido materno de patrocinador
	celular	VARCHAR	50	Celular internacional del patrocinador
	correo	VARCHAR	255	Correo electrónico del patrocinador
Patrocinador Participante	Id_patrocinador	INT	10	Identificador único del patrocinador (FK)
	Id_participante	INT	10	Identificador único del participante (FK)

4.3. Identificación de Riesgos

Se aplicó la técnica de análisis cualitativo de riesgos mediante el uso de una Matriz de Riesgos.

Esta técnica consistió en:

- **Listar los posibles riesgos** que podrían afectar el éxito del proyecto, basándose en entrevistas realizadas a miembros de la Asociación,

análisis del contexto organizacional y la experiencia del equipo de trabajo.

- **Evaluar la probabilidad** de ocurrencia de cada riesgo (Alta, Media, Baja).
- **Determinar el impacto** que tendría cada riesgo en los objetivos del proyecto (Alto, Medio, Bajo).
- **Clasificar los riesgos** en una matriz según su nivel de severidad (combinación de impacto y probabilidad).

La aplicación de esta técnica permitió priorizar los riesgos identificados y establecer planes de respuesta adecuados para su mitigación o control.

La tabla de identificación de riesgos resultante proporciona una visión clara de los riesgos más críticos y las acciones recomendadas para cada uno.

Tabla 7

Identificación de Riesgos

Riesgo	Descripción	Nivel	Mitigación
Falla de seguridad	Posibilidad de acceso no autorizado a información sensible	Alto	Implementación de roles y autenticación segura
Pérdida de datos	Riesgo durante la migración inicial	Alto	Pruebas controladas y respaldos periódicos
Falta de conectividad	Limitaciones de acceso a internet en algunas zonas	Medio	Posibilidad de uso local con sincronización posterior
Desactualización del sistema	Cambios tecnológicos futuros	Bajo	Documentación técnica completa para facilitar mantenimiento

4.4. Interfaz

Presenta las pantallas diseñadas para la interacción del usuario con el sistema de gestión documental. Estas interfaces han sido desarrolladas considerando los principios de usabilidad y accesibilidad, buscando facilitar la navegación y la eficiencia en la realización de las tareas. Se incluyen diseños de las principales funcionalidades del sistema, desde el inicio de sesión hasta la administración de usuarios y la gestión de la información de los participantes. El objetivo de esta sección es proporcionar una visión clara de cómo los usuarios interactuarán con el sistema, destacando la importancia de una interfaz intuitiva y amigable para garantizar una experiencia de usuario positiva.

Esta figura 31 muestra el diseño de la interfaz donde los usuarios ingresan sus credenciales (usuario y contraseña) para acceder al sistema.

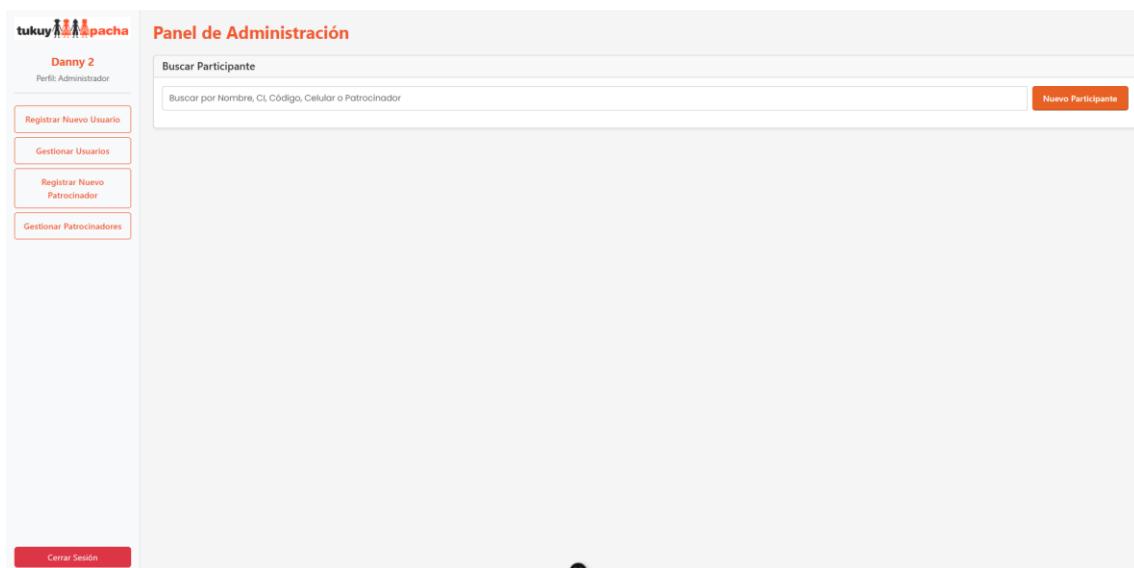
Figura 31

Pantalla de Inicio de Sesión



En la figura 32 presenta la interfaz principal que visualizan los usuarios con roles administrativos. Incluye accesos directos a las funciones de gestión del sistema, como la administración de usuarios, la configuración y otras herramientas administrativas.

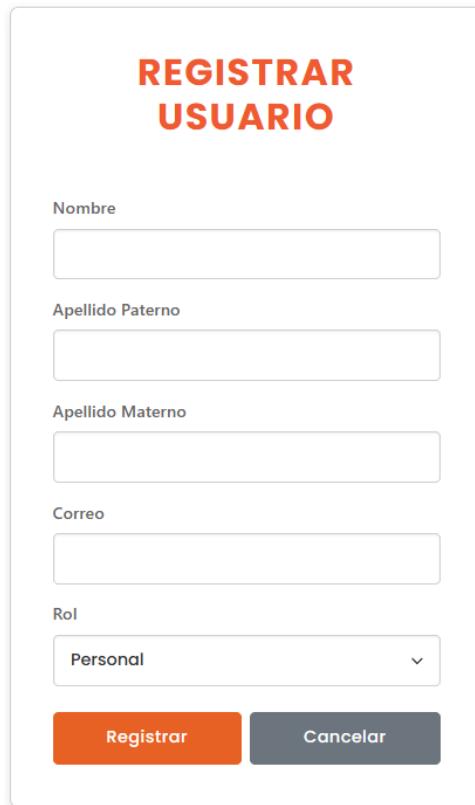
Figura 32
Panel de Administración del Sistema



La figura 33 muestra los campos necesarios para la creación de una cuenta de usuario, incluyendo información personal y permisos de acceso.

Figura 33

Formulario para Registrar Nuevo Usuario



REGISTRAR USUARIO

Nombre

Apellido Paterno

Apellido Materno

Correo

Rol

Registrar **Cancelar**

Esta figura 34 ilustra la visualización de la lista de todos los usuarios que tienen acceso al sistema. Puede incluir opciones editar o eliminar usuarios, facilitando la gestión de las cuentas.

Figura 34

Lista de Usuarios Registrados

Lista de Usuarios						
Nombre	Apellido Paterno	Apellido Materno	Correo	Rol	Restablecer Contraseña	Acciones
Danny 2	Gonzales 2	Medina 2	dan2@gmail.com	Administrador		 
Danny	Gonzales	Medina	dan9813gm@gmail.com	Personal		 

La siguiente figura 35 presenta el formulario diseñado para ingresar la información de un nuevo participante en el sistema. Detalla los campos necesarios para registrar los datos personales, información de contacto y otros detalles relevantes del participante.

Figura 35

Formulario para Registrar Nuevo Participante

Registrar Nuevo Participante

Código	Nombre	Apellido Paterno
Ej: TKC 0005 o TKE 1234		
Para niños, niñas y adolescentes es TKC, Para adultos mayores es TKE		
Apellido Materno		
<input type="text"/> <small>Apellido Materno</small>		
Foto		
<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado		
Cédula de Identidad	Fecha de Nacimiento	Dirección (Bolivia)
Ej: 8724526	dd/mm/aaaa <input type="button" value=""/>	Ej: Av. Arce 1234, La Paz
<small>Debe incluir un departamento boliviano: La Paz, Cochabamba, Santa Cruz, Oruro, Potosí, Tarija, Chuquisaca, Beni, Pando</small>		
Celular	Buscar Patrocinador	
<input type="text"/> Ej: 78765432	<input type="text"/> Buscar por nombre... <input type="button" value="Limpiar"/>	
Formato: 7XXXXXXXX o 6XXXXXXXX (8 dígitos)		
Nombre del Patrocinador	Apellido Paterno	
<input type="text"/>	<input type="text"/>	
Apellido Materno	Celular	
<input type="text"/>	<input type="text"/>	
Correo Electronico		
<input type="text"/>		
Documentos Acompañados		
Tipo de Documento	Archivo	
<input type="button" value="Seleccione un tipo"/>	<input type="button" value="Seleccionar archivo"/> Ning...ado <input type="button" value="Eliminar"/>	
<input type="button" value="+ Añadir Documento"/>		

Esta figura 36 muestra la interfaz donde se visualiza la información completa de un participante específico. Puede incluir diferentes secciones o pestañas para organizar la información en categorías (datos personales, historial, etc.).

Figura 36

Detalles del Participante

The screenshot shows a web-based application interface for managing participant details. At the top, there is a search bar labeled 'Buscar Participante' with the input 'dann'. To the right of the search bar are two buttons: 'Editar Participante' (in red) and 'Descargar PDF' (in green). Below the search bar, the participant's name 'danny 2 gonzales 2 medina 2' is displayed. A small profile picture of the participant is shown next to their name. To the right of the profile picture, their identification number 'CI: 11111111' is listed, along with their address 'Dirección: Av. panamericana, Cochabamba', and names of their parents 'Nombre del Padre/Madre: danny' and 'Apellido Materno: medina'. On the left side of the page, there is a section titled 'Documentos Relacionados' (Related Documents) which lists three files: 'Archivo: 13390919215683550.jpg' (Type: Imagen, Date: 11/8/2021), 'Archivo: Proyecto de Grado_Danny Gonzales Medina revisado.pdf' (Type: Documento, Date: 11/8/2021), and 'Archivo: Optimización del Sistema de Gestión Documental - Asociación Tukuy Pacha (Respuestas).pdf' (Type: Salir, Date: 11/8/2021). Each document entry includes 'Ver' and 'Descargar' buttons.

En la representación de esta figura 37 ilustra la interfaz que visualizan los usuarios con roles de personal del sistema. Se enfoca en las funcionalidades que estos usuarios necesitan para sus tareas diarias, como la consulta y registro de información de los participantes.

Figura 37

Panel del Personal del Sistema

The screenshot shows a web-based application interface for managing system staff. On the left side, there is a sidebar with the logo 'tukuy pacha' and the name 'Danny' under the heading 'Perfil Personal'. Below this, there is a button labeled 'Registrar Nuevo Participante' (in red). The main content area is titled 'Panel de Personal'. It features a search bar labeled 'Buscar Participante' with the input 'dann'. To the right of the search bar is a 'Descargar PDF' button. Below the search bar, the participant's name 'danny 2 gonzales 2 medina 2' is displayed, accompanied by a small profile picture. To the right of the profile picture, the same personal information is listed: 'CI: 11111111', 'Dirección: Av. panamericana, Cochabamba', 'Nombre del Padre/Madre: danny', and 'Apellido Materno: medina'. A section titled 'Documentos Relacionados' (Related Documents) is present, listing the same three files as in Figure 36: 'Archivo: 13390919215683550.jpg', 'Archivo: Proyecto de Grado_Danny Gonzales Medina revisado.pdf', and 'Archivo: Optimización del Sistema de Gestión Documental - Asociación Tukuy Pacha (Respuestas).pdf'. Each document entry includes 'Ver' and 'Descargar' buttons. At the bottom of the page, there is a red button labeled 'Cerrar sesión' (Close session).

4.5. Pruebas

Las pruebas se llevaron a cabo en varias fases del desarrollo para asegurar la calidad del sistema:

- **Pruebas de integración:** Evaluaron el correcto funcionamiento de las interacciones entre los módulos (API, base de datos, interfaz).
- **Pruebas de aceptación del usuario (UAT):** Fueron realizadas con personal de Tukuy Pacha, simulando escenarios reales de uso.
- **Pruebas de seguridad:** Se implementaron validaciones contra ataques como inyección SQL y XSS, así como revisión de autenticación y autorización.

