Desafíos y Decisiones del Backend - AI Challenge

1. Elección de Tecnologías

Java con Spring WebFlux fue elegido por su enfoque reactivo, ideal para manejar flujos de datos asincrónicos, escalabilidad y eficiencia en el consumo de recursos.

Arquitectura limpia se adoptó para mantener la separación clara de responsabilidades, facilitando el mantenimiento, las pruebas y la extensibilidad.

2. Modelado de Dominio

- Se definieron objetos de dominio como Item, Seller para representar las entidades claves de la aplicación.
- Se implementó un caso de uso principal: Buscar productos, con la posibilidad de escalar a otros como Ver detalle de producto y Enrolar favoritos.

3. Persistencia y Exposición

- Como el reto no requería una base de datos real, se simuló el comportamiento de una fuente de datos mediante un repositorio en memoria, preparados para ser reemplazados por una integración con una base de datos real o API externa.

4. Validaciones y Manejo de Errores

- Se implementó validación para detectar cuando no se encuentran resultados (ITEM_NOT_FOUND) y se devolvió el código correspondiente con un mensaje claro para el frontend.
- Se estructuraron respuestas de error mediante una clase personalizada que permite escalar el manejo de excepciones técnicas o de negocio.

5. Desafíos Encontrados

- **Transformación de datos**: Fue necesario mapear correctamente los datos a estructuras esperadas por el frontend, manteniendo un formato amigable para su visualización.

Desafíos y decisiones del Frontend

1. Elección de Tecnologías

- Se utilizó **React** con **Vite** como base del proyecto por su simplicidad y rendimiento.
- Para los estilos, fue elegido **Tailwind CSS** por su flexibilidad, rapidez en prototipado y capacidad responsiva.
- Se empleó **React Router DOM** para permitir una navegación fluida entre páginas (SPA).

2. Estructura del proyecto:

- **Componentes:** Se separaron elementos reutilizables como SearchBar, ItemCard, y EmptyResultMessage para mantener un código limpio y modular.
- **Páginas:** HomePage para la búsqueda y ItemDetail para el detalle del producto seleccionado.
- La organización en carpetas permitió escalar el código fácilmente.

3. UX/UI:

- Inspirado en el diseño de MercadoLibre, se incluyó una barra amarilla en el encabezado con la caja de búsqueda centrada.
- Las tarjetas de productos fueron diseñadas para mostrar la información clave como: título, descripción, precio, vendedor y formas de pago.
- Se agregó una página de detalle con una galería de imágenes y un botón para volver.

4. Responsividad:

- Todo el layout fue adaptado para funcionar en diferntes dispositivos móviles y de escritorio usando clases responsivas de Tailwind como sm:, md:, etc.

5. Estado y Almacenamiento:

- Se utilizó useState para el manejo del estado local y localStorage para almacenar la selección de producto, permitiendo mantener el detalle disponible tras la navegación.

Challenges and Solutions

1. Configuración de Tailwind:

- Inicialmente npx tailwindcss init -p no funcionaba en Windows.
- Solución: reinstalé Tailwind v3 con npm install -D tailwindcss@3 y configuré correctamente el archivo tailwind.config.js con la propiedad content.

2. Manejo de errores del backend:

- Cuando no había resultados, el backend respondía con un código 400.
- Se detectó este error en el frontend y se mostró un mensaje claro para el usuario usando EmptyResultMessage.

3. Navegación y persistencia:

- Quería evitar recargar la búsqueda anterior tras ir al detalle y volver.
- Solución: Se guardó la lista de resultados y el ítem seleccionado en localStorage, permitiendo persistencia entre vistas.

4. Ajuste visual en componentes:

- Los estilos inline dificultaban la responsividad.
- Se migraron los estilos a clases de Tailwind para facilitar el ajuste visual y consistencia.

Este desarrollo me permitió fortalecer habilidades en React moderno, gestión de estado, diseño responsivo con Tailwind y enrutamiento con SPA. El proyecto está preparado para escalar y permite incorporar mejoras futuras como paginación, historial de búsquedas o autenticación de usuarios.