

FORMATO N° 04
INFORME TÉCNICO DE PRÁCTICAS PRE PROFESIONALES
QUE PRESENTA EL ESTUDIANTE¹

1. PORTADA



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SOFTWARE

INFORME DE:

Pasantía

Ayudante de Cátedra

Práctica Pre Profesional No Remunerada

Ayudante de Investigación

Servicio a la comunidad

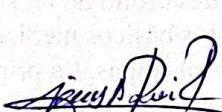
ADUFA

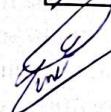
NOMBRES Y APELLIDOS DEL ESTUDIANTE: LUIS MARCELO ECHEVERRIA MOLINA

NOMBRES Y APELLIDOS DEL TUTOR ACADÉMICO: JENNY ALEXANDRA RUIZ ROBALINO

CALIFICACIÓN DEL INFORME

18
4
LNU


FIRMA DE TUTORA ACADÉMICA
Ing. Jenny Alexandra Ruiz Robalino


FIRMA DEL ESTUDIANTE
Luis Marcelo Echeverria Molina


FIRMA DEL TUTOR EMPRESARIAL
Ing. Luis Manuel Echeverria Yanez

Sangolquí, 29/07/2025

2. INTRODUCCIÓN

El presente informe describe el desarrollo de un sistema web para la gestión de quejas y peticiones, llevado a cabo por el estudiante Echeverría Luis, de la Carrera de Software de la Universidad de las Fuerzas Armadas ESPE, bajo la tutoría académica de la Ingeniera Jenny Ruiz y la tutoría empresarial de Ing. Luis Manuel Echeverría. Este proyecto se realizó en la Asociación de Docentes e Investigadores de la Universidad de las Fuerzas Armadas (Adufa). El sistema fue desarrollado utilizando tecnologías modernas como React, FastAPI, PostgreSQL y Tailwind CSS, siguiendo el marco de trabajo ágil scrum, entre el 21 de abril y el 17 de junio de 2025.

La iniciativa surgió debido a la necesidad de superar los métodos tradicionales ineficaces de gestión de quejas, como hojas físicas y correos desorganizados, que generaban pérdida de información y respuestas lentas. El propósito fue facilitar la comunicación entre usuarios y la institución, almacenar datos de manera segura, ofrecer visibilidad a los responsables mediante un panel administrativo y optimizar los tiempos de respuesta. Este trabajo se alinea con el perfil de egreso de la carrera, aplicando conocimientos en desarrollo web, bases de datos y pruebas de software.

3. DESARROLLO

Durante las prácticas preprofesionales, se ejecutaron diversas actividades en el área de desarrollo de software, aplicando metodologías ágiles, marco de trabajo ágil scrum basados en la ingeniería de software. A continuación, se describen las actividades realizadas, metodologías, recursos, limitaciones y logros.

En las prácticas preprofesionales se siguió el marco de trabajo ágil Scrum, aplicando principios de la ingeniería de software para el diseño y desarrollo de un sistema de gestión de quejas. Desde el inicio se definieron los requerimientos básicos mediante un documento ERS, lo cual permitió organizar y planificar el proyecto en etapas. La primera actividad fue la creación del repositorio remoto en GitHub, donde se centralizó todo el código del sistema como se puede observar en la Figura 1.

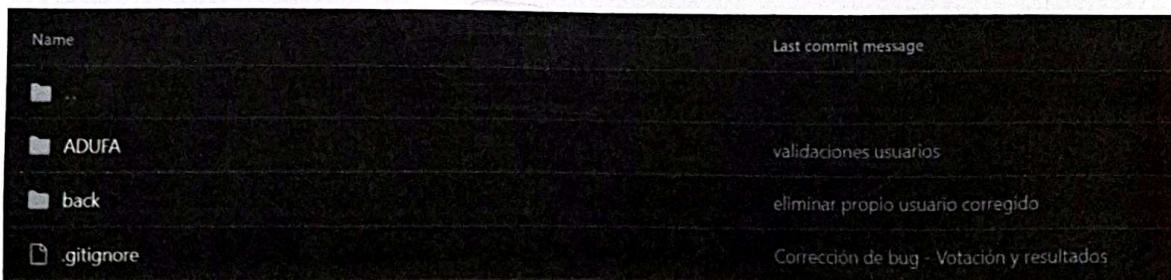


Figura 1. Creación del repositorio remoto en GitHub

Posteriormente, se procedió a la estructuración del proyecto en Visual Studio Code, organizando las carpetas de frontend y backend, lo cual permitió un desarrollo modular y claro como se puede observar en la Figura 2.

