

Prueba Técnica React + Python — Mini Ecommerce

Objetivo

Construir un pequeño **ecommerce** utilizando **React** para el frontend y **Python (Django o FastAPI)** para el backend.

La prueba busca evaluar tu conocimiento en **manejo de estado**, **comunicación con API**, **estructura de componentes** y **buenas prácticas de desarrollo FullStack**.

Requerimientos Funcionales

1. Página de listado de productos

- Mostrar productos obtenidos desde el backend (endpoint `/products`).
- Mostrar nombre, precio y botón para agregar al carrito.

2. Carrito de compras

- Permitir agregar productos.
- Mostrar nombre, precio, cantidad editable, subtotal y total general.
- Permitir eliminar productos del carrito.

3. Guardar carrito

- Incluir un botón o acción para **guardar el carrito**.
- Al presionar “Guardar carrito”, enviar los datos (productos y cantidades) al backend mediante un endpoint (`/cart` o similar) y **almacenarlos en la base de datos**.
- Mostrar un mensaje de confirmación al usuario (por ejemplo, “Carrito guardado correctamente”).

4. Persistencia

- Mantener el carrito en `localStorage` hasta que se guarde exitosamente en el backend.
-

Requerimientos No Funcionales

- **Frontend:** React (preferiblemente con TypeScript).
 - **Backend:** Python con Django (o Django REST Framework) o FastAPI.
 - Base de datos simple (SQLite o similar).
 - Código claro, modular y bien estructurado.
 - Uso de Hooks y manejo adecuado del estado global (Context o Zustand).
 - Estilos no obligatorios (plus si se agregan).
-

Criterios de Evaluación

- Lógica y manejo de estado en el frontend.
 - Comunicación y flujo de datos entre frontend y backend.
 - Estructura del proyecto y claridad del código.
 - Funcionalidad completa y persistencia de datos.
 - Buenas prácticas de desarrollo (nombres claros, componentes reutilizables, separación de capas).
 - Extras: TypeScript, pruebas con Jest/RTL., Docker, autenticación, etc.
-

Entregable

- Repositorio público con dos carpetas principales:
 - `/frontend` → Código React
 - `/backend` → Código Python (Django o FastAPI)
 - Archivo `README.md` con:
 - Instrucciones para instalación y ejecución.
 - Descripción técnica (frameworks, librerías, decisiones de arquitectura).
-

Plus Opcionales

- Backend con endpoints adicionales (editar/eliminar carritos, paginación, filtros).
- Context global (React Context, Zustand o Redux).
- Uso de TypeScript.
- Pruebas unitarias (Jest, React Testing Library, Pytest).
- Dockerfile o docker-compose.
- Estilos personalizados o uso de librerías UI (Material UI, Tailwind, etc.).

Tiempo sugerido

6 a 8 horas (más si desea agregar extras).

■ Resultado esperado

1. Ver productos obtenidos desde el backend.
2. Agregar/modificar cantidades en el carrito
3. Guardar el carrito en el backend y mostrar el mensaje de confirmación.
4. Persistir el carrito en localStorage antes de guardarlo.