

Actividad 2 – Proyecto: Sistema de gestión bibliotecaria

Abdiel Shamir León Manzano

Banner: 100211609

Correo institucional: aleonman@estudiante.ibero.edu.co

Carlos David Rodriguez Fontalvo

Banner: 100199487

Correo institucional: crodr213@estudiante.ibero.edu.co

Facultad Ingeniería

Universidad Iberoamericana

Ingeniería de Software

Estructura de Datos

William Ruiz

1. INTRODUCCIÓN

En esta actividad se desarrolló un sistema de gestión bibliotecaria por consola implementado en Python, que tiene como objetivo administrar de manera sencilla un catálogo de libros, usuarios y los movimientos de los libros, ya sea un préstamo o devolución.

El sistema se diseñó basados los requerimientos funcionales, tales como el registro de usuarios, autenticación, administración del inventario de libros y el control de préstamos realizados por los estudiantes.

Para la presentación de la información, se definieron tres estructuras principales:

- **Libros**, que contienen datos como título, autor, ISBN, cantidad disponible y fechas de registro.
- **Movimientos**, que registran préstamos y devoluciones asociadas a los estudiantes.
- **Usuarios**, encargados de la administración del sistema, con datos como nombre, correo electrónico y credenciales de acceso.

En cuanto a la estructura de datos, se utilizaron listas para almacenar las entidades en memoria, esto permite realizar cada una de las acciones dispuestas para estos modelos, tales como, búsqueda, inserción y eliminación de forma sencilla. En un futuro podrían implementarse estructuras como colas (para reservas de libros) o pilas (para el historial de operaciones), para esta primera versión el uso de listas resulta adecuado.

Finalmente, el sistema es ejecutado por consola, permitiendo al usuario interactuar mediante menús y comandos simples, garantizados una experiencia clara y directa para la gestión básica de una biblioteca.

2. REQUERIMIENTOS

A. Funcionales:

- **Gestión de usuarios:**
 - Registrar un nuevo usuario con email, contraseña y nombre.
 - Autenticar usuarios mediante email y contraseña.
 - Eliminar usuarios.
 - Listar usuarios.
- **Gestión de Libros:**
 - Registrar un nuevo libro con título, autor, fecha de publicación, ISBN y cantidad disponible.
 - Consultar todos los libros del catalogo.
 - Eliminar un libro.
 - Disminuir la cantidad disponible cuando se preste el libro.
 - Aumentar la cantidad disponible cuando se hace la devolución de un libro prestado.

- **Gestión de movimientos:**
 - Registrar un nuevo movimiento, validando:
 - Campos requeridos (Nombre del estudiante, identificación y fecha de devolución)
 - Que cuente con cantidad disponible.
 - Que no exista un préstamo duplicado del mismo libro para el mismo estudiante.
 - Registrar la devolución de un libro, marcando el movimiento como devuelto y actualizando la fecha de retorno.
 - Consultar todos los movimientos.
 - Validar los datos de préstamo.
 - Verificar si existe un movimiento activo para un estudiante y libro en particular.

B. No funcionales:

- **Rendimiento:**
 - Las operaciones de búsqueda y validación deben ser rápidas, incluso con listas largas.
 - El login y validación de movimientos deben ejecutarse en tiempo aceptable.
- **Confiabilidad:**
 - Debe evitar inconsistencias, como prestar un libro con stock cero.
 - No permitir préstamos duplicados de un mismo libro para un mismo estudiante.
- **Escalabilidad:**
 - Aunque se usa almacenamiento en memoria (listas), debe ser fácil de adaptar a una base de datos en el futuro.
- **Mantenibilidad:**
 - El código debe estar modularizado en servicios independientes.
 - Validaciones centralizadas para evitar duplicación de lógica.

