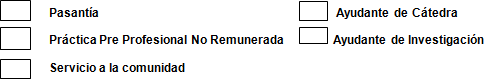
**FORMATO N° 04**

**INFORME TÉCNICO DE PRÁCTICAS PRE PROFESIONALES QUE PRESENTA EL ESTUDIANTE**[**1**](#_bookmark0)

1. **PORTADA**



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN CARRERA DE INGENIERÍA EN SOFTWARE**

**INFORME DE:**

X

ADUFA

**NOMBRES Y APELLIDOS DEL ESTUDIANTE: LUIS MARCELO ECHEVERRIA MOLINA**

**NOMBRES Y APELLIDOS DEL TUTOR ACADÉMICO: JENNY ALEXANDRA RUIZ ROBALINO**

**CALIFICACIÓN DEL INFORME**

**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, 21/07/2025**

# INTRODUCCIÓN

El presente informe describe el desarrollo de un sistema web para la gestión de quejas y peticiones, llevado a cabo por el estudiante Echeverria Luis, de la Carrera de Software de la Universidad de las Fuerzas Armadas ESPE, bajo la tutoría académica de Ruiz Jenny y la tutoría empresarial de Ing. Luis Manuel Echeverria. 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 5W+2H, 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.

# DESARROLLO

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

## Metodología y Técnicas de Trabajo

Se empleó el marco 5W+2H para estructurar las actividades, junto con conceptos de ingeniería de software como recolección de requisitos, diseño de arquitectura y pruebas. El control de versiones se gestionó con GitHub.

## Recursos Utilizados

**Hardware:** Computadora portátil Acer Aspire, Core i7-8565U, 4 GB RAM.

**Software:** Visual Studio Code, Node.js, PostgreSQL, Postman, JMeter, Git, Excel.

## Limitaciones Encontradas

El sistema no incluye autenticación ni notificaciones automáticas, limitaciones aceptadas dentro del alcance inicial. Está diseñado para guardar las quejas sin notificar por ningún medio al usuario o al administrador.

## Éxitos Alcanzados

Se implementó un sistema funcional que cumple los requisitos, validado mediante pruebas exhaustivas, demostrando la aplicación práctica de conocimientos técnicos y preparándolo para futuras expansiones.

## 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 tipo para refinar detalles. | 6 |
| 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 los entregables. Se configuró el repositorio del proyecto y se realizó un respaldo inicial del código. | 6 |
| 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, permitiendo recuperar todas las quejas registradas. Se realizaron pruebas con Postman y se implementó un manejo robusto de errores en el servidor. | 6 |
| 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 optimizó la carga de datos y se mejoró la presentación visual del panel. | 6 |
| 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 observaciones del tutor. Se actualizó el repositorio con los cambios finales y se documentó el proceso. | 6 |
| 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 |

# CONCLUSIONES

Durante el desarrollo del sistema de gestión de quejas y peticiones, se fortalecieron considerablemente las habilidades técnicas en el área de desarrollo web full stack. Particularmente, se adquirió experiencia en la construcción de APIs RESTful utilizando FastAPI y la integración con bases de datos PostgreSQL, aplicando principios de validación de datos, sanitización y control de errores en el backend.

Asimismo, se logró implementar exitosamente un frontend interactivo utilizando React.js, permitiendo a los usuarios registrar sus quejas de manera sencilla, mientras que el personal administrativo puede gestionar y visualizar el estado de cada petición.

Otro aspecto relevante fue el despliegue completo del sistema en un entorno real de producción, incluyendo la configuración de dominio personalizado, uso de SSL con Let’s Encrypt, y conexión entre frontend, backend y base de datos. También se

incorporó mecanismos de seguridad como validaciones en el lado del cliente y servidor, y realizó pruebas básicas ante ataques comunes como XSS y SQLi.

Gracias a estas prácticas, se adquirió una visión integral del ciclo de vida del desarrollo de software, desde la recolección de requisitos hasta la entrega de una solución funcional documentada, accesible y con un enfoque orientado al usuario.

# RECOMENDACIONES

Durante la experiencia, se identificaron oportunidades de mejora que pueden potenciar el sistema de gestión de quejas. En primer lugar, se recomienda implementar un sistema de notificaciones internas o vía correo electrónico para alertar al personal cuando se registre una nueva queja, mejorando así los tiempos de respuesta.

