

# **Departamento de Ciencias de la Computación (DCCO)**

## **Carrera de Software**

**A&D Sw**

### **Perfil del Proyecto**

Presentado por: Mideros Samir, Miranda Alison, Morán

David, Vivanco Gabriel (GRUPO 4)

Tutor académico: Ing. Jenny A Ruiz R

Ciudad: Sangolquí

Fecha: 11/12/2025

# Índice

---

## PERFIL DE PROYECTO

<b>1. Introducción.....</b>	<b>5</b>
<b>2. Planteamiento del trabajo.....</b>	<b>5</b>
2.1 Formulación del problema.....	5
2.2 Justificación.....	6
<b>3. Sistema de Objetivos.....</b>	<b>6</b>
3.1. Objetivo General.....	6
3.2. Objetivos Específicos.....	7
<b>4. Alcance.....</b>	<b>7</b>
<b>5. Marco Teórico.....</b>	<b>8</b>
5.1 Metodología (Marco de trabajo 5W+2H).....	8
<b>6. Ideas a Defender.....</b>	<b>8</b>
<b>7. Resultados Esperados.....</b>	<b>10</b>
<b>8. Viabilidad(Ej.).....</b>	<b>10</b>
8.1 Humana.....	11
8.1.1 Tutor Empresarial.....	11
8.1.2 Tutor Académico.....	11
8.1.3 Estudiantes.....	12
8.2 Tecnológica.....	12
8.2.1 Hardware.....	12
8.2.2 Software.....	13
<b>9. Conclusiones y recomendaciones.....</b>	<b>13</b>
9.1 Conclusiones.....	13
9.2 Recomendaciones.....	14
<b>10. Planificación para el Cronograma:.....</b>	<b>15</b>
<b>11. Referencias.....</b>	<b>15</b>

# 1. Introducción

En la actualidad, la gestión eficiente de eventos masivos y académicos requiere de herramientas tecnológicas actuales que garanticen la fluidez en los procesos administrativos y la seguridad de la información. La transformación digital ha dejado de ser una opción para convertirse en una necesidad operativa, especialmente en tareas críticas como la recaudación y validación de pagos, donde la precisión y la rapidez son fundamentales para la satisfacción de los asistentes y la transparencia organizacional.

Sin embargo, a pesar de los avances tecnológicos, muchas organizaciones continúan gestionando el registro de asistentes y la validación de comprobantes de pago mediante procesos manuales o herramientas ofimáticas desconectadas (como hojas de cálculo y correos electrónicos). Esta metodología tradicional presenta serias limitaciones: provoca cuellos de botella operativos debido a la carga manual de datos, aumenta significativamente el riesgo de errores humanos al conciliar información bancaria y dificulta la detección oportuna de comprobantes duplicados o fraudulentos. Además, la falta de un repositorio centralizado impide a los organizadores tener una visión en tiempo real del estado financiero y de aforo del evento.

Para mitigar estas deficiencias, el presente proyecto propone el desarrollo e implementación de un Sistema Web de Gestión y Validación de Comprobantes para Eventos. Esta solución tecnológica está diseñada para automatizar el flujo de trabajo de la tesorería y la secretaría del evento. El sistema permitirá a los usuarios cargar sus comprobantes digitales en una plataforma segura, mientras que, del lado administrativo, ofrecerá herramientas para una validación ágil tanto automática como manual, gestión de estados de pago y emisión de notificaciones. Con ello, se busca no sólo optimizar los tiempos de respuesta y reducir la carga operativa del personal, sino también garantizar la integridad de los datos y ofrecer una experiencia de usuario moderna y confiable.

