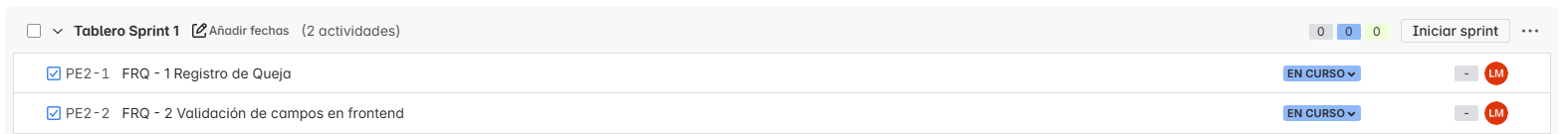
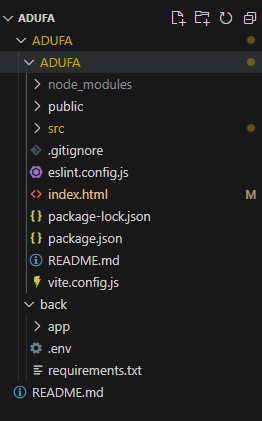
ACTIVIDADES DESARROLLADAS EN PRACTICAS PRE PROFESIONALES

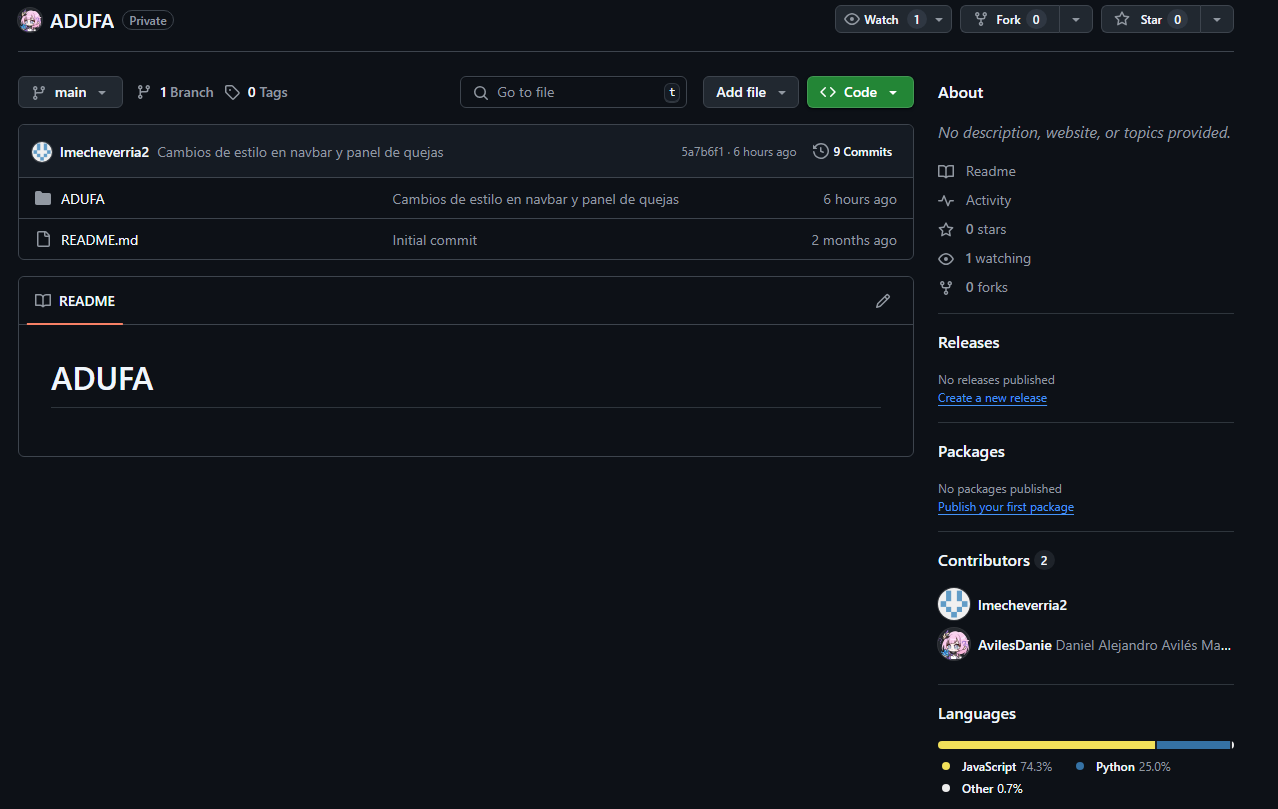
Sprint 1

**Lunes 21 de abril del 2025**



El primer día del sprint se dedicó a preparar el entorno de trabajo para el desarrollo del sistema de registro de quejas. Se instaló Node.js, una plataforma que permite ejecutar JavaScript en el servidor, junto con npm, su administrador de paquetes, para gestionar las bibliotecas y herramientas necesarias. Esta elección se basó en la popularidad de Node.js para aplicaciones web modernas y su compatibilidad con otras tecnologías del proyecto. Luego, se configuró Vite como la herramienta de empaquetado para el frontend, seleccionada por su velocidad en el desarrollo y compilación en comparación con otras alternativas como Webpack. Vite permite un servidor de desarrollo rápido y optimiza los tiempos de carga, lo que facilita pruebas rápidas durante la creación de la interfaz. Para la construcción de la interfaz, se optó por React.js, una biblioteca de JavaScript que permite crear componentes reutilizables y estructurar la aplicación de manera modular. React fue elegido por su flexibilidad y su amplio ecosistema, ideal para manejar interfaces dinámicas como formularios. Además, se integró Tailwind CSS, un framework de diseño basado en clases utilitarias, que agiliza la creación de estilos visuales sin necesidad de escribir CSS desde cero. Esto permitió diseñar una interfaz moderna y adaptable a diferentes dispositivos con un esfuerzo mínimo. También se creó un repositorio en GitHub para controlar las versiones del proyecto, asegurando que todos los cambios fueran rastreables y que el equipo pudiera colaborar de manera eficiente. Se configuró un archivo .gitignore para excluir archivos innecesarios, como la carpeta de dependencias (node\_modules), y se establecieron reglas de protección para la rama principal, garantizando que solo se aceptaran cambios revisados. Finalmente, se organizó la estructura inicial del proyecto, creando carpetas como src (para el código fuente), components (para componentes reutilizables de React), y pages (para páginas completas de la aplicación). Esta estructura se diseñó pensando en la escalabilidad, permitiendo agregar nuevas funcionalidades sin desorganizar el código. Se enfrentaron pequeños retos, como asegurar la compatibilidad de versiones entre Node.js y Vite, pero se resolvieron consultando la documentación oficial.