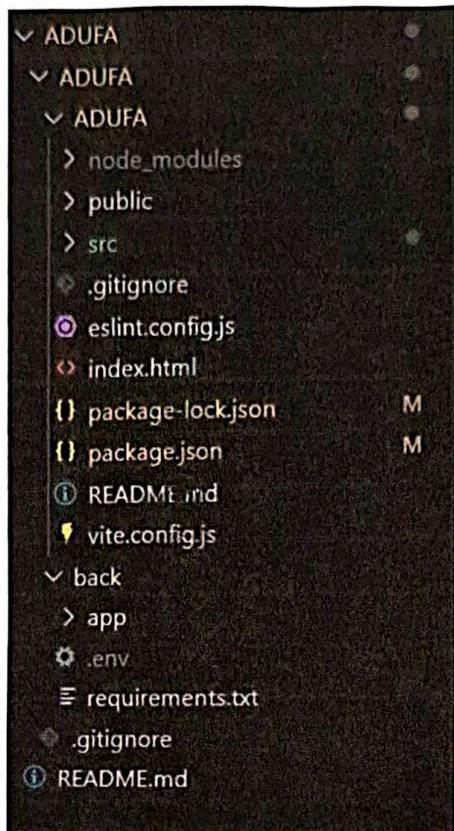


Figura 2. Organización del proyecto en Visual Studio Code

Luego se desarrollaron las interfaces principales, comenzando por el formulario de quejas en React, conectándolo al backend a través de peticiones HTTP usando fetch. Se añadieron validaciones en los campos de nombre, correo y mensaje para evitar entradas maliciosas y asegurar la integridad de los datos ingresados como se observa en la Figura 3.

```

ADUFA > ADOFA > ADOFA > src > pages > quejas.jsx > ComplaintForm
  5   function ComplaintForm() {
 14     useEffect(() => {
 23       fetchTotalQuejas();
 24     }, []);
 25   }
 26   const validateName = (name) => {
 27     const regex = /^[a-zA-Záéíóúññ]+$/;
 28     if (!name) return "El nombre es obligatorio.";
 29     if (name.length < 3) return "Debe tener al menos 3 caracteres.";
 30     if (name.length > 50) return "No debe exceder los 50 caracteres.";
 31     if (!regex.test(name)) return "Solo letras y espacios.";
 32     return "";
 33   }
 34
 35   const validateEmail = (email) => {
 36     const regex = /^[a-zA-Z0-9._]+@[a-zA-Z0-9._]+\.[a-zA-Z]{2,6}$/;
 37     if (!email) return "El correo es obligatorio.";
 38     if (!regex.test(email)) return "correo no válido.";
 39     return "";
 40   }
 41
 42   const handleSubmit = async (e) => {
 43     e.preventDefault();
 44     setIsSubmitting(true);
 45
 46     const nameError = validateName(name);
 47     const emailError = validateEmail(email);
  