# 2. Planteamiento del trabajo

## 2.1 Formulación del problema

Actualmente, el proceso de gestión y validación de comprobantes en los eventos presenta una serie de limitaciones que afectan directamente la eficiencia, la transparencia y la seguridad de la operación. El registro de gastos del personal, la validación de documentos y el procesamiento de pagos se realizan de manera manual, utilizando mensajes por WhatsApp, hojas de Excel independientes y verificaciones aisladas entre departamentos. Este método no solo es lento, sino también altamente propenso a errores humanos, pérdidas de información y casos de fraude o suplantación.

Las soluciones actuales carecen de mecanismos de trazabilidad, validación automática y control centralizado. No existe un sistema que cruce datos oficiales del staff con los comprobantes ingresados, ni que verifique que la información declarada coincida con la evidencia del documento físico. Tampoco existe un flujo digital que indique al personal el estado real de sus comprobantes o pagos, generando incertidumbre, reprocesos y dependencia del personal administrativo para responder consultas.

Frente a estas limitaciones, el proyecto propone la implementación de un Sistema de Gestión y Validación de Comprobantes para Eventos, diseñado específicamente para digitalizar y automatizar todo el flujo operativo. La solución plantea integrar en un mismo sistema: el registro digital de comprobantes, la carga de documentos de respaldo, la validación automática en múltiples capas, el procesamiento ordenado de pagos y un historial integral de gastos.

De esta manera, el proyecto aborda el problema al ofrecer una plataforma que elimina los riesgos del manejo manual, reduce el tiempo de validación, garantiza transparencia para el personal, asegura trazabilidad para auditorías y mejora el control financiero del evento. La propuesta supera las limitaciones actuales mediante automatización, centralización de datos y controles de seguridad que no son posibles bajo el proceso manual vigente.

## **2.2 Justificación**

Este proyecto resulta relevante para la comunidad investigadora porque aborda un problema real de gestión administrativa mediante la aplicación de principios de ingeniería de software, automatización y validación de información. La propuesta demuestra cómo un sistema web puede mejorar la eficiencia, reducir errores y fortalecer la transparencia en procesos que tradicionalmente se ejecutan de forma manual.

Desde el ámbito científico, el proyecto aporta valor al integrar arquitectura MVC, bases de datos NoSQL y validaciones automáticas en múltiples capas, elementos que pueden ser replicados en otros estudios sobre digitalización administrativa y trazabilidad documental. Asimismo, ofrece un caso práctico útil para analizar la transformación de flujos manuales en procesos controlados, seguros y auditables.

En consecuencia, esta propuesta no solo soluciona una necesidad operativa, sino que también genera conocimiento aplicable para investigaciones futuras relacionadas con automatización, control de información y desarrollo de sistemas orientados a eficiencia y confiabilidad.

## **3. Sistema de Objetivos**

### **3.1. Objetivo General**

Desarrollar un sistema web que permita gestionar de manera eficiente los comprobantes y gastos generados por el personal de eventos mediante módulos de registro digital, validación automática en múltiples capas y procesamiento de pagos, con el fin de optimizar los procesos administrativos, mejorar la trazabilidad y transparencia, reducir errores o fraudes y proporcionar una herramienta segura, ágil y accesible para todos los roles involucrados (Staff, Jefe de Ticketera y Contadora).

## 3.2. Objetivos Específicos

- Implementar un módulo de registro y autenticación que permita a los diferentes roles de usuario (Personal, Ticketera, Administrador) ingresar al sistema y cargar la información de los comprobantes de manera segura y estandarizada.
- Desarrollar un algoritmo de validación de pagos que verifique la unicidad y consistencia de los comprobantes registrados, alertando sobre duplicidad o inconsistencias antes de la aprobación final.
- Generar reportes automatizados sobre el estado de los pagos y la asistencia, proporcionando a los organizadores información en tiempo real para la toma de decisiones y el control de ingresos.

