Fase 1: Configuración del Entorno

1. Instalar Python:

Descarga la versión más reciente de Python desde python.org.

Durante la instalación, asegúrate de marcar la casilla "Add Python to PATH".

2. Instalar un Editor de Código:

Se recomienda usar VS Code, PyCharm, o Sublime Text.

3. Crear un Entorno Virtual (Recomendado):

Abre la terminal y navega hasta la carpeta de tu proyecto.

Ejecuta `python -m venv venv` (o `python3 -m venv venv` en algunos sistemas).

Activa el entorno virtual:

En Windows: `venv\Scripts\activate`

En macOS/Linux: `source venv/bin/activate`

4. Instalar Librerías Necesarias:

Si decides usar una base de datos SQLite: `pip install sqlalchemy`

Si desea una interfaz gráfica: `pip install tkinter` o `pip install kivvy`

Si desea trabajar con archivos excel: `pip install openpyxl`

Si desea trabajar con archivos csv: No necesita instalar nada, python tiene la libreria csv incluida.

Fase 2: Diseño de la Base de Datos (Opcional)

1. Elegir una Base de Datos:

SQLite (simple, archivos locales).

MySQL o PostgreSQL (para aplicaciones más grandes).

2. Diseñar las Tablas:

Productos, Categorías, Inventario, Compras, Detalle\\_Compras, Ventas, Detalle\\_Ventas, Usuarios, Roles.

Define las columnas y relaciones entre las tablas.

Fase 3: Estructura del Proyecto

1. Crear Carpetas:

`data/` (para archivos de base de datos o CSV).

`gui/` (para la interfaz gráfica).

`reports/` (para generar informes).

`models/` (para las clases de datos).

`main.py` (archivo principal).

2. Crear Archivos Python:

`models.py` (definición de las clases).

`database.py` (interacción con la base de datos).

`inventory.py` (funciones de manejo de inventario).

`sales.py` (funciones de manejo de ventas).

`purchases.py` (funciones de manejo de compras).

`reports.py` (funciones para generar reportes).

`gui.py` (implementación de la interfaz gráfica).

Fase 4: Implementación de la Lógica

1. Definir Clases en `models.py`:

Crea clases para Productos, Categorías, Inventario, etc.

Define los atributos y métodos necesarios.

2. Crear Funciones de Base de Datos en `database.py`:

Conexión a la base de datos.

Funciones para insertar, actualizar, eliminar y consultar datos.

3. Implementar la Lógica de Negocio:

`inventory.py`: Funciones para gestionar el inventario (entradas, salidas, stock).

`sales.py`: Funciones para registrar ventas, calcular totales, etc.

`purchases.py`: Funciones para registrar compras, calcular totales, etc.

`reports.py`: Funciones para generar informes (ventas, inventario).

4. Crear la Interfaz de Usuario (Opcional):

`gui.py`: Utiliza Tkinter, PyQt o Kivy para crear ventanas, formularios, etc.

Conecta la interfaz con las funciones de la lógica de negocio.

5. Archivo `main.py`:

Importa las funciones y clases necesarias.

Crea un menú principal o inicia la interfaz gráfica.

Fase 5: Pruebas y Mejoras

1. Pruebas Unitarias:

Escribe pruebas para cada función y clase.

2. Pruebas de Integración:

Prueba el flujo completo de la aplicación.

3. Mejoras Continuas:

Añade nuevas funcionalidades según sea necesario.

Optimiza el rendimiento.

Mejora la interfaz de usuario.

Herramientas Útiles

Git: Para el control de versiones.

SQLiteStudio o DB Browser for SQLite: Para visualizar y gestionar bases de datos SQLite.

Postman: Para probar APIs (si decides crear una API).