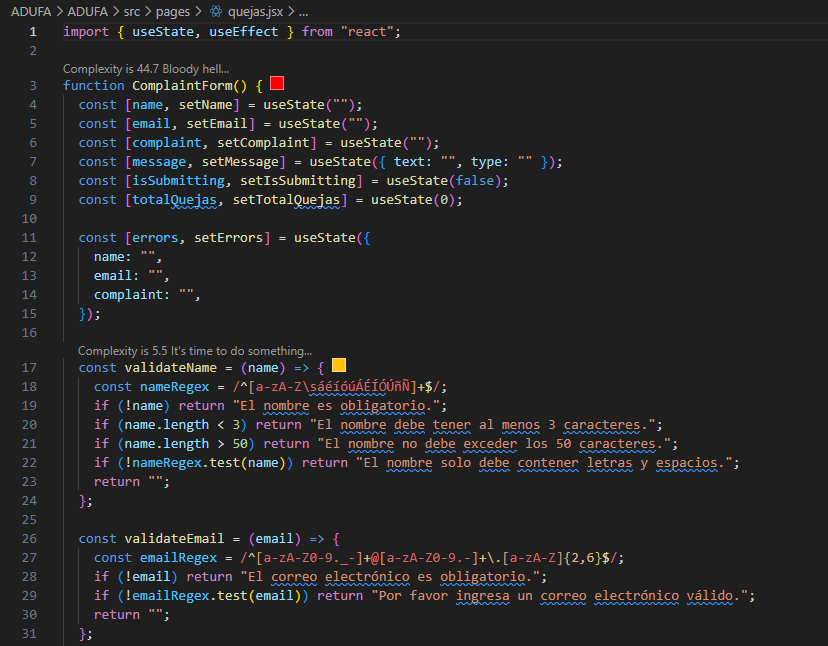




**Martes 22 de abril del 2025**

**Actividades realizadas:**

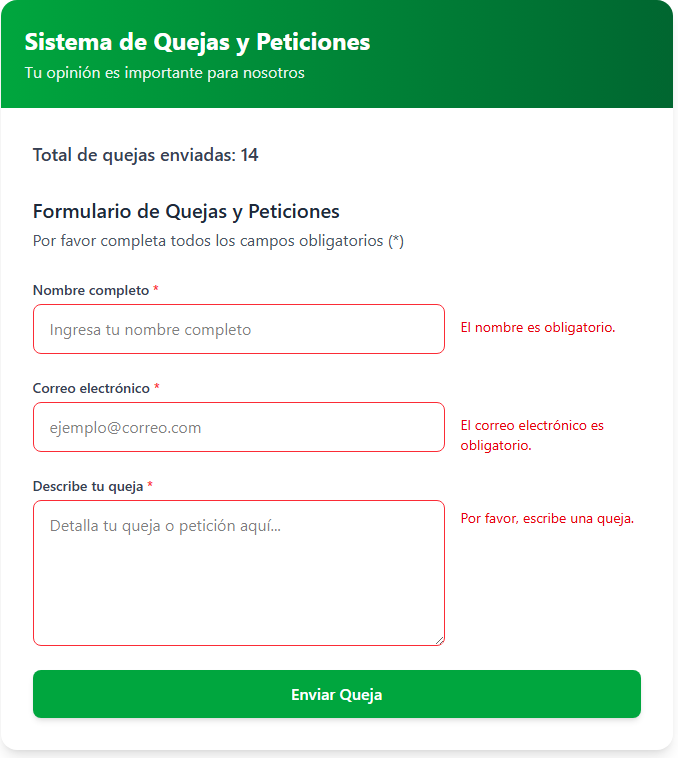
Desarrollo del formulario de ingreso de quejas:  
En este día, se desarrolló el componente principal del sistema: un formulario para que los usuarios ingresen sus quejas. Este formulario incluyó tres campos esenciales: un campo de texto para el nombre del usuario, un campo de correo electrónico con validación básica incorporada por el navegador, y un área de texto para el mensaje de la queja, que permite escribir textos largos. Cada campo se diseñó con un enfoque en la usabilidad, asegurando que los usuarios pudieran entender fácilmente qué información debían proporcionar. Para estilizar el formulario, se utilizaron clases de Tailwind CSS, que permitieron crear un diseño limpio, moderno y adaptable a diferentes tamaños de pantalla. Por ejemplo, se aplicaron bordes suaves, márgenes consistentes y colores que alinean con una experiencia de usuario agradable, como tonos neutros para los campos y botones destacados. El formulario se integró en la aplicación principal de React y se probó en el navegador para asegurar que se renderizara correctamente, verificando que los campos aparecieran en el orden correcto y que el diseño no se rompiera en pantallas más pequeñas, como las de dispositivos móviles. También se implementó un manejo básico de eventos para capturar lo que el usuario escribía en tiempo real, almacenando temporalmente los datos en el estado del componente. Esto permitió que el formulario fuera interactivo desde el inicio, mostrando al equipo un primer prototipo funcional. Los desafíos incluyeron ajustar los estilos para que el formulario fuera visualmente atractivo y funcional en diferentes navegadores, lo que requirió pruebas en Chrome, Firefox y Edge.



