Segunda Parcial – Programación 1

Contexto

La **biblioteca escolar** necesita **modernizar** su forma de administrar el **catálogo de libros** y el **stock de ejemplares disponibles**. Tu tarea es **desarrollar una aplicación de consola en Python** que permita **cargar, consultar y actualizar** el inventario de manera sencilla, manteniendo registros persistentes en un archivo **CSV**.

Objetivos de aprendizaje

- Manejar **listas** y **diccionarios**.
- Aplicar **control de flujo** (bucles, condicionales) y **menús interactivos**.
- Implementar **CRUD** sobre una colección.
- Validar entradas y mantener **consistencia**.
- **Persistir y recuperar** datos usando **CSV** (módulo estándar `csv`).

Rol del estudiante

Implementar un programa modular (con funciones) que muestre un **menú interactivo** en bucle hasta que el usuario decida salir. El sistema debe asegurar datos consistentes y operaciones claras para **altas, consultas, préstamos, devoluciones** y **reportes**.

Estructura de datos principal

El catálogo se gestionará con **una única lista** de **diccionarios**, donde **cada diccionario representa un libro**:

- Claves obligatorias por libro:
  - `TITULO` → `str` (nombre del libro)
  - `CANTIDAD` → `int` (número de ejemplares disponibles, `>= 0`)
- **Unicidad**: no pueden existir dos libros con el mismo `TITULO` (comparación **insensible a mayúsculas/minúsculas** y **espacios redundantes**).
- **Ejemplo:**

```
catalogo = [
    {"TITULO": "El Señor de los Anillos", "CANTIDAD": 5},
    {"TITULO": "1984",                    "CANTIDAD": 0},
    {"TITULO": "Ficciones",               "CANTIDAD": 3}
```

```
]
```

Persistencia (CSV)

- El programa **carga** el catálogo desde un archivo CSV al iniciar (si existe) y **guarda** los cambios **cada vez que se modifica** el inventario.
- Formato sugerido del CSV (con encabezado):
  - Columnas: `TITULO,CANTIDAD`

Ejemplo:

```
TITULO,CANTIDAD

El Señor de los Anillos,5

1984,0

Ficciones,3
```

Interacción por menú (flujo general)

- Al iniciar, se muestra un **menú numerado** y se solicita una opción.
- El menú debe **repetirse** (bucle `while`) hasta elegir **Salir**.
- Luego de **cada operación que cambie datos**, el programa debe **persistir** el catálogo en el CSV y mostrar un **mensaje de confirmación**.

Funcionalidades obligatorias del menú

1. **Ingresar títulos (múltiples)**: permite cargar varios libros de una vez, el usuario indica la cantidad de libros que quiere cargar. Por cada libro, pedir `TITULO` (no vacío, no duplicado) y `CANTIDAD` (`>= 0`).
2. **Ingresar ejemplares**: sumar una cantidad a un título existente.
3. **Mostrar catálogo**: listar todos los libros con su stock actual.
4. **Consultar disponibilidad**: buscar un `TITULO` y mostrar cuántos ejemplares hay disponibles.
5. **Listar agotados**: mostrar solo los títulos con `CANTIDAD == 0`.
6. **Agregar título**: alta de un libro individual (validar duplicados) con su cantidad inicial.
7. **Actualizar ejemplares (préstamo/devolución)**:
   - **Préstamo**: restar `1` solo si `CANTIDAD > 0`.
   - **Devolución**: sumar `1`.
8. **Salir**: finalizar la aplicación.

Reglas de negocio y validaciones

- **Títulos**: no se aceptan vacíos; comparar sin sensibilidad a mayúsculas y normalizando espacios.
- **Cantidades**: deben ser enteros `>= 0` al cargar/editar.
- **Préstamos**: no permitir valores negativos (si está en `0`, informar al usuario).
- **Mensajes claros**: informar siempre si una operación fue **exitosa** o **rechazada** y el **motivo**.

Requisitos y restricciones técnicas

- **Obligatorio**:
  - Bucle `while` para el menú.
  - `match`/`case` para direccionar opciones (Python 3.10+).
  - Funciones para **modularizar** (cargar/guardar CSV, validar entradas, buscar títulos, etc.).
  - Estructuras: **listas** y **diccionarios** (estructura principal descrita arriba).
  - Persistencia en **CSV**.
  - Se pueden usar validadores como `lower()`, `upper()`, `isdigit()`.
- **Prohibido** (penaliza a 10% máximo de nota si se usan): **funciones lambdas, excepciones, clases, variables globales**.

Comportamiento esperado (resumen)

- Si el CSV **no existe**, iniciar con **catálogo vacío**.
- Si el usuario ingresa un **título duplicado**, **rechazar** y volver a pedir.
- Tras **agregar/actualizar** datos, **guardar** automáticamente en el CSV.
- Las **búsquedas** de títulos deben ser **robustas** (insensibles a mayúsculas y espacios extra).

Entregables

- El estudiante deberá subir un único archivo del programa en lenguaje Python (.py) a la plataforma institucional.

- **NO SUBIR UN REPOSITORIO DE GITHUB. SOLO SUBIR EL ARCHIVO .PY.**

Rúbrica de Evaluación

Código	Criterio	Peso	Descripción detallada

C1	**Correctitud Funcional**	50%	Evalúa que todas las funcionalidades del sistema estén implementadas y funcionen correctamente. • Persistencia Robusta**:** Los datos se cargan desde catalogo.csv al iniciar y se guardan tras cada modificación. • Agregar título: se añade a titulos[] y ejemplares[] manteniendo la correspondencia de índices y evitando duplicados. • Consultar disponibilidad: muestra correctamente los ejemplares para un título válido. • Listar agotados: muestra títulos con ejemplares = 0. • Préstamo / Devolución: ajusta correctamente el stock. • Mostrar catálogo: lista todos los títulos con su stock.
C2	**Cumplimiento de Restricciones**	20%	Evalúa el respeto de las limitaciones de diseño. • Uso exclusivo de una única lista compuesta por diccionarios, uno para cada libro. • Mantener títulos con ejemplares = 0 en el catálogo. • Sincronía permanente entre titulos[] y ejemplares[].
C3	**Interacción y Validación**	10%	Evalúa la robustez de la interacción con el usuario. • 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%	Evalúa la calidad del código. • Variables descriptivas. • Indentación consistente. • Flujo claro y ordenado. • Mensajes coherentes y consistentes para el usuario.
C5	**Casos de Prueba / Cobertura**	5%	Evalúa la consideración de casos normales y extremos en las pruebas. • 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. • Préstamo cuando no hay ejemplares disponibles. • Préstamo con cantidades negativas o excesivas. • Devoluciones inválidas (cantidades negativas). • Operaciones sobre títulos inexistentes o vacíos.