

ANEP PET

“Los Arrayanes”

NOMBRE DEL PROYECTO

Proyecto biblioteca

ASIGNATURAS

Programación Avanzada, Base de Datos, Redes  
Informáticas, Matemáticas

INTEGRANTES DEL EQUIPO

Nahuel Montenegro

Gaspar Moreno

Juan Díaz

Germán Karamanoukián

GRUPO

2°do MC Tecnologías de la Información

AÑO

2025

# Índice

<b>Índice.....</b>	<b>2</b>
<b>1 Proyecto.....</b>	<b>3</b>
1.1 Descripción del problema.....	3
1.2 Objetivos.....	3
1.3 Lista de requerimientos.....	3
1.3.1 Requerimientos funcionales.....	3
1.3.2 Requerimientos No funcionales.....	4
1.4 Alcances y limitaciones.....	4
1.5 Descripción de las herramientas.....	4
2.Diagrama UML.....	4
2.3.2 Modelo Entidad Relación.....	5
2.3.3 Modelo de datos.....	5
2.3.4 Diccionario de datos.....	5
2.4 Pruebas(Testing).....	8
2.4.1 Juego de datos de prueba.....	8
<b>3 Redes Informáticas.....</b>	<b>8</b>
3.1Organización de los espacios.....	9
3.2Elementos de canalización.....	11
<b>4 Matemáticas.....</b>	<b>12</b>
<b>5 Conclusiones finales.....</b>	<b>13</b>
5.1 Nahuel Montenegro.....	13
5.2 Gaspar Moreno.....	13
5.3 Juan Díaz.....	13
5.4 Germán Karamanoukián.....	13
5.5 Del equipo.....	13
<b>6 Glosario.....</b>	<b>14</b>
<b>7 Bibliografía.....</b>	<b>15</b>
<b>8 Anexos.....</b>	<b>16</b>
8.1 Manual de usuario.....	17

# 1 Proyecto

## 1.1 Descripción del problema.

El problema en cuestión es una falta de fiabilidad, persistencia y accesibilidad por parte de la biblioteca y su método de guardado de datos, actualmente la herramienta utilizada para guardar datos de préstamos es un papel, una herramienta claramente insegura en ciertos aspectos con respecto a un software especializado en lo mismo, con su aspecto confiable, persistente y accesible tan necesitado.

## 1.2 Objetivos

OB1. Evitar la pérdida de datos

OB2. Digitalizar y facilitar la gestión de préstamos y del inventario

OB3. Mejorar la eficiencia con la que se maneja la biblioteca

## 1.3 Lista de requerimientos

1. Realizar Listar préstamos
2. Realizar Listar Inventario
3. Realizar préstamo de libros
4. Realizar préstamo de laptops
5. Realizar Cancelar préstamo de libros
6. Realizar Cancelar préstamo de laptops
7. Realizar Agregar libro al inventario
8. Realizar Agregar laptop al inventario
9. Realizar Eliminar libro del inventario
10. Realizar Eliminar laptop del inventario

### 1.3.1 Requerimientos funcionales

RF 1: Listar préstamos y inventario

RF 2: Realizar préstamos de libro y laptop

RF 3: Realizar eliminación de préstamo de libro y laptop

RF 4: Realizar eliminación de libro u laptop del inventario

## 1.3.2 Requerimientos No funcionales

RNF 1: El lenguaje de implementación debe de ser Java

RNF 2: El sistema operativo ubuntu

RNF 3: El sistema se alojará en MySQL

## 1.4 Alcances y limitaciones

1. Podrá dar de alta y baja libros y laptops
2. Podrá agregar libros y laptops o eliminarlos del inventario
3. Podrá listar la cantidad de prestamos, y en el inventario los libros y laptops
4. NO PODRÁ gestionar de ninguna manera las impresiones
5. NO PODRÁ gestionar dinero

## 1.5 Descripción de las herramientas

**Deben detallar las herramientas, aplicaciones utilizadas para el desarrollo del proyecto.(todas)**

# 2 Manual técnico

## 2.1 Metodología

La Programación Orientada a Objetos (POO) es un paradigma de programación, esto es, un modelo o un estilo de programación que proporciona unas guías acerca de cómo trabajar con él y que está basado en el concepto de clases y objetos. Este tipo de programación se emplea para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

Históricamente han ido surgiendo distintos paradigmas de programación. Por un lado, los lenguajes secuenciales como COBOL o procedimentales como Basic o C, se enfocan más en la lógica que en los datos. Por otro lado, otros más modernos como Java, C# y Python, usan paradigmas para definir los programas, siendo la POO la más popular.

Con el paradigma de POO lo que se busca es dejar de centrarse en la lógica pura de los programas, para comenzar a pensar en objetos, lo que forma la base de dicho paradigma. Esto ayuda bastante en sistemas grandes, pues en lugar de pensar en funciones, se piensa en las relaciones o interacciones de los distintos elementos del sistema.

Un programador diseña un programa de software organizando piezas de información y comportamientos relacionados en una plantilla denominada clase. Después, se crean objetos individuales a partir de la plantilla de clase. Todo el programa de software se ejecuta haciendo que diversos objetos interactúen entre sí para crear un programa mayor.

Por qué la Programación Orientada a Objetos

La POO permite que el código sea reutilizable, organizado y sencillo de mantener. Y es que sigue el principio de desarrollo de software empleado por muchos programadores DRY (Don't Repeat Yourself), para no duplicar el código y crear programas eficientes. Asimismo, previene el acceso no deseado a los datos o la exposición de código propietario a través de la encapsulación y la abstracción.

Clases, objetos e instancias

Resumiendo bastante, para crear POO se hacen clases y luego se crean objetos a partir de dichas clases, que constituyen el modelo a partir del que se estructuran los datos y los comportamientos. Y es que quizá el concepto más importante sea distinguir entre clase y objeto.

Una clase es una plantilla, que define de modo genérico cómo serán los objetos de un determinado tipo. Por ejemplo, una clase para representar a animales puede denominarse 'animal' y disponer de una serie de atributos, como 'nombre' o 'edad' (que habitualmente son propiedades), y una serie con los comportamientos que estos pueden tener, como caminar o comer, y que a su vez se implementan como métodos de la clase (funciones).

Un ejemplo sencillo de un objeto, podría ser un animal. Un animal tiene una edad, por lo que se crea un nuevo atributo de 'edad', y, además, puede envejecer, por lo que se define un nuevo método. Datos y lógica. Esto es lo que se define en muchos programas como la definición de una clase, que es la definición global y genérica de muchos objetos.

