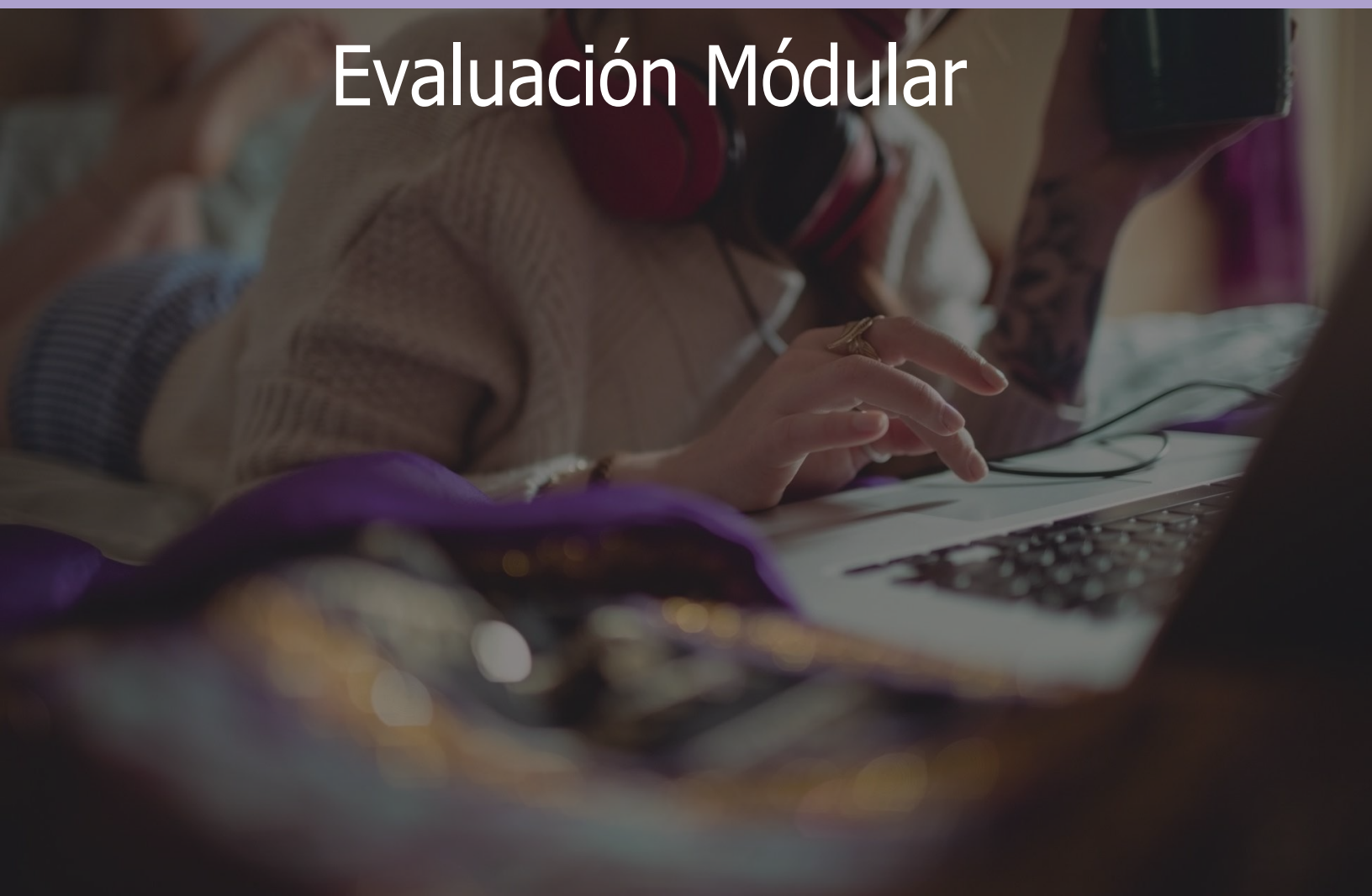


Módulo 2

Evaluación Modular



ACTIVIDAD:

Sistema de Gestión de libros, usuarios y préstamos

Objetivo:

Desarrollar una aplicación en Python aplicando funciones, estructuras de datos, POO y manejo de errores para gestionar libros, usuarios y préstamos de forma automatizada.