4.5.1. Pruebas de Seguridad

Durante la implementación del sistema de sistematización y organización documental para la Asociación Tukuy Pacha, se establecieron e incorporaron diversas medidas de seguridad orientadas a garantizar la integridad, confidencialidad y disponibilidad de la información. A continuación, se presentan las pruebas realizadas para comprobar el cumplimiento de dichas medidas:

4.5.1.1. Prueba de Integración

El objetivo fue verificar que los diferentes componentes del backend (servidor, rutas, middlewares) interactúan correctamente entre sí.

Se utilizó **Vitest** como ejecutor de pruebas y **Supertest** para simular peticiones HTTP a la API de Express, verificando el correcto funcionamiento de las rutas, los *middlewares* y las interacciones con la base de datos.

En el *frontend*, se realizaron pruebas de integración con **Vue.js** para confirmar que las rutas y las vistas funcionan de manera correcta. Esto incluyó la verificación de que el sistema maneja la navegación, el control de acceso de usuarios y las interacciones de los componentes con el *backend*, asegurando que la experiencia del usuario sea fluida y que el sistema sea estable.

Esta combinación permitió probar el comportamiento de los *endpoints* sin necesidad de un servidor en funcionamiento, como se puede observar en la figura 38.

Las pruebas realizadas fueron:

Verificación de rutas inexistentes: Se validó que el servidor responde con el código de estado **404** para cualquier ruta no definida. Esto previene que se expongan rutas o información de manera accidental.

Validación de métodos HTTP: Se comprobó que el *endpoint* de autenticación (/api/auth/login) solo responde a peticiones **POST**, rechazando cualquier otro método. Esto es una medida de seguridad fundamental para prevenir accesos no autorizados.

Existencia y respuesta de endpoints: Se aseguró que las rutas principales como /api/participantes, /api/documentos, /api/usuarios y /api/patrocinadores están correctamente configuradas y responden con un código de estado válido, evitando que se expongan errores internos.

Manejo de archivos estáticos: Se verificó que el servidor maneja correctamente la ruta de archivos estáticos (/uploads), devolviendo un **404** cuando se solicita un archivo que no existe.

Figura 38

Prueba de Integración de App.js

```
C:\laragon\www\tukuy pacha 2\back(main -> origin) (back@1.0.0)
λ npm test
    4.5.1. Pruebas de Seguridad
  > back@1.0.0 test
  > vitest
    Durante la implementación del sistema de sistematización y organización documental para la
    Asociación Tukuy Pacha, se establecieron e incorporaron diversas medidas de seguridad
    orientadas a garantizar la integridad, confidencialidad y disponibilidad de la información. A
    Multer intentará guardar en (upload.js): C:\laragon\www\tukuy pacha 2\back\uploads
    stdout | test/app.test.js
    Controlador de documentos está utilizando uploadsDir: C:\laragon\www\tukuy pacha 2\back\uploads
    stdout | test/app.test.js
    ⚡ Sirviendo estáticos desde: C:\laragon\www\tukuy pacha 2\back\uploads
    GET /esta-ruta-no-existe 404 1.954 ms - 158 Verificar que los diferentes componentes del backend (servidor, rutas,
    POST /api/auth/login 500 5.570 ms - 1267
    GET /api/participantes 200 25.905 ms - 842 interactúan correctamente entre sí
    GET /api/documentos 200 1.765 ms - 1364
    GET /api usuarios 200 1.154 ms - 289
    GET /api/patrocinadores 200 1.096 ms - 292
    GET /uploads/archivo-inexistente.txt 404 1.162 ms - 170
    GET /uploads/prueba.txt 404 0.451 ms - 157
    ✓ test/app.test.js (8 tests) 79ms
      ✓ Pruebas de la API de Express > Debería retornar 404 para una ruta inexistente 19ms
      ✓ Pruebas de la API de Express > Debería manejar la ruta POST /api/auth/login 10ms
      ✓ Pruebas de la API de Express > Debería poder acceder a la ruta GET /api/participantes 30ms
      ✓ Pruebas de la API de Express > Debería poder acceder a la ruta GET /api/documentos 5ms
      ✓ Pruebas de la API de Express > Debería poder acceder a la ruta GET /api/usuarios 4ms
      ✓ Pruebas de la API de Express > Debería poder acceder a la ruta GET /api/patrocinadores 4ms
      ✓ Pruebas de la API de Express > Debería retornar 404 para un archivo estático inexistente 4ms
      ✓ Pruebas de la API de Express > Debería retornar 200 para un archivo estático que existe 3ms
    Test Files 1 passed (1)
    Tests 8 passed (8)
    Start at 15:15:24
    Duration 6.17s (transform 300ms, setup 0ms, collect 5.61s, tests 79ms, environment 0ms, prepare 219ms)

    PASS Waiting for file changes...
    press h to show help, press q to quit
```

La figura 39 muestra que las pruebas de integración han pasado con éxito.

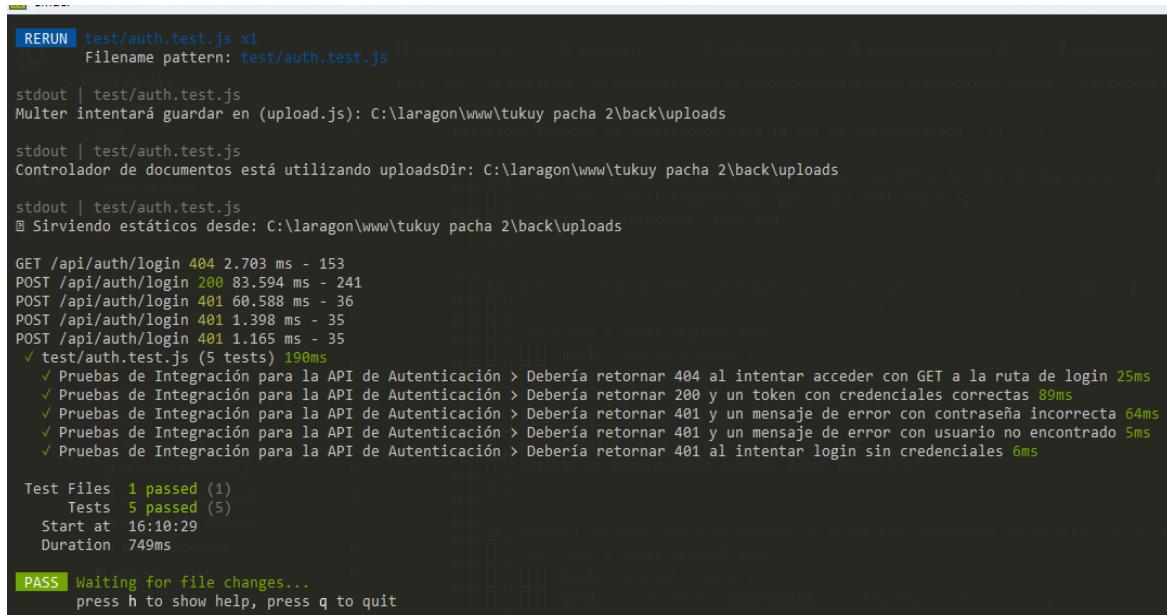
El resultado PASS indica que las cinco pruebas que tienes definidas en tu archivo test/auth.test.js se ejecutaron correctamente y todas las aserciones que se definió se cumplieron.

- Ruta incorrecta (GET): La prueba para la ruta /api/auth/login con un método GET devolvió correctamente un 404.
- Login exitoso: La prueba con credenciales correctas devolvió un 200 y un token válido.
- Contraseña incorrecta: La prueba con una contraseña incorrecta devolvió un 401 y el mensaje de error esperado.

- Usuario no encontrado: La prueba con un correo electrónico que no existe en tu base de datos devolvió un 401 y el mensaje de error esperado.
- Sin credenciales: La prueba sin enviar credenciales devolvió un 401.

Figura 39

Prueba de Integración de Auth



```
RERUN test/auth.test.js x1
Filename pattern: test/auth.test.js

stdout | test/auth.test.js
Multer intentará guardar en (upload.js): C:\laragon\www\tukuy pacha 2\back\uploads

stdout | test/auth.test.js
Controlador de documentos está utilizando uploadsDir: C:\laragon\www\tukuy pacha 2\back\uploads

stdout | test/auth.test.js
@ Sirviendo estáticos desde: C:\laragon\www\tukuy pacha 2\back\uploads

GET /api/auth/login 404 2.703 ms - 153
POST /api/auth/login 200 83.594 ms - 241
POST /api/auth/login 401 60.588 ms - 36
POST /api/auth/login 401 1.398 ms - 35
POST /api/auth/login 401 1.165 ms - 35
✓ test/auth.test.js (5 tests) 190ms
  ✓ Pruebas de Integración para la API de Autenticación > Debería retornar 404 al intentar acceder con GET a la ruta de login 25ms
  ✓ Pruebas de Integración para la API de Autenticación > Debería retornar 200 y un token con credenciales correctas 89ms
  ✓ Pruebas de Integración para la API de Autenticación > Debería retornar 401 y un mensaje de error con contraseña incorrecta 64ms
  ✓ Pruebas de Integración para la API de Autenticación > Debería retornar 401 y un mensaje de error con usuario no encontrado 5ms
  ✓ Pruebas de Integración para la API de Autenticación > Debería retornar 401 al intentar login sin credenciales 6ms

Test Files 1 passed (1)
Tests 5 passed (5)
Start at 16:10:29
Duration 749ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

En la siguiente figura 40 muestra que las pruebas de integración han pasado con éxito. El resultado PASS indica que las cinco pruebas que se definieron en el archivo test/documentos.test.js se ejecutaron correctamente y todas las aserciones que se definieron se cumplieron.

El API de documentos está funcionando como se espera:

- Subir un nuevo documento: La prueba para la ruta /api/documentos con un método POST devolvió un 201 (creado) correctamente.
- Actualizar un documento: La prueba para la ruta /api/documentos/:id con un método PUT devolvió un 200 y el mensaje esperado de actualización correcta.

- Descargar un documento: La prueba para la ruta /api/documentos/download/:filename con un método GET devolvió un 200 y el mensaje de descarga exitosa.
- Listar documentos: La prueba para la ruta /api/documentos con un método GET devolvió una lista de documentos, incluyendo el recién creado.
- Eliminar un documento: La prueba para la ruta /api/documentos/:id con un método DELETE devolvió un 200 y el mensaje de eliminación correcta.

Figura 40

Prueba de Integración de Documentos

```
stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos > PUT /api/documentos/:id - Debería actualizar un documento (200)
Archivo antiguo eliminado: C:\laragon\www\tukuy pacha 2\back\uploads\1754948722361-701296567.pdf

PUT /api/documentos/38 200 10.457 ms - 103
GET /api/documentos/download/1754948722361-701296567.pdf 200 3.154 ms - 44
DELETE /api/documentos/38 200 6.044 ms - 47
stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos > PUT /api/documentos/:id - Debería actualizar un documento (200)
🕒 Respuesta de actualización: {
  status: 200,
  body: {
    mensaje: 'Documento actualizado correctamente.',
    nuevo_nombre_archivo: '1754948722361-701296567.pdf'
  }
}

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos > GET /api/documentos/download/:filename - Debería descargar un documento (200)
Intentando descargar el archivo desde la ruta: C:\laragon\www\tukuy pacha 2\back\uploads\1754948722361-701296567.pdf

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos > GET /api/documentos/download/:filename - Debería descargar un documento (200)
Archivo '1754948722361-701296567.pdf' enviado para descarga exitosamente.

