**DOCUMENTACIÓN INTEGRACIÓN**

**PONENTES**

**JUAN DAVID SERRANO**

**MIGUEL ANGEL ROMERO**

**LAURA GAONA CASTAÑO**

**INGENIERIA DE SOFTWARE**

**UNIVERSIDAD MANUELA BELTRAN**

**2025**

**1. Introducción**

Se encarga de unir los diferentes componentes del proyecto de la tienda online de libros "Entrelineas". Su objetivo principal es asegurar que la comunicación entre el FrontEnd, el BackEnd y la base de datos sea fluida y eficiente, garantizando la estabilidad y el correcto funcionamiento del sistema.

**2. Responsabilidades del Grupo**

El grupo tiene las siguientes responsabilidades claves:

**a) Conectar BackEnd y FrontEnd**

* Uso de fetch () o Axios para realizar solicitudes al servidor.
* Verificación de logs en el servidor para detectar errores de conexión.
* Pruebas de comunicación mediante Postman para validar endpoints.

**b) Pruebas de Funcionamiento**

* Realizar pruebas completas de integración para verificar la funcionalidad de cada módulo.
* Validar que los endpoints definidos devuelvan respuestas correctas.
* Confirmar el correcto almacenamiento y recuperación de datos en la base de datos.

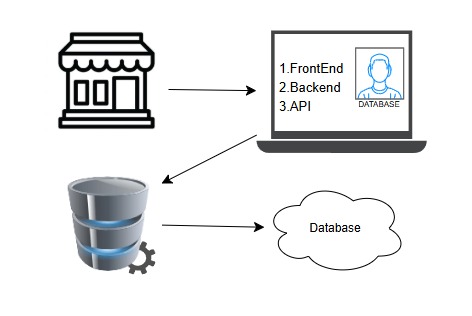
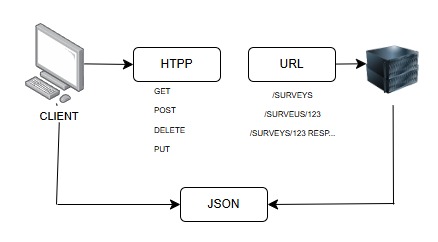
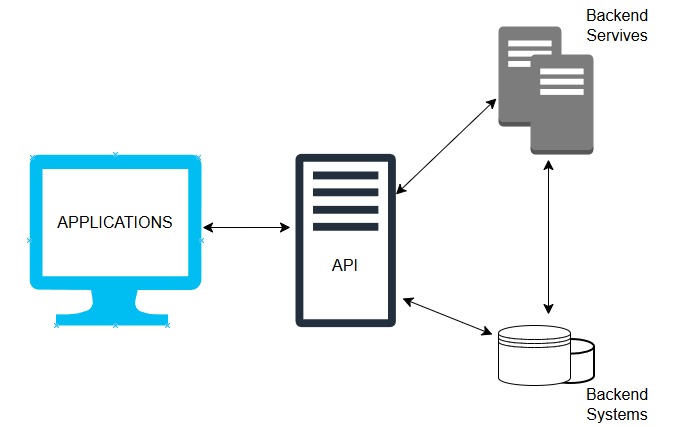
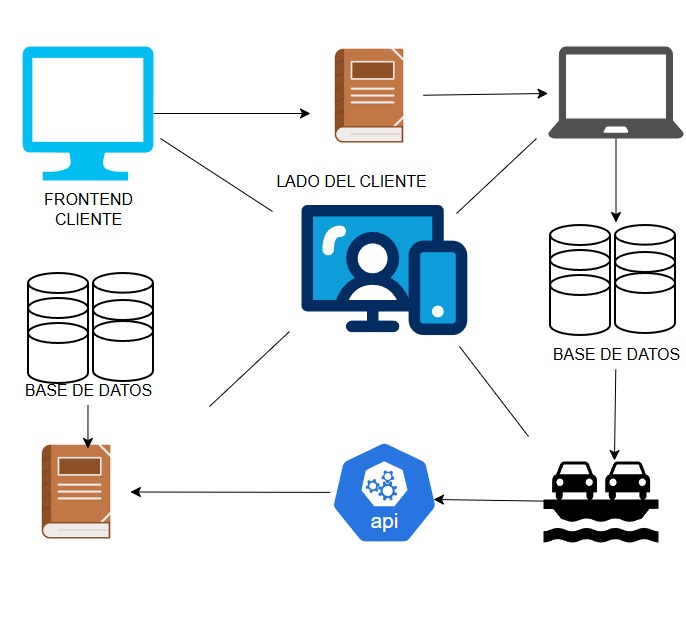
**c) Identificación de Errores y Documentación**

* Documentar los problemas encontrados durante las pruebas.
* Coordinar ajustes entre el BackEnd y el FrontEnd si es necesario.
* Generar reportes con posibles mejoras y soluciones.

