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

**Carrera de Software**

**Prácticas Pre Profesionales (PPP)**

Perfil del Proyecto

Presentado por: Echeverria Luis

Tutor académico: Ruiz Jenny

Nrc: 24421

Ciudad: Quito

Fecha: 08/05/2025

**Índice Pág.**

Contenido

[1. Introducción 5](#_Toc203689164)

[2. Planteamiento del trabajo 5](#_Toc203689165)

[2.1 Formulación del problema 5](#_Toc203689166)

[2.2 Justificación 5](#_Toc203689167)

[3. Sistema de Objetivos 5](#_Toc203689168)

[3.1. Objetivo General 5](#_Toc203689169)

[3.2. Objetivos Específicos 6](#_Toc203689170)

[4. Alcance 6](#_Toc203689171)

[5. Marco Teórico 6](#_Toc203689172)

[5.1 Metodología (Marco de trabajo 5W+2H) 7](#_Toc203689173)

[6. Ideas a Defender 8](#_Toc203689174)

[7. Resultados Esperados 8](#_Toc203689175)

[8. Viabilidad 9](#_Toc203689176)

[9. Conclusiones y recomendaciones 11](#_Toc203689177)

[**9.1** Conclusiones 11](#_Toc203689178)

[**9.2** Recomendaciones 11](#_Toc203689179)

[10. Planificación para el Cronograma: 13](#_Toc203689180)

[11. Referencias 13](#_Toc203689181)

[12. Anexos 14](#_Toc203689182)

[Anexo I. Crono 14](#_Toc203689183)

## Introducción

En el entorno actual, tanto instituciones públicas como privadas deben gestionar adecuadamente las quejas, sugerencias y peticiones de sus usuarios o ciudadanos. Sin embargo, muchas de estas organizaciones aún utilizan métodos poco eficaces, como hojas físicas, correos desorganizados o sistemas manuales que dificultan el seguimiento de estos reclamos. Esto genera insatisfacción, pérdida de confianza y falta de respuesta efectiva.  
El proyecto propuesto pretende responder a esta necesidad, desarrollando una plataforma digital que permita a los usuarios registrar sus quejas de manera simple y a las instituciones gestionarlas eficazmente.

## 2. Planteamiento del trabajo

### 2.1 Formulación del problema

Actualmente, existe una deficiente gestión de quejas en muchas organizaciones, con procesos fragmentados, sin trazabilidad ni seguimiento. Esto causa pérdida de información, lentitud en las respuestas y baja satisfacción del usuario.  
Este proyecto plantea como solución el desarrollo de una aplicación web que recoja, almacene y administre estas quejas de manera automatizada. La plataforma contará con un formulario amigable, validación de datos, almacenamiento en base de datos y un panel administrativo para la revisión de los registros. Esta propuesta supera a los sistemas tradicionales ofreciendo inmediatez, organización, facilidad de uso y escalabilidad.

### 2.2 Justificación

Este proyecto busca precisamente resolver ese problema mediante el desarrollo de una aplicación web sencilla, accesible y enfocada en la experiencia del usuario. La herramienta no solo digitaliza el proceso de recolección de quejas, sino que también permite organizar, visualizar y dar seguimiento a los reportes, mejorando la comunicación interna y la capacidad de respuesta de la organización.

## 3. Sistema de Objetivos

### 3.1. Objetivo General

Desarrollar un sistema web funcional que permita a los usuarios registrar quejas o peticiones de forma eficiente y que facilite a los administradores su almacenamiento, visualización y gestión, a fin de mejorar los procesos de atención institucional y garantizar respuestas más oportunas, utilizando el marco de trabajo 5W+2H y tecnologías como React, FastAPI y PostgreSQL.

### 3.2. Objetivos Específicos

1. Identificar los requisitos funcionales y no funcionales del sistema mediante entrevistas con el tutor empresarial y análisis de procesos actuales.
2. Implementar el sistema utilizando tecnologías modernas como React, FastAPI, PostgreSQL y Tailwind, garantizando una arquitectura escalable y segura.
3. Validar el correcto funcionamiento del sistema mediante pruebas funcionales (Postman), de carga (JMeter) y pruebas de aceptación.

## 4. Alcance

**Funcionalidades incluidas:**

1. Formulario digital para el registro de quejas.
2. Validación de campos en frontend y backend.
3. Almacenamiento en base de datos PostgreSQL.
4. Panel de administración para visualizar quejas recibidas.
5. Confirmación al usuario tras envío exitoso.

