

Primer Parcial – Programación 1

Enunciado

La biblioteca escolar necesita un sistema de gestión sencillo para su catálogo de libros y las copias disponibles. Se pide desarrollar un programa con una interfaz basada en menú que utilice listas paralelas (una para títulos[] y otra para ejemplares[]). Cada título debe estar vinculado a su número correspondiente de copias utilizando el mismo índice en ambas listas. Se debe utilizar un bucle while para navegar por las opciones del menú hasta que el usuario elija salir.

Requisitos y Restricciones Técnicas

- **Estructuras Permitidas:** Debes usar un bucle while para navegar por las opciones del menú hasta que el usuario elija salir. Se debe utilizar una estructura case (o su equivalente con if/elif/else) para el menú, así como estructuras condicionales y secuenciales de Python. Se permiten funciones para validar cadenas de texto como lower(), upper() o isdigit().
- **Estructuras Prohibidas:** Está estrictamente prohibido usar excepciones, clases, diccionarios, funciones de creación propia y estructuras de datos avanzadas. Si se utilizan, la calificación máxima será de 10%.
- **Listas Paralelas:** Las listas títulos[] y ejemplares[] deben estar sincronizadas, de modo que el título en un índice corresponda a la cantidad de copias en el mismo índice de la otra lista

Ejemplo:

- títulos[] = ["El Señor de los Anillos", "Orgullo y Prejuicio", "Matar un Ruiseñor"]
- ejemplares[] = [5, 3, 7]

En este ejemplo, "El Señor de los Anillos" tiene 5 copias, "Orgullo y Prejuicio" tiene 3 copias, y "Matar un Ruiseñor" tiene 7 copias.

Requerimientos del Menú

1. **Ingresar títulos** → Para agregar los títulos iniciales de los libros, el usuario indica la cantidad inicial.
2. **Ingresar ejemplares** → Para agregar una cantidad de copias para cada título.
3. **Mostrar catálogo** → Muestra todos los libros y su stock actual.
4. **Consultar disponibilidad** → Busca un título específico y muestra cuántos ejemplares hay.
5. **Listar agotados** → Muestra los títulos que tienen 0 ejemplares.

6. **Agregar título** → Permite añadir un nuevo libro y sus ejemplares disponibles al catálogo, respetando la sincronía de los índices en las listas.
7. **Actualizar ejemplares (préstamo/devolución)** → Permite modificar el stock de un libro, elegido por el usuario, para registrar préstamos o devoluciones.
 - **Préstamo** -> Disminuye en 1 la cantidad de ejemplares del libro seleccionado, si hay unidades suficientes.
 - **Devolución** -> Aumenta en 1 la cantidad de ejemplares del libro seleccionado.
8. **Salir** → Termina la ejecución del programa.

Entregables

El estudiante deberá **subir el archivo del programa en lenguaje Python** a la plataforma institucional. **NO SUBIR UN REPOSITORIO DE GITHUB. SOLO SUBIR EL ARCHIVO.PY**

El código debe cumplir con:

- Todas las funcionalidades solicitadas reflejadas en el menú.
- Buena ejecución sin errores.
- Nomenclatura clara en el nombre de las variables.
- Legibilidad general y buenas prácticas de codificación.

Rúbrica de Evaluación

Código	Criterio	Peso	Descripción detallada
C1	Correctitud Funcional	50%	<p>Evalúa que todas las funcionalidades del sistema estén implementadas y funcionen correctamente:</p> <ul style="list-style-type: none">• Agregar título: se añade a <code>titulos[]</code> y <code>ejemplares[]</code> manteniendo la correspondencia de índices y evitando duplicados.• Consultar disponibilidad: muestra correctamente los ejemplares para un título válido.• Listar agotados: muestra títulos con <code>ejemplares = 0</code>.• Préstamo / Devolución: ajusta correctamente el stock.• Mostrar catálogo: lista todos los títulos con su stock.

C2	Cumplimiento de Restricciones	20%	<p>Evalúa el respeto de las limitaciones de diseño:</p> <ul style="list-style-type: none"> • Uso exclusivo de listas paralelas. • Mantener títulos con ejemplares = 0 en el catálogo. • No usar diccionarios, clases, funciones ni estructuras avanzadas. • Sincronía permanente entre <code>títulos[]</code> y <code>ejemplares[]</code>.
C3	Interacción y Validación	10%	<p>Evalúa la robustez de la interacción con el usuario:</p> <ul style="list-style-type: none"> • Validar que el título no sea vacío. • Validar que las cantidades sean enteros y positivas (o cero si corresponde). • No permitir prestar más de lo disponible. • Verificar la existencia del título antes de cualquier operación. • Menú persistente hasta elegir salir. • Manejo de opciones inválidas con mensajes claros.
C4	Estructura y Legibilidad	10%	<p>Evalúa la calidad del código:</p> <ul style="list-style-type: none"> • Variables descriptivas. • Indentación consistente. • Flujo claro y ordenado. • Mensajes coherentes y consistentes para el usuario.
C5	Casos de Prueba / Cobertura	5%	<p>Evalúa la consideración de casos normales y extremos en las pruebas:</p> <ul style="list-style-type: none"> • Pruebas con títulos y ejemplares válidos. • Título con ejemplares = 0. • Intento de préstamo mayor que el disponible. • Título inexistente en cualquier operación.

C6	Gestión de Casos Borde	5%	Evalúa el manejo de escenarios críticos: <ul style="list-style-type: none">• Préstamo cuando no hay ejemplares disponibles.• Préstamo con cantidades negativas o excesivas.• Devoluciones inválidas (cantidades negativas).• Operaciones sobre títulos inexistentes o vacíos
----	------------------------	----	---