INCLUIR %3 REFREENTES A LA ARQUITECTURA

## 4. Alcance

El proyecto se centra en el desarrollo de un Sistema de Gestión y Validación de Comprobantes para Eventos, cuyo propósito es digitalizar y automatizar las actividades relacionadas con el registro de gastos del personal y el procesamiento de pagos. El sistema permitirá que el staff registre sus comprobantes y cargue documentos de respaldo de forma sencilla, evitando así el uso de métodos manuales que suelen generar errores y retrasos. Todo este proceso se realizará dentro de una plataforma segura que diferencia las funciones según el rol del usuario.

Dentro del alcance se incluye la capacidad del sistema para realizar validaciones automáticas en dos niveles. Primero, los comprobantes registrados serán comparados con la información oficial del staff proveniente del archivo Excel cargado por el Jefe de Ticketera. Luego, el sistema verificará que los datos ingresados coinciden con la información que aparece en el documento real subido por el usuario, lo que brinda mayor confianza y reduce las inconsistencias.

El alcance también comprende la gestión administrativa realizada por el Jefe de Ticketera y la Contadora. Ambos roles podrán revisar y validar comprobantes, procesar pagos al personal, registrar pagos excepcionales y llevar un control de los gastos operativos de cada evento. Además, el sistema permitirá consultar un historial completo de comprobantes y gastos, con opciones de filtrado que facilitan auditorías internas y análisis financieros.

Por otro lado, el personal del evento podrá consultar en cualquier momento el estado de sus comprobantes y el estado de sus pagos, lo que aporta transparencia y elimina la necesidad de depender de mensajes o confirmaciones manuales. Esta facilidad mejora la comunicación y la confianza entre el personal y el área administrativa.

En general, el alcance del proyecto se limita al registro, validación y control de comprobantes y pagos relacionados exclusivamente a los eventos. No contempla funcionalidades adicionales como manejo de nómina, administración de proveedores o integración con sistemas contables externos. El objetivo es mantener un enfoque claro y práctico que cubra las necesidades reales del proceso sin extenderse a áreas que no forman parte de su propósito principal.

## 5. Marco Teórico

Para el desarrollo del proyecto se utilizaron diversas herramientas que facilitan la programación, organización y prueba del sistema. Visual Studio Code fue el entorno principal debido a su ligereza, soporte de extensiones y capacidad para trabajar con diferentes lenguajes de programación. Junto a esto, se empleó Node.js y su framework Express, que permiten construir aplicaciones web y APIs de forma rápida utilizando JavaScript en el servidor.

Como sistema de almacenamiento se usó MongoDB, una base de datos NoSQL flexible y escalable que facilita el manejo de datos en formato de documentos. Para el control de versiones se trabajó con Git y GitHub, herramientas esenciales para mantener un historial de cambios y permitir la colaboración durante el desarrollo.

Además, se utilizó Postman para realizar pruebas de las rutas y funcionalidades de la API, asegurando que cada endpoint responda correctamente antes de integrar todo el sistema. Finalmente, Docker permitió ejecutar el proyecto en contenedores aislados, garantizando que funcione de la misma manera en cualquier equipo y facilitando su despliegue.

### 5.1 Metodología (Marco de trabajo 5W+2H)

<b>What?</b> <b>(¿Qué?)</b>	Desarrollo de un sistema web que permita registrar, validar y gestionar comprobantes de pago del personal de eventos, reemplazando el proceso manual actual.
<b>Why?</b> <b>(¿Por qué?)</b>	Para reducir errores, evitar fraudes, centralizar la información y mejorar la eficiencia y trazabilidad del proceso administrativo.
<b>Who?</b> <b>(¿Quién?)</b>	Intervienen tres roles: Staff (registra sus comprobantes), Jefe de Ticketera (valida y gestiona pagos), Contadora (procesa y confirma los pagos). El equipo desarrollador implementa todo el sistema.
<b>When?</b> <b>(¿Cuándo?)</b>	Durante la fase de desarrollo del proyecto académico, siguiendo el cronograma establecido, y utilizado durante los eventos donde se requiera control de comprobantes.