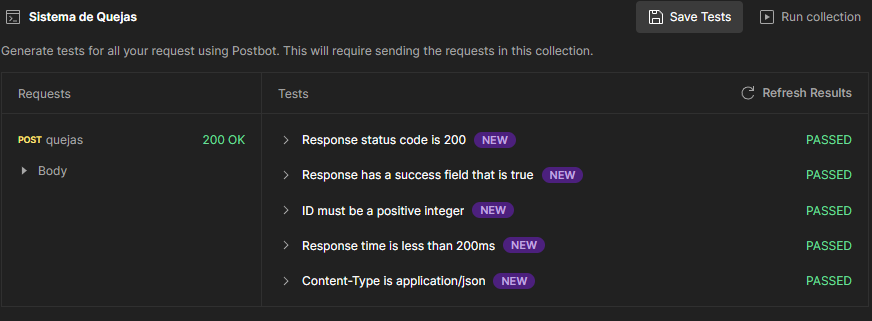
Otra sugerencia es incluir métricas estadísticas que permitan generar reportes automáticos sobre el número de quejas, su clasificación y tiempos de resolución, lo cual facilitaría la toma de decisiones estratégicas dentro de la organización.

Adicionalmente, se recomienda implementar un sistema de seguimiento para los usuarios que registran quejas, permitiéndoles consultar el estado de su solicitud mediante un código único o correo electrónico. Esto aumentaría la transparencia del proceso y fomentaría la participación ciudadana, ya que los usuarios sentirán que su mensaje está siendo atendido. Para esto, se podrían generar vistas públicas limitadas, protegidas por captchas o autenticación sencilla, garantizando al mismo tiempo la seguridad de los datos.

RESTful, lo que permitirá a la empresa mantenerse competitiva y alineada con las tendencias tecnológicas actuales.

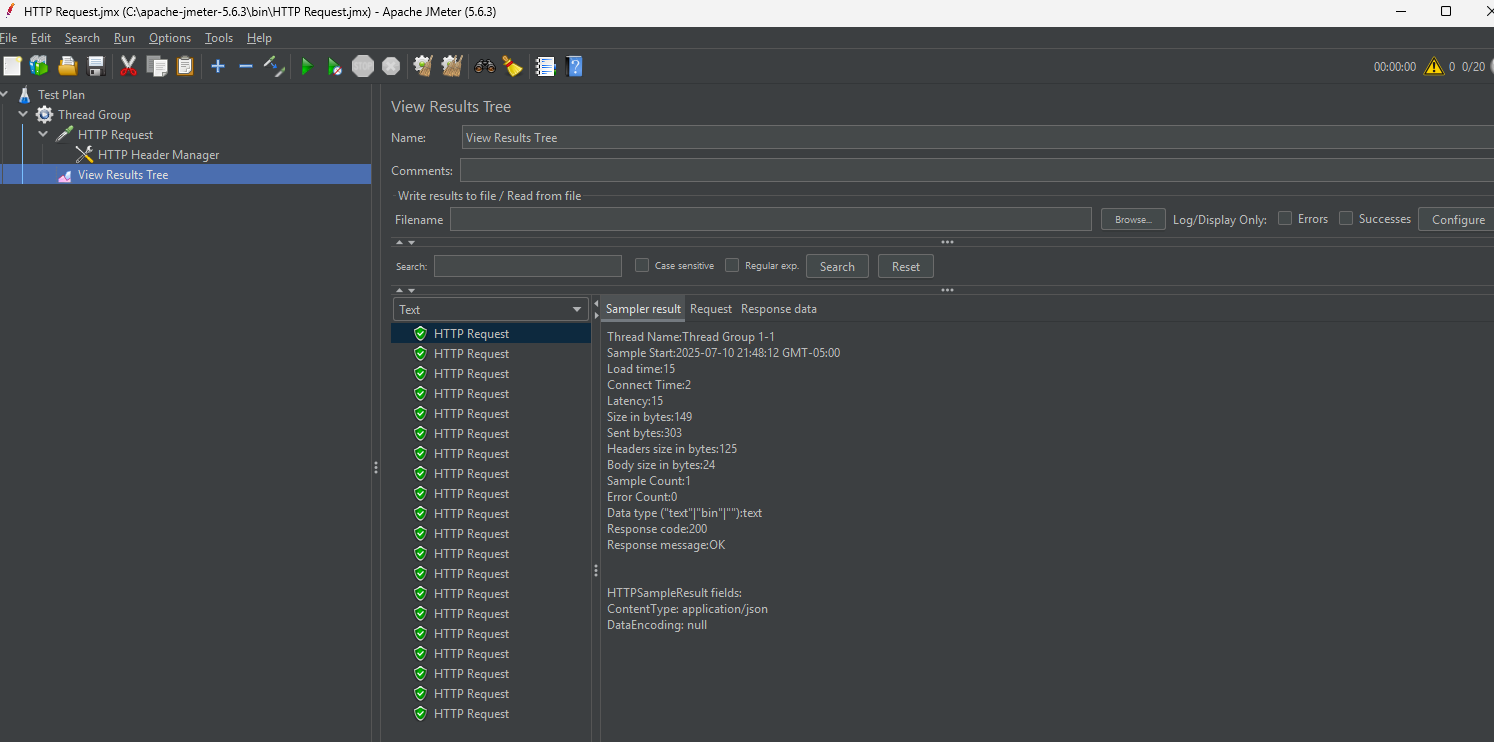
# ANEXOS

En la Fig. 1 se muestran las pruebas realizadas por medio de PostmanRunner. Estas pruebas tuvieron como propósito verificar que las quejas se envíen correctamente sin errores y devuelva respuestas claras y controladas. De esta manera, se asegura que el sistema sea capaz de gestionar situaciones anómalas sin comprometer su estabilidad o funcionalidad.



