

---

---

# LIBRARY MANAGER - APLICACIÓN JAVA PARA LA GESTIÓN DE SISTEMAS BIBLIOTECARIOS

---

---



TRABAJO FIN DE GRADO

CURSO 2024-2025

AUTOR

DAVID BEAMONDE SALINAS

DIRECTOR

RODRIGO CABELLO HERNÁNDEZ

GRADO EN DESARROLLO DE APLICACIONES

MULTIPLATAFORMAS

I.E.S VALLE DEL CIDACOS

## **AGRADECIMIENTOS**

A mi tutor, Rodrigo Cabello Hernández, por guiarme durante la realización de este proyecto y la resolución de dudas.

---

---

"Este trabajo se lo dedico a mis padres y a mi hermana, por el apoyo que me han dado durante el proceso, y por haber participado como testers durante la producción para controlar los errores de mi proyecto." - David Beamonde Salinas

---

---

"Nada en la vida debe ser temido, solamente comprendido. Ahora es el momento de comprender más para temer menos" – Marie Curie

## **RESUMEN**

Library Manager es una aplicación de escritorio desarrollada en Java, cuyo objetivo principal es facilitar la gestión de préstamos y devoluciones de libros en una biblioteca. El sistema permite a los usuarios consultar los libros disponibles en el catálogo, filtrando la búsqueda según diferentes características como el autor, el género o el año de publicación.

Además de las funcionalidades para los usuarios, la aplicación incorpora un sistema de gestión administrativa que permite al administrador supervisar y modificar la información almacenada en la base de datos. A través de esta interfaz, se pueden añadir, editar o eliminar registros de libros, autores y cuentas de usuario. Todo el sistema está conectado a una base de datos MySQL que mantiene la persistencia de la información.

El proyecto combina una interfaz gráfica amigable con una estructura lógica clara y robusta, lo que lo convierte en una herramienta útil tanto para bibliotecas pequeñas como para entornos educativos. Library Manager ofrece una experiencia completa de gestión bibliotecaria, integrando funcionalidades esenciales para el control del inventario y la interacción con los usuarios.

### **Palabras clave**

Java, base de datos, biblioteca, libros, préstamos, devoluciones, usuario, administrador.

## ÍNDICE DE CONTENIDOS

Agradecimientos.....	2
Resumen .....	3
Capítulo 1: Introducción.....	8
1.1 Motivación .....	8
1.2 Objetivos.....	9
1.3 Metodología .....	10
1.3.1 Herramientas utilizadas.....	12
1.3.2 Plan de Trabajo .....	12
1.3.3 Presupuesto horario .....	13
1.4 Estructura del documento .....	14
Capítulo 2: Arquitectura general del sistema.....	15
2.1 Visión general.....	15
2.2 Capa de presentación .....	15
2.3 Capa de lógica de negocio .....	16
2.4 Capa de acceso a datos .....	16
2.5 Comunicación entre capas .....	16
2.6 Consideraciones de diseño .....	17
Capítulo 3: Desarrollo .....	18
3.1 Inicio del desarrollo.....	18
3.1.1 Funcionalidad sobre diseño.....	18
3.2 Diseño estructurado y escalable.....	19
3.2.1 Rediseño de la interfaz .....	19
3.2.2 Elección del tipo de base de datos, proveedor y configuración .....	19
3.3 Desarrollo del sistema de gestión de bibliotecas .....	20
3.3.1 Estructura de datos y relaciones .....	20
3.4 Interfaz y controladores .....	20
3.4.1 Diseño modular .....	20
3.4.2 Desarrollo progresivo de las secciones .....	21