Con la clase pueden crearse instancias de un objeto, cada uno de ellos con sus atributos definidos independientemente. Con esto se puede crear un gato llamado Pepe, con 5 años de edad, y otro animal, este tipo perro y llamado Lucho, con una edad de 3 años. Los dos están definidos por la clase animal, pero son dos instancias diferentes. Así, llamar a sus métodos puede tener resultados distintos. Los dos comparten la lógica, pero cada uno tiene su estado de manera independiente.

Todo esto son herramientas que ayudan a escribir un código mejor, más limpio y reutilizable.

## 4 principios de la Programación Orientada a Objetos

Hay cuatro principios de la POO.

### La encapsulación

La encapsulación presenta toda la información importante de un objeto dentro del mismo y solo expone la información elegida al mundo exterior.

Esta propiedad permite asegurar que la información de un objeto esté oculta para el mundo exterior, agrupando en una clase las características o atributos que tienen un acceso privado, y los comportamientos o métodos que cuenta con un acceso público.

La encapsulación de cada objeto es responsable tanto de su información y de su estado. El único modo en la que esta se puede modificar es a través los propios métodos del objeto. De esta manera, los atributos internos de un objeto son ser inaccesibles desde fuera, pudiéndose modificar sólo llamando a las funciones correspondientes. Así se consigue mantener el estado a salvo de usos indebidos o que puedan resultar inesperados.

Para explicar la encapsulación usaremos un coche de ejemplo. El coche comparte información pública mediante las luces de freno o intermitentes para indicar los giros (interfaz pública). Por contra, la interfaz interna, el mecanismo propulsor del coche, está oculto bajo el capó. Al conducir un automóvil es necesario indicar a otros conductores los movimientos, pero no exponer datos privados sobre el tipo de combustible o la temperatura del motor, ya que son muchos datos, lo que confundiría a los demás conductores.

### La abstracción

Otro de los principios de la Programación Orientada a Objetos es la abstracción, que se produce cuando el usuario interactúa solo con los atributos y métodos seleccionados de un objeto, usando herramientas simplificadas de alto nivel para acceder a un objeto complejo.

En la POO, los programas suelen ser muy grandes y los objetos se comunican bastante entre sí. De este modo, la abstracción facilita el mantenimiento de un código de gran tamaño, donde pueden surgir distintos cambios con el paso del tiempo.

Así, la abstracción está basada en utilizar cosas simples para representar la complejidad. Los objetos y las clases representan código subyacente, ocultando los detalles complejos al usuario. Por ende, se trata de una extensión de la

encapsulación. Continuando con el ejemplo anterior, no es necesario que conozcas todos los detalles sobre cómo funciona el motor de un coche para poder conducirlo.

## La herencia

La herencia define relaciones jerárquicas entre clases, de modo que atributos y métodos comunes puedan ser reutilizados. Las clases principales extienden atributos y comportamientos a las clases secundarias. Mediante la definición en una clase de los atributos y comportamientos básicos, pueden crearse clases secundarias, ampliando la funcionalidad de la clase principal y añadiendo atributos y comportamientos extra. Es una de las claves de la Programación Orientada a Objetos.

Siguiendo el ejemplo de los animales, puede emplearse una única clase de animal y añadir un atributo de tipo de animal que especifique el tipo de animal. Los distintos tipos de animales requerirán diferentes métodos, por ejemplo, los reptiles deben poder poner huevos y los peces nadar. Incluso si los animales disponen de un método en común, como moverse, la implementación requeriría muchas declaraciones “si” para garantizar el comportamiento de movimiento idóneo. Por ejemplo, las ranas saltan, mientras que las serpientes se deslizan. El principio de herencia permite solucionar dicho problema.

## El polimorfismo

El polimorfismo reside en diseñar objetos para compartir comportamientos, lo que permite procesar objetos de distintos modos. Es la capacidad de presentar la misma interfaz para distintas maneras subyacentes o tipos de datos. Al usar la herencia, los objetos pueden anular los comportamientos principales compartidos, con comportamientos secundarios específicos. El polimorfismo permite que el mismo método ejecute distintos comportamientos de dos modos: anulación de método y sobrecarga de método.

## Beneficios de Programación Orientada a Objetos

- Reutilización del código.
- Convierte cosas complejas en estructuras simples reproducibles.
- Evita la duplicación de código.
- Permite trabajar en equipo gracias al encapsulamiento, puesto que minimiza la posibilidad de duplicar funciones cuando distintas personas trabajan sobre un mismo objeto al mismo tiempo.

- Al estar la clase bien estructurada permite la corrección de errores en diversos lugares del código.
- Protege la información mediante la encapsulación, pues solo se puede acceder a los datos del objeto mediante propiedades y métodos privados.
- La abstracción nos permite construir sistemas más complejos y de un modo más sencillo y organizado.

Hoy en día, la POO es el paradigma que más se usa para diseñar aplicaciones y programas informáticos. Sus ventajas son evidentes, especialmente cuando se necesita resolver retos de programación complejos. Además, permite una mejor estructura de datos y reutilización del código, lo que facilita el ahorro de tiempo a largo plazo.

Ahora bien, para ello se necesita pensar bien en la estructura del programa, planificar al inicio de la codificación, además de analizar los requisitos en clases simples y reutilizables que pueden usarse para diseñar instancias de objetos.

## 2.2 Lenguaje

¿Qué es Java?

Java es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del servidor.

¿Para qué se utiliza el lenguaje de programación Java?

¿Para qué se utiliza el lenguaje de programación Java?

Debido a que Java es un lenguaje versátil y de uso gratuito, crea software localizado y distribuido. Algunos usos comunes de Java incluyen:



## 1. Desarrollo de juegos

Muchos videojuegos, así como juegos para móviles y computadoras, se crean con Java. Incluso los videojuegos modernos que integran tecnología avanzada, como el [machine learning](#) o la realidad virtual, se crean con la tecnología de Java.

## 2. Computación en la nube

Java a menudo se conoce como WORA: escribir una vez y ejecutar en cualquier lugar (por sus siglas en inglés “*Write Once and Run Anywhere*”), lo que lo hace perfecto para aplicaciones descentralizadas basadas en la nube. Los proveedores de la nube eligen el lenguaje Java para ejecutar programas en una amplia gama de plataformas subyacentes.

## 3. Big data

Java se usa para motores de procesamiento de datos que pueden trabajar con conjuntos de datos complejos y cantidades masivas de datos en tiempo real.