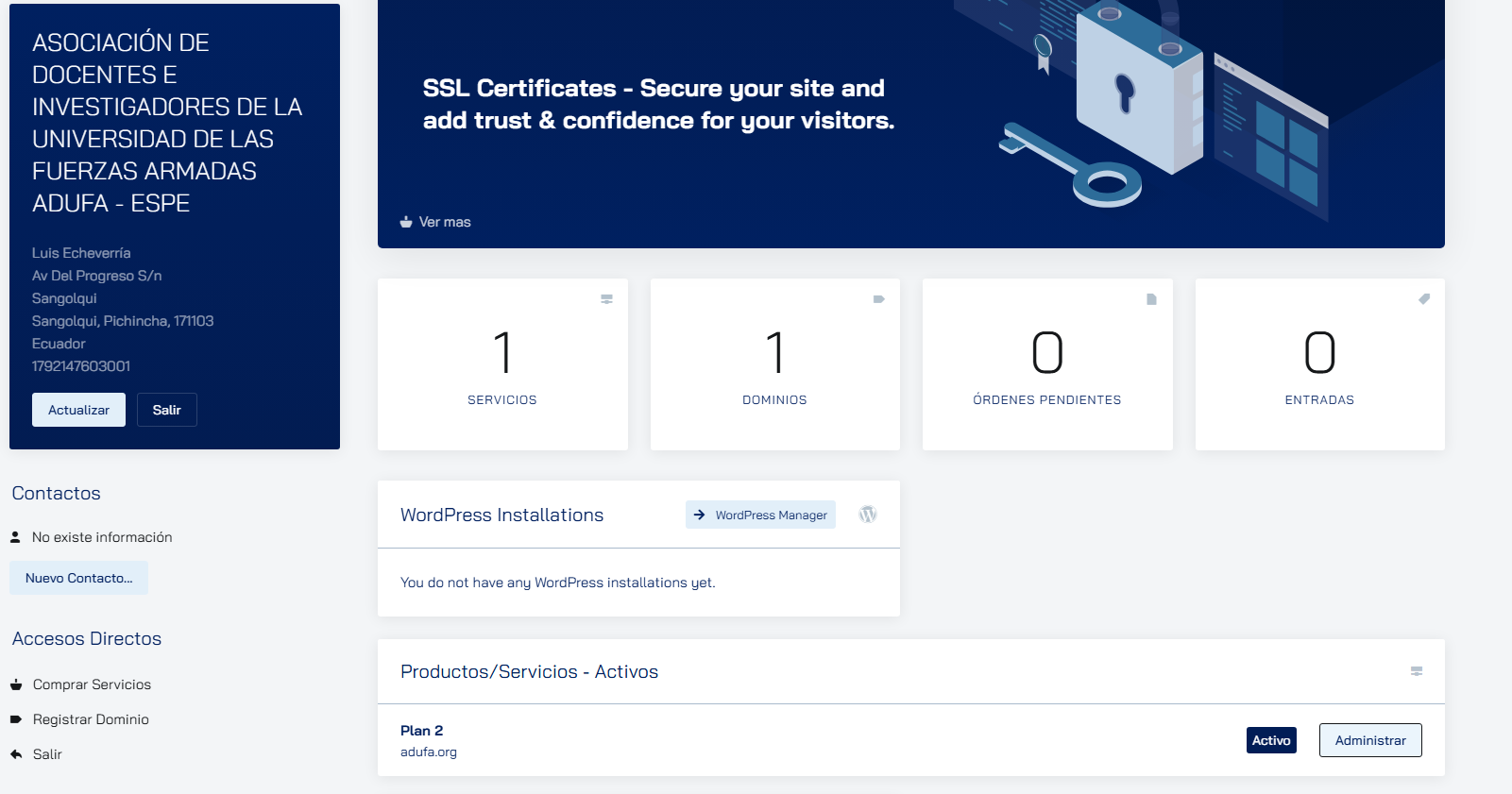
*Figura 1 Pruebas con Postman*

La Fig. 2 muestra las pruebas de carga simulando múltiples usuarios enviando quejas al mismo tiempo, usando herramientas especializadas para medir cómo respondía el servidor bajo presión