**Limitaciones:**

1. No incluye autenticación de usuarios.
2. No incorpora notificaciones automáticas.
3. No cuenta con análisis estadístico de las quejas.

**Escalabilidad:**

1. Diseñado inicialmente para entorno local.
2. Preparado para migrar a servidores remotos en el futuro.

## 5. Marco Teórico

Durante el desarrollo del sistema de gestión de quejas y peticiones se utilizaron las siguientes herramientas e IDEs:

* **Visual Studio Code v1.89.1**: Entorno principal de desarrollo tanto para frontend como backend. Ofrece soporte para JavaScript, React y Node.js.
* **Node.js v20.11.1**: Utilizado para crear el servidor backend, configurar rutas REST y gestionar la conexión con la base de datos.
* **Postgrest v15.4**: Base de datos relacional utilizada para almacenar las quejas registradas. Se eligió por su robustez, seguridad, soporte a largo plazo y compatibilidad con sistemas de producción.
* **Vite v5.1 + React.js v18.2.0**: Framework moderno y eficiente para el desarrollo del frontend. Se eligió Vite por su rapidez en el arranque del entorno.
* **Tailwind CSS v3.4.1**: Framework CSS utilizado para estilizar el sistema de manera rápida y adaptable.
* **Postman v10.21.1:**: Herramienta usada para probar las API REST del backend durante el desarrollo.
* **JMeter v5.6.3**: Para pruebas de carga.
* **Git v2.44 y GitHub**: Utilizados para control de versiones y respaldo del proyecto durante su avance.
* **Excel**: Para el diseño de cronogramas, planificación de actividades (como la guía 5W+2H), y registro del itinerario de trabajo diario.

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

Debe explicar paso a paso el desarrollo de la guía con la herramienta de Excel aplicando el marco de trabajo de las 5W y 2H

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ¿QUÉ? | ¿CÓMO? | ¿QUIÉN? | ¿CUÁNDO? | ¿POR QUÉ? | ¿CUÁNTO? | % DE CUMPLIMIENTO |
| Levantamiento de requerimientos | Entrevistas con el tutor empresarial y revisión de necesidades del sistema actual | Estudiante, tutor empresarial | Semana 1 | Para identificar funciones necesarias del sistema de quejas |  | 100% |
| |  | | --- | | Diseño de base de datos y modelos |  |  | | --- | |  | | |  | | --- | | Usando SQLAlchemy con PostgreSQL para estructurar la persistencia |  |  | | --- | |  | | Estudiante | Semana 2 | |  | | --- | | Para garantizar una estructura sólida y escalable de datos |  |  | | --- | |  | |  | 100% |
| |  | | --- | | Desarrollo del backend con FastAPI |  |  | | --- | |  | | |  | | --- | | Implementación de endpoints REST y lógica de negocio |  |  | | --- | |  | | Estudiante | Semana 3 y 4 | |  | | --- | | Para permitir que el sistema reciba, almacene y gestione las quejas desde el frontend |  |  | | --- | |  | |  | 100% |
| |  | | --- | | Desarrollo del frontend con React |  |  | | --- | |  | | |  | | --- | | Construcción de formularios y paneles de administración |  |  | | --- | |  | | Estudiante | Semana 5 y 6 | |  | | --- | | Para permitir la interacción del usuario con el sistema |  |  | | --- | |  | |  | 100% |
| |  | | --- | | Pruebas funcionales y de seguridad |  |  | | --- | |  | | |  | | --- | | Postman para pruebas de API, JMeter para pruebas de carga y validaciones contra ataques |  |  | | --- | |  | | Estudiante | Semana 7 | |  | | --- | | Para asegurar que el sistema funcione correctamente y sea seguro |  |  | | --- | |  | |  | 100% |
| |  | | --- | | Documentación y despliegue local |  |  | | --- | |  | | |  | | --- | | Redacción del manual técnico y del usuario, despliegue en entorno controlado |  |  | | --- | |  | | Estudiante, tutor empresarial | Semana 8 | |  | | --- | | Para finalizar el proyecto y garantizar su entendimiento por parte del usuario y posibles mantenedores |  |  | | --- | |  | |  | 100% |

Tabla 1 Marco de trabajo 5W+2H

## 6. Ideas a Defender

Las ideas clave que se defienden en este proyecto son:

1. **Fundamentos de Programación**: Uso de estructuras de control, validaciones, conexión con base de datos.
2. **Ingeniería de requisitos de software**: Recolección de requisitos, arquitectura, pruebas funcionales.
3. **Desarrollo Web**: Aplicación práctica de React, Tailwind y APIs REST.
4. **Bases de Datos**: Modelado y consultas en PostgreSQL.
5. **Pruebas de Software**: Aplicación de Postman, JMeter.