<b>Where?</b> <b>(¿Dónde?)</b>	El sistema funciona en entorno web, accesible desde cualquier computadora con navegador; el desarrollo se realiza en ambiente local con herramientas como VS Code, Git, MongoDB y Docker.
<b>How?</b> <b>(¿Cómo?)</b>	Mediante un flujo digital que incluye: registro de datos, carga de documentos, validación automática en dos capas, revisión administrativa y actualización del estado de pago. Se desarrolla bajo arquitectura MVC con base de datos NoSQL.
<b>How much?</b> <b>(¿Cuánto?)</b>	El costo es mínimo debido al uso de software gratuito. Solo se requiere hardware de uso doméstico y una pequeña inversión opcional en licencias (Windows). La inversión total estimada es de USD 1385.

*Tabla 1 Marco de trabajo 5W+2H*

## 6. Ideas a Defender

En la defensa de este proyecto, se presentan las decisiones de diseño y arquitectura que el equipo adoptó para construir un sistema de gestión y control de comprobantes sólido, escalable y alineado con los principios de Ingeniería de Software y Programación Aplicada. Estas ideas sustentan por qué la solución es técnicamente coherente, mantenible y adecuada para los procesos administrativos de la empresa.

### 1. La Solidez de la Arquitectura MVC para Sistemas Empresariales

El equipo defiende que la arquitectura Modelo–Vista–Controlador (MVC) es la más adecuada para un sistema empresarial que gestiona información crítica como comprobantes, gastos y pagos del personal. Se argumenta que MVC permite separar de manera clara la lógica del negocio, la interfaz de usuario y el manejo de datos, lo cual reduce la complejidad del código, facilita el mantenimiento y mejora la escalabilidad del proyecto.

Desde la Ingeniería de Software, se justifica que una estructura desacoplada favorece la evolución futura del sistema, permitiendo agregar módulos sin afectar otros componentes. Desde la programación aplicada, MVC permite organizar de manera clara controladores, modelos y vistas, garantizando un código limpio, reutilizable y más fácil de depurar.

## **2. La Estructuración del Flujo Administrativo como Mejora Organizacional**

El equipo sostiene que un sistema para gestión de comprobantes debe organizar sus procesos de forma estructurada y coherente, aplicando técnicas de análisis y diseño propias de la Ingeniería de Software.

Aquí se defiende que el proyecto no solo digitaliza tareas, sino que modela el flujo real de trabajo: registro, almacenamiento, validación, consulta y pago.

Este modelado se apoya en principios de programación como manejo correcto de datos, uso de clases y funciones específicas para cada responsabilidad y control adecuado de errores.

De esta manera, el sistema aporta mejoras tangibles a la organización: elimina tareas manuales, reduce tiempos de procesamiento y brinda trazabilidad completa del ciclo administrativo.

## **3. MongoDB como Base para un Sistema Flexible y Orientado a Trazabilidad**

El equipo sostiene que el uso de MongoDB es la decisión más apropiada considerando que el sistema debe manejar comprobantes, pagos, gastos y eventos que pueden variar ampliamente en estructura. MongoDB permite trabajar con documentos flexibles y escalables, lo que facilita adaptarse a diferentes formatos de comprobantes y necesidades futuras sin modificar esquemas rígidos.

La elección también se fundamenta en la necesidad de preservar trazabilidad completa de cada acción del sistema: quién registró un comprobante, cuándo fue modificado, quién procesó un pago y cuáles documentos fueron adjuntados. Esta capacidad es esencial para procesos administrativos y auditorías internas. Con MongoDB, el sistema puede almacenar históricos, objetos incrustados y metadatos útiles sin afectar el rendimiento.

Desde la programación, se demuestra la facilidad de manipular documentos JSON, integrarlos con controladores y aplicar operaciones CRUD de manera eficiente.