**Miercoles 23 de abril del 2025**

**Actividades realizadas:**

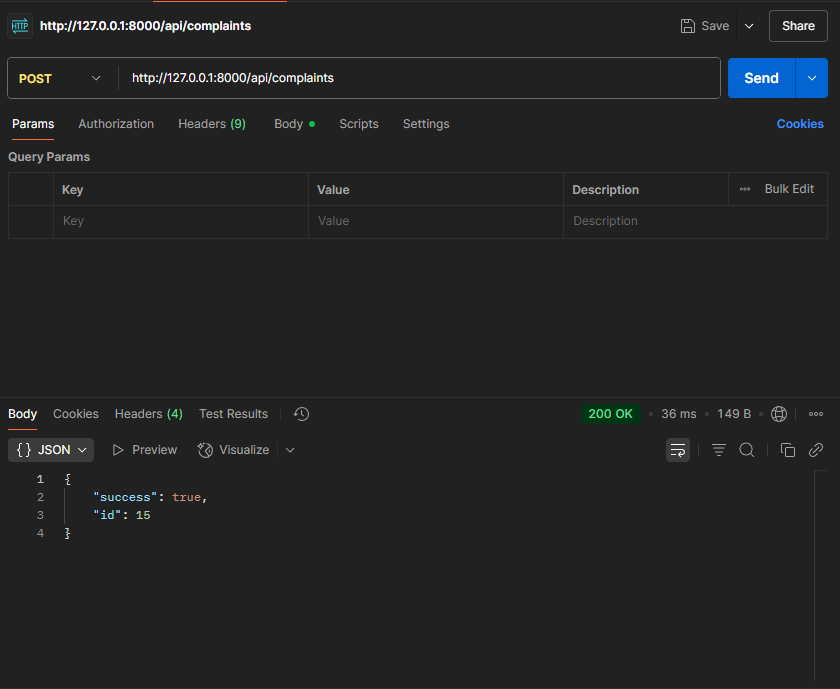
Aplicación de validaciones de los campos en el frontend:  
El enfoque de este día fue garantizar que los datos ingresados por los usuarios fueran válidos antes de enviarlos al servidor. Se implementaron reglas de validación en el frontend para los tres campos del formulario. Para el campo de nombre, se estableció que debía tener entre 3 y 50 caracteres, evitando entradas demasiado cortas o excesivamente largas que pudieran ser problemáticas. Para el correo, se verificó que cumpliera con un formato estándar de email, como “usuario@dominio.com”, utilizando técnicas de validación en el navegador y lógica adicional en JavaScript. El campo de mensaje se marcó como obligatorio, asegurando que los usuarios no pudieran enviar el formulario sin escribir una queja. Estas validaciones se diseñaron para mostrar mensajes de error en tiempo real, es decir, mientras el usuario escribía. Por ejemplo, si el nombre era demasiado corto, aparecía un mensaje como “El nombre debe tener al menos 3 caracteres” debajo del campo. Esto se logró utilizando las herramientas de estado y efectos de React, que permitieron actualizar dinámicamente la interfaz según los datos ingresados. Además, se desactivó el botón de envío del formulario cuando había errores, evitando que los usuarios enviaran información inválida. Este enfoque mejoró la experiencia del usuario al proporcionar retroalimentación inmediata. Las pruebas se realizaron ingresando datos incorrectos, como nombres de un solo carácter o correos sin el símbolo “@”, para confirmar que los mensajes de error aparecían correctamente y que el botón de envío permanecía deshabilitado hasta corregir los errores. Un desafío fue ajustar el tiempo de respuesta de las validaciones para que no fueran intrusivas, logrando un equilibrio entre rapidez y usabilidad.



**Jueves 24 de abril del 2025**

**Actividades realizadas:**

Configuración de backend con Node.js y Express:  
Se dio un paso importante al configurar el backend del sistema utilizando Node.js y Express, un framework que simplifica la creación de servidores web. Se creó un servidor básico que pudiera recibir solicitudes HTTP desde el frontend. El primer endpoint, o punto de conexión, fue POST /api/quejas, diseñado para recibir los datos del formulario en formato JSON, como el nombre, correo y mensaje del usuario. Para conectar el frontend con el backend, se utilizó una herramienta como fetch o axios, siendo esta última preferida por su facilidad para manejar respuestas y errores. Se realizaron pruebas enviando datos desde el formulario al servidor y verificando que el endpoint los recibiera correctamente, aunque en esta etapa los datos solo se registraban en la consola, ya que aún no se había configurado una base de datos. Se implementó un manejo inicial de errores en el servidor, como devolver mensajes claros si los datos enviados eran incorrectos. Un desafío fue configurar correctamente las políticas de acceso entre el frontend y el backend (CORS), ya que el navegador bloqueaba las solicitudes entre diferentes puertos locales. Esto se resolvió instalando un middleware de CORS en Express y asegurando que las conexiones fueran seguras. Las pruebas iniciales confirmaron que el frontend podía enviar datos al backend sin problemas, sentando las bases para la integración con la base de datos en el siguiente sprint.



**Viernes 25 de abril del 2025**

**Actividades realizadas:**

El último día del sprint se dedicó a probar todo el flujo del formulario, desde que el usuario lo llenaba hasta que los datos llegaban al servidor. Se verificó que las validaciones del frontend funcionaran correctamente, que los datos se enviaran al endpoint del backend, y que el servidor respondiera adecuadamente. Para mejorar la experiencia del usuario, se añadieron notificaciones visuales en la interfaz. Cuando una queja se enviaba con éxito, aparecía un mensaje verde con el texto “Queja enviada correctamente”, diseñado con colores y estilos que destacaban en la pantalla. Si ocurría un error, como una falla en la conexión al servidor, se mostraba un mensaje rojo con una explicación clara, como “Error de red, intenta de nuevo”. Estas notificaciones se diseñaron para ser temporales, desapareciendo después de unos segundos para no saturar la interfaz. Finalmente, se realizó el primer envío completo de los cambios al repositorio de GitHub, incluyendo tanto el código del frontend como del backend. El equipo escribió un mensaje de commit detallado, explicando que se había completado la funcionalidad básica del formulario y su conexión con el servidor. Las pruebas finales aseguraron que no hubiera errores críticos, y el sprint se cerró con un sistema funcional, aunque básico, listo para expandirse en las siguientes semanas.