```

Figura 3. Validaciones en los campos

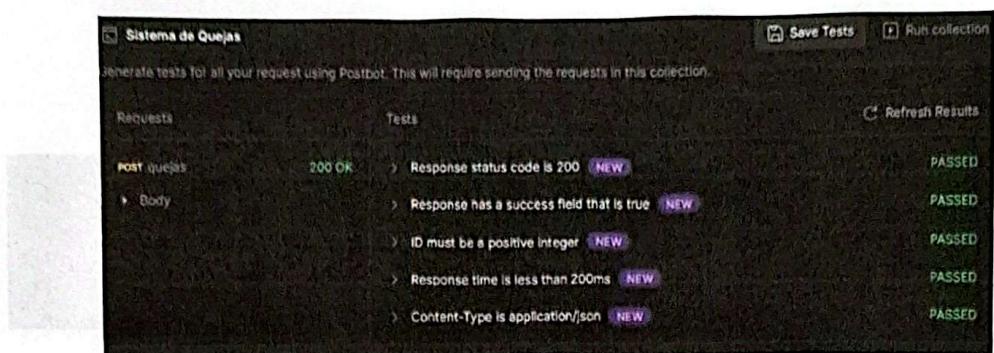
Paralelamente, en el backend desarrollado con FastAPI y SQLAlchemy, se construyeron los modelos de base de datos y los endpoints RESTful, incluyendo /api/complaints para registrar las quejas en PostgreSQL como se puede observar en la Figura 4.

```

ADUFA > ADOFA > back > app > quejas.py > contar_quejas
  15
 16   @router.post("/complaints", response_model=dict)
 17   def registrar_queja(queja: QuejaCreate, db: Session = Depends(get_db)):
 18     nueva = Queja(**queja.dict())
 19     db.add(nueva)
 20     db.commit()
 21     db.refresh(nueva)
 22     return {"success": True, "id": nueva.id}
 23
 24
 25   @router.get("/complaints/count", response_model=dict)
 26   def contar_quejas(db: Session = Depends(get_db)):
 27     total = db.query(Queja).count()
 28     return {"total_quejas": total}
 29
 30
 31   @router.get("/complaints", response_model=list[QuejaOut])
 32   def listar_quejas(
 33     db: Session = Depends(get_db),
 34     admin: Usuario = Depends(admin_required),
 35   ):
 36     return db.query(Queja).order_by(Queja.fecha_creacion.desc()).all()
 37
  
```

Figura 4. Backend y endpoints

Una vez terminada la lógica básica, se realizaron pruebas funcionales utilizando Postman Runner. Estas pruebas permitieron comprobar que las solicitudes POST eran procesadas correctamente, devolviendo respuestas claras incluso ante datos incompletos o inválidos como se puede observar en la Figura 5.



Tests	Status
Response status code is 200	PASSED
Response has a success field that is true	PASSED
ID must be a positive integer	PASSED
Response time is less than 200ms	PASSED
Content-Type is application/json	PASSED

Figura 5. Pruebas en PostmanRunner

Más adelante, se aplicaron pruebas de carga con Apache JMeter, simulando múltiples usuarios enviando quejas simultáneamente, lo cual permitió analizar el rendimiento del sistema bajo condiciones de estrés como se puede observar en la Figura 6.

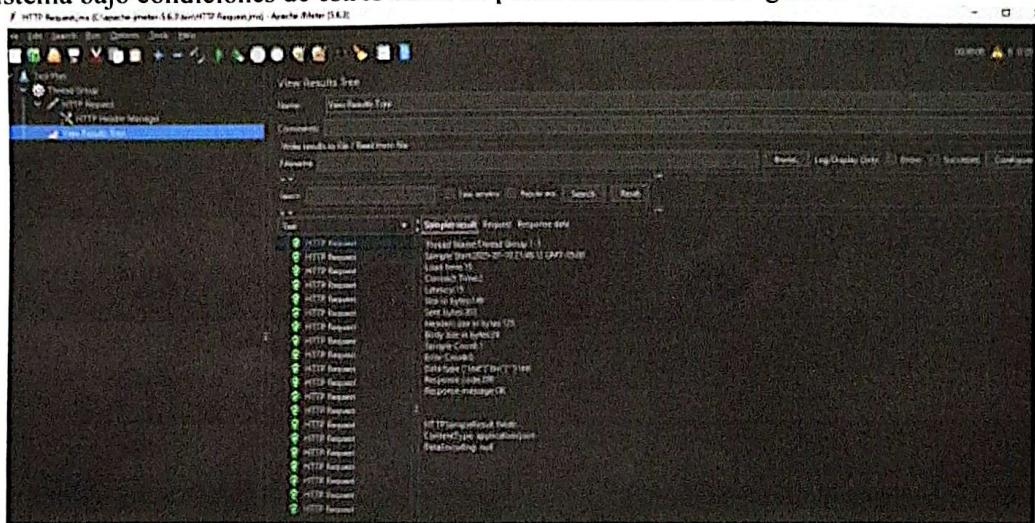
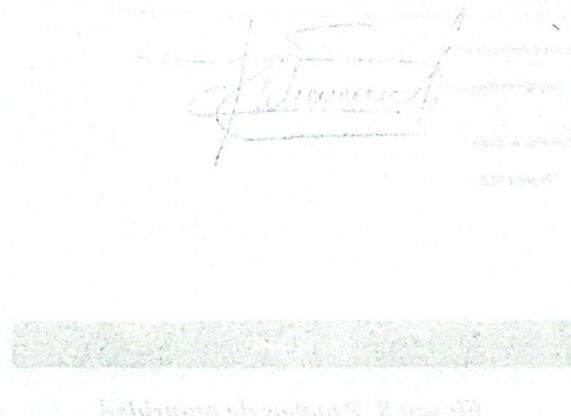