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos > DELETE /api/documentos/:id - Debería eliminar un documento (200)
✖ Respuesta de eliminación: { status: 200, body: { mensaje: 'Documento eliminado correctamente' } }

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos
🕒 Limpieando después de las pruebas...
🕒 Eliminando usuario ID: 56

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos
Archivo físico eliminado: C:\laragon\www\tukuy pacha 2\back\uploads\1754948722361-701296567.pdf

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos
🕒 Eliminando participante ID: 64

stdout | test/documentos.test.js > Pruebas de Integración para la API de Documentos
🕒 Archivo de prueba eliminado: C:\laragon\www\tukuy pacha 2\back\test\test-file.pdf
🕒 Limpieza completada

✓ test/documentos.test.js (5 tests) 135ms
✓ Pruebas de Integración para la API de Documentos > POST /api/documentos - Debería subir un nuevo documento (201) 41ms
✓ Pruebas de Integración para la API de Documentos > GET /api/documentos - Debería listar documentos incluyendo el recién creado (200) 9ms
✓ Pruebas de Integración para la API de Documentos > PUT /api/documentos/:id - Debería actualizar un documento (200) 15ms
✓ Pruebas de Integración para la API de Documentos > GET /api/documentos/download/:filename - Debería descargar un documento (200) 7ms
✓ Pruebas de Integración para la API de Documentos > DELETE /api/documentos/:id - Debería eliminar un documento (200) 11ms

Test Files 1 passed (1)
Tests 5 passed (5)
Start at 17:45:21
Duration 79ms

PASSED Waiting for file changes...
press h to show help, press q to quit
```

Las pruebas de integración para la API de participantes han pasado con éxito. El resultado PASS indica que las siete pruebas definidas en el archivo test/participantes.test.js se ejecutaron correctamente y todas las aserciones se cumplieron según la figura 41.

Dáandonos con el resultado de que la API de participantes funciona como se espera en los siguientes escenarios:

Escenarios de Pruebas Exitosas

- Creación: La prueba para la ruta /api/participantes con un método POST devolvió un código 201 (creado) correctamente.
- Listado: La prueba para la ruta /api/participantes con un método GET devolvió una lista de participantes, incluyendo el recién creado.
- Búsqueda: La prueba para la ruta /api/participantes/buscar?termino=TEST-1754... devolvió un 200 con el participante encontrado.
- Obtención por ID: La prueba para la ruta /api/participantes/:id con un método GET devolvió un 200 y la información del participante correspondiente.
- Actualización: Las pruebas para la ruta /api/participantes/:id con un método PUT devolvieron un 200 y el mensaje esperado de actualización correcta, tanto para la actualización con foto como sin ella.
- Eliminación: La prueba para la ruta /api/participantes/:id con un método DELETE devolvió un 200 y el mensaje de eliminación correcta.

Figura 41

Prueba de Integración de Participantes

```
c:\laragon\www\tukuy pacha 2\back(main -> origin) (back@1.0.0)
λ npx vitest test/participantes.test.js

DEV v3.2.4 C:/laragon/www/tukuy pacha 2/back

stdout | test/participantes.test.js
Multer intentará guardar en (upload.js): C:\laragon\www\tukuy pacha 2\back\uploads

stdout | test/participantes.test.js
Controlador de documentos está utilizando uploadsDir: C:\laragon\www\tukuy pacha 2\back\uploads

stdout | test/participantes.test.js
└ Sirviendo estáticos desde: C:\laragon\www\tukuy pacha 2\back\uploads

stdout | test/participantes.test.js > Pruebas de Integración para la API de Participantes
  └ Configurando entorno de pruebas para participantes...
    └ Archivo de imagen de prueba creado: C:\laragon\www\tukuy pacha 2\back\test\test-image.png
    └ Limpiando datos anteriores...

stdout | test/participantes.test.js > Pruebas de Integración para la API de Participantes
  └ Creando patrocinador de prueba...

stdout | test/participantes.test.js > Pruebas de Integración para la API de Participantes
  └ Patrocinador creado con ID: 7

POST /api/participantes 201 18.967 ms - 84
GET /api/participantes 200 2.059 ms - 1142
GET /api/participantes/buscar?termino=TEST-1754949771304 200 2.283 ms - 530
GET /api/participantes/66 200 1.458 ms - 474
PUT /api/participantes/66 200 7.272 ms - 68
PUT /api/participantes/66 200 8.633 ms - 68
DELETE /api/participantes/66 200 6.024 ms - 67
GET /api/participantes/66 404 0.669 ms - 40
stdout | test/participantes.test.js > Pruebas de Integración para la API de Participantes
  └ Limpiando después de las pruebas de participantes...

stdout | test/participantes.test.js > Pruebas de Integración para la API de Participantes
  └ Limpieza de participantes completada

  ✓ test/participantes.test.js (7 tests) 124ms
    ✓ Pruebas de Integración para la API de Participantes > POST /api/participantes - Debería crear un nuevo participante (201) 39ms
    ✓ Pruebas de Integración para la API de Participantes > GET /api/participantes - Debería listar participantes incluyendo el recién creado (200) 6ms
    ✓ Pruebas de Integración para la API de Participantes > GET /api/participantes/buscar - Debería encontrar el participante por su código (200) 5ms
    ✓ Pruebas de Integración para la API de Participantes > GET /api/participantes/:id - Debería obtener un participante por su ID (200) 4ms
    ✓ Pruebas de Integración para la API de Participantes > PUT /api/participantes/:id - Debería actualizar un participante (200) 11ms
    ✓ Pruebas de Integración para la API de Participantes > PUT /api/participantes/:id - Debería actualizar un participante sin foto (200) 12ms
    ✓ Pruebas de Integración para la API de Participantes > DELETE /api/participantes/:id - Debería eliminar un participante (200) 13ms

Test Files 1 passed (1)
  Tests 7 passed (7)
  Start at 18:02:50
  Duration 1.25s (transform 123ms, setup 0ms, collect 742ms, tests 124ms, environment 0ms, prepare 128ms)

PASS Waiting for file changes...
press h to show help, press q to quit
```

El reporte de la figura 42 confirma que las pruebas de integración para la API de patrocinadores fueron exitosas. El resultado PASS indica que las ocho pruebas definidas en el archivo `test/patrocinadores.test.js` se ejecutaron correctamente, cumpliendo con todas las aserciones establecidas.

De esta manera, se valida que la API de patrocinadores opera como se espera en los siguientes escenarios:

Escenarios de Pruebas Exitosos

- Creación de patrocinador: La prueba para la ruta `/api/patrocinadores` con un método POST devolvió un código 201 (creado) y el mensaje de "Patrocinador creado exitosamente".

- Listado de todos los patrocinadores: La prueba para la ruta /api/patrocinadores con un método GET devolvió un 200, indicando que se listaron correctamente.
- Búsqueda de patrocinador: La prueba para la ruta /api/patrocinadores/busqueda devolvió un 200 con el patrocinador encontrado por su nombre.
- Obtención por ID: La prueba para la ruta /api/patrocinadores/:id con un método GET devolvió un 200 y la información del patrocinador correspondiente.
- Creación de patrocinio: La prueba para la ruta /api/patrocinadores/patrocinar con un método POST devolvió un 201 y el mensaje de "Patrocinio registrado exitosamente".
- Obtención de participantes patrocinados: La prueba para la ruta /api/patrocinadores/:id/participantes con un método GET devolvió un 200, indicando que se obtuvieron los participantes patrocinados.
- Actualización de patrocinador: La prueba para la ruta /api/patrocinadores/:id con un método PUT devolvió un 200 y el mensaje esperado de actualización correcta.
- Eliminación de patrocinador: La prueba para la ruta /api/patrocinadores/:id con un método DELETE devolvió un 200 y el mensaje de eliminación correcta.

Figura 42

Prueba de Integración de Patrocinadores

```
stdout | test/patrocinador.test.js > Pruebas de Integración para la API de Patrocinadores
  Creando participante de prueba para patrocinio...
    ✘ Participante creado con ID: 70

  POST /api/patrocinadores 201 7.909 ms - 53
  stdout | test/patrocinador.test.js > Pruebas de Integración para la API de Patrocinadores
    ✘ Respuesta POST patrocinador: { mensaje: 'Patrocinador creado exitosamente', id: 9 }

  GET /api/patrocinadores 200 2.881 ms - 469
  GET /api/patrocinadores?busqueda=testPatrocinador 200 0.991 ms - 178
  GET /api/patrocinadores/9 200 0.952 ms - 176
  POST /api/patrocinadores/patrocinar 201 5.132 ms - 48
  stdout | test/patrocinador.test.js > Pruebas de Integración para la API de Patrocinadores > POST /api/patrocinadores/patrocinar - Debería crear un nuevo patrocinio (201)
    ✘ Respuesta POST patrocinio: { mensaje: 'Patrocinio registrado exitosamente' }

  GET /api/patrocinadores/9/participantes 200 1.677 ms - 276
  PUT /api/patrocinadores/9 200 4.595 ms - 51
  DELETE /api/patrocinadores/9 200 5.289 ms - 68
  GET /api/patrocinadores/9 404 0.798 ms - 40
  stdout | test/patrocinador.test.js > Pruebas de Integración para la API de Patrocinadores
    ✘ Limpiando después de las pruebas de patrocinadores...

  stdout | test/patrocinador.test.js > Pruebas de Integración para la API de Patrocinadores
    ✘ Limpieza de patrocinadores completada

    ✓ test/patrocinador.test.js (8 tests) 107ms
    ✓ Pruebas de Integración para la API de Patrocinadores > POST /api/patrocinadores - Debería crear un nuevo patrocinador (201) 26ms
    ✓ Pruebas de Integración para la API de Patrocinadores > GET /api/patrocinadores - Debería listar todos los patrocinadores (200) 7ms
    ✓ Pruebas de Integración para la API de Patrocinadores > GET /api/patrocinadores?busqueda= - Debería encontrar patrocinadores por su nombre (200) 4ms
    ✓ Pruebas de Integración para la API de Patrocinadores > GET /api/patrocinadores/:id - Debería obtener un patrocinador por su ID (200) 4ms
    ✓ Pruebas de Integración para la API de Patrocinadores > POST /api/patrocinadores/patrocinar - Debería crear un nuevo patrocinio (201) 9ms
    ✓ Pruebas de Integración para la API de Patrocinadores > GET /api/patrocinadores/:id/participantes - Debería obtener los participantes de un patrocinador (200) 4ms
    ✓ Pruebas de Integración para la API de Patrocinadores > PUT /api/patrocinadores/:id - Debería actualizar un patrocinador existente (200) 7ms
    ✓ Pruebas de Integración para la API de Patrocinadores > DELETE /api/patrocinadores/:id - Debería eliminar el patrocinador y su patrocinio (200) 13ms

  Test Files: 1 passed (1)
  Tests: 8 passed (8)
  Start at: 18:23:39
  Duration: 1.215 (transform 116ms, setup 0ms, collect 719ms, tests 107ms, environment 0ms, prepare 125ms)

  PASS Waiting for file changes...
  press h to show help, press q to quit
```

El reporte de la figura 43 confirma que las pruebas de integración para la API de usuarios fueron exitosas. El resultado PASS indica que las nueve pruebas definidas en el archivo test/usuarios.test.js se ejecutaron correctamente y todas las aserciones se cumplieron.

Con base en estos resultados, se confirma el correcto funcionamiento de la API de usuarios en las siguientes funciones:

Escenarios de Pruebas Exitosos

- Creación de usuario: La prueba para la ruta /api/usuarios con un método POST devolvió un código 201 (creado) y un mensaje confirmando la creación y el envío de un correo electrónico.
- Listado de usuarios: La prueba para la ruta /api/usuarios con un método GET devolvió un 200, indicando que se listan todos los usuarios correctamente.
- Obtención por ID: La prueba para la ruta /api/usuarios/:id con un método GET devolvió un 200 y la información del usuario correspondiente.

- Usuario no encontrado: La prueba para la ruta /api/usuarios/:id devolvió un 404 cuando se buscó un usuario que no existe.
- Actualización de rol: La prueba para la ruta /api/usuarios/:id/rol con un método PUT devolvió un 200 al actualizar correctamente el rol de un usuario.
- Actualización de contraseña: La prueba para la ruta /api/usuarios/:id/contraseña con un método PUT devolvió un 200 al actualizar la contraseña de un usuario.
- Restablecimiento de contraseña: La prueba para la ruta /api/usuarios/:id/restablecer-contraseña con un método POST devolvió un 200, confirmando que la contraseña se restableció y se envió un correo.
- Actualización de usuario: La prueba para la ruta /api/usuarios/:id con un método PUT devolvió un 200 al actualizar la información de un usuario existente.
- Eliminación de usuario: La prueba para la ruta /api/usuarios/:id con un método DELETE devolvió un 200 y el mensaje de eliminación correcta.

Figura 43

Prueba de Integración de Usuarios