## **4. El Patrón Factory Method como Facilitador de Módulos Escalables**

El equipo defiende el uso del patrón de diseño Factory Method como una solución efectiva para instanciar objetos relacionados con comprobantes, pagos y registros del sistema sin acoplar la lógica del negocio a clases rígidas. Este patrón permite crear diferentes tipos de entidades (por ejemplo: comprobantes, pagos procesados, gastos operativos) mediante un único punto de creación.

Desde Ingeniería de Software, se argumenta que el Factory Method aporta extensibilidad: si en el futuro se agrega un nuevo tipo de comprobante o un nuevo tipo de gasto, se puede integrar sin romper el código existente. Desde el enfoque de programación, se demuestra

cómo el uso de este patrón simplifica la construcción de objetos, mejora la organización del código y hace que los módulos sean más fáciles de probar, mantener y expandir.

## 7. Resultados Esperados

Como resultado del proyecto, se espera obtener un sistema funcional que permita mejorar significativamente la gestión de comprobantes y pagos dentro de los eventos. En primer lugar, se espera que la solución automate tareas que actualmente se realizan de manera manual, logrando procesos más rápidos, ordenados y con menos errores. El sistema debe permitir que el personal pueda registrar sus comprobantes y consultar el estado de sus pagos sin necesidad de intermediarios, lo que aporta transparencia y reduce el tiempo que se pierde en coordinaciones internas.

En el ámbito de Ingeniería de Software, se espera haber aplicado correctamente todos los fundamentos trabajados en clase: análisis del problema, levantamiento de requerimientos, diseño de casos de uso, definición de roles, modelamiento, validación, planificación y documentación formal del sistema. El proyecto debe reflejar la capacidad de transformar un problema real en una solución estructurada, documentada y fundamentada, siguiendo buenas prácticas y estándares básicos.

Desde Fundamentos de Programación, se espera haber construido una aplicación que implemente la lógica necesaria para que las funcionalidades principales del sistema operen correctamente. Esto incluye procesos como el registro de información, la validación automática de datos, la carga de documentos, la consulta de estados y la gestión de pagos. El proyecto debe evidenciar el uso adecuado de estructuras de control, manejo de datos, validaciones, funciones y comunicación con una base de datos o sistema de almacenamiento.

En general, el resultado esperado es contar con un prototipo o sistema funcional que no solo cumpla los requerimientos, sino que también demuestre la capacidad de aplicar los conocimientos teóricos y prácticos adquiridos durante el ciclo. Además, se busca que el sistema sea una solución útil, coherente y alineada con las necesidades reales del proceso que se pretende mejorar.

## 8. Viabilidad(Ej.)

Cantidad	Descripción	Valor Unitario (USD)	Valor Total (USD)
	<b>Equipo en casa</b>		
2	Laptop ASUS vivobook / 16gb RAM / 512gb SSD	600	1200
1	Router WiFi 5	40	40
	<b>Software</b>		
1	Sistema operativo Windows 10	145	145
1	Ubuntu Linux	0	0
1	Git	0	0
1	Visual Studio Code	0	0
1	Docker	0	0
1	FileZilla	0	0
	TOTAL		1385

Tabla 2 Presupuesto del proyecto

### 8.1 Humana

#### 8.1.1 Tutor Empresarial

##### Miguel Vivanco

- Explicar al equipo de desarrollo el flujo real del proceso administrativo.
- Validar que el sistema cumpla con los requisitos funcionales internos.
- Asegurar que las validaciones correspondan a la operación real.
- Supervisar los módulos de registro, validación y pagos desde la perspectiva empresarial.
- Proveer retroalimentación continua durante el desarrollo.
- Confirmar que las funcionalidades implementadas facilitan el trabajo del staff, de la Jefe de Ticketera y de la Contadora.
- Su rol es clave para asegurar que el sistema sea útil, aplicable y alineado con los objetivos financieros de la empresa.