Figura 6. Pruebas en JMeter

Con el sistema funcionando correctamente en entorno local, se inició el proceso de implementación en línea mediante un servicio de hosting compartido con cPanel. Se adquirió un dominio propio (adufa.org) y se realizó la subida de los archivos del frontend a la carpeta `public_html`, mientras que el backend en Python fue configurado en la sección de aplicaciones Python del servidor como se puede observar en la Figura 7.



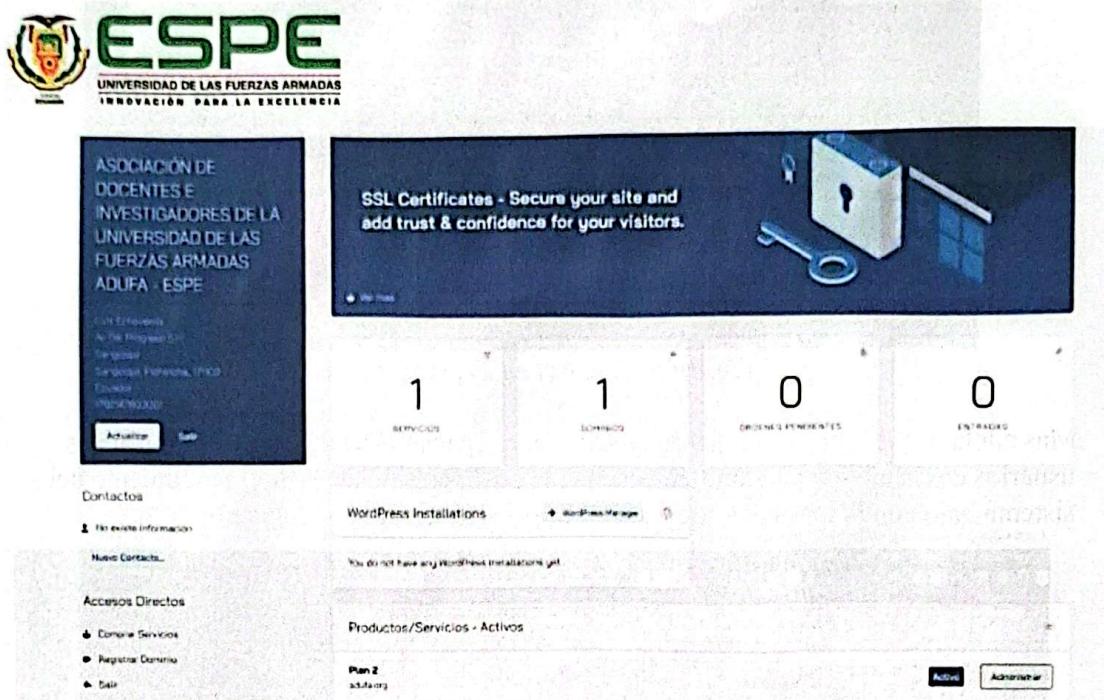


Figura 7. Despliegue del sistema

Además, se realizaron pruebas de seguridad básica, como un intento de inyección SQL simulada utilizando la cadena '`OR 1=1; --`' en el campo de nombre. El sistema rechazó esta entrada gracias a las validaciones implementadas tanto en frontend como en backend, lo que demuestra que las medidas de protección fueron aplicadas correctamente como se puede observar en la Figura 8.