```
stdout | test/usuarios.test.js > Pruebas de Integración para la API de Usuarios
@ Configurando entorno de pruebas para usuarios...
  ✘ Limpiando datos anteriores...
POST /api/usuarios 201 2521.766 ms - 72
stdout | test/usuarios.test.js > Pruebas de Integración para la API de Usuarios > POST /api/usuarios - Debería crear un nuevo usuario y enviar un correo (201)
  ✘ Respuesta POST usuarios: {
    mensaje: 'Usuario creado y credenciales enviadas por correo.',
    id: 60
  }
GET /api/usuarios 200 2.780 ms - 454
GET /api/usuarios/60 200 1.131 ms - 164
GET /api/usuarios/999999 404 0.691 ms - 35
PUT /api/usuarios/60/rol 200 6.104 ms - 29
PUT /api/usuarios/60/contraseña 200 64.388 ms - 37
POST /api/usuarios/60/restablecer-contraseña 200 2449.407 ms - 68
PUT /api/usuarios/60 200 5.577 ms - 47
DELETE /api/usuarios/60 200 5.951 ms - 73
stdout | test/usuarios.test.js > Pruebas de Integración para la API de Usuarios
  ✘ Limpiando después de las pruebas de usuarios...
stdout | test/usuarios.test.js > Pruebas de Integración para la API de Usuarios
  ✘ Limpieza de usuarios completada

  ✓ test/usuarios.test.js (9 tests) 5133ms
    ✓ Pruebas de Integración para la API de Usuarios > POST /api/usuarios - Debería crear un nuevo usuario y enviar un correo (201) 2548ms
    ✓ Pruebas de Integración para la API de Usuarios > GET /api/usuarios - Debería listar todos los usuarios (200) 7ms
    ✓ Pruebas de Integración para la API de Usuarios > GET /api/usuarios/:id - Debería obtener un usuario por su ID (200) 5ms
    ✓ Pruebas de Integración para la API de Usuarios > GET /api/usuarios/:id - Debería retornar 404 si el usuario no existe (404) 3ms
    ✓ Pruebas de Integración para la API de Usuarios > PUT /api/usuarios/:id/rol - Debería actualizar el rol de un usuario (200) 10ms
    ✓ Pruebas de Integración para la API de Usuarios > PUT /api/usuarios/:id/contraseña - Debería actualizar la contraseña de un usuario (200) 67ms
    ✓ Pruebas de Integración para la API de Usuarios > POST /api/usuarios/:id/restablecer-contraseña - Debería restablecer la contraseña y enviar un correo (200) 2453ms
    ✓ Pruebas de Integración para la API de Usuarios > PUT /api/usuarios/:id - Debería actualizar un usuario existente (200) 9ms
    ✓ Pruebas de Integración para la API de Usuarios > DELETE /api/usuarios/:id - Debería eliminar un usuario (200) 10ms

Test Files 1 passed (1)
Tests 9 passed (9)
Start at 18:47:53
Duration 6.275 (transform 158ms, setup 0ms, collect 773ms, tests 5.13s, environment 0ms, prepare 127ms)

PASS | Waiting for file changes...
  press h to show help, press q to quit
```

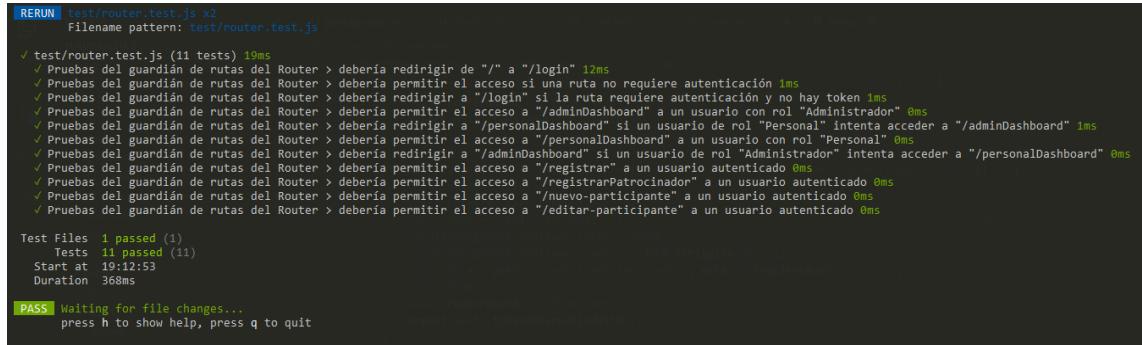
Las pruebas de integración de la figura 44 confirma que las rutas del frontend resultaron exitosas. El resultado PASS indica que las 11 pruebas definidas en el archivo test/router.test.js se ejecutaron correctamente y todas las aserciones se cumplieron.

Esto confirma el funcionamiento correcto del guardián en las siguientes funciones: Escenarios de Pruebas Exitosas

- Redirección a /login sin autenticación: La prueba verificó que un usuario no autenticado es redirigido correctamente a la página de login al intentar acceder a rutas protegidas.
- Acceso a rutas sin autenticación: La prueba confirmó que el guardián permite el acceso a rutas que no requieren autenticación.
- Redirección de usuario autenticado sin rol de administrador: Se verificó que un usuario con rol "Personal" es redirigido a /personalDashboard si intenta acceder a /adminDashboard.
- Acceso a rutas de administración: Se confirmó que un usuario con rol "Administrador" puede acceder a la ruta /adminDashboard.
- Acceso a rutas de usuario personal: Se verificó que un usuario con rol "Personal" puede acceder a la ruta /personalDashboard.
- Redirección de usuario con rol "Administrador" a rutas de "Personal": La prueba confirmó que un usuario con rol "Administrador" es redirigido a /adminDashboard si intenta acceder a /personalDashboard.
- Acceso a rutas protegidas con autenticación: Se verificó que un usuario autenticado puede acceder a rutas como /registrar, /registrarPatrocinador, /nuevo-participante y /editar-participante.

Figura 44

Prueba de Integración de Routes del Fronted



```
RERUN host/router.test.js X2
Filename pattern: test/router.test.js

✓ test/router.test.js (11 tests) 10ms
  ✓ Pruebas del guardián de rutas del Router > debería redirigir de "/" a "/login" 12ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso si una ruta no requiere autenticación 1ms
  ✓ Pruebas del guardián de rutas del Router > debería redirigir a "/Login" si la ruta requiere autenticación y no hay token 1ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso a "/adminDashboard" a un usuario con rol "Administrador" 0ms
  ✓ Pruebas del guardián de rutas del Router > debería redirigir a "/personalDashboard" si un usuario de rol "Personal" intenta acceder a "/adminDashboard" 1ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso a "/personalDashboard" a un usuario con rol "Personal" 0ms
  ✓ Pruebas del guardián de rutas del Router > debería redirigir a "/adminDashboard" si un usuario de rol "Administrador" intenta acceder a "/personalDashboard" 0ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso a "/registro" a un usuario autenticado 0ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso a "/registrarPatrocinador" a un usuario autenticado 0ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso a "/nuevo-participante" a un usuario autenticado 0ms
  ✓ Pruebas del guardián de rutas del Router > debería permitir el acceso a "/editar-participante" a un usuario autenticado 0ms

Test Files: 1 passed (1)
Tests: 11 passed (11)
Start at 19:12:53
Duration 368ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Las pruebas del componente de login del frontend resultaron exitosas. El resultado PASS indica que las cuatro pruebas definidas en el archivo test/login.test.js se ejecutaron correctamente. De acuerdo a la figura 45.

Esto valida el correcto funcionamiento del componente de login en las siguientes funciones:

Escenarios de Pruebas Exitosos

- Renderizado de la interfaz: Se verificó que el formulario de login se renderiza de manera correcta.
- Inicio de sesión de administrador: Se probó que un usuario con credenciales de administrador puede iniciar sesión exitosamente y redirigir a la página correspondiente.
- Inicio de sesión de personal: Se validó que un usuario con credenciales de personal puede iniciar sesión y ser redirigido a su página.
- Credenciales incorrectas: Se confirmó que el componente maneja correctamente el caso de credenciales incorrectas, mostrando la alerta esperada (response: { status: 401 }) y sin fallar la prueba.

Figura 45

Prueba de Integración del Login

```
RERUN test/login.test.js x2
Archivo: Filename pattern: test/login.test.js
stderr | test/login.test.js > Login.vue > debería mostrar una alerta en caso de credenciales incorrectas
{ response: { status: 401 } }

✓ test/Login.test.js (4 tests) 61ms
  ✓ Login.vue > debería renderizar el formulario de login correctamente 32ms
  ✓ Login.vue > debería iniciar sesión correctamente para un Administrador 13ms
  ✓ Login.vue > debería iniciar sesión correctamente para un usuario Personal 6ms
  ✓ Login.vue > debería mostrar una alerta en caso de credenciales incorrectas 8ms

Test Files 1 passed (1)
Tests 4 passed (4)
Start at 19:25:50
Duration 308ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

El test del componente AdminDashboard resultó exitoso. El resultado PASS indica que las 14 pruebas definidas en el archivo test/adminDashboard.test.js se ejecutaron correctamente.

Esto confirma en la figura 46 que las siguientes funcionalidades operan como se espera:

Escenarios de Pruebas Exitosas

- Renderizado y rol de usuario: El componente se renderiza correctamente y muestra el nombre y el rol del usuario.
- Redirección a login: El componente redirige a la página de login si no hay un usuario en el localStorage.
- Cerrar sesión: Se probó que cerrar la sesión redirige al usuario a la página de login.
- Navegación: El usuario puede navegar a las rutas para registrar un nuevo usuario o patrocinador, y para crear un nuevo participante.
- Funcionalidades de búsqueda: Se validó que el componente puede buscar participantes de forma correcta.
- Descarga de documentos: El componente permite descargar el PDF de un participante seleccionado.

- Carga de datos: Se verificó que el componente carga la lista de usuarios al abrir el modal.
- Edición y eliminación: El componente permite editar, guardar y eliminar usuarios y patrocinadores.

Figura 46

Prueba de Integración del Panel del Administrador

```
RERUN test/adminDashboard.test.js
Filename pattern: test/adminDashboard.test.js

stdout | test/adminDashboard.test.js > AdminDashboard.vue > debería permitir editar y guardar un usuario
Cambios de usuario guardados en la base de datos

stdout | test/adminDashboard.test.js > AdminDashboard.vue > debería permitir editar y guardar un patrocinador
Cambios de patrocinador guardados en la base de datos

stdout | test/adminDashboard.test.js > AdminDashboard.vue > debería eliminar un usuario
Solicitud de eliminación de usuario para el ID: 1

stdout | test/adminDashboard.test.js > AdminDashboard.vue > debería eliminar un usuario
Usuario eliminado de la base de datos

stdout | test/adminDashboard.test.js > AdminDashboard.vue > debería eliminar un patrocinador
Patrocinador eliminado de la base de datos

✓ test/adminDashboard.test.js (14 tests) 154ms
  ✓ AdminDashboard.vue > debería renderizar y mostrar el nombre y rol del usuario 34ms
  ✓ AdminDashboard.vue > debería redirigir a /login si no hay usuario en localStorage 5ms
  ✓ AdminDashboard.vue > debería cerrar sesión y redirigir a /login 6ms
  ✓ AdminDashboard.vue > debería navegar a la ruta /registrar al hacer clic en "Registrar Nuevo Usuario" 5ms
  ✓ AdminDashboard.vue > debería navegar a la ruta /registrarPatrocinador al hacer clic en "Registrar Nuevo Patrocinador" 4ms
  ✓ AdminDashboard.vue > debería navegar a la ruta /nuevo-participante al hacer clic en "Nuevo Participante" 3ms
  ✓ AdminDashboard.vue > debería buscar participantes después de 300ms de escribir 3 o más caracteres 9ms
  ✓ AdminDashboard.vue > debería seleccionar un participante y cargar sus documentos si hay un solo resultado 10ms
  ✓ AdminDashboard.vue > debería descargar el PDF del participante seleccionado 11ms
  ✓ AdminDashboard.vue > debería cargar la lista de usuarios al abrir el modal 10ms
  ✓ AdminDashboard.vue > debería permitir editar y guardar un usuario 15ms
  ✓ AdminDashboard.vue > debería permitir editar y guardar un patrocinador 29ms
  ✓ AdminDashboard.vue > debería eliminar un usuario 7ms
  ✓ AdminDashboard.vue > debería eliminar un patrocinador 5ms

Test Files 1 passed (1)
Tests 14 passed (14)
Start at 20:13:12
Duration 420ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

El test del componente PersonalDashboard resultó exitoso. El resultado PASS indica que las 8 pruebas definidas en el archivo test/personalDashboard.test.js se ejecutaron correctamente como se puede observar en la figura 47.