3.5 Pruebas y mejoras finales .....	21
3.5.1 Optimización de las conexiones a la base de datos .....	21
3.5.2 Últimas correcciones y añadidos.....	22
Capítulo 4: Interfaz .....	23
4.1 Ventana de Acceso .....	23
4.2 Ventana Principal .....	24
4.3 Ventanas Interiores .....	25
4.3.1 Ventana de libros y consultas .....	26
4.3.2 Ventana de autores .....	27
4.3.3 Ventana de categorías.....	27
4.3.4 Ventana de editoriales.....	28
4.3.5 Ventana de idiomas .....	28
4.3.6 Ventana de usuarios.....	29
4.3.7 Ventana de préstamos.....	29
4.3.8 Ventana de devoluciones .....	30
Capítulo 5: Base de datos .....	31
5.1 Tablas.....	31
5.2 Funcionalidades principales .....	32
5.3 Modificación de base de datos.....	33
Capítulo 6: Requisitos del Sistema .....	34
6.1 Requisitos de Hardware .....	34
6.2 Requisitos de Software .....	34
6.3 Requisitos de Conectividad.....	34
Capítulo 7: Realización de página Web .....	35
7.1 Página principal.....	35
7.2 Sobre mi .....	36
7.3 Documentación .....	36
7.4 Guía de uso .....	36
Capítulo 8: Discusión .....	37
8.1 Ventajas.....	37

8.2 Limitaciones.....	37
8.3 Líneas de trabajo futuro .....	37
8.3.1 Mejoras para la aplicación .....	37
8.3.2 Implementaciones para la página web.....	38
Capítulo 9: Conclusión .....	39
Capítulo 10: Bibliografía .....	40

## **TABLA DE ILUSTRACIONES**

1 Ventana de acceso. Registro de usuarios.....	23
2 Ventana de acceso. Inicio de sesión.....	24
3 Ventana principal. Administrador.....	24
4 Ventana principal. Usuario.....	25
5 Ventana de Libros. Ventana para consultar libros.....	26
6 Ventana de consultas. ....	26
7 Ventana de Autores. Ventana para modificar autores. ....	27
8 Ventana de Categorías. Ventana para eliminar categorías. ....	27
9 Ventana de Editoriales. Ventana para registrar editoriales. ....	28
10 Ventana de Idiomas. Ventana para modificar editoriales. ....	28
11 Ventana de Usuarios. Ventana para registrar usuarios.....	29
12 Ventana de Préstamos. Ventana para modificar préstamos.....	29
13 Ventana de Devoluciones. Ventana para registrar devoluciones.....	30
14 Diagrama de flujo de la base de datos.....	32
15 Página principal de la página web. ....	35
16 Información personal y proyectos realizados.....	36

## **CAPÍTULO 1: INTRODUCCIÓN**

La gestión eficiente de bibliotecas es un aspecto fundamental en entornos educativos, institucionales y culturales. A medida que las bibliotecas evolucionan, se hace cada vez más necesario contar con herramientas digitales que faciliten tanto el acceso a la información como la administración del catálogo bibliográfico. La organización manual de préstamos, devoluciones y registros puede resultar ineficiente y propensa a errores, especialmente cuando el volumen de usuarios y libros es considerable.

Library Manager es una aplicación de escritorio desarrollada en Java que busca ofrecer una solución práctica y completa para la gestión de bibliotecas. Este software permite a los usuarios consultar los libros disponibles mediante filtros personalizados, así como realizar préstamos y devoluciones de manera intuitiva. Por otro lado, proporciona al administrador un conjunto de herramientas para gestionar el sistema: desde el control de cuentas de usuario hasta la creación, modificación o eliminación de registros de libros y autores en la base de datos.

El objetivo principal de este proyecto es mejorar la experiencia tanto del usuario como del administrador, ofreciendo una interfaz clara, una estructura lógica sólida y una integración completa con una base de datos persistente. De esta forma, Library Manager representa un avance hacia la automatización y digitalización de los servicios bibliotecarios tradicionales.

### **1.1 MOTIVACIÓN**

La motivación detrás de este proyecto surge tanto de una necesidad técnica como de una experiencia personal. En un mundo donde la digitalización avanza rápidamente, muchas bibliotecas aún dependen de métodos manuales para gestionar sus catálogos, préstamos y devoluciones. Esta falta de automatización puede generar desorganización, pérdida de información y una experiencia insatisfactoria para clientes y trabajadores.

Desde una perspectiva más personal, el hecho de que mi hermana haya publicado un libro despertó en mí un mayor interés por el mundo editorial y la importancia de preservar, organizar y difundir obras literarias. La posibilidad de contribuir con una herramienta que facilite el acceso a libros y fomente la lectura y el conocimiento resulta especialmente significativa.

