Crodi Shop

Tienda Online

Metodologías:

* Kanban: Mantener un flujo constante de trabajo (Excel).
* MVP (Minimum Viable Product): Progresar de manera priorizada y eficiente (Word).
* Timeboxing: Usar tus 2 horas diarias de manera estructurada (Reloj).

Herramientas: Python, Visual Studio Code, Django, React, APIs, GitHub

Alcance del proyecto:

* Mostrar un catálogo de productos.
* Agregar productos al carrito.
* Procesar pedidos.

Etapas del MVP:

* Etapa 1: Backend básico con Django.

Configura el entorno y crea las APIs para manejar productos, usuarios y pedidos.

* Etapa 2: Frontend básico con React.

Diseña las páginas principales y conecta las APIs con el front-end.

* Etapa 3: Integración.

Prueba y ajusta la funcionalidad del flujo completo (usuarios, carrito, pedidos).

Escalabilidad del proyecto:

* Sistema de pagos.
* Diseño responsivo.
* Funcionalidades avanzadas como filtros de búsqueda.

Timeboxing:

* Bloque 1 (90 minutos): Trabaja en tareas prácticas del proyecto.
* Bloque 2 (30 minutos): Haz pruebas, documentación o revisa el trabajo del día.

1. Configura el entorno de Django y React

Objetivo: Preparar los entornos de desarrollo para que puedas comenzar a trabajar tanto en el backend (Django) como en el frontend (React).

1. Crea modelos para productos y usuarios en Django

Objetivo: Definir los modelos de Producto y Usuario en Django, configurando las bases de datos y aplicando migraciones para que los datos puedan almacenarse correctamente.

1. Implementa vistas y APIs básicas con Django REST Framework.

Objetivo: Configurar Django REST Framework (DRF) y crear endpoints básicos para manejar usuarios y productos en la tienda online.

1. Diseña el esquema inicial de la base de datos

Objetivo: Definir y estructurar la base de datos para la tienda online, modelando correctamente las relaciones entre productos, usuarios y pedidos.

1. Diseña una página básica con React y conecta al backend.

Objetivo: Crear la estructura inicial del frontend con React, diseñar una página básica y establecer la conexión con el backend de Django mediante API REST.

1. Implementa el carrito de compras en el backend

Objetivo: Crear la lógica del carrito de compras en Django, permitiendo a los usuarios agregar, actualizar y eliminar productos del carrito.