**3. Flujo de Integración**

1. **Planificación**
   * Definir endpoints y documentarlos.
   * Crear una guía de consumo de APIs.
2. **Verificación del BackEnd**
   * Asegurar que los endpoints funcionen correctamente.
   * Documentar respuestas esperadas y errores posibles.
3. **Implementación en el FrontEnd**
   * Visualizar datos en la interfaz.
   * Agregar botones y acciones necesarias.
   * Ajustar diseño y comportamiento según pruebas.
4. **Pruebas con Postman**
   * Validar API sin depender del FrontEnd.
   * Evaluar seguridad y manejo de autenticación.
   * Detectar errores en la comunicación con la base de datos.
5. **Validación Final**
   * Asegurar integridad de los datos.
   * Revisar errores y documentar mejoras.
   * Ajustes finales antes del despliegue.

**4. Diagramas**

****

**Errores generados por POSTMAN**

Desde la integración continua del software el equipo de trabajo definió posibles errores comunes, explicando la posible causa que puede generar los diferentes tipos de errores. Posteriormente se pretende implementar las pruebas en el entorno de pruebas PostMan, documentando los errores que se generen en ejecución.

**Errores de Cliente:**

* **400 Bad Request**:

Estos errores son errores comunes, que se generan por la falta de especificación de algunos datos o mal manejo de excepciones que detecten posibles fallos del usuario.

* + Datos faltantes en el cuerpo de la petición (ejemplo: registrar un libro sin título).
  + Formato incorrecto en los datos enviados (ejemplo: enviar un string en un campo numérico).
* **401 Unauthorized**:

Estos errores pretenden advertir de una vulnerabilidad en el sistema, que no permite el correcto flujo del programa, porque lasa API encargadas de la autenticación no están bien definidas.

* + No se envió un token de autenticación en endpoint protegidos.
  + Token JWT inválido o expirado.
* **403 Forbidden**:

Son errores de prohibición, quiere decir, que no todos los usuarios pueden realizar acciones a las cuales no tienen permisos por parte del administrador. Tienen como fin, delimitar las acciones permitidas para diferentes usuarios.

* + El usuario intenta acceder a un recurso sin permisos (ejemplo: un cliente intenta eliminar un libro).
* **404 Not Found**:

Los Errores endpoint, se producen por un error en la base de datos, generalmente por datos duplicados, relaciones lógicas de la base de datos mal establecidas, por consiguiente, la deficiente normalización de esta provoca errores de elementos vacíos.

* + Endpoint incorrecto en la solicitud.
  + Intentar obtener un libro con un ID inexistente.
* **405 Method Not Allowed**:

Los errores de métodos http se producen por un incorrecto manejo en los endpoint de los formularios o peticiones que se requieran por parte del backend, desde una consulta a la base de datos, hasta la entrega de la información requerida.

* + Uso incorrecto del método HTTP (ejemplo: enviar POST en un endpoint que solo permite GET).

**Errores de servidor:**

**500 Internal Server Error:**

Los errores internos del servidor se generan por un control de versiones mal configurado, o un manejo de excepciones, también es producido por que errores en las credenciales o ejecución del PostgreSQL

* Un error inesperado en el código (ejemplo: intentar acceder a una variable undefined).
* Errores no manejados en controladores o middleware.

**502 Bad Gateway:**

Los errores de conexión son un problema común en las paginas web, especialmente por errores en la arquitectura del software, o algún error en los puertos de express.

* El servidor Express no está corriendo o no está disponible.
* Problemas de conexión entre el backend y la base de datos.

**503 Service Unavailable:**

Error común debido a la falta de escalabilidad del sistema, una incorrecta documentación en la capacidad de respuesta del software ante una alta demanda, genera errores en el servidor, también ocurre cuando la infraestructura de red no es lo suficientemente robusta, para mantener el sistema un 99.9% disponible.

* Sobrecarga del servidor o mantenimiento del sistema.

**6. Conclusión**

Garantizar la estabilidad y funcionalidad del sistema. A través de pruebas exhaustivas y documentación detallada, se busca optimizar el desarrollo colaborativo y asegurar una experiencia de usuario eficiente en la tienda online de libros "Entrelineas".