Esto confirma que las siguientes funcionalidades operan como se espera:

Escenarios de Pruebas Exitosos

- Renderizado de la interfaz: Se verificó que el componente se renderiza correctamente, mostrando el nombre y el rol del usuario.

- Redirección a login: El componente redirige a la página de login si no hay un usuario en el localStorage.
- Cerrar sesión: Se probó que al cerrar la sesión se redirige al usuario a la página de login.
- Navegación: El usuario puede navegar a la ruta para registrar un nuevo participante.
- Búsqueda de participantes: Se validó que el componente busca participantes de forma correcta, tanto al escribir 3 o más caracteres como al no buscar si la cadena de búsqueda es menor.
- Selección y descarga: Se verificó que el componente puede seleccionar un participante, cargar sus documentos y descargar un PDF específico.

Figura 47

Prueba de Integración del Panel del Personal

```
c:\laragon\www\tukuy pacha 2\front(main -> origin)
λ npx vitest test/personalDashboard.test.js

DEV v3.2.4 C:/laragon/www/tukuy pacha 2/front

✓ test/personalDashboard.test.js (8 tests) 94ms
  ✓ PersonalDashboard.vue > debería renderizar y mostrar el nombre y rol del usuario 37ms
  ✓ PersonalDashboard.vue > debería redirigir a /login si no hay usuario en localStorage 6ms
  ✓ PersonalDashboard.vue > debería cerrar sesión y redirigir a /login 7ms
  ✓ PersonalDashboard.vue > debería navegar a la ruta /nuevo-participante 5ms
  ✓ PersonalDashboard.vue > debería buscar participantes después de escribir 3 o más caracteres 16ms
  ✓ PersonalDashboard.vue > no debería buscar participantes si la cadena de búsqueda es menor a 3 caracteres 4ms
  ✓ PersonalDashboard.vue > debería seleccionar un participante y cargar sus documentos si hay un solo resultado 10ms
  ✓ PersonalDashboard.vue > debería descargar el PDF del participante seleccionado 7ms

Test Files 1 passed (1)
Tests 8 passed (8)
Start at 20:21:30
Duration 1.54s (transform 242ms, setup 0ms, collect 474ms, tests 94ms, environment 588ms, prepare 128ms)

PASS Waiting for file changes...
press h to show help, press q to quit
```

El informe de los tests del componente NuevoParticipante indica un éxito rotundo. El resultado PASS significa que las 17 pruebas definidas en el archivo test/nuevoParticipante.test.js se ejecutaron sin errores como se puede observar en la figura 48.

Los resultados verifican la correcta operación de las siguientes funcionalidades:

Escenarios de Pruebas Exitosas

- Interfaz de usuario: El formulario se renderiza correctamente con campos vacíos.
- Redirección: Al hacer clic en "Cancelar", el usuario es redirigido a /AdminDashboard.
- Validación de campos:
 - El formato de los campos código y celular se valida correctamente, mostrando errores cuando el formato es inválido.
 - El campo dirección se valida correctamente, asegurando que se incluya un departamento boliviano.
- Patrocinadores: El componente permite buscar patrocinadores, seleccionar uno para completar los campos y limpiar la selección.
- Documentos: Se pueden añadir y eliminar campos para documentos, y el componente muestra una alerta si se sube un tipo de archivo inválido.
- Envío de formulario: El componente registra exitosamente un participante con sus documentos y maneja los errores de la API mostrando un mensaje al usuario.

Figura 48

Prueba de Integración al Registrar Participante

```
RERUN src/views/NuevoParticipante.vue x15
Filename pattern: test/nuevoParticipante.test.js

stderr | test/nuevoParticipante.test.js > NuevoParticipante.vue > debería mostrar un mensaje de error si la API falla
Error al registrar participante o documentos: { response: { data: { mensaje: 'Error de servidor' } } }

✓ test/nuevoParticipante.test.js (17 tests) 289ms
  ✓ NuevoParticipante.vue > debería renderizar el formulario con campos vacíos 54ms
  ✓ NuevoParticipante.vue > debería redirigir a /adminDashboard al hacer clic en Cancelar 17ms
  ✓ NuevoParticipante.vue > debería mostrar un error si el código tiene un formato inválido 17ms
  ✓ NuevoParticipante.vue > debería validar el código correctamente para un formato válido 12ms
  ✓ NuevoParticipante.vue > debería mostrar un error si la dirección no incluye un departamento boliviano 13ms
  ✓ NuevoParticipante.vue > debería validar la dirección correctamente si incluye un departamento boliviano 10ms
  ✓ NuevoParticipante.vue > debería mostrar un error si el celular tiene un formato inválido 13ms
  ✓ NuevoParticipante.vue > debería validar el celular correctamente para un formato válido 15ms
  ✓ NuevoParticipante.vue > debería buscar patrocinadores y mostrar los resultados 18ms
  ✓ NuevoParticipante.vue > debería completar los campos de patrocinador al seleccionar uno 15ms
  ✓ NuevoParticipante.vue > debería limpiar los campos de patrocinador al hacer clic en Limpiar 12ms
  ✓ NuevoParticipante.vue > debería añadir un nuevo campo de documento 11ms
  ✓ NuevoParticipante.vue > debería eliminar un campo de documento 17ms
  ✓ NuevoParticipante.vue > debería mostrar una alerta si se sube un tipo de archivo de documento inválido 10ms
  ✓ NuevoParticipante.vue > debería mostrar una alerta si se sube un tipo de archivo de foto inválido 7ms
  ✓ NuevoParticipante.vue > debería registrar un participante y sus documentos con éxito 21ms
  ✓ NuevoParticipante.vue > debería mostrar un mensaje de error si la API falla 25ms

Test Files 1 passed (1)
Tests 17 passed (17)
Start at 20:42:44
Duration 788ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

De acuerdo a la figura 49, el reporte de la ejecución de tests para el componente EditarParticipante muestra un éxito total. El resultado PASS indica que las 22 pruebas definidas en el archivo test/editarParticipante.test.js se ejecutaron correctamente.

Los resultados detallan que las siguientes funcionalidades operan correctamente:

Escenarios de Pruebas Exitosos

- Estado de la interfaz: El componente muestra un mensaje de carga inicial y luego los datos del participante a editar.
- Validación de campos: Se validan los campos de código, nombre, apellidos, dirección y celular, mostrando los errores correctos para entradas inválidas y no mostrando errores para entradas válidas.

- Gestión de patrocinadores: El componente busca patrocinadores, muestra los resultados y permite seleccionar uno de la lista para llenar los campos o limpiar la selección.
- Gestión de documentos: Se pueden agregar y eliminar documentos existentes del participante.
- Funcionalidades de la API: Se probó que el componente actualiza al participante con éxito y muestra un mensaje de error si la actualización falla.
- Navegación: El usuario puede navegar al dashboard al cancelar la edición.

Figura 49

Prueba de Integración al Editar Participante

```
RERUN src/views/EditarParticipante.vue x8
Filename pattern: test/editarParticipante.test.js

stderr | test/editarParticipante.test.js > editarParticipante.vue > muestra error si la actualización falla
Error al actualizar participante o documentos: { response: { data: { error: 'Error de actualización' } } }

✓ test/editarParticipante.test.js (22 tests) 1200ms
  ✓ editarParticipante.vue > muestra un mensaje de carga inicial y luego los datos
  ✓ editarParticipante.vue > muestra error si el código es inválido
  ✓ editarParticipante.vue > no muestra error si el código es válido
  ✓ editarParticipante.vue > muestra error si el nombre del participante está vacío
  ✓ editarParticipante.vue > no muestra error si el nombre del participante es válido
  ✓ editarParticipante.vue > muestra error si el apellido paterno del participante está vacío
  ✓ editarParticipante.vue > no muestra error si el apellido paterno del participante es válido
  ✓ editarParticipante.vue > muestra error si el apellido materno del participante está vacío
  ✓ editarParticipante.vue > no muestra error si el apellido materno del participante es válido
  ✓ editarParticipante.vue > muestra error si la dirección no contiene departamento válido
  ✓ editarParticipante.vue > no muestra error si la dirección contiene departamento válido
  ✓ editarParticipante.vue > muestra error si el celular es inválido
  ✓ editarParticipante.vue > no muestra error si el celular es válido
  ✓ editarParticipante.vue > actualiza participante con éxito
  ✓ editarParticipante.vue > muestra error si la actualización falla
  ✓ editarParticipante.vue > realiza una búsqueda de patrocinadores y muestra los resultados 300ms
  ✓ editarParticipante.vue > selecciona un patrocinador de la lista y rellena los campos 300ms
  ✓ editarParticipante.vue > muestra mensaje si no hay patrocinadores 300ms
  ✓ editarParticipante.vue > limpia los campos de patrocinador al hacer clic en el botón "Limpiar" 300ms
  ✓ editarParticipante.vue > agrega un documento
  ✓ editarParticipante.vue > elimina documento existente
  ✓ editarParticipante.vue > navega al dashboard al cancelar

Test Files 1 passed (1)
  Tests 22 passed (22)
Start at 23:56:30
Duration 1.20s

PASS Waiting for file changes...
press h to show help, press q to quit
```

La ejecución del test para el componente de registro fue exitosa. El resultado PASS indica que las 9 pruebas definidas en el archivo test/registrar.test.js se ejecutaron correctamente como se muestra en la figura 50.

Los hallazgos de las pruebas indican que el componente se comporta de manera esperada en las siguientes situaciones:

Escenarios de Pruebas Exitosos

- Renderizado: El formulario de registro se renderiza correctamente.
- Registro exitoso: El formulario registra un usuario con éxito y luego limpia todos los campos.
- Manejo de errores: El componente muestra un mensaje de error si el registro falla en el servidor.
- Navegación: Al hacer clic en el botón "Cancelar", el usuario navega de regreso al AdminDashboard.
- Validación de campos: El formulario no se envía si el nombre, apellido paterno o apellido materno están vacíos.
- Validación de formato: El formulario no se envía si el correo es inválido.
- Validación de rol: El formulario no se envía si el rol no ha sido seleccionado.

Figura 50

Prueba de Integración al Registrar Nuevo Usuario

```
RERUN test/registrar.test.js x10
Filename pattern: test/registrar.test.js

stderr | test/registrar.test.js > Registrar.vue > muestra un mensaje de error si el registro falla
{ response: { data: { error: 'Error del servidor' } } }

✓ test/registrar.test.js (9 tests) 123ms
✓ Registrar.vue > renderiza el formulario de registro correctamente 48ms
✓ Registrar.vue > registra un usuario con éxito y limpia los campos 20ms
✓ Registrar.vue > muestra un mensaje de error si el registro falla 12ms
✓ Registrar.vue > navega a adminDashboard al hacer clic en el botón Cancelar 5ms
✓ Registrar.vue > no envía el formulario si el nombre está vacío 7ms
✓ Registrar.vue > no envía el formulario si el apellido paterno está vacío 6ms
✓ Registrar.vue > envía el formulario con éxito si el apellido materno está vacío 9ms
✓ Registrar.vue > no envía el formulario si el correo es inválido 9ms
✓ Registrar.vue > no envía el formulario si el rol no está seleccionado (aunque el valor por defecto lo evita) 6ms

Test Files 1 passed (1)
Tests 9 passed (9)
Start at 00:38:22
Duration 418ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

La ejecución del test para el componente RegistrarPatrocinador fue un éxito. El resultado PASS indica que las 7 pruebas definidas en el archivo test/registrarPatrocinador.test.js se ejecutaron sin errores como se demuestra en la figura 51.