## 4. Inteligencia artificial

Java es una fuente inagotable de bibliotecas de machine learning. Su estabilidad y velocidad lo hacen perfecto para el desarrollo de aplicaciones de [inteligencia artificial](#) como el procesamiento de lenguaje natural y el [aprendizaje profundo](#).

## 5. Internet de las cosas

Java se ha utilizado para programar sensores y hardware en dispositivos de periferia que pueden conectarse de forma independiente a Internet.

¿Por qué Java es una opción tan popular entre los desarrolladores de software de hoy en día?

Java es popular porque se ha diseñado para que sea fácil de usar. Algunas razones por las que los desarrolladores siguen eligiendo Java sobre otros lenguajes de programación incluyen:

#### Recursos de aprendizaje de alta calidad

Java existe desde hace mucho tiempo, por lo que hay muchos recursos de aprendizaje disponibles para los nuevos programadores. La documentación detallada, los libros completos y los cursos ayudan a los desarrolladores a lo largo de la curva de aprendizaje. Además, los principiantes pueden comenzar a escribir código en Core Java antes de pasar a Advanced Java.

#### Funciones y bibliotecas incorporadas

Al usar Java, los desarrolladores no necesitan escribir cada nueva función desde cero. En cambio, Java proporciona un rico ecosistema de funciones y bibliotecas integradas para desarrollar una variedad de aplicaciones.

#### Apoyo comunitario activo

Java tiene muchos usuarios activos y una comunidad que puede ayudar a los desarrolladores cuando enfrentan desafíos de codificación. El software de la plataforma Java también se mantiene y actualiza periódicamente.

#### Herramientas de desarrollo de alta calidad

Java ofrece varias herramientas para admitir la edición automatizada, la depuración, las pruebas, la implementación y la administración de cambios. Estas herramientas hacen que la programación de Java sea más rápida y rentable.

#### Plataforma independiente

El código de Java puede ejecutarse en cualquier plataforma subyacente, como Windows, Linux, iOS o Android, sin tener que volver a escribir. Esto lo hace especialmente poderoso en el entorno actual, donde queremos ejecutar aplicaciones en múltiples dispositivos.

## Seguridad

Los usuarios pueden descargar código Java que no sea de confianza a través de una red y ejecutarlo en un entorno seguro en el que no puede causar ningún daño. El código que no es de confianza no puede infectar el sistema host con un virus ni puede leer o escribir archivos del disco duro. Los niveles de seguridad y las restricciones en Java también son altamente configurables.

### ¿Cómo funciona Java?

Todos los lenguajes de programación son un medio para comunicarse con las máquinas. El hardware de la máquina solo responde a la comunicación electrónica. Los lenguajes de programación de alto nivel como Java actúan como un puente entre el lenguaje humano y el lenguaje del hardware. Para usar Java, un desarrollador debe comprender dos cosas:

#### 1. Lenguaje Java y API

Esta es la comunicación de frontend entre el desarrollador y la plataforma de Java.

#### 2. Java Virtual Machine

Esta es la comunicación de backend entre la plataforma de Java y el hardware subyacente. A continuación, veremos cada uno de estos aspectos con más detalle.

### ¿Qué es la API de Java?

Java define la sintaxis y la semántica del lenguaje de programación Java. Esto incluye el vocabulario básico y las reglas utilizadas para escribir algoritmos, como tipos de datos inmutables, bloques if/else, bucles, etc.

Las API son componentes de software importantes que se incluyen con la plataforma de Java. Estos son programas de Java preescritos que pueden conectar y reproducir la funcionalidad existente en su propio código. Por ejemplo, podría usar las API de Java para obtener la fecha y la hora, realizar operaciones matemáticas o manipular texto.

Por lo general, cualquier código de aplicación de Java escrito por un desarrollador combinará código nuevo y preexistente de las API de Java y las bibliotecas de Java.

#### ¿Qué es Java Virtual Machine?

Java Virtual Machine actúa como una capa de abstracción adicional entre la plataforma de Java y el hardware de la máquina subyacente. El código fuente de Java solo puede ejecutarse en aquellas máquinas en las que se haya instalado JVM. La respuesta a por qué se necesita Java Virtual Machine se encuentra en la historia de la programación.

#### La historia de la programación

Cuando se desarrollaron por primera vez los lenguajes de programación naturales, se dividieron en dos grandes categorías, según cómo se comunicaran con el hardware subyacente.

1. **Compiladores:** se escribe el programa completo en una sintaxis similar a la del inglés natural con compiladores, y el lenguaje posteriormente compila (o traduce) todo el código a código máquina. A continuación, el código compilado se ejecuta en el hardware.
2. **Intérpretes:** mediante intérpretes, cada instrucción de código de nivel superior se interpreta en código máquina sobre la marcha. El hardware ejecuta inmediatamente las declaraciones escritas antes de ver la siguiente declaración.

#### El entorno en tiempo de ejecución de Java

El programa Java fue el primer lenguaje en combinar los dos métodos anteriores mediante Java Virtual Machine (JVM). El compilador de código Java se llama Java Virtual Machine. Cualquier archivo Java se compila primero en código de bytes. El código de bytes de Java solo se puede ejecutar en la JVM. Luego, la JVM interpreta el código de bytes para ejecutarlo en la plataforma de hardware subyacente. Entonces, si la aplicación se ejecuta en una máquina con Windows, la JVM la interpretará para Windows. Pero si se ejecuta en una plataforma de código abierto como Linux, la JVM lo interpretará para Linux.

¿Cómo se programa en Java?

Para comenzar a programar en Java, debe instalar una edición de Java en su sistema. Hay cuatro ediciones principales de Java:

1. Java Standard Edition (Java SE)
2. Java Enterprise Edition (Java EE)
3. Java Micro Edition (Java ME)

¿Qué es Java SE?

Java Standard Edition es la plataforma central de programación de Java. Contiene todas las bibliotecas y API que cualquier programador necesita para el desarrollo de Java. Open Java Development Kit (OpenJDK) es la implementación gratuita y de código abierto de Java SE.

[Amazon Corretto](#) es una distribución de OpenJDK sin costo, multiplataforma y lista para producción. Se puede usar como un reemplazo directo para muchas distribuciones de Java SE y cuenta con soporte a largo plazo y gratuito de Amazon. Le permite ejecutar el mismo entorno en la nube, en las instalaciones y en su máquina local.

¿Qué es AWS SDK para Java?