Contexto:

La empresa **DataSolutions Academy**, dedicada a la formación continua de profesionales en áreas tecnológicas, ha lanzado una nueva plataforma educativa enfocada en el aprendizaje autónomo. Una de sus unidades, llamada “**Biblioteca Digital para Ingenieros de Datos**”, necesita una aplicación interna que permita **gestionar libros, usuarios y préstamos de manera automatizada y escalable**. Hasta ahora, todo el proceso se realizaba manualmente, lo que ha provocado errores frecuentes, pérdida de información y lentitud en la atención a estudiantes. Por ello, se ha decidido desarrollar una aplicación en **Python** que permita:

- Registrar y consultar libros según diferentes criterios (autor, género, disponibilidad).
- Registrar usuarios y diferenciar su rol (por ejemplo: estudiante, instructor).
- Gestionar préstamos de libros con validación de disponibilidad.
- Controlar errores como préstamos duplicados o usuarios no registrados.

La aplicación debe ser simple, clara y robusta, aplicando los conocimientos adquiridos en este módulo.



Instrucciones:

Desarrolla un programa en Python que cumpla con los siguientes requerimientos:

1. Estructura General

- El proyecto debe estar organizado en **módulos**.
- Todo el código debe estar comentado y documentado (usando **docstrings** y comentarios tipo PEP8).

2. Funciones

- Uso de **funciones definidas por el usuario** para dividir la lógica del programa.
- Empleo de **funciones con argumentos, retorno de valores, argumentos por defecto y argumentos variables**.

3. Estructuras de Datos

- Uso de **listas y diccionarios** para almacenar información de libros y usuarios.
- Uso de **tuplas o sets** donde sea apropiado.

4. Programación Orientada a Objetos


- Definición de **clases** para representar libros, usuarios y préstamos.
- Uso de **atributos, métodos y constructores**.
- Implementación de **herencia** para diferenciar tipos de usuarios.
- Aplicación de **polimorfismo** en métodos comunes.

5. Manejo de Excepciones

- Uso de estructuras **try-except** para capturar y gestionar errores comunes como:
 - Libro no encontrado
 - Usuario no registrado
 - Préstamo duplicado

6. Operaciones del sistema (Ejecutables desde menú):

- Registrar nuevo libro
- Registrar nuevo usuario
- Consultar libros por autor, género o disponibilidad
- Realizar un préstamo
- Listar libros prestados



Rúbrica:

Indicador de logro/criterio	Insuficiente (0%-20%)	Por lograrlo (21%-40%)	Medianamente logrado (41%-60%)	Logrado (61%-80%)	Sobresaliente (81%-100%)
1. Estructuras básicas del lenguaje	Usa estructuras básicas de forma deficiente o ausente.	Uso limitado, con errores de sintaxis o lógica frecuentes.	Uso funcional pero con errores menores.	Usa correctamente variables, condicionales, bucles con buena lógica.	Uso correcto, variado y robusto de variables, condicionales, operadores y bucles.
2. Funciones y módulos	No hay uso de funciones/módulos.	Uso de funciones sin modularización clara o reutilización.	Uso de funciones con estructura básica y modularización parcial.	Funciones claras, bien organizadas y reutilizables.	Funciones bien diseñadas, con argumentos variados, documentadas y distribuidas en módulos.
3. Estructuras de datos	Estructuras mal empleadas o ausentes.	Uso parcial o con errores importantes en listas y diccionarios.	Uso limitado con errores menores.	Listas y diccionarios usados adecuadamente; estructuras adicionales bien utilizadas.	Uso correcto y variado de estructuras (listas, diccionarios, tuplas, sets) de forma eficiente.
4. Clases, herencia y polimorfismo	Poco o nulo uso de clases; sin herencia ni métodos claros.	Clases presentes pero con errores graves o sin herencia.	Clases funcionales, herencia parcial o incompleta.	Clases bien diseñadas con herencia funcional.	Clases bien estructuradas, herencia clara y polimorfismo aplicado correctamente.
5. Manejo de excepciones	No se maneja ningún tipo de error.	Manejo básico o incompleto con try-except.	Manejo limitado a uno o dos errores esperados.	Manejo completo de errores comunes definidos en el proyecto.	Manejo robusto y adecuado con try-except y mensajes informativos claros.
6. Documentación y buenas prácticas	Código sin documentar, nombres sin significado, estilo desordenado.	Comentarios dispersos, nombres poco descriptivos.	Uso parcial de docstrings y estilo inconsistente.	Uso de docstrings, nombres claros y estilo PEP8 en la mayor parte del código.	Documentación completa, nombres claros, estilo PEP8 coherente, código limpio y mantenible.
7. Resultado funcional del programa	El sistema no ejecuta o presenta errores graves.	Varias funcionalidades con errores.	Algunas funciones presentan errores leves.	Ejecuta correctamente las funcionalidades principales.	El sistema ejecuta todas las funcionalidades de forma correcta y fluida.