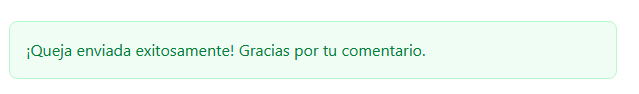
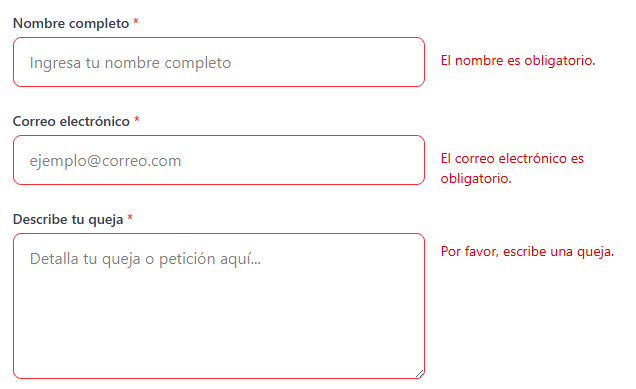
ACTIVIDADES DESARROLLADAS EN PRACTICAS PRE PROFESIONALES   
Sprint 3

Lunes 5 de mayo de 2025

**Actividades realizadas:**

Implementación de alertas visuales en el formulario:  
El sprint comenzó mejorando la experiencia del usuario al agregar notificaciones visuales en el formulario de quejas. Estas alertas se diseñaron para informar al usuario sobre el resultado de su acción, ya fuera un envío exitoso o un error. Para un envío exitoso, se mostró una alerta verde con un mensaje claro, como “Tu queja ha sido enviada con éxito”, que aparecía en la parte superior o inferior de la pantalla. Para errores, se diseñó una alerta roja con mensajes específicos, como “Por favor, corrige los errores en el formulario”. Estas notificaciones se crearon utilizando componentes personalizados en React, estilizados con Tailwind CSS para mantener la consistencia visual con el resto de la aplicación. Se probaron diferentes escenarios, como enviar un formulario válido o uno con errores, para asegurar que las alertas aparecieran en el momento correcto y desaparecieran después de unos segundos para no interrumpir la experiencia del usuario. Los desafíos incluyeron ajustar el tiempo de las notificaciones y garantizar que no se superpusieran con otros elementos de la interfaz, lo que se resolvió con pruebas exhaustivas en diferentes dispositivos.

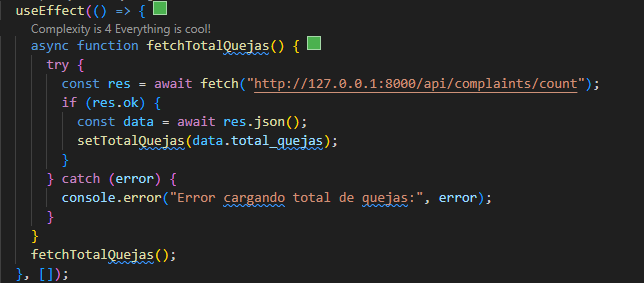




Martes 6 de mayo de 2025

**Actividades realizadas:**

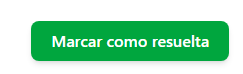
Manejo completo de errores en frontend:  
Se trabajó en fortalecer el manejo de errores en la interfaz para cubrir todas las posibles fallas que un usuario pudiera encontrar. Esto incluyó errores del servidor (como cuando el backend no respondía), problemas de conexión a internet, y validaciones incorrectas en el formulario. Cada tipo de error se asoció con un mensaje claro y específico, como “No se pudo conectar con el servidor, verifica tu conexión” o “El correo ingresado no es válido”. Para mejorar la visibilidad de los errores, los campos con problemas se resaltaron visualmente, por ejemplo, con un borde rojo alrededor del campo afectado. Esto ayudó a los usuarios a identificar rápidamente qué debían corregir. Se probaron escenarios extremos, como desconectar el servidor o enviar datos inválidos, para asegurar que los mensajes fueran útiles y que el sistema no se rompiera. Un desafío fue garantizar que los mensajes fueran comprensibles para usuarios no técnicos, lo que llevó a reescribir algunos textos para hacerlos más amigables.