3. EVALUACIÓN DE ESTRUCTURAS LINEALES

Para esta primera versión del sistema resulta viable utilizar listas debido a su simplicidad y flexibilidad. Python ofrece listas como una estructura de datos nativa que permite almacenar, recorrer, insertar y eliminar elementos de manera directa, lo cual se adapta perfectamente a los requerimientos básicos de registrar usuarios, libros y movimientos, además de listar o buscar datos sin restricciones de orden. Esto facilita la implementación inicial y permite centrar los esfuerzos en la lógica de negocio sin añadir complejidad innecesaria.

Además, el uso de listas es suficiente en términos de rendimiento para un sistema pequeño o en etapa de desarrollo. Aunque las búsquedas son lineales, el volumen de datos esperado en esta primera versión es bajo, por lo que no supone un problema de eficiencia. En caso de que el proyecto crezca, la migración

hacia estructuras más eficientes será un paso natural, pero para esta etapa inicial las listas ofrecen el equilibrio ideal entre sencillez y funcionalidad.

	VENTAJAS	DESVENTAJAS
LISTAS	Flexibles: permiten insertar, eliminar y recorrer sin restricciones de orden. Útiles para almacenar entidades principales (usuarios, libros, movimientos). Búsqueda secuencial sencilla de implementar.	La búsqueda puede ser lenta con muchos registros. Eliminar por ID requiere recorrer toda la lista hasta encontrar el elemento. No tienen control de acceso (cualquiera puede insertar en cualquier parte).
PILAS	Útiles para gestionar historial (ejemplo: últimas operaciones, deshacer acciones). Acceso rápido al último elemento insertado.	No son adecuadas para búsquedas arbitrarias (solo se accede al tope). No sirven bien para entidades como usuarios o libros que requieren acceso directo.
COLAS	Útiles para gestionar procesos en orden de llegada (ejemplo: lista de espera de libros prestados). Acceso rápido al primer elemento insertado.	No son adecuadas para búsquedas aleatorias o modificaciones directas. Se pierde flexibilidad, ya que solo se puede acceder al frente o al final.

4. ESTRUCTURA DE DATOS

En esta sección se presentan las estructuras de datos utilizadas en el sistema, describiendo la forma en que se organizaron los distintos elementos de información (usuarios, libros y movimientos) para garantizar un manejo ordenado y coherente de los registros.

- **Libros:** representa los ejemplares disponibles en la biblioteca.
 - id (int): Identificador único del libro.
 - title (str): Título del libro.
 - author (str): Autor del libro.
 - published_date (str): Fecha de publicación (ejemplo: "2020-05-10").
 - isbn (str): Número ISBN del libro (único).
 - quantity (int): Cantidad de ejemplares disponibles en stock.
 - created_at (date): Fecha en que se creó el registro.
 - updated_at (date): Fecha de última actualización del registro.

- **Movimientos:** registra los préstamos y devoluciones de los libros.
 - id (int): Identificador único del movimiento.
 - book_id (int): ID del libro prestado (relación con Libros).
 - student_name (str): Nombre del estudiante que realiza el préstamo.
 - student_identification (str): Número de identificación del estudiante.
 - loan_date (date): Fecha en que se realizó el préstamo.
 - return_date (date): Fecha programada o efectiva de devolución.
 - returned (bool): Indica si el libro fue devuelto (True / False).
 - created_at (date): Fecha de creación del registro.
 - updated_at (date): Fecha de última actualización del registro.
- **Usuarios:** representa los usuarios registrados en el sistema (administradores o encargados).
 - id (int): Identificador único del usuario.
 - name (str): Nombre completo del usuario.
 - email (str): Dirección de correo electrónico (única).
 - password (str): Contraseña del usuario (actualmente en texto plano, debería almacenarse con hash en producción).
 - created_at (date): Fecha de creación del registro.
 - updated_at (date): Fecha de última actualización del registro.