Desde el punto de vista académico y profesional, Library Manager busca aportar una solución concreta a la gestión bibliotecaria mediante el desarrollo de una aplicación de escritorio robusta y funcional. El uso de Java como lenguaje de programación, junto con una base de datos MYSQL para la persistencia de información, permite ofrecer una experiencia



completa tanto para los usuarios como para los administradores. A nivel social, el proyecto promueve el acceso organizado y eficiente a los libros, facilitando así el desarrollo educativo, cultural y personal de sus usuarios.

En definitiva, este proyecto no solo representa un ejercicio técnico, sino también una forma de acercar la tecnología al mundo de los libros y la educación, con la intención de que sistemas como este puedan ser utilizados en bibliotecas escolares, centros culturales o pequeñas instituciones que buscan modernizar sus servicios sin recurrir a soluciones complejas o costosas.

## **1.2 OBJETIVOS**

A partir de la motivación expuesta, este proyecto plantea los siguientes objetivos **generales**:

1. Desarrollar una aplicación de escritorio multiplataforma (en este caso, enfocada en Windows) que permita gestionar de forma integral el funcionamiento de una biblioteca.
2. Ofrecer una herramienta de uso sencillo y bajo requerimiento técnico, adaptable a distintos entornos bibliotecarios.
3. Facilitar la organización y el control de libros, usuarios, préstamos y devoluciones mediante una interfaz clara y funcional.

Más en detalle, estos objetivos generales comprenden los siguientes objetivos **específicos**:

- A. Diseñar una interfaz gráfica limpia, clara e intuitiva que permita a cualquier usuario interactuar con el sistema sin necesidad de conocimientos técnicos previos.
- B. Diseñar una base de datos estructurada y segura que permita almacenar la información relativa a libros, autores, usuarios y registros de préstamo.
- C. Implementar funcionalidades CRUD (crear, leer, actualizar y eliminar) tanto para libros como para usuarios, facilitando su gestión.
- D. Desarrollar un sistema de búsqueda que permita a los usuarios consultar el catálogo por diferentes características (autor, título, género, etc.) antes de acudir a la biblioteca.
- E. Implementar un sistema de préstamos y devoluciones que registre las fechas correspondientes, mantenga actualizado el estado de los libros y permita saber en todo momento a quién se ha prestado cada ejemplar.
- F. Garantizar la compatibilidad con sistemas operativos Windows y asegurar que la instalación y uso del sistema no requiera configuraciones técnicas complejas.

- G. Diseñar la arquitectura de la aplicación pensando en futuras ampliaciones, como la integración de recomendaciones, reportes estadísticos o notificaciones automáticas de devolución.
- H. Diseñar la página web para mostrar documentación y una guía para que los usuarios aprendan a usar la aplicación correctamente.

### 1.3 METODOLOGÍA

Para el desarrollo de la aplicación se empleó una metodología de tipo incremental, que ha permitido construir el sistema por etapas, validando cada módulo antes de avanzar al siguiente. Este enfoque ha facilitado la detección temprana de errores y la mejora continuada del producto.

En primer lugar, se llevó a cabo una **selección de las herramientas y tecnologías** más adecuadas para garantizar el correcto funcionamiento de la aplicación. Para ello, se recopiló información sobre distintos entornos de desarrollo, bases de datos y frameworks, evaluando su compatibilidad, rendimiento y facilidad de integración en sistemas Windows.

Posteriormente, se realizó un **análisis de requisitos**, identificando las necesidades esenciales de una biblioteca. Entre ellas, se destacó la importancia de contar con funcionalidades como el registro y consulta de libros, la gestión de usuarios, el control de préstamos y devoluciones, y una interfaz gráfica que resultase intuitiva para el usuario. Asimismo, se estableció como criterio fundamental que la aplicación no requiriera configuraciones complejas y funcionase correctamente en sistemas operativos Windows.

El siguiente paso fue el **diseño de la base de datos**, en el que se definieron siete tablas principales: libros, autores, categorías, editoriales, idiomas, usuarios y préstamos. Estas tablas se organizaron mediante relaciones lógicas bien definidas que garantizan la integridad referencial de los datos y evitan redundancias innecesarias, permitiendo un acceso eficiente y coherente a la información almacenada.