## 7. Resultados Esperados

A través del desarrollo e implementación del sistema de gestión de quejas y peticiones, se espera lograr los siguientes resultados:

* **Facilitar la comunicación con los usuarios:** Que cualquier persona pueda enviar fácilmente una queja o sugerencia desde la página web sin necesidad de registros ni procesos complejos.
* **Almacenar la información de forma segura:** Que todas las quejas ingresadas queden registradas correctamente en una base de datos, sin pérdidas de información, asegurando integridad y trazabilidad.
* **Visibilidad de las quejas para los responsables:** Que el personal designado (como un administrador o secretaria) pueda ver, clasificar y dar seguimiento a las quejas en un panel organizado y funcional.
* **Interacción clara entre el sistema y los usuarios:** Que el sistema muestre mensajes comprensibles (confirmación, errores, validaciones) para asegurar una buena experiencia de uso.
* **Optimización del tiempo de respuesta institucional:** Que el uso del sistema permita reducir el tiempo que toma recibir, revisar y gestionar una queja o petición, mejorando así la atención ciudadana o institucional.
* **Aplicación práctica de conocimientos adquiridos:** Que el proyecto refleje el uso correcto de herramientas de programación (React, FastAPI, PostgreSQL), y conceptos clave de Ingeniería de Software, como estructura del sistema, validación, pruebas y seguridad básica.
* **Preparación para entornos reales de trabajo:** Que la experiencia sirva como base para enfrentar desarrollos web reales, incluyendo el despliegue en un servidor, uso de dominios, SSL y conexión a bases de datos en la nube.
* **Documentación completa del proyecto:** Que se entregue un proyecto acompañado de toda su documentación técnica, cronograma de desarrollo, manual de usuario y evidencias de pruebas funcionales y de seguridad.

## 8. Viabilidad

|  |  |  |  |
| --- | --- | --- | --- |
| Cantidad | Descripción | Valor Unitario (USD) | Valor Total (USD) |
|  | **Equipo en casa** |  |  |
| 1 | Computadora portátil Acer Aspire, Core i7-8565U | 800 | 800 |
|  |  |  |  |
|  | **Software** |  |  |
| 1 | Sistema operativo Windows 11 | 139 | 139 |
| 1 | Visual Studio Code | 0 | 0 |
| 1 | Postgrest | 0 | 0 |
| 1 | Postman, JMeter | 0 | 0 |
|  | | TOTAL | 939 |

Tabla 2 Presupuesto del proyecto

Debe explicar los recursos necesarios para su proyecto y adicionalmente la viabilidad del punto 8.1. y 8.2

**8.1** Humana

8.1.1 **Tutor Empresarial**

Ing. Luis Manuel Echeverria

• **Responsabilidades**

Supervisar el cumplimiento de los requisitos funcionales

Validar el sistema implementado y aprobar pruebas

8.1.2 **Tutor Académico**

Ing. Jenny Ruiz

* **Responsabilidades**

Acompañamiento técnico y metodológico de ingeniería de software

Revisión de entregables, documentación y asesoría continua

8.1.3 **Estudiantes**

Luis Marcelo Echeverria

• **Responsabilidades**

Diseño, desarrollo y pruebas del sistema

Redacción de documentación

Implementación de medidas de seguridad

**8.2** Tecnológica

**8.2.1 Hardware**

|  |  |  |
| --- | --- | --- |
|  | Requisitos mínimos | Disponibilidad |
| Memoria RAM | 4 GB de RAM | Alta |
| Almacenamiento | 10 GB de espacio de almacenamiento | Alta |

Tabla 3 Requisitos de Hardware

**8.2.2 Software**

|  |  |  |
| --- | --- | --- |
|  | Requisitos mínimos | Disponibilidad |
| Sistema Operativo | Se recomienda Windows 10 u 11, macOS  10.10 o Ubuntu 16 | Alta |
| IDE | |  | | --- | |  |  |  | | --- | | Visual Studio Code | |  | |  | | Alta |
| |  | | --- | | Base de Datos | |  |  |  | | --- | |  | | PostgreSQL | Alta |
| Backend | |  | | --- | |  |  |  | | --- | | FastAPI + SQLAlchemy | | Alta |
| Frontend | React + Vite + Tailwind CSS | Alta |
| |  | | --- | | Herramientas extra |  |  | | --- | |  | | Postman, Apache JMeter, Git | Alta |