The figure shows a screenshot of a web form titled "Sistema de Quejas y Peticiones". The header says "Tu opinión es importante para nosotros". The form has a green header bar. Below it, the text "Total de quejas enviadas: 61" is displayed. The form fields include "Nombre completo *", "Correo electrónico *", and "Describe tu queja *". In the "Nombre completo *" field, the user has entered "`* OR 1=1; --`". A red validation message appears next to the field: "El nombre solo debe contener letras y espacios.". The "Correo electrónico *" field contains "test@correo.com" and the "Describe tu queja *" field contains "'Prueba SQL'". At the bottom is a green button labeled "Enviar Queja".

Figura 8. Pruebas de seguridad

Finalmente, se validó el correcto almacenamiento de los datos en la base de datos, observando que cada queja ingresada incluye nombre, correo, mensaje, fecha de creación y

Prácticas Pre Profesionales Formato N°04

estado inicial “Abierta”, como se puede observar en la Figura 9. Estas evidencias demuestran el cumplimiento de los objetivos establecidos y reflejan la aplicabilidad de los conocimientos adquiridos en un entorno real de desarrollo.

ID [PK] integer	Nombre character varying (100)	email character varying (150)	mensaje text	estado character varying	fecha_creacion timestamp without time zone
1	Juan Perez	juang@example.com	Esta es una queja de prueba	Cerrada	2025-07-06 16:42:01.22592
2	Santiago Bermudez	juang@example.com	Esta es una queja de prueba	Cerrada	2025-07-06 17:00:00.192325
3	Luis Echeverria	luismecheverriam@gmail.com	mas eventos	Cerrada	2025-07-06 17:02:25.831517
4	Luis Echeverria	lmecheverria2@espe.edu.ec	Mas eventos	Cerrada	2025-07-06 17:08:02.761178
5	Luis Echeverria	luismecheverriam@gmail.com	mas eventos	Cerrada	2025-07-09 20:36:10.117099
6	Santiago Perez	sang@gmail.com	Mejorar los eventos	Cerrada	2025-07-09 22:24:21.162918
7	Rodrigo Perez	sang@gmail.com	menos eventos con fiestas	Cerrada	2025-07-09 23:22:58.398473
8	Rodrigo Perez	sang@gmail.com	aaaa	Cerrada	2025-07-09 23:23:19.681072
9	Santiago Perez	lmecheverria2@espe.edu.ec	bbbbb	Cerrada	2025-07-09 23:23:28.936504
10	Luis Echeverria	lmecheverria2@espe.edu.ec	eventos	Cerrada	2025-07-09 23:27:45.590952
11	Ayme Escobar	aymedespe.edu.ec	muchos eventos	Cerrada	2025-07-09 23:28:11.338191

Figura 9. Validación de guardado en la base de datos

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

Tabla 1.
Lista de tareas

Ord.	Fecha	Dia	Actividad Realizada	Horas
1	21/4/2025	Lunes	Inicio del proyecto con una reunión clave donde se presentaron los objetivos generales a los stakeholders. Se realizó un levantamiento exhaustivo de requisitos funcionales y no funcionales, identificando las necesidades principales del sistema. Además, se definió el alcance del proyecto, estableciendo metas claras y plazos de entrega.	6
2	22/4/2025	Martes	Se organizaron los requisitos recopilados, diferenciando entre funcionales (como el registro de quejas) y no funcionales (como la seguridad y rendimiento). Se priorizaron los requisitos críticos para garantizar que el desarrollo se enfocara en las funcionalidades más importantes. También se elaboró un borrador del documento de requerimientos.	6
3	23/4/2025	Miércoles	Se revisaron los requisitos con el tutor, asegurando su viabilidad y alineación con los objetivos del proyecto. Se analizaron casos de uso para cubrir todos los escenarios posibles y se ajustó el alcance funcional según las observaciones recibidas.	6
4	24/4/2025	Jueves	Se redactó formalmente el alcance del sistema, describiendo sus límites y funcionalidades clave. Se identificaron los módulos principales, como el formulario de quejas y el panel de administración. También se realizó una entrevista complementaria con un usuario	6