5. VALIDACIONES

- **Prestar un libro:** En la imagen se muestra la validación realizada al intentar registrar un préstamo con un identificador de libro inexistente (ID = 4). El sistema responde con el mensaje “Libro no encontrado” y cancela la operación, evidenciando la existencia de controles que garantizan la integridad de los datos y evitan inconsistencias en el registro de movimientos.

```
MOVIMIENTOS
7. Prestar un libro
8. Ver Todos los Movimientos
9. Devolver Libro
-----
SALIR
10. Salir
Ingresa una opción: 7
-----
ID: 1 - Título: Cien años de soledad - Autor: Gabriel García Márquez - Fecha de Publicación: 1967-05-30 - ISBN: 9780307474728 - Cantidad: 5 - Fecha de Creación: 2025-09-29
ID: 2 - Título: 1984 - Autor: George Orwell - Fecha de Publicación: 1949-06-08 - ISBN: 9780451524935 - Cantidad: 4 - Fecha de Creación: 2025-09-29
ID: 3 - Título: Don Quijote de la Mancha - Autor: Miguel de Cervantes - Fecha de Publicación: 1605-01-16 - ISBN: 9788420412145 - Cantidad: 3 - Fecha de Creación: 2025-09-29
-----
Ingresa el ID del libro: 4
Ingresa el nombre del estudiante: Shamir
Ingresa la identificación del estudiante: 1234567891
Ingresa la fecha de devolución (YYYY-MM-DD): 2025-11-11
XXX Libro no encontrado XXX
Error al agregar movimiento
```

- **Agregar usuario:** En la figura se observa la validación aplicada al intentar registrar un nuevo usuario con una contraseña inferior a seis caracteres. El sistema muestra el mensaje “La contraseña debe tener al menos 6 caracteres” y rechaza la operación, garantizando así el cumplimiento de reglas básicas de seguridad en la gestión de usuarios.

```
Bienvenido Admin
-----
Menú de Administrador
-----
USUARIOS
1. Agregar Usuario
2. Ver Todos los Usuarios
3. Eliminar Usuario
-----
LIBROS
4. Agregar Libro
5. Ver Todos los Libros
6. Eliminar Libro
-----
MOVIMIENTOS
7. Prestar un libro
8. Ver Todos los Movimientos
9. Devolver Libro
-----
SALIR
10. Salir
Ingresa una opción: 1
Ingresa el nombre del usuario: Shamir
Ingresa el email del usuario: shamir.com
Ingresa la contraseña del usuario:

XXX La contraseña debe tener al menos 6 caracteres XXX

Error al agregar usuario X
-----
```

- **Agregar libro:** En la imagen se observa el registro exitoso de un nuevo libro en el sistema. El usuario ingresa correctamente todos los datos requeridos: título, autor, fecha de publicación, ISBN y cantidad disponible. Como resultado, el sistema confirma la operación mostrando el mensaje “Libro El Principito agregado exitosamente”. Esta validación evidencia que, al cumplir con los criterios establecidos, el sistema procesa de manera adecuada la información y actualiza el catálogo de libros, garantizando un flujo correcto en la gestión bibliotecaria.

LIBROS

- 4. Agregar Libro
 - 5. Ver Todos los Libros
 - 6. Eliminar Libro
-

MOVIMIENTOS

- 7. Prestar un libro
 - 8. Ver Todos los Movimientos
 - 9. Devolver Libro
-

SALIR

- 10. Salir

Ingresar una opción: 4

Ingresar el título del libro: El Principito

Enter the author of the book: Antoine de Saint-Exupéry

Ingresar la fecha de publicación del libro: 1943-04-06

Ingresar el ISBN del libro: 9780156012

Ingresar la cantidad disponible del libro: 7

Libro El Principito agregado exitosamente 🎉✅✅

6. CONCLUSIÓN

El sistema de gestión bibliotecaria evidencia la importancia de seleccionar adecuadamente las estructuras de datos en el desarrollo de software. En esta primera versión, el uso de listas permitió almacenar y manipular de manera sencilla los registros de usuarios, libros y movimientos, ofreciendo la flexibilidad necesaria para operaciones básicas como inserción, búsqueda y eliminación. Aunque no es la opción más eficiente para grandes volúmenes de información, su simplicidad la convierte en una solución viable para un prototipo funcional, que puede evolucionar en el futuro hacia estructuras más especializadas o bases de datos según las necesidades del sistema.