Boletín ejercicios - App web

1ª PARTE: Gestión de Biblioteca (POO en Python)

Enunciado

Crea un programa en Python utilizando programación orientada a objetos que modele una pequeña biblioteca, almacenando los datos en una base de datos SQLite . Como Implementación de referencia puedes usar las siguientes clases con sus atributos y métodos:

1. Libro
   - ➔ Atributos: titulo (str), autor (str), isbn (str), disponible (bool, por defecto True)
   - ➔ Métodos:
     - ◆ __init__(self, titulo, autor, isbn)
     - ◆ prestar(self) → cambia disponible a False si está disponible, retorna True si se prestó, False si ya estaba prestado.
     - ◆ devolver(self) → cambia disponible a True si estaba prestado.
     - ◆ __str__(self) → retorna una cadena con la información del libro (ej: "Título: El Quijote, Autor: Cervantes, ISBN: 12345, Disponible: Sí")
2. Biblioteca
   - ➔ Atributos: nombre (str), libros (lista de objetos Libro)
   - ➔ Métodos:
     - ◆ __init__(self, nombre)
     - ◆ agregar_libro(self, libro) → añade un libro a la lista
     - ◆ mostrar_libros(self) → imprime todos los libros con su estado
     - ◆ buscar_por_titulo(self, titulo) → retorna el libro si existe, o None si no
     - ◆ prestar_libro(self, isbn) → busca por ISBN y presta el libro si está disponible

Programa

- ➔ Crea una biblioteca llamada "Biblioteca Central".
- ➔ Añade al menos 3 libros.
- ➔ Muestra todos los libros.
- ➔ Intenta prestar un libro (por ISBN), muestra el resultado.
- ➔ Vuelve a mostrar los libros.

Requisitos

- ➔ Usa encapsulamiento adecuado (atributos privados si es necesario).

Boletín ejercicios - App web

- ➔ Maneja errores básicos (ej: intentar prestar un libro no disponible o inexistente).
- ➔ Código claro, comentado y bien estructurado.
- ➔ Usa SQLite para almacenar los datos.
- ➔ Entrega: Un archivo zip con todas los clases y el programa principal funcionando

Boletín ejercicios - App web

2ª PARTE: Web de Gestión de Biblioteca con Flask (usando el código POO + SQLite)

Objetivo

Crear una aplicación web sencilla con Flask que permita gestionar la biblioteca implementada previamente (clases Libro y Biblioteca con SQLite), exponiendo las funcionalidades principales a través de una interfaz web.

Enunciado

Usa el código de las clases Libro y Biblioteca (con persistencia en SQLite) como base y crea una aplicación web con Flask que cumpla con lo siguiente:

Requisitos funcionales

1. Rutas y páginas:
   - ➔ / → Inicio: Muestra el nombre de la biblioteca y un listado de todos los libros (título, autor, ISBN, estado).
   - ➔ /prestar/<isbn> → Presta el libro con ese ISBN. Redirige al inicio con mensaje de éxito o error.
   - ➔ /devolver/<isbn> → Devuelve el libro. Redirige al inicio con mensaje.
   - ➔ /agregar → Formulario para agregar un nuevo libro (método POST).

2. Plantillas HTML simples (Jinja2):
   - ➔ Usa al menos dos plantillas:
     - ◆ base.html (herencia)
     - ◆ index.html (hereda de base.html)
     - ◆ agregar.html

Estructura propuesta del proyecto:

```
biblioteca_web/
├── app.py              # Aplicación Flask
├── biblioteca.py       # Clases Libro y Biblioteca (copia del código anterior)
├── templates/
│   ├── base.html
│   ├── index.html
│   └── agregar.html
└── biblioteca.db       # (se genera automáticamente)
```

Boletín ejercicios - App web

Requisitos técnicos

- ➔ Usa Flask (sin frameworks adicionales).
- ➔ No uses SQLAlchemy → sigue usando sqlite3 directamente.
- ➔ Instancia una sola Biblioteca global en app.py.
- ➔ Las plantillas deben ser limpias y responsive (puedes usar Bootstrap CDN).
- ➔ Manejo básico de errores (ej: ISBN duplicado, libro no encontrado).

Ejemplo de comportamiento:

Acción	**Resultado**
Visitar /	Ver lista de libros
Hacer clic en “Prestar”	Libro pasa a “No disponible” + mensaje
Intentar prestar un libro prestado	Mensaje de error
Agregar libro con ISBN repetido	Mensaje de error

Programa

Un directorio biblioteca_web/ con:

- ➔ app.py (código Flask completo)
- ➔ biblioteca.py (clases Libro y Biblioteca)
- ➔ Carpeta templates/ con al menos los 3 archivos HTML
- ➔ Instrucciones en README.md para ejecutar

¡Desarrolla una web funcional, simple y elegante para gestionar tu biblioteca!