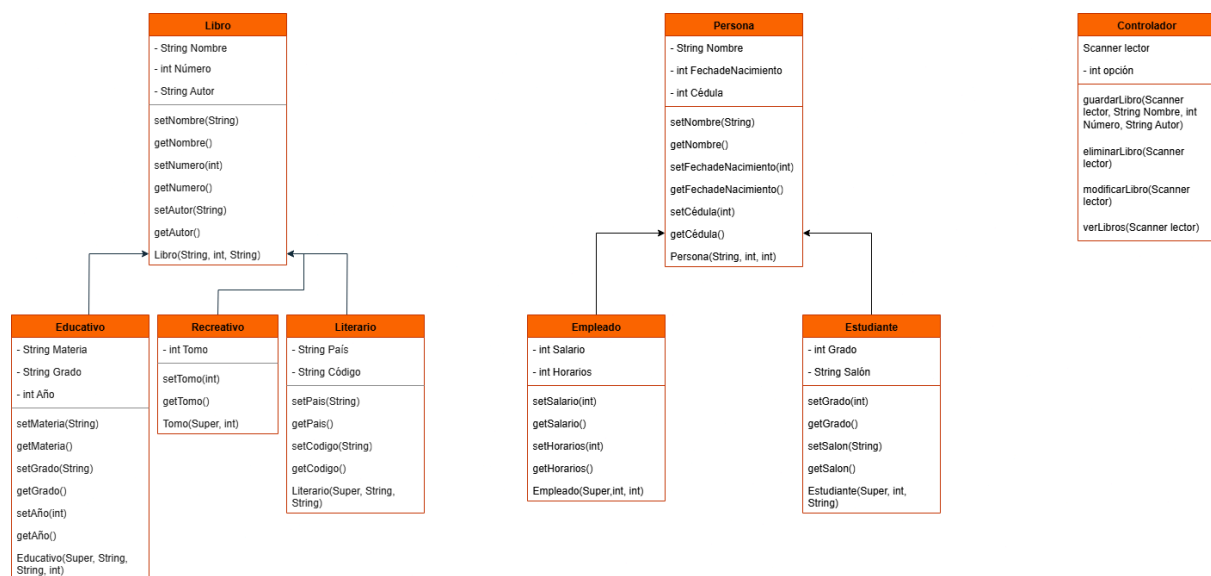
[AWS SDK para Java](#) simplifica el uso de los servicios de AWS porque ofrece un conjunto de bibliotecas que a los desarrolladores que usan Java les resultan lógicas y

familiares. Admite abstracciones de alto nivel para un desarrollo simplificado. Las bibliotecas de Java de código abierto centradas en AWS están disponibles junto con ejemplos de código y una guía de referencia de la API de Java.

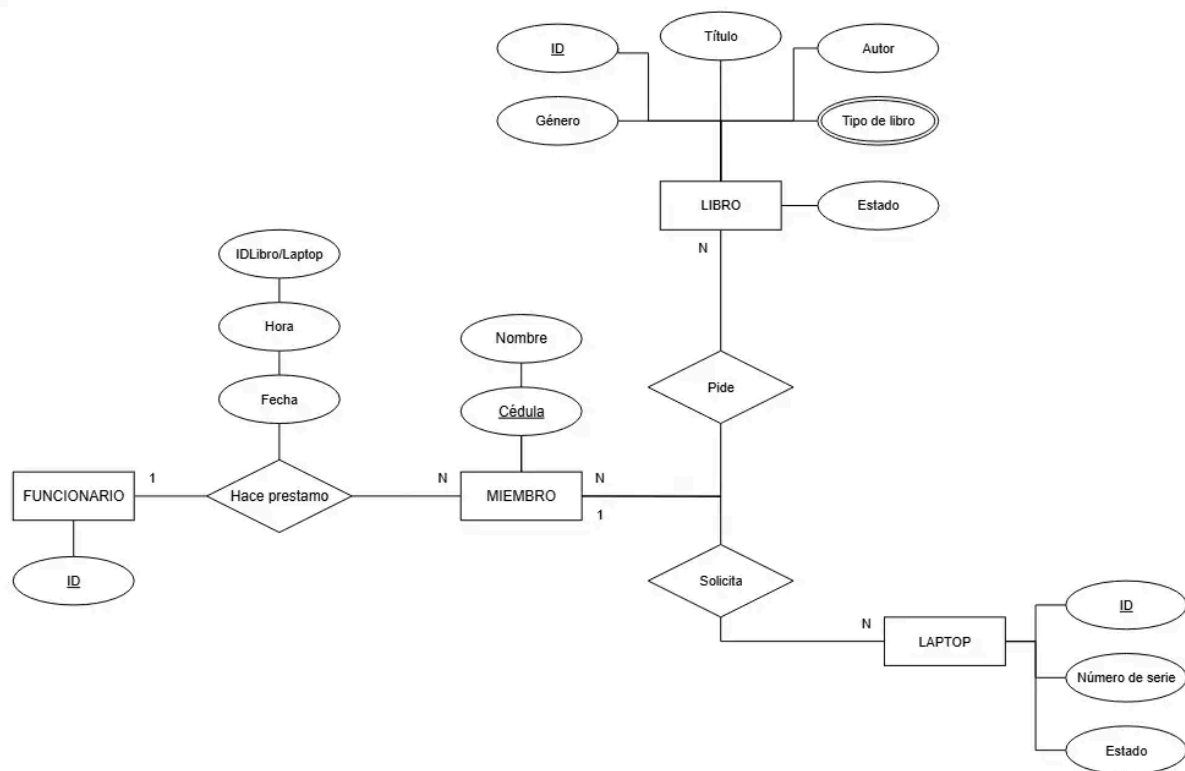
Los [ejemplos de código de SDK](#) contienen ejemplos de código de Java y casos de uso reales para que los servicios de AWS ayuden a acelerar el desarrollo de sus aplicaciones. Además, la [Guía de referencia de la API de Java](#) describe las operaciones de la API para la versión más reciente de AWS SDK para Java. La guía de referencia también ofrece ejemplos de solicitudes, respuestas y errores de los protocolos de servicios web compatibles.