			tipo para refinar detalles.	
5	25/4/2025	Viernes	Se completó la redacción del documento de alcance, detallando cada módulo y su propósito. Se enfatizó en la identificación de componentes críticos y se incorporaron los comentarios del usuario tipo para asegurar que el sistema cumpliera con las expectativas.	6
6	28/4/2025	Lunes	Se consolidó toda la información recopilada en un documento de requisitos completo. Tras una validación final con el tutor, se entregó el documento, marcando el fin de la fase de planificación y el inicio del desarrollo.	6
7	29/4/2025	Martes	Se diseñó el modelo de la base de datos, definiendo tablas, relaciones y campos necesarios para almacenar la información de quejas y usuarios. Se revisó la estructura con el tutor para garantizar su eficiencia y escalabilidad.	6
8	30/4/2025	Miércoles	Se esbozó la arquitectura del software, seleccionando tecnologías como React para el frontend, FastAPI para el backend y PostgreSQL para la base de datos. Se definieron los componentes principales y su interacción.	6
9	1/5/2025	Jueves	Se crearon wireframes detallados para el formulario de quejas y el panel de administración, enfocándose en la usabilidad y experiencia del usuario. Se realizaron ajustes visuales basados en feedback inicial.	6
10	5/5/2025	Lunes	Se validó la arquitectura propuesta con el tutor y se ajustó el modelo de datos según sus recomendaciones. Se documentó el diseño técnico para futuras referencias.	6
11	6/5/2025	Martes	Se revisaron los wireframes y la arquitectura, consolidando	6

			los entregables. Se configuró el repositorio del proyecto y se realizó un respaldo inicial del código.	
12	7/5/2025	Miércoles	Se configuró el entorno local con FastAPI y Vite, inicializando los proyectos de frontend y backend. Se creó la base de datos en PostgreSQL y se estableció la estructura básica de archivos.	6
13	8/5/2025	Jueves	Se organizaron las carpetas del proyecto y se instalaron las dependencias necesarias. Se verificó la conexión con la base de datos y se realizaron pruebas iniciales de escritura y lectura.	6
14	9/5/2025	Viernes	Se desarrollaron rutas básicas en Express para manejar las solicitudes HTTP. Se insertaron datos de prueba en la base de datos y se validó su correcto funcionamiento mediante pruebas locales.	6
15	12/5/2025	Lunes	Se revisó el modelo de datos en PostgreSQL, realizando ajustes para optimizar su rendimiento. Se documentó la configuración técnica del proyecto, incluyendo pasos para replicar el entorno.	6
16	13/5/2025	Martes	Se realizaron pruebas exhaustivas del entorno, asegurando que todos los componentes funcionaran correctamente. Se generaron respaldos de la base de datos y se revisó el avance general del proyecto.	6
17	14/5/2025	Miércoles	Se implementaron los endpoints POST y GET para el registro y consulta de quejas. Se integró express-validator para validar los datos ingresados y se aseguró su correcto almacenamiento en la base de datos.	6
18	15/5/2025	Jueves	Se completó el desarrollo del endpoint GET /api/complaints,	6

			permitiendo recuperar todas las quejas registradas. Se realizaron pruebas con Postman y se implementó un manejo robusto de errores en el servidor.	
19	16/5/2025	Viernes	Se sanitizaron los datos ingresados para prevenir inyecciones SQL y otros ataques comunes. Se escribieron pruebas unitarias para validar el funcionamiento de los endpoints y se refactorizó el código para mejorar su legibilidad.	6
20	19/5/2025	Lunes	Se agregaron medidas de seguridad básicas, como CORS y body-parser, para proteger la API. Se revisó la funcionalidad con el tutor y se documentó el backend para facilitar su mantenimiento.	6
21	20/5/2025	Martes	Se verificó la integración entre la base de datos y la API, asegurando que los datos fluyeran correctamente. Se confirmó que todos los endpoints funcionaran como se esperaba y se realizó una revisión final del backend.	6
22	21/5/2025	Miércoles	Se estructuró el formulario de quejas en React, conectándolo con la API REST. Se implementaron validaciones en el frontend para garantizar que los datos ingresados cumplieran con los requisitos antes de ser enviados.	6
23	22/5/2025	Jueves	Se mejoró el diseño del formulario utilizando Tailwind CSS, asegurando una interfaz atractiva y responsive. Se agregaron mensajes visuales para feedback de éxito o error, mejorando la experiencia del usuario.	6
24	26/5/2025	Lunes	Se desarrolló el panel de administración, permitiendo visualizar las quejas registradas de manera dinámica. Se	6

