**DOCUMENTACIÓN INTEGRACIÓN**

**PONENTES**

**JUAN DAVID SERRANO**

**MIGUEL ANGEL ROMERO**

**LAURA GAONA CASTAÑO**

**INGENIERIA DE SOFTWARE**

**UNIVERSIDAD MANUELA BELTRAN**

**2025**

**WebSite Entrelineas**

**1. Introducción**

Esta documentación detalla el proceso de integración del sitio web de la tienda online de libros, enfocándose en la comunicación entre el **frontEnd, backEnd y la base de datos**, así como en el control de versiones mediante Git. La integración de estos componentes es fundamental para garantizar un funcionamiento estable y eficiente del sistema, facilitando el desarrollo colaborativo y asegurando que todas las partes del proyecto estén correctamente sincronizadas.

**2. Estructura del Proyecto**

El repositorio del proyecto está organizado en diferentes ramas, cada una con un propósito específico:

* **FrontEnd:** Contiene los archivos **HTML, CSS y JavaScript**, utilizando frameworks como React o Vue.js para la construcción de la interfaz de usuario.
* **Backend:** Almacena la lógica del servidor, implementada en **Node.js con Express.js**, así como la configuración de la base de datos.
* **Documentación:** Incluye todos los archivos relacionados con la documentación del proyecto, garantizando un desarrollo ordenado y comprensible.
* **Integración:** Es la rama encargada de unir las distintas partes del proyecto y realizar pruebas de funcionamiento antes de desplegar nuevas versiones.

**3. Pruebas de Integración**

Antes de realizar el despliegue de nuevas versiones, es fundamental ejecutar pruebas en la rama de integración para validar el correcto funcionamiento del sistema. Entre las pruebas clave se incluyen:

* **Verificación de la comunicación entre el FrontEnd y el BackEnd:** Se realizarán pruebas para asegurarse de que las solicitudes y respuestas entre ambos componentes sean correctas.
* **Validación del almacenamiento de datos en la base de datos:** Se comprobará que la información se almacene y recupere correctamente.
* **Revisión del control de versiones con Git:** Se asegurará que todas las modificaciones se integren de manera organizada y sin conflictos.

**4. Pruebas con Postman**

Para la validación del **BackEnd**, se utilizará la herramienta **Postman**, que permite probar APIs RESTful de manera eficiente. A través de Postman, se realizarán peticiones HTTP (GET, POST, PUT, DELETE) para verificar que el servidor responda correctamente antes de integrar el FrontEnd.

El uso de Postman proporciona varias ventajas, tales como:

* **Pruebas sin dependencia del FrontEnd:** Permite evaluar la funcionalidad del BackEnd sin necesidad de una interfaz de usuario.
* **Depuración de errores:** Facilita la identificación y corrección de errores en la comunicación con la base de datos.
* **Automatización de pruebas:** Se pueden crear colecciones de pruebas para ejecutar de forma automatizada.

**5. APIs de Testing y Conexión entre Componentes**

Para la integración entre la base de datos y el BackEnd, se empleará una API REST desarrollada en **Node.js con Express.js**. Esta API será responsable de manejar **autenticaciones, validación de datos y operaciones CRUD** sobre la base de datos.

Para la integración entre el **FrontEnd y el BackEnd**, se utilizará la API fetch() de **JavaScript**, que permite enviar y recibir datos de manera asincrónica. Este mecanismo es clave para garantizar un flujo de información eficiente y una experiencia de usuario fluida.

Las principales consideraciones en esta integración incluyen:

* **Envío correcto de peticiones HTTP desde el FrontEnd al BackEnd.**
* **Manejo adecuado de respuestas para actualizar la interfaz de usuario dinámicamente.**
* **Almacenamiento y gestión de tokens JWT para autenticación segura y autorización de peticiones.**

**6. Uso de Fetch en la Integración**

La función fetch() es una herramienta clave en la comunicación entre el FrontEnd y el BackEnd. A continuación, se demostraremos como se implementaría:



**7. Conclusión**

Una integración adecuada entre el **FrontEnd y el BackEnd** es crucial para garantizar el correcto funcionamiento de la aplicación. El uso de herramientas como **Postman para pruebas de API** y fetch() para la comunicación asincrónica permite optimizar la validación de la funcionalidad antes de realizar despliegues. Asimismo, la gestión de control de versiones mediante Git asegura que el desarrollo sea ordenado y colaborativo.

Con este enfoque, buscamos garantizar una integración efectiva, segura y escalable del sistema, permitiendo que la tienda online de libros **Entrelineas** ofrezca una experiencia de usuario fluida y eficiente.