## 2.3 Implementacion

### 2.3.1.Diagrama UML



## 2.3.2 Modelo Entidad Relación



## 2.3.3 Modelo Relacional

**Funcionarios**(ID, Cédula)

**Miembros**(Cedula, IDLibro/Laptop, Hora, Fecha, nombre)

**Laptops**(ID, Cédula, NúmeroDeSerie, Estado)

**Libros**(ID, Título, Autor, Género, EstadoFisico)

**TipodeLibro**(IDLibro, TipoDeLibro)

**Pide**(Cédula, ID)

## 2.3.4 Diccionario de datos

### Miembros

Nombre del atributo	Tipo de dato	Longitud	Observaciones
<u>CI</u>	INT	11	PK, NOT NULL
Nombre Estudiante	VARCHAR	45	NOT NULL
Grupo	VARCHAR	45	NOT NULL

## Inventario Libros

Nombre del atributo	Tipo de dato	Longitud	Observaciones
<u>ISBN</u>	VARCHAR	45	PK,NOT NULL
Título	VARCHAR	50	NOT NULL
Autor	VARCHAR	50	NOT NULL
AÑO	VARCHAR	45	NOT NULL
Tipo	VARCHAR	50	NOT NULL
Género	VARCHAR	45	NULL

## Inventario Laptops

Nombre del atributo	Tipo de dato	Longitud	Observaciones
<u>ID</u>	INT	50	PK,NOT NULL
NumeroDeSerie	VARCHAR	50	NOT NULL

## Préstamo Libro

Nombre del atributo	Tipo de dato	Longitud	Observaciones
<u>IDPrestamo</u>	INT	45	PK,Auto Increment, NOT NULL



ISBN	VARCHAR	45	FK-INVENTARIO LIBROS.ISBN,NU LL
CI	INT	11	FK-Miembros.CI,N ULL
FechaPrestamo	DATE	10	NOT NULL
FechaDevolucion	DATE	10	NOT NULL
Estado	VARCHAR	20	NOT NULL

## **Préstamo Laptop**

<b>Nombre del atributo</b>	<b>Tipo de dato</b>	<b>Longitud</b>	<b>Observaciones</b>
<u>IDPrestamoLaptop</u>	INT	11	PK,Auto Increment, NOT NILL
CI	INT	11	FK-MIEMBROS.C, NULL
FechaPrestamo	DATE	10	NOT NULL
FechaDevolucion	DATE	10	NOT NULL
IDlaptop	INT	50	FK-InventarioLapt ops.ID,NOT NULL

## 2.4 Pruebas(Testing)

### 2.4.1 Juego de datos de prueba

Nº	Dato Ingresado	Acción / Funcionalidad	Resultado Esperado	Resultado Obtenido	Resultado de Prueba
1	<b>Miembro:</b> Nombre: <i>Juan Pérez</i> , DNI: <i>12345678</i>	Alta de miembro (VentanaMiembros)	Miembro registrado correctamente en base de datos	Se muestra mensaje de confirmación y el miembro aparece en la lista	✓
2	<b>Miembro existente:</b> DNI: <i>12345678</i>	Alta de miembro duplicado	Sistema muestra error o evita duplicado	Mensaje "Miembro ya registrado" mostrado	✓
3	<b>Campos vacíos:</b> sin nombre o DNI	Alta de miembro	Sistema impide el alta y solicita completar campos	Aparece mensaje de validación	✓
4	<b>Miembro:</b> <i>Juan Pérez</i>	Baja de miembro	Registro eliminado de la base de datos	Miembro eliminado correctamente	✓
5	<b>Miembro inexistente:</b> DNI <i>99999999</i>	Baja de miembro	Error "Miembro no encontrado"	Mensaje de error mostrado	✓
6	<b>Libro:</b> Título <i>El Quijote</i> , Autor <i>Cervantes</i> , ISBN <i>978-84-376-0494-7</i>	Alta de libro (AltaLibro)	Libro agregado al inventario	Libro aparece en tabla de inventario	✓

7	<b>Libro existente (mismo ISBN)</b>	Alta duplicada	Sistema impide duplicado	Mensaje “Libro ya registrado” mostrado	✓
8	<b>Libro:</b> <i>El Quijote</i>	Modificar inventario (cambiar cantidad a 0)	Inventario actualizado	Cambio reflejado en lista	✓
9	<b>Libro inexistente:</b> ISBN 0000	Modificar inventario	Error “Libro no encontrado”	Mensaje de error mostrado	✓
10	<b>Libro:</b> <i>El Quijote</i>	Eliminar libro del inventario	Registro eliminado	Libro ya no aparece en lista	✓
11	<b>Cliente:</b> <i>Juan Pérez</i> , <b>Libro:</b> <i>El Quijote</i>	Registrar préstamo (VentanaPrestamo)	Préstamo registrado con fecha de salida	Registro agregado a la tabla de préstamos	✓
12	<b>Mismo libro</b> ya prestado	Nuevo préstamo	Sistema impide préstamo (libro no disponible)	Mensaje “Libro prestado actualmente” mostrado	✓
13	<b>Cliente:</b> <i>Juan Pérez</i> , devolución de <i>El Quijote</i>	Registrar devolución	Estado de libro cambia a disponible	Registro de préstamo actualizado	✓
14	<b>Cliente con 3 préstamos activos</b>	Nuevo préstamo	Restricción por límite de préstamos	Mensaje “Límite de préstamos alcanzado” mostrado	✓
15	<b>Conexión MySQL</b> interrumpida	Cualquier operación (alta, baja, etc.)	Error de conexión detectado	Mensaje de error SQL mostrado	✓
16	<b>Cerrar sesión / volver al menú</b>	Navegación (VentanaMenu → VentanaInicio)	Retorna correctamente al inicio	VentanaInicio mostrada	✓

### 3 Redes Informáticas

Deben colocar todo lo solicitado por la asignatura Redes Informáticas



#### Organización de los espacios

La biblioteca es de aproximadamente 47,914 m<sup>2</sup>. El espacio se organiza en diferentes zonas funcionales:

En el centro se ubican cuatro mesas redondas con cuatro sillas cada una, destinadas a la lectura o el trabajo grupal

. En la parte inferior (zona sur) se encuentra un mostrador o escritorio largo, donde están instalados los equipos principales de trabajo: una computadora (PC0) y una impresora (Printer0).

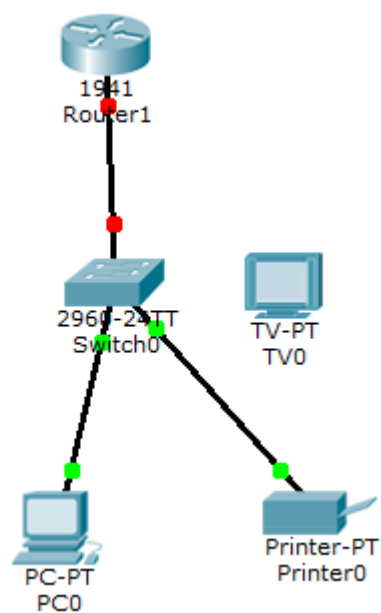
En la parte superior (zona norte) se localiza una televisión (TV0) sobre una mesa, utilizada para proyecciones o presentaciones. Finalmente, hacia el sector izquierdo

hay un área amueblada con sofás o estanterías, que cumple la función de zona de descanso o lectura informal.