## 8.1.2 Tutor Académico

Ing. Jenny A Ruiz R

- Supervisar el desarrollo del proyecto asegurando que se apliquen correctamente los fundamentos de Análisis y diseño de la ingeniería del Software.
- Revisar y validar el backlog del proyecto y las historias de usuario propuestas por el equipo.
- Evaluar la calidad de la documentación, el análisis y los entregables presentados en cada etapa.
- Brindar retroalimentación técnica y metodológica para mejorar la estructura y coherencia del proyecto.
- Guiar al equipo en la correcta elaboración de modelos, diagramas y procesos según los estándares vistos en clase.
- Asegurar que el proyecto mantenga consistencia con las buenas prácticas de desarrollo de software.
- Realizar el seguimiento académico del avance y tomar decisiones de mejora cuando sea necesario.
- Calificar el proyecto de acuerdo con los criterios establecidos en la asignatura

## 8.1.3 Estudiantes

Estudiante	Rol Principal	Responsabilidades y Actividades
Mideros, Samir	Desarrollador Backend / DB	Diseño del esquema de base de datos en MongoDB y desarrollo de lógica de validación en el servidor.
Miranda, Alison	Analista / QA	Levantamiento de requerimientos, redacción de historias de usuario y ejecución de pruebas de software.
Morán, David	Desarrollador Frontend	Implementación de interfaces de usuario para el registro de comprobantes y paneles administrativos.

Vivanco, Gabriel	Arquitecto de Software	Definición de la arquitectura MVC, integración de módulos y despliegue del sistema.
------------------	------------------------	-------------------------------------------------------------------------------------

## 8.2 Tecnológica

### 8.2.1 Hardware

	Requisitos mínimos	Disponibilidad
Memoria RAM	<b>4 GB de RAM.</b> Este valor garantiza que el usuario pueda ejecutar el navegador y el sistema web sin ralentizaciones, permitiendo cargar formularios, comprobantes y vistas con fluidez.	Alta
Almacenamiento	<b>10 GB de espacio disponible.</b> Es suficiente para almacenar archivos temporales, descargas de comprobantes, documentos generados y la caché del navegador necesaria para el uso continuo del sistema.	Alta
Procesador	<b>Procesador Dual Core 2.0 GHz o superior.</b> Esto asegura que las validaciones internas, el renderizado del frontend y el manejo de archivos se ejecuten sin retrasos perceptibles.	Alta
Conectividad	<b>Conexión a Internet estable.</b> El sistema funciona completamente en línea, por lo que se requiere acceso constante para autenticación, carga de documentos y consultas de estado.	Alta
Pantalla	<b>Resolución mínima 1366x768.</b> Permite visualizar correctamente todas las interfaces del sistema sin que los elementos se solapen o queden fuera de la vista	Alta

Tabla 3 Requisitos de Hardware

## 8.2.2 Software

Software / Herramienta	Requisitos mínimos / Descripción	Disponibilidad
Sistema Operativo	Windows 10 / 11 o Linux. Ambiente base para el desarrollo y despliegue local de la aplicación.	Alta
Visual Studio Code	Última versión estable. IDE ligero recomendado por su capacidad de integración con FTP y extensiones para desarrollo web.	Alta
MongoDB	Versión 5.0 o superior. Base de datos NoSQL requerida para almacenar documentos flexibles y garantizar la trazabilidad de los comprobantes.	Alta
Git / GitHub	Control de versiones. Necesario para la gestión colaborativa del código fuente y el control de cambios del proyecto.	Alta
Navegador Web	Google Chrome / Brave / Edge. Requerido para la ejecución de pruebas del cliente y depuración de la interfaz gráfica.	Alta

Tabla 4 Requisitos de Software

## 9. Conclusiones y recomendaciones

### 9.1 Conclusiones

El desarrollo del Sistema de Gestión y Validación de Comprobantes para Eventos permitió abordar de manera estructurada los problemas existentes en el proceso manual que utilizaba la organización. Durante el proyecto se logró automatizar el registro de gastos, implementar un sistema de validación en dos capas y generar un flujo completo de pago

con evidencia digital, contribuyendo significativamente a la transparencia y trazabilidad de la información financiera.

