

La siguiente práctica simula una tarea que nuestro jefe de proyecto nos ha encargado. Se debe seguir al pie de la letra las exigencias del enunciado.

---

## Enunciado de la práctica

### Título: Sistema de Gestión de Biblioteca

**Descripción:** Eres el desarrollador de un sistema básico de gestión para una biblioteca. Debes crear un programa en Python que permita registrar libros y gestionar préstamos a usuarios. El programa debe cumplir con los siguientes requisitos:

#### 1. Clase Libro:

- Crea una clase Libro con los atributos **título** (str), **autor** (str), **isbn** (str) y **disponible** (bool, inicialmente **True**).
- Incluye un método **agregar()** que permita introducir un nuevo libro con sus características.
- Incluye un método **prestar()** que cambie el estado de disponible a False si el libro está disponible, y muestre un mensaje si ya está prestado.
- Incluye un método **devolver()** que cambie el estado de disponible a True si estaba prestado, y muestre un mensaje si ya estaba disponible.
- Incluye un método **mostrar()** que devuelva una lista con todos los libros de la biblioteca y los muestre en pantalla con todos sus datos y diga si están disponibles o no.
- Incluye un método **buscar()** que busque un libro en concreto por su ISBN y lo muestre en pantalla con todos sus datos y diga si está disponible o no.

#### 2. Gestión del inventario:

- Usa una lista para almacenar objetos de la clase Libro.
- Implementa un bucle que permita al usuario interactuar con el programa mediante un menú con las siguientes opciones:
  - a) Agregar un nuevo libro ingresando título, autor e ISBN.

- b) Prestar un libro buscando por ISBN.
- c) Devolver un libro buscando por ISBN.
- d) Mostrar todos los libros y su estado (disponible o no).
- e) Salir del programa.

### 3. Condiciones:

- Valida que el ISBN ingresado exista en la lista antes de prestar o devolver un libro.
- Si el usuario ingresa una opción inválida en el menú, muestra un mensaje de error y vuelve a pedir una opción.

### Entregable:

- Un script en Python que implemente todas las funcionalidades descritas.
- El código debe ser claro, con comentarios explicativos y usando buenas prácticas.

**Ejemplo de uso. Al ejecutar el programa el resultado debería ser como se ve en el siguiente ejemplo:**

### Bienvenido al Sistema de Gestión de Biblioteca

1. Agregar libro
2. Prestar libro
3. Devolver libro
4. Mostrar libros
5. Buscar
6. Salir

**Elige una opción: 1**

**Título:** El Quijote

**Autor:** Cervantes

**ISBN:** 12345

**Libro agregado con éxito.**

**Elige una opción: 4**

**- El Quijote (Cervantes) - ISBN: 12345 - Disponible: Sí**

Elige una opción: 2

Ingresa el ISBN: 12345

Libro prestado con éxito.

Elige una opción: 4

- El Quijote (Cervantes) - ISBN: 12345 - Disponible: No

---

**Consideraciones a tener en cuenta:**

Se creará la clase:

- Libro

Con los atributos:

- Titulo
- Autor
- isbn
- disponible

Se crearán los métodos:

- agregar
- prestar
- devolver
- Mostrar
- buscar

El ejercicio se entregará en **un solo archivo** con extensión **.py**

El nombre de la clase, atributos y métodos será exactamente el que se pide en el enunciado.

Solo se admitirá una sola subida por alumno. En el caso de que un alumno suba más de una versión del ejercicio, solo se calificará la primera que se subiese, ignorando el sistema las demás. Por lo que se recomienda encarecidamente que se revisen detalladamente los ejercicios antes de proceder a su entrega.

Cualquier ejercicio que no cumpla escrupulosamente los requisitos pedidos **NO SERÁ CORREGIDO**.

Todo aquel ejercicio que sea entregado posteriormente a las 00:00 horas del día dd/mm/yyyy no será corregido por el sistema.