#### Organización y ubicación de los equipamientos de la biblioteca

Equipo	Ubicación	Funcion	Conectividad
PC-PT (PC0)	Mostrador sur	Equipo de administración o consulta de catalogo	Conectada al switch principal mediante cableado estructurado
Printer-PT (Printer0)	Junto a la PC	Impresora compartida en red local	Conexión directa al switch
TV-PT (TV0)	Pared Norte	Presentación de contenidos	Conectada por cable al switch o al router

## Esquema de Red



## Estimación aproximada

Conexión	Distancia aproximada	Tipo de cable
Switch - PC0	2 m	Cable UTP Cat 6
Switch - Printer 0	1 m	Cable UTP Cat 6
Switch - TV0	8 m	Cable UTP Cat 6
Switch -Router	1m	Cable UTP Cat 6

**Total estimado de cable: 12 metros de cableado estructurado (categoría 6).**

Cantidad de fichas o bocas estáticas:

**Total:** 3 a 4 bocas RJ45.

## Elementos de canalización

Canaletas de PVC para conducción superficial del cableado hacia la TV y el escritorio.

Zócalos técnicos o ductos en el mostrador para organizar las conexiones de la PC y la impresora.

Posibles pasacables en el piso si se agregan nuevos puntos de red hacia el centro del salón.

Elemento	Descripción
Rack mural	Instalado en la pared posterior al mostrador, para alojar el switch, patchera y router
Patchera Cat 6 de 12 puertos	Facilita la organización y mantenimiento del cableado estructurado
Switch 8 puertos (puede ser gigabit)	Para conectar los dispositivos principales
Router wifi	Acceso a internet inalámbrico

## 4 Matemáticas

## 5 Conclusiones finales

### 5.1 Nahuel Montenegro

### 5.2 Gaspar Moreno

En mi opinión este proyecto fue un desafío desde cualquier punto de vista, primero por el corto plazo en el que nos tuvimos que adaptar a programar la base de datos y aprender cosas nuevas también dentro del ámbito de programación dentro de java, de todas maneras esta propuesta me pareció muy interesante y sentí que logró desarrollar mis capacidades en todos los ámbitos, con esto me refiero a programación y SQL.

### 5.3 Juan Díaz

A mi parecer, fue un reto desde distintos puntos de vista, tanto visual como técnico. Hubo algunas dificultades de mi parte, pero se pudieron solucionar. En cuanto a la organización del equipo, considero que estuvo muy bien y no hubo ningún problema. Más que nada, quiero agradecer a mi equipo por la ayuda y por el resultado final del proyecto.



## 5.4 Germán Karamanoukián

Gracias a este proyecto logre reflexionar que la programación y yo no nos llevamos bien pero algunas cosas he logrado hacer, considero que nuestro equipo es bastante bueno, hemos tenido algunos conflictos pero nada fuera de lo común dentro de un equipo, Personalmente soy consciente de que tuve dificultades, pero por suerte mi equipo me ha ayudado bastante con eso, la verdad fue un gran desafío este proyecto pero considero que fue muy útil y se adapta al día a día de un programador.

## 5.5 Del equipo

# 6 Glosario

**INSTANCIA:** una instancia es un objeto concreto que se crea a partir de una clase, la cual actúa como una plantilla o plano

**ATRIBUTO:** un atributo es una variable que pertenece a una clase y representa una característica o propiedad de un objeto, definiendo así su estado

**FUNCIÓN:** una función es un bloque de código reutilizable que realiza una tarea específica, a menudo de forma intercambiable, aunque técnicamente una función es un método que se puede invocar sin un objeto, y se pueden asignar a variables o utilizar en programación funcional a través de interfaces funcionales

## 7 Bibliografía

W3schools: <https://www.w3schools.com/>

Eclipse: <https://eclipseide.org/>

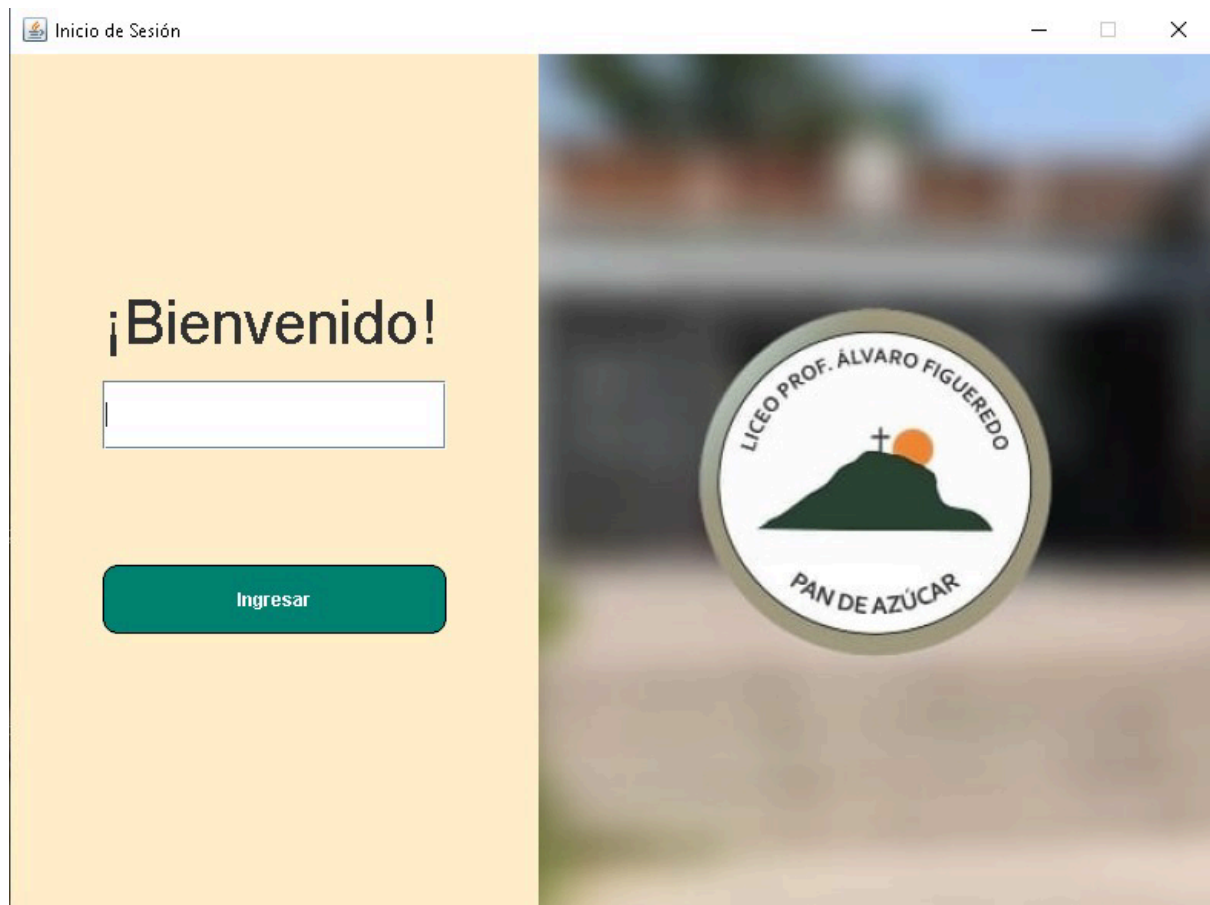
Xampp: <https://www.apachefriends.org/es/index.html>