El análisis realizado al inicio permitió identificar claramente las necesidades del Staff, de la Jefe de Ticketera y de la Contadora. Gracias a ello, el sistema fue diseñado siguiendo una arquitectura modular que separa responsabilidades por roles, cumpliendo el objetivo general planteado: automatizar y asegurar el proceso de registro, validación y pago de comprobantes. Asimismo, los objetivos específicos —mejorar la trazabilidad, reducir errores, fortalecer la seguridad y facilitar auditorías— se cumplieron satisfactoriamente mediante la inclusión de validaciones automáticas, almacenamiento digital y un historial de consultas filtrables.

Desde una perspectiva crítica, el sistema proporciona un avance significativo respecto al proceso manual previo. Sin embargo, también se identificaron desafíos, como la integración con documentos oficiales, la necesidad de estandarizar los formatos de Excel y las limitaciones al realizar validaciones sobre documentos escaneados. Estos aspectos sugieren oportunidades futuras para optimizar la precisión de la validación y mejorar la experiencia del usuario en la carga de documentos.

En síntesis, el proyecto alcanza los objetivos planteados y propone una solución tecnológica funcional, escalable y alineada con los procesos reales de la organización, generando un impacto positivo en tiempos de procesamiento, seguridad y confiabilidad de la información financiera.

## 9.2 Recomendaciones

### **1. Mejorar el proceso de validación automatizada:**

Aunque el sistema cumple con las validaciones básicas, se recomienda incorporar herramientas más avanzadas de reconocimiento de texto para optimizar la exactitud al comparar datos entre el formulario y los documentos digitales.

### **2. Estandarizar los formatos de los archivos del staff:**

Se sugiere elaborar una plantilla oficial que sea utilizada en todos los eventos, con el fin de evitar inconsistencias durante la importación y reducir errores de estructura.

### **3. Capacitar a los usuarios finales:**

Se recomienda realizar capacitaciones para el Staff, la Jefe de Ticketera y la Contadora, con el fin de asegurar un uso adecuado del sistema y aprovechar al máximo las funcionalidades implementadas.

### **4. Implementar alertas y notificaciones automatizadas:**

Para mejorar la experiencia del Staff, se recomienda agregar un módulo de notificaciones por correo o dentro del sistema que informe cambios de estado en comprobantes y pagos.

### **5. Monitorear periódicamente el sistema:**

Para mantener la estabilidad y seguridad, se aconseja realizar auditorías internas y actualizaciones periódicas, asegurando que los datos se mantengan íntegros y protegidos.

### **6. Considerar futuras integraciones con sistemas externos:**

A mediano plazo, se recomienda explorar la posibilidad de integrar el sistema con herramientas contables utilizadas por la empresa para automatizar aún más el flujo de gasto y pago.

## 10. Planificación para el Cronograma:

Se dividieron las tareas por actividades del proyecto: Perfil del Proyecto, Matrices, Maquetado y Diagramas (revisar Anexo I).

