

# Gestor de Productos con Tkinter y SQLAlchemy

---

## 1. Introducción

Esta aplicación permite gestionar productos personales con las funcionalidades de crear, editar, listar y eliminar productos. Cada producto tiene un nombre, precio, categoría y stock. La aplicación está desarrollada con Tkinter para la interfaz gráfica y SQLAlchemy para la gestión de la base de datos SQLite.

## 2. Estructura del Proyecto

- app.py: Archivo principal con la lógica de la interfaz y gestión de eventos.
- models.py: Definición del modelo Producto para la base de datos.
- database.py: Configuración de la base de datos SQLite y conexión con SQLAlchemy.
- productos.db: Archivo SQLite donde se almacenan los datos de productos.

## 3. Funcionamiento de la Aplicación

### 3.1 Ventana principal (Listado y formulario)

- Muestra un formulario para ingresar datos del producto: nombre, precio, categoría y stock.
- Debajo se visualiza una lista con los productos registrados.
- Se pueden seleccionar productos para editar o eliminar.

The screenshot shows a Tkinter window titled 'Registrar / Editar producto'. It contains a form with the following fields: 'Nombre:' (text input), 'Precio:' (text input), 'Categoría:' (dropdown menu with 'Otros' selected), and 'Stock:' (text input). Below the form are two buttons: 'Guardar producto' and 'Actualizar producto'. Below the form is a table with the following columns: 'Ordenador', 'Nombre', 'Precio', 'Categoría', and 'Stock'. The table contains one row of data: 'Ordenador', 'Nombre', 'Precio 1500', 'Categoría Electrónica', and 'Stock 45'. At the bottom of the window are two buttons: 'Eliminar producto' and 'Limpiar campos'.

Ordenador	Nombre	Precio	Categoría	Stock
		1500	Electrónica	45

### 3.2 Crear un nuevo producto

- Se completa el formulario con los datos del producto.
- Se pulsa el botón Guardar producto para añadirlo a la base de datos.
- El producto aparece en la lista actualizada.

Registrar / Editar producto

Nombre:

Precio:

Categoría:

Stock:

Ordenador	Nombre	Precio	Categoría	Stock
		1500	Electrónica	45

### 3.3 Seleccionar producto para editar

- Al hacer clic en un producto de la lista, el formulario se rellena con sus datos.
- Se puede modificar cualquiera de los campos.
- Se pulsa Actualizar producto para guardar los cambios.

Gestor de Productos con SQLAlchemy

Registrar / Editar producto

Nombre:

Precio:

Categoría:

Stock:

Ordenador	Nombre	Precio	Categoría	Stock
		3444	Electrónica	21

### 3.4 Eliminar un producto

- Se selecciona un producto en la lista.
- Se pulsa el botón Eliminar producto.
- El producto se elimina de la base de datos y desaparece de la lista.

The screenshot shows a web application interface for managing products. At the top, there is a form titled 'Registrar / Editar producto' with fields for 'Nombre' (Ordenador), 'Precio' (3444), 'Categoría' (Electrónica), and 'Stock' (21). Below the form are buttons for 'Guardar producto' and 'Actualizar producto'. A red status message 'Editando producto ID 1' is displayed. Below this is a table with columns 'Nombre', 'Precio', 'Categoría', and 'Stock', containing one row: 'Ordenador', '3444', 'Electrónica', '21'. A confirmation dialog box is centered on the screen, asking '¿Estás seguro de que deseas eliminar este producto?' with 'Sí' and 'No' buttons. At the bottom of the interface are two buttons: 'Eliminar producto' and 'Limpiar campos'.

### 3.5 Limpiar formulario

- El botón Limpiar campos borra los datos del formulario para crear un nuevo producto o cancelar edición.

## 4. Descripción técnica del código

### 4.1 database.py

- Configura la conexión con SQLite.
- Crea el motor y la sesión de SQLAlchemy para gestionar la base de datos.
- Define la base declarativa para los modelos.

### 4.2 models.py

- Define el modelo Producto con los campos:
  - id (clave primaria)
  - nombre (cadena, obligatorio)
  - precio (entero, obligatorio)

- categoría (cadena, opcional)
- stock (entero, opcional)

### 4.3 app.py

- Usa Tkinter para crear la interfaz gráfica.
- Usa SQLAlchemy para interactuar con la base de datos.
- Gestiona eventos de botones para CRUD (Crear, Leer, Actualizar, Eliminar).
- Incluye validaciones básicas para entradas de usuario.

## 5. Base de datos

- Se almacena localmente en productos.db.
- Las tablas se crean automáticamente al iniciar la app si no existen.
- Cada producto se guarda con sus atributos para consulta y modificación.

## 6. Cómo ejecutar la aplicación

1. Clonar o descargar el proyecto.
2. Crear y activar un entorno virtual (opcional pero recomendado).
3. Instalar dependencias:  
`pip install sqlalchemy`
4. Ejecutar la aplicación:  
`python app.py`
5. Usar la aplicación en la ventana gráfica que se abre.