MySQL worckbench: <https://dev.mysql.com/downloads/workbench/>

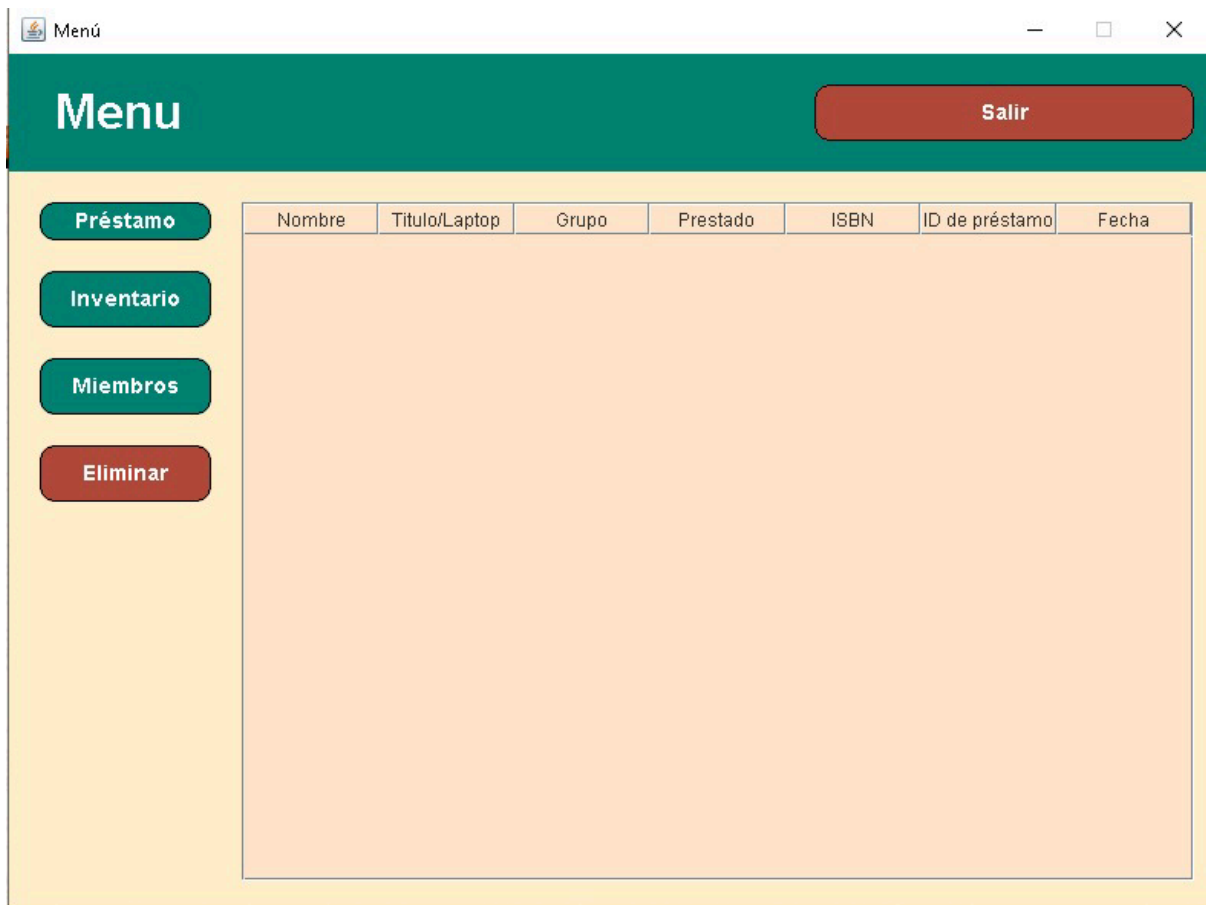
API de Java <https://docs.oracle.com/javase/8/docs/api/>

## 8 Anexos

## 8.1 Manual de usuario



Permite acceder al programa después de introducir la contraseña del usuario



**Salir:** Cierra la sesión o la ventana principal de la aplicación.

**Préstamo:** Navega a la sección donde se gestionan y registran nuevos préstamos de libros o laptops.

**Inventario:** Navega a la sección donde se puede ver, agregar, modificar y eliminar ítems del inventario.

**Miembros:** Navega a la sección donde se gestiona la lista de miembros, incluyendo la posibilidad de registrarlos o eliminarlos.

**Eliminar:** Elimina el registro de préstamo seleccionado en la tabla principal de esta pantalla, que lista los préstamos activos.

Préstamos

Volver

Libros

Laptops

CI del Estudiante

ISBN del Libro

Realizar Prestamo

The screenshot shows a web browser window titled 'Préstamos'. The interface has a dark green header bar with the title 'Préstamos' on the left and a red 'Volver' button on the right. Below the header is a light yellow bar containing two tabs: 'Libros' and 'Laptops'. The 'Laptops' tab is currently selected. The main content area is divided into three sections: the first is labeled 'CI del Estudiante' and contains a white input field; the second is labeled 'ID de la Laptop' and also contains a white input field; the third is a large, empty light orange rectangular area. At the bottom of the form is a dark green button labeled 'Realizar Prestamo'.