Tabla 4 Requisitos de Software

## 9. Conclusiones y recomendaciones

## **9.1** Conclusiones

Se logró identificar de forma precisa los requerimientos del sistema mediante entrevistas, permitiendo diseñar un producto ajustado a las necesidades institucionales.

Se implementó correctamente el sistema con herramientas modernas y eficientes, cumpliendo con estándares actuales de desarrollo web.

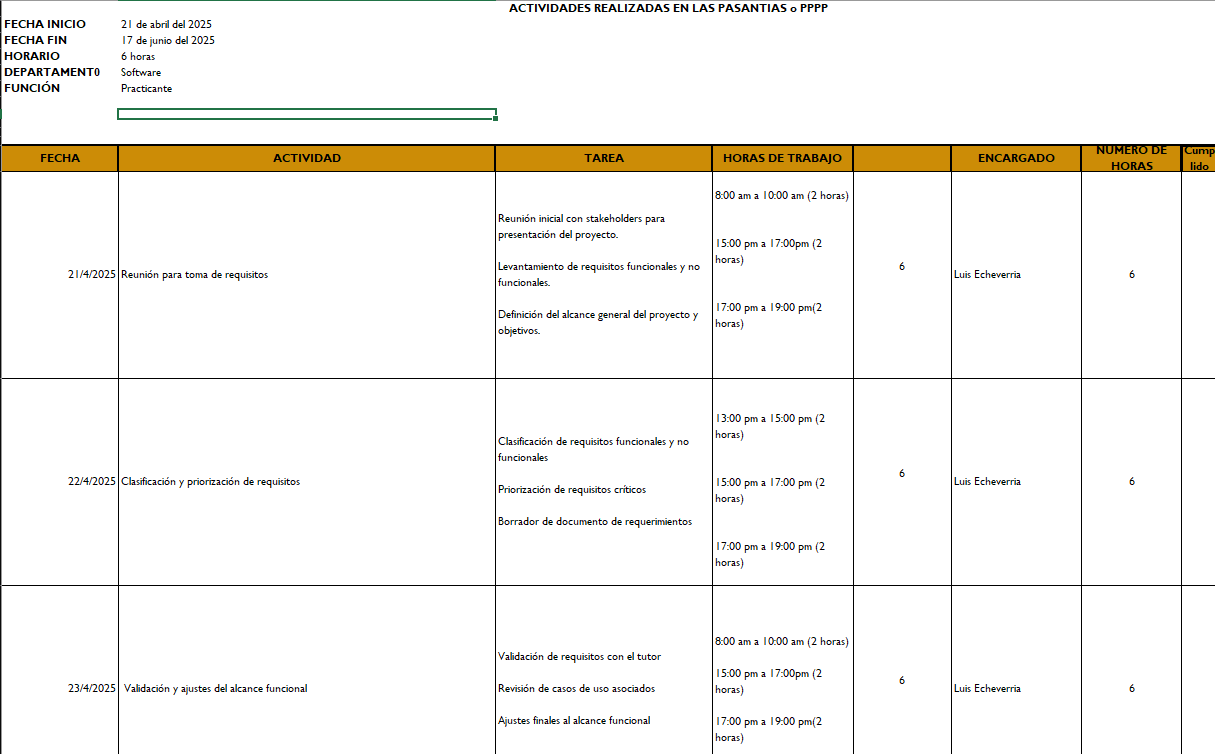
Las pruebas realizadas confirmaron el correcto funcionamiento del sistema en condiciones normales y de carga, con respuestas esperadas y sin vulnerabilidades detectadas.