			optimizó la carga de datos y se mejoró la presentación visual del panel.	
25	27/5/2025	Martes	Se revisó la interactividad del sistema, optimizando componentes para mayor fluidez. Se realizó una revisión de código con el tutor, incorporando sus sugerencias para mejorar el rendimiento.	6
26	28/5/2025	Miércoles	Se aseguró que la conexión entre frontend y backend funcionara correctamente. Se realizaron ajustes menores y se consolidó el entregable de frontend para su revisión final.	6
27	29/5/2025	Jueves	Se ejecutaron pruebas completas del sistema, simulando el flujo de un usuario real. Se recopiló feedback para identificar áreas de mejora y se ajustó el sistema según los comentarios.	6
28	30/5/2025	Viernes	Se solucionaron los errores identificados durante las pruebas, realizando ajustes en la presentación visual y refactorizando código para mayor eficiencia.	6
29	2/6/2025	Lunes	Se validó el sistema con usuarios reales, confirmando que cumpliera con los requisitos iniciales. Se preparó la entrega final, organizando los archivos y documentación necesaria.	6
30	3/6/2025	Martes	Se redactó la documentación técnica del sistema y se realizaron pruebas supervisadas por el tutor. Se confirmó que todas las funcionalidades estuvieran operativas y listas para su entrega.	6
31	4/6/2025	Miércoles	Se generó un respaldo completo del proyecto y se realizó la entrega formal al tutor. Se revisaron los detalles de cierre, asegurando que nada quedara pendiente.	6

32	5/6/2025	Jueves	Se incorporaron las sugerencias recibidas durante las pruebas, realizando mejoras visuales y funcionales. Se ejecutaron pruebas finales para garantizar la estabilidad del sistema.	6
33	6/6/2025	Viernes	Se documentó el sistema en un manual técnico, incluyendo capturas de pantalla y explicaciones detalladas de cada funcionalidad. Se validó la experiencia de usuario para asegurar claridad.	6
34	9/6/2025	Lunes	Se revisó el manual técnico con el tutor, incorporando sus observaciones. Se finalizó la documentación, asegurando que estuviera completa y precisa.	6
35	10/6/2025	Martes	Se realizaron pruebas de despliegue local y se limpió el código para eliminar redundancias. Se subió la versión final al repositorio y se verificó el cumplimiento de todos los requisitos.	6
36	11/6/2025	Miércoles	Se realizó una última revisión del sistema, organizando los archivos para su entrega. Se envió la documentación y los respaldos finales al tutor.	6
37	12/6/2025	Jueves	Se realizaron ajustes basados en la retroalimentación final y se llevó a cabo una demostración técnica del sistema. Se presentó el proyecto a los stakeholders, destacando sus funcionalidades y beneficios.	6
38	13/6/2025	Viernes	Se integró el módulo de quejas en la página web existente, ajustando estilos y rutas para garantizar coherencia visual. Se probó la navegación en diferentes resoluciones para asegurar compatibilidad.	6
39	16/6/2025	Lunes	Se implementó un contador de quejas registradas y se ajustaron los estilos según las	6

			observaciones del tutor. Se actualizó el repositorio con los cambios finales y se documentó el proceso.	
40	17/6/2025	Martes	Se validó la funcionalidad del contador y se redactó un instructivo actualizado. Se subió la versión final al repositorio, marcando la conclusión exitosa del proyecto.	6
Total de horas:				240

4. 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.

5. RECOMENDACIONES

Implementar un sistema de notificaciones internas o por correo para alertar al personal sobre nuevas quejas.

Mantener reuniones periódicas con el tutor empresarial para recibir retroalimentación continua y aclarar dudas.

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 para mejorar el servicio.