Los resultados de las pruebas confirman que el componente funciona como se espera en las siguientes situaciones:

Escenarios de Pruebas Exitosos

- Renderizado y registro: El formulario se renderiza correctamente y, al ser completado, registra un patrocinador con éxito y limpia los campos.
- Camposopcionales: El formulario permite que campos como el apellido paterno, materno, correo y celular queden vacíos sin impedir el registro.
- Validación:
 - No se envía el formulario si el campo de nombre está vacío.
 - Muestra un mensaje de error si el formato del correo es inválido.
- Manejo de errores: El componente muestra un mensaje de error al usuario si el registro falla en el servidor.
- Navegación: Al hacer clic en el botón "Cancelar", el usuario navega de regreso al AdminDashboard.

Figura 51

Prueba de Integración al Registrar Nuevo Patrocinador

```
RERUN test/registrarPatrocinador.test.js xii
Filename pattern: test/registrarPatrocinador.test.js

stderr | test/registrarPatrocinador.test.js > RegistrarPatrocinador.vue > muestra mensaje de error si el registro falla
{ response: { data: { error: 'Error del servidor' } } }

✓ test/registrarPatrocinador.test.js (7 tests) 108ms
  ✓ RegistrarPatrocinador.vue > renderiza el formulario correctamente 45ms
  ✓ RegistrarPatrocinador.vue > registra un patrocinador con éxito y limpia los campos 21ms
  ✓ RegistrarPatrocinador.vue > permite campos opcionales vacíos (apellido paterno, materno, correo, celular) 7ms
  ✓ RegistrarPatrocinador.vue > muestra mensaje de error si el registro falla 11ms
  ✓ RegistrarPatrocinador.vue > no envía el formulario si el nombre está vacío 8ms
  ✓ RegistrarPatrocinador.vue > muestra un mensaje de error si el correo es inválido 7ms
  ✓ RegistrarPatrocinador.vue > navega a adminDashboard al hacer clic en el botón Cancelar 8ms

Test Files 1 passed (1)
Tests 7 passed (7)
Start at 01:13:18
Duration 403ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

4.5.1.2. Prueba de seguridad contra inyecciones SQL

Para evaluar la seguridad de la aplicación frente a ataques de inyección SQL, se utilizó la herramienta ZAP by Checkmarx, anteriormente conocida como OWASP ZAP (Zed Attack Proxy). Esta herramienta es un escáner de vulnerabilidades gratuito y de código abierto, diseñado específicamente para detectar debilidades en aplicaciones web.

El objetivo principal fue identificar posibles vectores de ataque por **inyección SQL**, así como otras vulnerabilidades relacionadas con la entrada de datos del usuario.

La herramienta fue configurada para escanear de forma activa el entorno local de desarrollo (<http://localhost:5173>) y evaluó múltiples tipos de ataques, como se puede observar en la figura 52, incluidos:

- SQL Injection (MySQL, Oracle, PostgreSQL, SQLite, MsSQL)
- XSS (Cross Site Scripting)
- Path Traversal
- Command Injection
- Directory Browsing
- Server Side Template Injection
- Información filtrada en archivos .env, .htaccess, entre otros.

Resultados:

- Se realizaron **2,029 solicitudes** durante el análisis.
- Se evaluaron más de **30 vectores diferentes de ataque SQL**, incluyendo motores como MySQL, Oracle y SQLite.
- **No se detectaron vulnerabilidades de tipo SQL Injection**, lo cual demuestra una implementación segura del manejo de consultas a la base de datos.
- Solo se reportaron **4 alertas leves**, relacionadas con exposición de archivos (.env, archivos ocultos) y configuración de seguridad.
- La prueba concluyó en aproximadamente **33 segundos**, con un estado **positivo para todas las pruebas SQL**.

Figura 52

Progreso del escaneo de seguridad realizado con ZAP by Checkmarx

Progress Response Chart						
Host	Strength	Progress	Elapsed	Reqs	Alerts	Status
Analyser			00:00:390	68		
Plugin						
Path Traversal	Medium		00:00:006	0	0	✓
Remote File Inclusion	Medium		00:00:002	0	0	✓
Heartbleed OpenSSL Vulnerability	Medium		00:00:002	0	0	✓
Source Code Disclosure - /WEB-INF Folder	Medium		00:00:255	7	0	✓
Source Code Disclosure - CVE-2012-1823	Medium		00:00:255	0	0	✓
Remote Code Execution - CVE-2012-1823	Medium		00:00:005	2	0	✓
External Redirect	Medium		00:00:002	0	0	✓
Server Side Include	Medium		00:00:002	0	0	✓
Cross Site Scripting (Reflected)	Medium		00:00:002	0	0	✓
Cross Site Scripting (Persistent) - Prime	Medium		00:00:001	0	0	✓
Cross Site Scripting (Persistent) - Spider	Medium		00:00:007	1	0	✓
Cross Site Scripting (Persistent)	Medium		00:00:002	0	0	✓
SQL Injection	Medium		00:00:002	0	0	✓
SQL Injection - MySQL	Medium		00:00:001	0	0	✓
SQL Injection - Hyperersonic SQL	Medium		00:00:002	0	0	✓
SQL Injection - Oracle	Medium		00:00:002	0	0	✓
SQL Injection - PostgreSQL	Medium		00:00:002	0	0	✓
SQL Injection - SQLite	Medium		00:00:001	0	0	✓
Cross Site Scripting (DOM Based)	Medium		00:31:651	1871	0	✓
SQL Injection - MsSQL	Medium		00:00:002	0	0	✓
Log4Shell	Medium		00:00:001	0	0	✗
Spring4Shell	Medium		00:00:005	2	0	✓
Server Side Code Injection	Medium		00:00:002	0	0	✓
Remote OS Command Injection	Medium		00:00:000	0	0	✓
XPath Injection	Medium		00:00:001	0	0	✓
XML External Entity Attack	Medium		00:00:001	0	0	✓
Generic Padding Oracle	Medium		00:00:001	0	0	✓
Cloud Metadata Potentially Exposed	Medium		00:00:031	9	0	✓
Server Side Template Injection	Medium		00:00:032	0	0	✓
Server Side Template Injection (Blind)	Medium		00:00:001	0	0	✓
Directory Browsing	Medium		00:00:003	1	0	✓
Buffer Overflow	Medium		00:00:001	0	0	✓
		Copy	Close			
Format String Error	Medium		00:00:001	0	0	✓
CRLF Injection	Medium		00:00:000	0	0	✓
Parameter Tampering	Medium		00:00:001	0	0	✓
ELMAH Information Leak	Medium		00:00:004	1	0	✓
Trace.axd Information Leak	Medium		00:00:009	1	0	✓
.htaccess Information Leak	Medium		00:00:005	1	0	✓
.env Information Leak	Medium		00:00:004	1	0	✓
Hidden File Finder	Medium		00:00:209	50	4	✓
Spring Actuator Information Leak	Medium		00:00:008	2	0	✓
XSLT Injection	Medium		00:00:197	0	0	✓
GET for POST	Medium		00:00:001	0	0	✓
User Agent Fuzzer	Medium		00:00:047	12	0	✓
Script Active Scan Rules	Medium		00:00:001	0	0	✗
SOAP Action Spoofing	Medium		00:00:001	0	0	✓
SOAP XML Injection	Medium		00:00:001	0	0	✓
Totals			00:33:441	2029	4	
		Copy	Close			

Tabla 8

*Tipos de ataques evaluados durante la prueba de seguridad con OWASP ZAP*Ataques evaluados

Ataque / Prueba	Descripción resumida
Path Traversal	Intenta acceder a archivos fuera del directorio permitido mediante rutas como ../.
Remote File Inclusion	Busca incluir archivos remotos maliciosos dentro del servidor.
Heartbleed OpenSSL Vulnerability	Explota una vulnerabilidad crítica en OpenSSL para obtener datos sensibles del servidor.
Source Code Disclosure - /WEB-INF Folder	Trata de acceder a carpetas internas como /WEB-INF que contienen lógica confidencial.
Source Code Disclosure - CVE-2012-1823	Intenta explotar una vulnerabilidad específica en PHP para mostrar código fuente.
Remote Code Execution - CVE-2012-1823	Busca ejecutar código remoto en el servidor mediante una vulnerabilidad conocida.
External Redirect	Comprueba si el servidor permite redirecciones a sitios externos, facilitando phishing.
Server Side Include	Intenta ejecutar comandos a través de inclusiones del lado del servidor (<!-- #include-->).
Cross Site Scripting (Reflected)	Inyecta scripts que se reflejan directamente en las respuestas del servidor.

Cross Site Scripting (Persistent - Prime, Spider, etc.)	Inyecta scripts que se almacenan en el servidor y se ejecutan al visitar ciertas páginas.
SQL Injection (Genérico y por motores: MySQL, Oracle, MSSQL, PostgreSQL, SQLite, Hypersonic SQL)	Inyecta código SQL malicioso en formularios o parámetros para acceder o modificar bases de datos.
Cross Site Scripting (DOM Based)	Inyecta scripts aprovechando la manipulación del DOM en el navegador.
Log4Shell	Explota una vulnerabilidad crítica en bibliotecas Log4j para ejecutar código remoto.
Spring4Shell	Explota una vulnerabilidad en aplicaciones Spring para ejecutar comandos maliciosos.
Server Side Code Injection	Intenta ejecutar código arbitrario directamente en el servidor.
Remote OS Command Injection	Busca ejecutar comandos del sistema operativo mediante entradas del usuario.
XPath Injection	Inyecta código malicioso en consultas XPath (usadas en XML).
XML External Entity (XXE) Attack	Inyecta entidades externas maliciosas en documentos XML.
Generic Padding Oracle	Prueba si el sistema es vulnerable a ataques criptoanalíticos tipo padding oracle.
Cloud Metadata Potentially Exposed	Busca acceso a metadatos de servicios en la nube como AWS, Azure, etc.

Server Side Template Injection (Blind y visible)	Inyecta código en plantillas de servidor como Ninja, Thymeleaf o EJS.
Directory Browsing	Verifica si el servidor permite ver el contenido de directorios.
Buffer Overflow	Envía datos excesivos con la intención de sobrecargar la memoria del sistema.
Format String Error	Prueba errores en el manejo de cadenas de formato (%s, %d, etc.) que pueden generar fallos.
CRLF Injection	Inyecta caracteres de nueva línea (\r\n) para manipular encabezados HTTP.
Parameter Tampering	Modifica parámetros para alterar el comportamiento de funciones.
ELMAH Information Leak	Intenta acceder a registros de errores de ELMAH (ASP.NET).
Trace.axd Information Leak	Prueba si está habilitada la traza de solicitudes en ASP.NET (trace.axd).
.htaccess Information Leak	Busca archivos .htaccess que revelen reglas del servidor.
.env Information Leak	Intenta obtener archivos .env que contienen variables sensibles como claves API.
Hidden File Finder	Detecta archivos ocultos que podrían contener información confidencial.
Spring Actuator Information Leak	Verifica si endpoints de Spring Boot están expuestos sin seguridad.

XSLT Injection	Inyecta código en transformaciones XSLT para ejecutar instrucciones peligrosas.
GET for POST	Comprueba si solicitudes POST también responden a solicitudes GET (riesgo de exposición).
User Agent Fuzzer	Modifica el encabezado User-Agent para detectar errores en el tratamiento de agentes.
Script Active Scan Rules	Ejecuta scripts personalizados para probar múltiples vectores de ataque
SOAP Action Spoofing	Manipula acciones SOAP para probar seguridad de servicios web.
SOAP XML Injection	Inyecta estructuras XML maliciosas en mensajes SOAP.

Tabla 9*Pruebas de seguridad*

Medida de Seguridad	Descripción	Verificación Técnica	Resultado
Autenticación y control de acceso	Se utiliza JWT (JSON Web Tokens) para gestionar sesiones y restringir el acceso a rutas protegidas según el rol del usuario (Admin / Personal).	Se verificó la presencia del token JWT en el almacenamiento local y el correcto redireccionamiento de rutas según rol. El backend protege rutas con middleware.	Cumplido

Cifrado de contraseñas	Las contraseñas de los usuarios son cifradas antes de ser almacenadas en la base de datos.	Se inspeccionó directamente la base de datos y se comprobó que las contraseñas están cifradas (hash bcrypt). No se almacenan en texto plano.	Cumplido
Protección contra inyección SQL	Las consultas SQL se realizan utilizando parámetros preparados.	Se revisó el código backend en los controladores. Las consultas hacia la base de datos utilizan pool.query(..., [param]) evitando así la inserción directa de datos.	Cumplido
Validación de entradas	Los formularios validan que los datos requeridos sean correctos antes de enviarlos al servidor.	Se realizaron pruebas con entradas erróneas, vacías y scripts maliciosos. El sistema impide el registro y muestra mensajes de error apropiados.	Cumplido
Seguridad en el enrutamiento (frontend)	Se impide que usuarios no autenticados accedan a rutas restringidas en la interfaz de usuario.	Se validó que el router.beforeEach redirige a /login si no hay token o si el rol no es válido para la ruta requerida.	Cumplido
Restricción de funciones por rol	Los usuarios sólo pueden acceder a	Se realizó una prueba de inicio de sesión con	Cumplido

	funcionalidades específicas según su rol.	distintos usuarios y se comprobó que los botones y rutas no autorizadas se ocultan o redirigen correctamente.	
Gestión de archivos segura	Los documentos y fotos subidos se almacenan en un directorio específico y son servidos de forma controlada desde el servidor.	Se revisó que el middleware de multer limite el tipo de archivos permitidos y que las rutas de acceso sean protegidas correctamente.	Cumplido

4.6. Planificación

El desarrollo del sistema siguió un esquema iterativo basado en la metodología de prototipos, que permitió ajustar los requerimientos a medida que se obtenía retroalimentación del cliente. La planificación usa el modelo Prototipado Evolutivo, se organizó en las siguientes fases:

Tabla 10*Esquema Basado en la Metodología de Prototipos*

Fase	Actividades clave
Fase 1: Análisis y Diseño	Recolección de requerimientos, diseño de prototipos
Fase 2: Desarrollo Inicial	Implementación del backend y estructura básica del frontend
Fase 3: Validación de Prototipo	Pruebas preliminares con usuarios, ajustes
Fase 4: Desarrollo Avanzado	Integración de funcionalidades completas
Fase 5: Pruebas y Seguridad	Pruebas de rendimiento, usabilidad y seguridad

4.6.1. Fases del Desarrollo del Proyecto

4.6.1.1. Fase Inicial: Escuchar al Cliente y Recolección de Requisitos

En esta fase crucial, se establece una comunicación directa y constante con la Asociación Tukuy Pacha. El objetivo principal es obtener una comprensión profunda de sus procesos de gestión documental actuales, identificar los desafíos que enfrentan y definir claramente los objetivos que buscan alcanzar con la implementación del nuevo sistema. Se emplean diversas técnicas de recopilación de información, incluyendo:

- **Entrevistas:** Se realizan entrevistas estructuradas y semiestructuradas con el personal clave de la asociación para profundizar en sus necesidades específicas y obtener información detallada sobre sus flujos de trabajo.
- **Encuestas:** Se distribuyen encuestas para recopilar información de un grupo más amplio de usuarios, permitiendo obtener una visión general de sus requerimientos y preferencias.

- **Observación Directa:** Se observan los procesos de trabajo diarios de la asociación para comprender cómo se gestionan actualmente los documentos y dónde se presentan los cuellos de botella.
- **Análisis de Documentos:** Se revisan los documentos existentes de la asociación (formularios, registros, etc.) para identificar la información que debe ser gestionada por el sistema.