Miercoles 7 de mayo de 2025

**Actividades realizadas:**

Agregado de columna editable de estado en QuejasPanel.jsx:  
En el panel administrativo, se añadió la capacidad de modificar el estado de las quejas directamente desde la tabla. Se incluyó una columna con un menú desplegable o botones que permitían cambiar el estado entre “Abierta” y “Cerrada”. Esta funcionalidad se diseñó para ser intuitiva, con indicadores visuales claros, como un icono que mostraba que el campo era editable o botones con colores distintivos. Cuando un administrador seleccionaba un nuevo estado, se activaba un botón de “Guardar” para confirmar el cambio, evitando modificaciones accidentales. La interfaz se diseñó con Tailwind CSS para que los elementos fueran visualmente consistentes con el resto del sistema, utilizando colores y estilos que facilitaban la interacción. Las pruebas iniciales aseguraron que el cambio de estado se reflejara correctamente en la interfaz, aunque aún no se actualizaba en la base de datos, lo que se abordó el día siguiente.

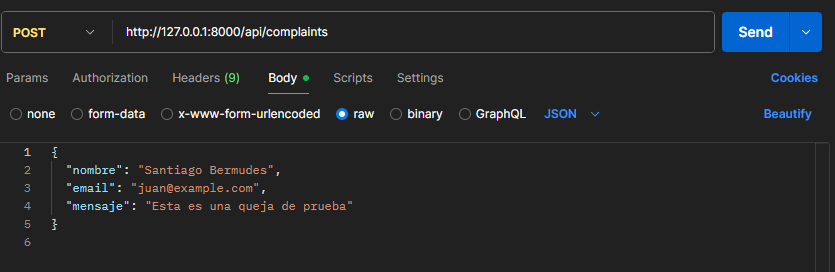




Jueves 8 de mayo de 2025

**Actividades realizadas:**

Validación del nuevo estado recibido:  
Se desarrolló la lógica en el backend para actualizar el estado de una queja en la base de datos. Se creó un nuevo endpoint que permitía modificar el campo de estado para una queja específica, asegurando que solo se aceptaran valores válidos (“Abierta” o “Cerrada”). Si se enviaba un valor incorrecto, el servidor devolvía un mensaje de error claro, como “Estado no válido”. Para probar esta funcionalidad, se usó Postman para simular solicitudes al servidor, enviando diferentes estados y verificando que la base de datos se actualizara correctamente. También se comprobó que no se pudieran modificar quejas inexistentes, manejando estos casos con mensajes de error apropiados. Los desafíos incluyeron garantizar que las actualizaciones fueran seguras y no afectaran otros datos, lo que se logró con pruebas exhaustivas y una lógica robusta en el servidor.



Viernes 9 de mayo de 2025

**Actividades realizadas:**

Pruebas completas de cambio de estado:  
El último día se dedicó a probar toda la funcionalidad de cambio de estado desde el panel administrativo. Se verificó que los administradores pudieran seleccionar un nuevo estado, guardar el cambio, y que este se reflejara tanto en la interfaz como en la base de datos. Se realizaron pruebas con múltiples quejas para asegurar que el sistema manejara correctamente grandes volúmenes de datos. También se revisó el código para garantizar que fuera claro y bien organizado, eliminando cualquier redundancia. Se documentaron todos los avances del sprint en un archivo detallado, y se subieron los cambios al repositorio de GitHub con un mensaje que resumía las nuevas funcionalidades, como las notificaciones y la capacidad de actualizar estados. El sprint se cerró con un sistema más interactivo y funcional, listo para las pruebas finales en los siguientes sprints.

