

**Nombre:** Stefano Falvo

**Código:** 6872

## Arquitectura del Proyecto

### 1. Arquitectura General

El proyecto sigue una arquitectura modular basada en una separación clara entre frontend, backend y servicios externos como PayPal y MongoDB Atlas. Se organiza de la siguiente manera:

- **Frontend:** React.js, manejando la interfaz de usuario y la lógica del cliente.
- **Backend:** Node.js con Express, gestionando la lógica de negocio y la comunicación con la base de datos.
- **Base de Datos:** MongoDB Atlas, donde se almacenan los datos de usuarios, productos, pedidos y más.
- **Sistema de Pagos:** Integración con PayPal para procesar transacciones seguras.
- **Machine Learning:** Algoritmo de recomendaciones para mejorar la experiencia del usuario.

### 2. Frontend

El frontend está desarrollado en **React.js**, estructurado en carpetas organizadas de la siguiente manera:

- **components/**: Contiene los componentes reutilizables de la aplicación, como botones y formularios.
- **pages/**: Incluye las vistas principales como Login, Dashboard y Checkout.
- **store/**: Gestión del estado con Redux para administrar datos compartidos.
- **config/**: Configuraciones generales del proyecto.

El frontend se comunica con el backend a través de **API RESTful** y consume datos mediante `fetch` o `axios`.

### 3. Backend

El backend está desarrollado en **Node.js con Express**, estructurado en módulos:

- **controllers/**: Implementan la lógica de negocio para manejar productos, usuarios y pedidos.

- **models**: Definen la estructura de la base de datos con MongoDB.
- **routes**: Gestionan las solicitudes API con endpoints RESTful.

El backend maneja la autenticación, el procesamiento de pedidos y la lógica de negocio de la tienda en línea.

#### 4. Base de Datos en MongoDB Atlas

La base de datos está organizada en colecciones:

- **Users**: Almacena información de los usuarios registrados.
- **Products**: Contiene datos de los productos en la tienda.
- **Orders**: Guarda los pedidos realizados por los clientes.
- **Cart**: Administra los productos añadidos al carrito de compras.
- **Reviews**: Contiene reseñas de productos realizadas por clientes.

Las relaciones entre colecciones se manejan mediante referencias (`ObjectId``) para optimizar las consultas.

#### 5. Integración con PayPal

Para permitir pagos seguros, se ha integrado la API de PayPal en el backend. La funcionalidad incluye:

- Redirección del usuario a PayPal para completar el pago.
- Confirmación y validación de pagos exitosos.
- Registro del estado de la transacción en la base de datos.

#### 6. Sistema de Recomendaciones con Machine Learning

El sistema de recomendaciones utiliza técnicas de Machine Learning para sugerir productos a los usuarios basados en sus preferencias y comportamiento. El proceso incluye:

- **Análisis de datos**: Captura de datos de interacción del usuario con la tienda.
- **Modelo de recomendación**: Algoritmo que predice productos relevantes para cada usuario.
- **Optimización y despliegue**: Integración con el backend para ofrecer recomendaciones en tiempo real.

#### 7. Seguridad y Optimización

Para garantizar un alto rendimiento y seguridad en MongoDB Atlas y el backend:

- Se utilizan **\*\*índices\*\*** en los campos más consultados, como `email` en Users y `category` en Products.
- Se implementa **\*\*cifrado de contraseñas\*\*** con `bcrypt`.
- Se restringen accesos a la base de datos mediante **\*\*roles y permisos\*\*** en MongoDB Atlas.