Se identifican y documentan tanto los **requerimientos funcionales** (las acciones que el sistema debe realizar, como registrar participantes, buscar documentos, generar informes) como los **requerimientos no funcionales** (las cualidades que debe tener el sistema, como usabilidad, seguridad, rendimiento, escalabilidad). Se presta especial atención a la identificación de las áreas que requieren una especificación más detallada en las iteraciones posteriores.

Actividades: Identificar objetivos generales, recopilar requisitos iniciales y definir áreas que necesitan especificación.

Resultado: Comprensión clara de las necesidades del cliente y primeros requisitos documentados.

- **Iteración 1: Diseño Rápido Y Construcción**

Con los requisitos iniciales documentados, se inicia la primera iteración de desarrollo, cuyo objetivo principal es crear un prototipo inicial del sistema. Este prototipo no es una versión completa y final, sino una maqueta funcional que permite a los usuarios visualizar y experimentar con las principales funcionalidades e interfaces de usuario. Se prioriza la rapidez en la construcción del prototipo para facilitar la obtención de retroalimentación temprana y minimizar el tiempo de desarrollo inicial. Se seleccionan herramientas y tecnologías de desarrollo que permitan una rápida creación y modificación del prototipo, favoreciendo la agilidad y la flexibilidad. El enfoque se centra en representar las funciones esenciales del sistema, como la gestión de participantes, la búsqueda de documentos y la generación de informes básicos.

Actividades: Crear un prototipo inicial que refleje las funcionalidades principales del sistema.

Resultado: Primera versión del prototipo con funciones básicas.

- **Iteración 2: Evaluación y Ajustes Iniciales**

El prototipo inicial se presenta a los usuarios clave de la Asociación Tukuy Pacha para su evaluación y obtención de retroalimentación. Se organizan sesiones de demostración y pruebas donde los usuarios pueden interactuar directamente con el prototipo, realizar tareas comunes y expresar sus opiniones sobre su usabilidad, funcionalidad y cumplimiento de los requisitos. Se fomenta la participación activa de los usuarios y se les anima a proporcionar comentarios detallados y constructivos. Se registran cuidadosamente todas las observaciones, sugerencias, críticas y problemas identificados por los usuarios. Esta retroalimentación se analiza y se utiliza para realizar ajustes y mejoras en el prototipo, corrigiendo errores, modificando el diseño de la interfaz y adaptando las funcionalidades según las necesidades y preferencias de los usuarios.

Actividades: Presentar el prototipo inicial al cliente para obtener retroalimentación y realizar los ajustes necesarios.

Resultado: Prototipo ajustado con mejoras basadas en las observaciones del cliente.

- **Iteración 3: Refinamiento y Extensión de Funcionalidades**

En esta iteración, se continúa refinando y mejorando el prototipo, incorporando funcionalidades adicionales y optimizando las existentes según la retroalimentación recibida en la iteración anterior. Se profundiza en el desarrollo de las características más complejas del sistema, se mejora la interfaz de usuario para hacerla más intuitiva y amigable, y se optimiza el rendimiento y la eficiencia del prototipo. Se realizan pruebas más exhaustivas para asegurar la calidad y estabilidad del prototipo. Se mantiene una comunicación constante con la Asociación Tukuy Pacha para asegurar que el

desarrollo se alinea con sus expectativas y necesidades. El resultado es un prototipo avanzado que representa una versión más completa y funcional del sistema, acercándose a la versión final.

Actividades: Incorporar funcionalidades adicionales y optimizar las existentes según los comentarios del cliente.

Resultado: Prototipo avanzado que representa una versión más completa del sistema.

4.6.1.2. Fase Final: Desarrollo Completo e Implementación

Una vez que el prototipo ha sido validado y aprobado por la Asociación Tukuy Pacha, se procede al desarrollo completo del sistema final. Se construye el sistema completo, incluyendo todas las funcionalidades y características definidas en los requisitos y refinadas durante las iteraciones anteriores. Se optimiza el sistema para garantizar su rendimiento, seguridad, escalabilidad y mantenibilidad. Se realizan pruebas exhaustivas de todo el sistema (pruebas de integración, pruebas de sistema, pruebas de aceptación) para asegurar su correcto funcionamiento y la ausencia de errores. Se prepara el entorno de producción para la implementación del sistema. Se realiza la migración de los datos existentes de la Asociación Tukuy Pacha al nuevo sistema. Se capacita al personal de la asociación en el uso del sistema, proporcionándoles la formación y el apoyo necesarios para su adopción exitosa. Se pone en marcha el sistema en el entorno real de la asociación y se brinda soporte técnico y seguimiento durante un período de tiempo para asegurar su correcto funcionamiento y resolver cualquier problema que pueda surgir. Se recopila retroalimentación continua de los usuarios para realizar mejoras y mantenimiento del sistema a largo plazo.

Actividades: Convertir el prototipo final en el sistema operativo completo, realizar pruebas exhaustivas y capacitar al personal en su uso.

Resultado: Sistema implementado, funcional y alineado con las expectativas del cliente.

Conclusiones

Se concluye con el cumplimiento de los objetivos planteados en el presente proyecto, desarrollando un Sistema de Gestión Documental Web, cuyo propósito es mejorar los procesos de registro, almacenamiento, consulta y administración de documentos digitales, aportando a la optimización del flujo de información interna y preservación de documentos importantes.

Se puede concluir que el **objetivo general** fue alcanzado, logrando diseñar y desarrollar el Sistema de Gestión Documental, lo cual ha permitido organizar la información documental de manera centralizada, brindar un acceso más eficiente a los archivos y reducir el uso de documentos físicos.

En cuanto a los objetivos específicos, se justifica el cumplimiento de cada uno de ellos en los siguientes párrafos:

Los procesos manuales existentes en la institución fueron identificados y analizados, particularmente aquellos relacionados con el manejo documental, lo que permitió detectar deficiencias como pérdida de documentos, dificultad de acceso.

La arquitectura del sistema fue diseñada tomando en cuenta criterios de funcionalidad, escalabilidad y simplicidad, lo cual resultó en una estructura clara y adaptada a las necesidades del entorno institucional.

Los módulos necesarios fueron desarrollados para registrar, almacenar, visualizar y consultar documentos, además de gestionar usuarios y establecer niveles de acceso, contribuyendo así a una administración de la información más segura y eficiente.

El sistema incorpora una búsqueda avanzada **con múltiples filtros** (nombre, código, carnet de identidad, etc.), lo cual facilita la localización de documentos al permitir una recuperación de información precisa basada en los datos del participante.

El diseño e implementación del sistema informático sientan las bases para una significativa optimización de los procesos operativos. Se proyecta que la digitalización de los documentos reducirá el consumo de papel y el espacio físico de almacenamiento, lo que generará un **ahorro** económico a largo plazo. Además, las funcionalidades de búsqueda y recuperación **eficiente** de la información permitirán mejorar la productividad del personal, liberándolo de la gestión manual para que pueda enfocarse en tareas de mayor valor estratégico.

El sistema cumple con los principios de seguridad y privacidad necesarios, al implementar una estrategia de seguridad robusta. Las medidas incluyen la autenticación segura con JWT, la protección contra vulnerabilidades comunes como la inyección SQL, y una validación integral a través de pruebas de integración para asegurar la estabilidad del sistema. Estas pruebas demuestran que las medidas de seguridad funcionan como se espera.

En síntesis, el sistema desarrollado cumple con los requerimientos planteados, mejora sustancialmente los procesos documentales de la institución, y representa una solución tecnológica efectiva y adecuada al contexto real del entorno institucional.

Recomendaciones

A partir de los resultados obtenidos y el análisis realizado, se presentan las siguientes recomendaciones para la Asociación Tukuy Pacha, con el objetivo de optimizar su gestión documental y aprovechar al máximo el sistema implementado:

Migración del sistema a un servidor en línea: Actualmente, el sistema funciona en un entorno local, lo cual limita su disponibilidad, acceso remoto y escalabilidad. Se recomienda realizar la migración a un servidor en la nube para garantizar mayor estabilidad, respaldo automático y acceso desde múltiples ubicaciones.

Una opción recomendable es DigitalOcean, que ofrece instancias virtuales (droplets) de bajo costo, adecuadas para aplicaciones Node.js con bases de datos MySQL.

También se puede considerar Render o Heroku para una configuración más sencilla con integración continua.

En caso de requerir mayor seguridad y respaldo institucional, [Amazon Web Services (AWS)] con EC2 y RDS sería ideal, aunque su gestión técnica requiere mayor experiencia.

Implementación Completa y Adopción del Sistema: La Asociación Tukuy Pacha llevar a cabo la implementación completa del sistema de gestión documental en todas sus áreas y promover su adopción por parte de todo el personal. El soporte técnico necesario y asegurar que el sistema se integre de manera efectiva en los flujos de trabajo existentes. La adopción completa del sistema es crucial para maximizar sus beneficios y garantizar una transición exitosa hacia una gestión documental más eficiente.

Digitalización Sistématica de Documentos: Es fundamental establecer un plan de digitalización sistemática de los documentos físicos existentes. Esto permitirá reducir el espacio de almacenamiento físico, facilitar el acceso a la

información y mejorar la preservación de los documentos a largo plazo. Se recomienda priorizar la digitalización de los documentos más importantes y de mayor uso, y establecer un proceso claro para la digitalización, indexación y almacenamiento de los documentos digitales.

Definición de Políticas y Procedimientos de Gestión Documental: Se recomienda definir políticas y procedimientos claros y documentados para la gestión documental. Esto incluye establecer normas para la creación, almacenamiento, acceso, modificación, eliminación y conservación de los documentos, tanto físicos como digitales. La definición de estas políticas y procedimientos garantizará la consistencia y la estandarización en la gestión de la información.

Mantenimiento y Actualización del Sistema: Es recomendable realizar un mantenimiento y actualización periódicos del sistema de gestión documental para asegurar su correcto funcionamiento, su seguridad y su adaptación a las necesidades cambiantes de la asociación. Esto incluye la corrección de errores, la implementación de nuevas funcionalidades, la actualización de la tecnología y la realización de copias de seguridad de los datos.

Evaluación Periódica del Sistema: Aplicar evaluaciones periódicas del sistema de gestión documental para identificar posibles mejoras y asegurar que siga cumpliendo con las necesidades de la asociación. Esto puede incluir la recopilación de retroalimentación de los usuarios, el análisis de los indicadores de rendimiento del sistema y la revisión de las políticas y procedimientos de gestión documental.

Bibliografía

- Altamirano Bajaña, D. F., & Palma Murillo, D. E. (Marzo de 2018). *DSpace*. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/15572>
- Axmark, D., Larsson, A., & Widenius, M. (s.f.). *MySql*. Obtenido de <https://www.mysql.com/why-mysql/>
- Blanco, Y. C. (2018). *Aplicación web para gestionar información sobre personas discapacitadas*. Obtenido de SCIELO: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1028-99332018000400710#aff2
- Dahl, R. (s.f.). *Nodejs*. Obtenido de <https://nodejs.org/en/about>
- Gonzales, G. (2025). *PLAN ESTRATÉGICO INSTITUCIONAL*.
- Hollowaychuk, T. (s.f.). *geeksforgeeks*. Obtenido de <https://www.geeksforgeeks.org/express-js/>
- Khoa, L. (s.f.). Obtenido de <https://laragon.org/why-laragon/>
- Kong. (s.f.). *Insomnia*. Obtenido de <https://docs.insomnia.rest/insomnia/get-started>
- Microsoft. (s.f.). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com/>
- Miguel. (09 de 12 de 2015). *Qué es MySQL y cómo funciona*. Obtenido de MLO: <https://migueleonardortiz.com.ar/mysql/que-es-mysql-y-como-funciona/994>
- Mozilla. (s.f.). *Express Node.js: Routes*. Obtenido de MDN Web Docs: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/routes
- Otto, M., & Thornton, J. (s.f.). *Bootstrap*. Obtenido de <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

- Palacios, D. (12 de 04 de 2016). *Styde*. Obtenido de
<https://styde.net/introduccion-a-vue-js/>
- Pfleeger, C. P., & Pfleeger, S. L. (2002). *Security in Computing* (Tercera ed.). Prentice Hall.
- Pfleeger, S. L. (2002). *Ingeniería de software: teoría y práctica*. Pearson Education.
- Preston-Werner, T., Wanstrath, C., & Hyett, P. (s.f.). *GitHub*. Obtenido de
<https://github.com/>
- Sommerville, I. (2017). *Software Engineering* (Decima ed.). Pearson.
- Torvalds, L. (s.f.). *Git*. Obtenido de <https://git-scm.com/>
- Tukuy Pacha; Save the Children. (2024). “*PROCESOS DE INCLUSIÓN EDUCATIVA DE NIÑAS Y NIÑOS CON DISCAPACIDAD Y RETRASO EN EL DESARROLLO EN EL NIVEL INICIAL Y PRIMARIO*”. Cochabamba: Virmegraf.
- welearndata.com. (s.f.). *welearndata.com*. Obtenido de Git y GitHub – Conceptos Principales: <https://welearndata.com/git/git-github-conceptos-principales/>
- Widenius, M., Axmark, D., & Arno, K. (2002). *MySQL Reference Manual*. O'Reilly Media.
- You, E. (s.f.). *Vue.js*. Obtenido de <https://vuejs.org/guide/introduction.html>

ANEXOS

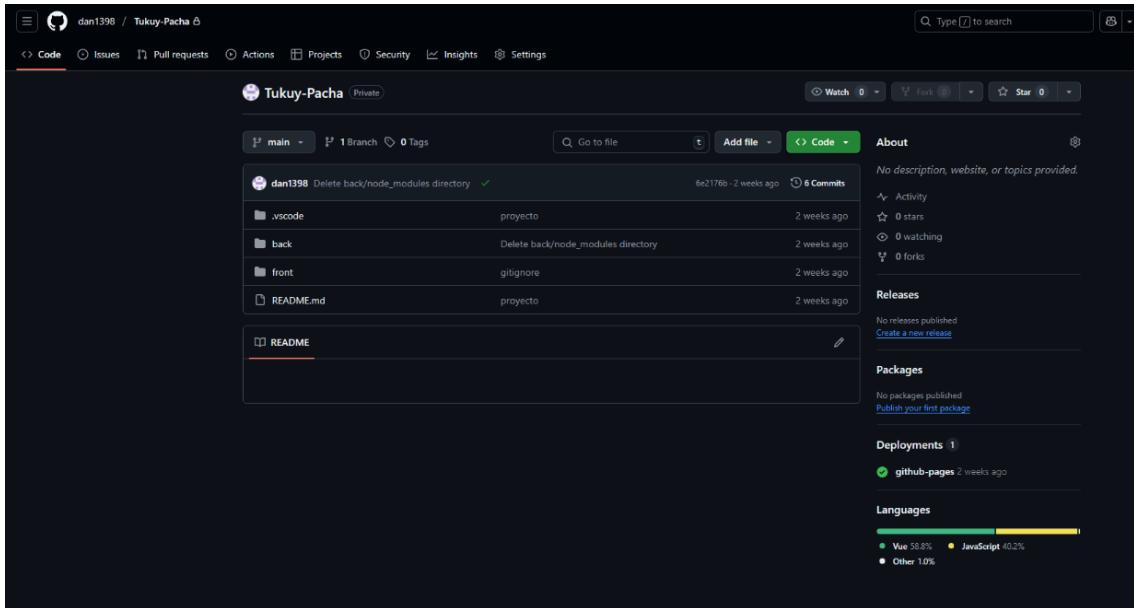
Anexos 1

Partipantes de la encuesta

NOMBRE COMPLETO	CARGO EN LA ASOCIACIÓN
Gianluca Scannapieco	Coordinador
Lizeth Turumaya Cuba	Trabajador de campo
António Yapu Alcocer	Trabajador de campo
Astrid Calle	Técnico de Campo
Angel Villarroel	Trabajador de campo
Patricia Coca Villarroel	Coordinador
Cinthya Isabel Cervantes Apaza	Trabajador Social
Richard Luis Aranda Fernandez	Trabajador de campo
Marlene Chávez Bautista	Trabajador de campo

Anexos 2

Repositorio de GitHub del proyecto Tukuy-Pacha



Anexos 3

Autenticación de Usuario con JWT y bcrypt en Node.js