## **9.2** Recomendaciones

* Implementar un sistema de notificaciones internas o por correo para alertar al personal sobre nuevas quejas.
* Mantener reuniones periódicas con el tutor o supervisor para recibir retroalimentación continua y aclarar dudas.
* Establecer canales de comunicación claros (correo, mensajería instantánea) para reportar avances y obstáculos en el proyecto.
* Implementar un sistema de notificaciones internas o por correo para alertar al personal sobre nuevas quejas.
* Documentar detalladamente los procesos y decisiones tomadas durante el desarrollo para facilitar futuras referencias.
* Expandir el sistema con métricas estadísticas o reportes automáticos para apoyar la toma de decisiones.
* Considerar la integración futura con herramientas de análisis de satisfacción ciudadana para mejorar el servicio.
* Realizar pruebas periódicas de seguridad y rendimiento tras cada actualización del sistema.
* Promover la autoevaluación constante y la búsqueda de aprendizaje adicional para mejorar las competencias profesionales.
* Fomentar la colaboración entre compañeros y áreas para enriquecer el proyecto con diferentes perspectivas.

## 10. Planificación para el Cronograma:

Debe insertar una imagen clara y legible de la planificación del proyecto a desarrollar.



## 11. Referencias

**Normas APA**: <http://normasapa.com/>

**FastAPI documentation**: https://fastapi.tiangolo.com

**ReactJS documentation**: <https://reactjs.org>

**Tailwind CSS documentation**: <https://tailwindcss.com>

**PostgreSQL documentation**: <https://www.postgresql.org/docs/>

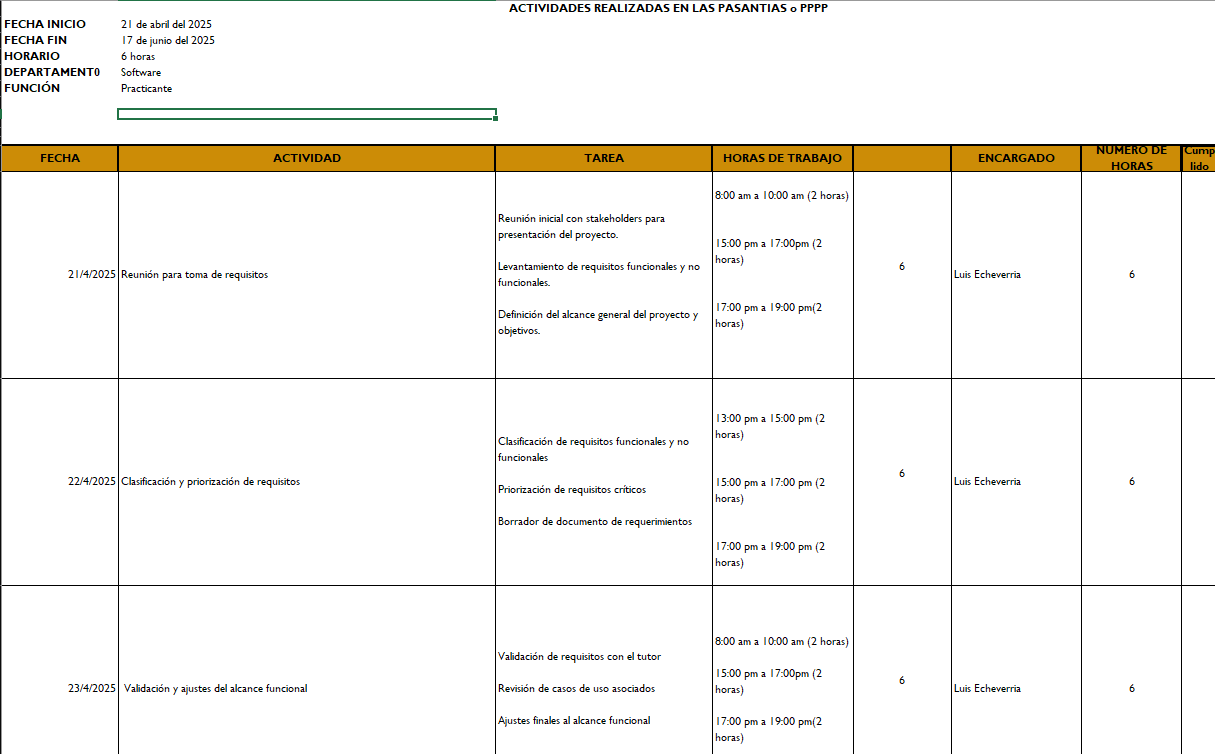
**SQLAlchemy documentation**: https://docs.sqlalchemy.org

**OWASP XSS Prevention**: https://owasp.org/www-community/xss

**OWASP SQL Injection Prevention**: https://owasp.org/www-community/attacks/SQL\_Injection

12. Anexos**.**

### Anexo I. Crono

****