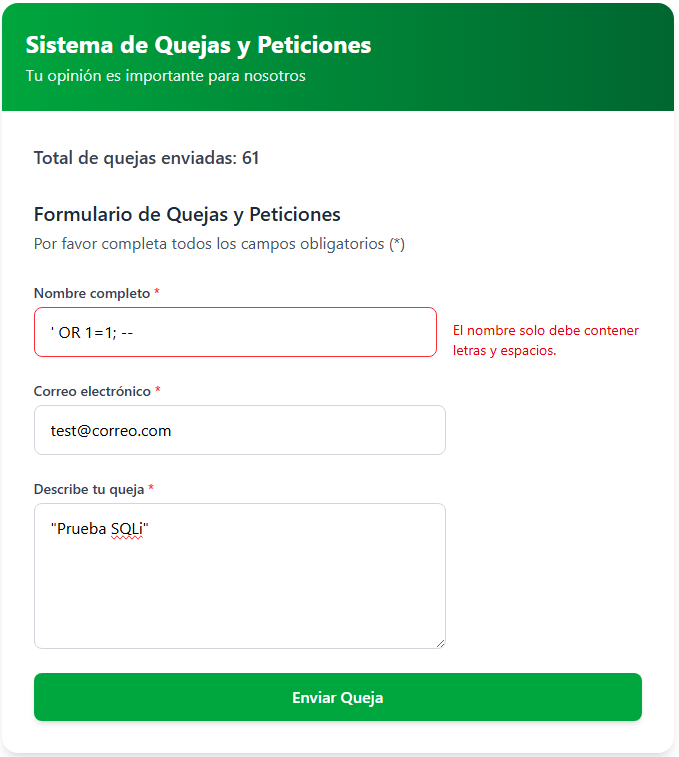
*Figura 2 Prueba con JMeter*

La Fig. 3 se puede observar el respectivo hosting que se contrato para poder subir la pagina web con los sistemas implementados durante las practicas no remuneradas



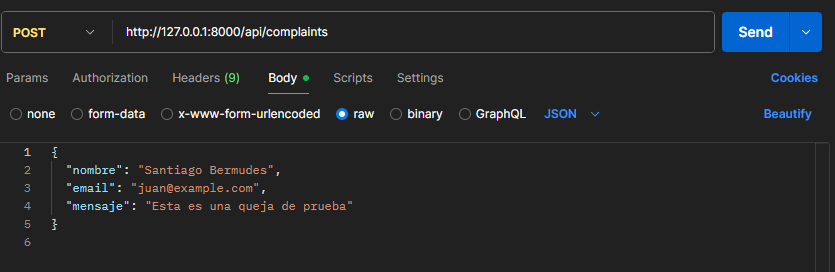
*Figura 3 Hosting de la pagina web*

En la Figura 4 se muestra un intento de inyección SQL utilizando una cadena maliciosa (' OR 1=1; --) con el fin de alterar la consulta a la base de datos. Esta prueba permitió validar que el sistema cuenta con medidas de protección adecuadas, como validaciones en los campos de entrada y uso seguro del ORM, bloqueando correctamente la amenaza y garantizando la integridad del sistema.



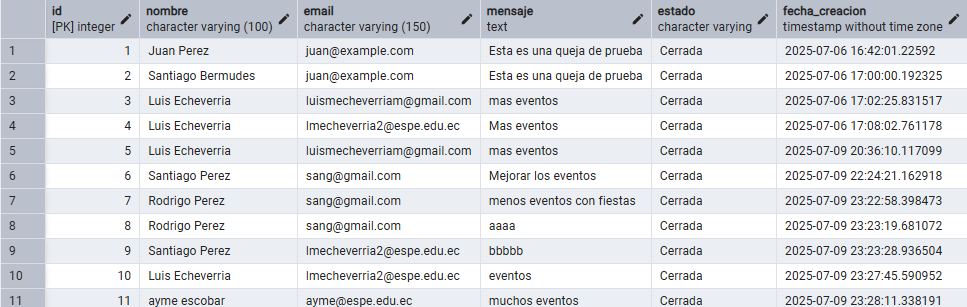
*Figura 4 Simulación de un ataque SQL*

La Figura 5 muestra el envío de una queja utilizando Postman, con el propósito de validar que el endpoint del sistema procesa correctamente los datos recibidos y almacena la información en la base de datos de forma adecuada.



*Figura 5 Envió de una queja por Postman*

La Figura 6 muestra el correcto almacenamiento de la información de las quejas en la base de datos, incluyendo el nombre del usuario, el correo electrónico, el mensaje enviado, la fecha de creación y el estado actual de la queja.



*Figura 6 Prueba del proceso Scripting a la senescyt*