```

1 import pool from '../db.js'
2 import bcrypt from 'bcrypt'
3 import jwt from 'jsonwebtoken'
4
5 export const login = async (req, res) => {
6   const { correo, contraseña } = req.body
7
8   try {
9     const [rows] = await pool.query('SELECT * FROM Usuario WHERE correo = ?', [correo])
10    if (rows.length === 0) return res.status(401).json({ mensaje: 'Usuario no encontrado' })
11
12    const usuario = rows[0]
13    const match = await bcrypt.compare(contraseña, usuario.contraseña)
14    if (!match) return res.status(401).json({ mensaje: 'Contraseña incorrecta' })
15
16    const token = jwt.sign({ id: usuario.id_usuario, rol: usuario.rol }, process.env.JWT_SECRET, { expiresIn: '8h' })
17    res.json({ token, usuario: { id: usuario.id_usuario, nombre: usuario.nombre, rol: usuario.rol } })
18  } catch (err) {
19    res.status(500).json({ mensaje: 'Error en el servidor', error: err.message })
20  }
21}

```

Nota, implementa un sistema de autenticación de usuario que verifica credenciales (correo y contraseña), compara la contraseña con bcrypt y genera un JSON Web Token (JWT) para la sesión.

Anexos 4

Visualización de Usuarios y Contraseñas Hasheadas con bcrypt en HeidiSQL

id_usuario	nombre	correo	contraseña	rol
3	celes	cel@gmail.com	\$2b\$10\$vp5W0xyQWPhdaYCb71jm.uYVPIQwiE7ANyiX.82oDgDT.IDt4Gxiu	Admin
4	Carlos	carlos@tukuy.bo	\$2b\$10\$ef51S1qgTlFKzU7C0t.6001ULDQOrg7iInMTxGMqpi0nJhaefka	Admin
5	danny	d1303gm@gmail.com	\$2b\$10\$npnpxQvLxBbd7014Yta6hU.aJ3Hg4DNld7W5MHWB/XNKZIQk0j95S	Personal
6	danny	danny@gmail.com	\$2b\$10\$SrFcYaObfOxQBbwCtvsy.uDgaHuTkfJh2yMze8F/AdubEtusui	Admin
7	Celeste	celeste@gmail.com	\$2b\$10\$G2AruVo.om9ThzVwFlDV7erpF/G.Vw0Y/6DgCPPk.iv4OdJtzxS/a	Personal

Nota, esta imagen muestra la tabla usuario de la base de datos tukuypacha en HeidiSQL, donde las contraseñas de los usuarios están hasheadas utilizando bcrypt.

Anexos 5

API REST para Obtener Participantes con Consulta SQL Segura en Node.js

```

1  export const getParticipantes = async (req, res) => {
2    try {
3      const [rows] = await pool.query('SELECT * FROM Participante')
4      res.json(rows)
5    } catch (err) {
6      res.status(500).json({ mensaje: 'Error al obtener participantes', error: err.message })
7    }
8  }

```

Nota, este código define una función asíncrona para obtener todos los participantes de una base de datos, manejando errores y utilizando una consulta SQL básica que, en este contexto, no es susceptible a inyecciones SQL directas al no incluir parámetros dinámicos de entrada.

Anexos 6

Navegación Protegida por Roles con Vue Router y Local Storage

```
1 router.beforeEach((to, from, next) => {
2     const token = localStorage.getItem('token')
3     const userData = localStorage.getItem('usuario')
4     const isAuthenticated = token && token !== 'undefined' && token.trim() !== ''
5
6     if (to.meta.requiresAuth && !isAuthenticated) {
7         next({ name: 'login' })
8     } else if (to.path === '/adminDashboard') {
9         if (!userData) return next({ name: 'login' })
10        const user = JSON.parse(userData)
11        if (user.rol === 'Admin') {
12            next()
13        } else {
14            next({ name: 'personalDashboard' })
15        }
16    } else if (to.path === '/personalDashboard') {
17        if (!userData) return next({ name: 'login' })
18        const user = JSON.parse(userData)
19        if (user.rol === 'Personal') {
20            next()
21        } else {
22            next({ name: 'adminDashboard' })
23        }
24    } else {
25        next()
26    }
27 })
```

Nota, se implementa un router.beforeEach para proteger rutas, redirigiendo a la página de login si la autenticación falla o el token no existe, y gestionando el acceso a diferentes paneles (admin/personal) según el rol del usuario almacenado en el Local Storage.