Libros / Laptops: Permiten cambiar entre el formulario para prestar un libro y el formulario para prestar una laptop.

Realizar Préstamo: Ejecuta y registra la transacción de préstamo en el sistema, una vez que se han ingresado los datos requeridos.

Volver: Regresa a la pantalla del Menú Principal,



Inventario

Volver

Libros

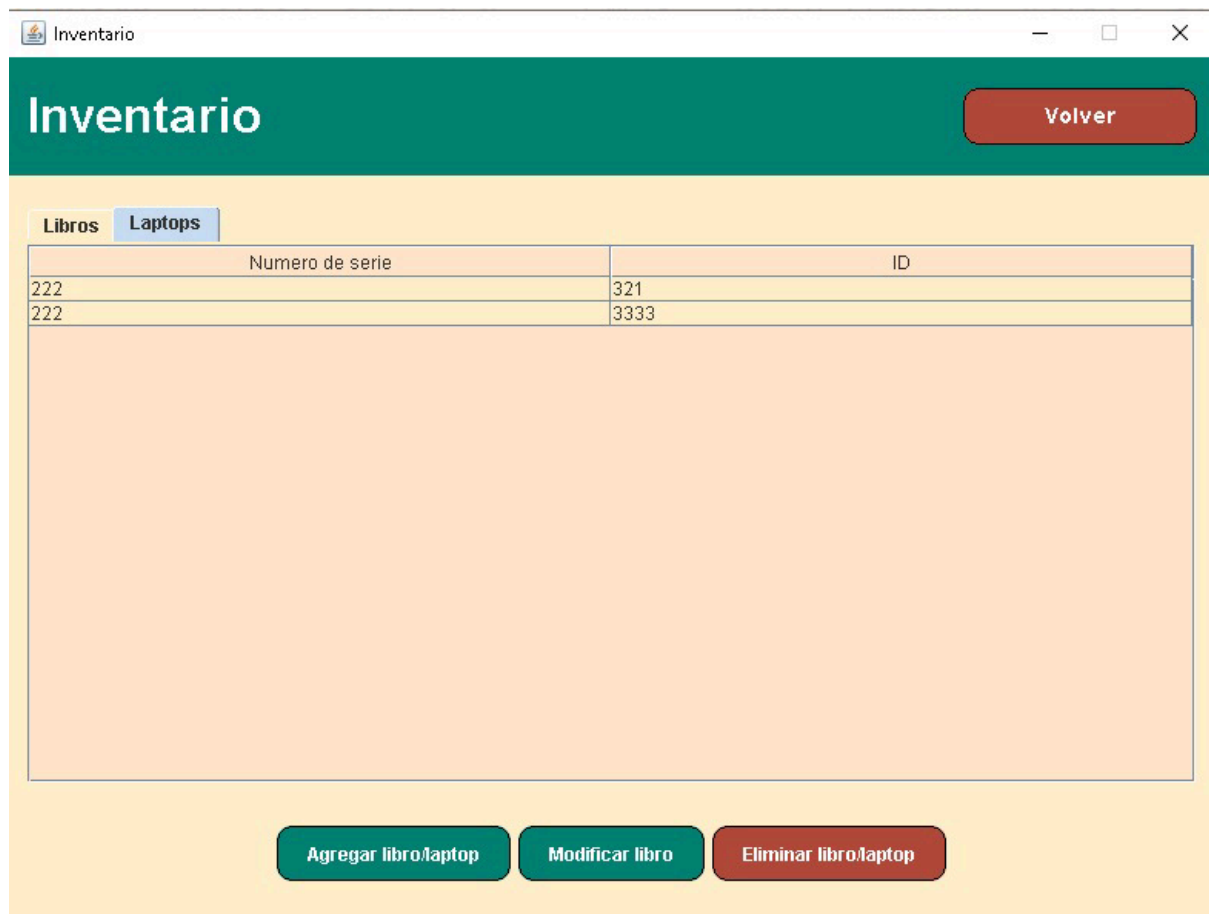
Laptops

Título	Autor	Año	Tipo	Género	ISBN
aaa	aaa	213	Estudio	www	123

Agregar libro/laptop

Modificar libro

Eliminar libro/laptop



Numero de serie	ID
222	321
222	3333

Libros / Laptops: Permiten cambiar entre la lista de ítems en el inventario de Libros y el inventario de Laptops.

Volver: Regresa a la pantalla del Menú Principal.

Agregar libro/laptop: Abre un nuevo formulario para registrar un nuevo libro o laptop en el inventario.

Modificar libro: Abre un formulario para editar los detalles de un libro o laptop ya existente en el inventario.

Eliminar libro/laptop: Elimina el registro del libro o laptop seleccionado de la lista de inventario.

Agregar libro/laptop

Agregar

Volver

Libros Laptops

Numero de serie

ID

Registrar

**Registrar:** Guarda y confirma la adición de los datos del nuevo libro o la nueva laptop al inventario.

**Volver:** Regresa a la pantalla principal de Inventario.

ID del libro para modificar: Es el espacio donde escribes la identificación del libro que quieres editar. Es necesario para que el sistema sepa qué registro va a cambiar.

Otros campos: Te permiten reescribir la información para corregir o actualizar los datos del libro.

Tipo de libro: Te permite cambiar la clasificación del libro, eligiendo entre: Recreativo, Estudio, o Literario.

Modificar : Guarda todos los cambios que hiciste y actualiza la información del libro en la lista.

Volver: Regresa a la pantalla principal de Inventario.

Miembros

volver

CI	Nombre	Grupo
333	gaspar	2do
56557162	Emiliano	1ero
57317945	rarar	2do

Registrar

Eliminar

Miembros

volver

CI	Nombre	Grupo
333	gaspar	2do
56557162	Emiliano	1ero
57317945	rarar	2do

Registrar miembro

Cedula:

Nombre:

Grupo:

OK

Cancel

Registrar

Eliminar

**Miembros**

volver

CI	Nombre	Grupo
333	gaspar	2do
56557162	Emiliano	1ero
57317945	rarar	2do

Registrar Eliminar

**Registrar:** Este botón abre una ventana aparte para que puedas escribir la información de una persona nueva y añadirla a la lista de miembros.

**Eliminar:** Su función es borrar de la lista el registro del miembro que hayas seleccionado previamente.

**Volver:** Este botón te lleva de regreso a la pantalla principal, que es el Menú Principal.