ACTIVIDAD	TAREA	Prioridad	ENCARGADO	ESTADO	FECHA DE INICIO	FECHA DE FINALIZACIÓN
PERFIL PROYECTO	Introducción	Alta	ALISON SOLANGE MIRANDA CREAMER	Completada	4/12/2025	11/12/2025
	Planteamiento del Trabajo	Alta	LENIN DAVID MORAN PALOMO	Completada	4/12/2025	11/12/2025
	Sistema de Objetivos	Alta	SAMIR ESTEBAN MIDEROS CRUZ	Completada	4/12/2025	11/12/2025
	Alcance	Alta	LENIN DAVID MORAN PALOMO	Completada	4/12/2025	11/12/2025
	Marco Teórico	Alta	GABRIEL NICOLAS VIVANCO RAZA	Completada	4/12/2025	11/12/2025
	Ideas a Defender	Alta	ALISON SOLANGE MIRANDA CREAMER	Completada	4/12/2025	11/12/2025
	Resultados Esperados	Alta	SAMIR ESTEBAN MIDEROS CRUZ	Completada	4/12/2025	11/12/2025
	Viabilidad	Alta	GABRIEL NICOLAS VIVANCO RAZA	Completada	4/12/2025	11/12/2025
	Conclusiones y Recomendaciones	Alta	ALISON SOLANGE MIRANDA CREAMER	Completada	4/12/2025	11/12/2025
MATRIZ	Planificación para el Cronograma	Alta	ALISON SOLANGE MIRANDA CREAMER	Completada	4/12/2025	11/12/2025
	Históricas de Usuario	Alta	LENIN DAVID MORAN PALOMO	Completada	4/12/2025	11/12/2025
	Backlog (excel)	Alta	GABRIEL NICOLAS VIVANCO RAZA	Completada	4/12/2025	11/12/2025
	Backlog (jira)	Alta	SAMIR ESTEBAN MIDEROS CRUZ	Completada	4/12/2025	11/12/2025

Figura 1: Cronograma del proyecto (1).

MAQUETADO	Interfaz 1: Pantalla de Inicio	Alta	LENIN DAVID MORAN PALOMO	Completada	4/12/2025	11/12/2025
	Interfaz 2: Registro de Comprobante	Alta	ALISON SOLANGE MIRANDA CREAMER	Completada	4/12/2025	11/12/2025
	Interfaz 3: Carga de Archivo	Alta	LENIN DAVID MORAN PALOMO	Completada	4/12/2025	11/12/2025
	Interfaz 4: Validación de Comprobante	Alta	GABRIEL NICOLAS VIVANCO RAZA	Completada	4/12/2025	11/12/2025
DIAGRAMAS	Casos de Uso	Alta	ALISON SOLANGE MIRANDA CREAMER	Completada	4/12/2025	11/12/2025
	Contexto	Alta	LENIN DAVID MORAN PALOMO	Completada	4/12/2025	11/12/2025
	Arquitectura	Alta	GABRIEL NICOLAS VIVANCO RAZA	Completada	4/12/2025	11/12/2025
	Patrón de Diseño	Alta	SAMIR ESTEBAN MIDEROS CRUZ	Completada	4/12/2025	11/12/2025

Figura 2: Cronograma del proyecto (2).

## 11. Referencias

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley. (Referencia para Patrón Factory Method y MVC).

MongoDB Inc. (2024). *The MongoDB 5.0 Manual*. Recuperado de <https://www.mongodb.com/docs/manual/> (Documentación oficial para la base de datos).

Pressman, R. S., & Maxim, B. R. (2021). *Ingeniería del software: Un enfoque práctico* (9na ed.). McGraw-Hill Interamericana. (Fundamento para el ciclo de vida y calidad de software).

Sommerville, I. (2016). *Software Engineering* (10ma ed.). Pearson Education. (Referencia para levantamiento de requisitos y arquitectura).

## **Anexos.**

### **Anexo I. Cronograma**

[https://docs.google.com/spreadsheets/d/1rNu7pYNBGTQLf3dgYBIHJP5XCEyaPo3DTx9jGZFC\\_yc/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1rNu7pYNBGTQLf3dgYBIHJP5XCEyaPo3DTx9jGZFC_yc/edit?usp=sharing)

### **Anexo II. MTZ de Historias de Usuarios**

<https://docs.google.com/document/d/1n6pb19oPJnshMZH147Cvd4UEk5y6J4p1T0BKDuCIIww/edit?usp=sharing>