En cuanto al **desarrollo del backend**, se optó por utilizar Java, lenguaje de programación adquirido durante el grado, y debido a su robustez, portabilidad y amplia compatibilidad con Windows. Para la gestión de los datos se empleó MySQL, una base de datos relacional ligera y fácil de implementar. Además, se desarrolló una página web complementaria utilizando HTML, CSS y JavaScript, con el objetivo de explorar una posible futura expansión a entornos web, ofreciendo estilos personalizados y funcionalidades dinámicas. La web se ha desarrollado con el objetivo de documentar la aplicación, como instalarla y como utilizarla.

El **desarrollo de la interfaz gráfica** se realizó mediante JavaFX, un framework que permite construir interfaces modernas y visualmente agradables en Java. Se priorizó la claridad visual, una navegación sencilla y el acceso rápido a las funcionalidades más importantes, de modo que incluso usuarios sin experiencia técnica pudieran utilizar el sistema sin dificultad.

Para asegurar el correcto funcionamiento de la aplicación, se realizaron **pruebas y validaciones** exhaustivas. Estas incluyeron pruebas funcionales en todos los módulos del sistema: registro y búsqueda de libros, gestión de préstamos y devoluciones, manejo de errores y validaciones de entrada. Además, se comprobó su rendimiento en distintos equipos con sistema operativo Windows, asegurando su estabilidad y compatibilidad.

Finalmente, se elaboró la **documentación del sistema**, explicando su instalación y uso. También se generó un archivo ejecutable que permite distribuir la aplicación fácilmente, facilitando su instalación en cualquier equipo sin necesidad de un entorno de desarrollo adicional.

### 1.3.1 HERRAMIENTAS UTILIZADAS

Herramientas	Uso
IntelliJ Idea Ultimate	Entorno de desarrollo integrado para crear la aplicación en Java
JavaFX	Framework de Java para el desarrollo de interfaces gráficas
Maven	Herramienta de automatización y gestión de proyectos
Clever Cloud	Plataforma de servicio para guardar bases de datos en la nube
MySQL	Sistema de almacenamiento de datos
HeidiSQL	Herramienta para gestión de bases de datos
GitHub	Plataforma de control de versiones y almacenamiento de archivos
Visual Studio Code	Programa de edición de código Java, HTML, CSS y JS
Microsoft Word	Herramienta utilizada para la creación de este documento

### 1.3.2 PLAN DE TRABAJO

El plan de trabajo seguido durante la realización del proyecto ha permitido una organización eficiente y una correcta gestión del tiempo. Las tareas han sido distribuidas de manera estratégica de forma que el avance sea lineal y coherente.

Objetivo	Diciembre	Enero	Febrero	Marzo	Abril	Mayo
A						
B						
C						
B						
E						
F						
G						
H						

### 1.3.3 PRESUPUESTO HORARIO

El desarrollo total de la aplicación estaba planificado en 70 horas, distribuidas de la siguiente manera:

Planificar la interfaz gráfica	5h
Crear la interfaz gráfica	15h
Planificar y crear la base de datos	3h
Codificar la aplicación, añadir las funciones y vincularlas con la interfaz	24h
Conectar la aplicación con la base de datos, crear consultas y ver que funcionan	8h
Hacer pruebas y arreglar errores	15h

Finalmente, el tiempo invertido en el desarrollo total ha sido de 87 horas, que se distribuyes de esta manera:

Planificar la interfaz gráfica	4h
Crear la interfaz gráfica	12h
Planificar y crear la base de datos	9h
Codificar la aplicación, añadir las funciones y vincularlas con la interfaz	31h
Conectar la aplicación con la base de datos, crear consultas y ver que funcionan	17h
Hacer pruebas y arreglar errores	6h
Hacer la página web	8h

#### 1.4 ESTRUCTURA DEL DOCUMENTO

Este documento se estructura de la siguiente manera:

- En el primer capítulo, esta introducción, se presentan brevemente la motivación, así como los objetivos propuestos para cumplir durante el desarrollo, las herramientas utilizadas y el plan de trabajo.
- En el segundo capítulo se presenta la arquitectura general del sistema para poder formar una visión de mapa completa del proyecto y de la base de datos utilizada.
- En el tercer capítulo se describe el proceso del desarrollo que he seguido para llevar a cabo el proyecto.
- En el cuarto capítulo se muestran imágenes de la interfaz, explicando brevemente su función.
- En el quinto capítulo se detalla la estructura y relaciones de la base de datos que se ha creado para el proyecto.
- En el sexto capítulo se muestran los requisitos del sistema mínimos y recomendados para que la aplicación funcione correctamente.
- En el séptimo capítulo se ha desarrollado el proceso de implementación de la web de Library Manager a modo de documentación y guía de instalación.
- En el octavo capítulo se discuten desde un punto de vista técnico, las ventajas y limitaciones de mi trabajo y posibles mejoras a futuro.
- Por último, en el noveno capítulo se detalla la conclusión de mi proyecto.

Finalmente, se presenta la bibliografía completa referenciada por el trabajo.

## CAPÍTULO 2: ARQUITECTURA GENERAL DEL SISTEMA

Este capítulo detalla la arquitectura general del sistema desarrollado en el proyecto, proporcionando una visión completa y estructurada de los componentes involucrados y sus interacciones. La comprensión de esta arquitectura es fundamental para entender el flujo.

### 2.1 VISIÓN GENERAL

El sistema **Library Manager** ha sido diseñado siguiendo una arquitectura modular que separa claramente las responsabilidades de cada componente, facilitando así su mantenimiento, escalabilidad y comprensión. La aplicación se compone de tres capas principales: la capa de presentación, la capa de lógica de negocio y la capa de acceso a datos. Esta estructura permite una interacción eficiente entre el usuario y la base de datos, asegurando una experiencia fluida y coherente.

### 2.2 CAPA DE PRESENTACIÓN

La interfaz gráfica de usuario (GUI) se ha desarrollado utilizando **JavaFX**, un framework de Java que permite la creación de interfaces modernas y responsivas. Esta capa es responsable de la interacción directa con el usuario, proporcionando ventanas y controles intuitivos para realizar operaciones como el registro de libros, gestión de usuarios, y control de préstamos y devoluciones.

Para el diseño de las interfaces, se ha utilizado **FXML**, un lenguaje basado en XML que permite describir la estructura de la GUI de manera declarativa. Esto facilita la separación entre el diseño de la interfaz y la lógica de negocio, permitiendo que diseñadores y desarrolladores trabajen de manera independiente y eficiente.

## 2.3 CAPA DE LÓGICA DE NEGOCIO

La lógica de negocio se encuentra encapsulada en una serie de clases Java que gestionan las operaciones fundamentales del sistema. Esta capa actúa como intermediaria entre la interfaz de usuario y la base de datos, procesando las solicitudes del usuario, aplicando las reglas de negocio y coordinando las respuestas adecuadas.

Por ejemplo, cuando un usuario solicita el préstamo de un libro, esta capa verifica la disponibilidad del ejemplar, actualiza el estado del préstamo en la base de datos y proporciona la retroalimentación correspondiente a la interfaz de usuario. Esta separación asegura que la lógica del sistema sea coherente y reutilizable en diferentes contextos.

## 2.4 CAPA DE ACCESO A DATOS

La persistencia de la información se gestiona a través de **MySQL**, un sistema de gestión de bases de datos relacional que ofrece fiabilidad y rendimiento. La aplicación se conecta a la base de datos utilizando **JDBC** (Java Database Connectivity), una API que permite la ejecución de sentencias SQL desde Java.

La base de datos está estructurada en varias tablas que representan las entidades principales del sistema, como libros, autores, categorías, editoriales, idiomas, usuarios y préstamos. Las relaciones entre estas tablas están definidas mediante claves primarias y foráneas, asegurando la integridad referencial y facilitando consultas eficientes.

La aplicación busca el archivo "database.properties" el cual contiene toda la información para la conexión con la base de datos. Esto se hace para permitir al usuario modificar los datos de la conexión, y de esta forma poder cambiar los datos de acceso para conectarse con otra base de datos diferente.

## 2.5 COMUNICACIÓN ENTRE CAPAS

La interacción entre las diferentes capas del sistema se realiza de manera estructurada y coherente. La capa de presentación captura las acciones del usuario y las transmite a la capa de lógica de negocio, que procesa la información y, si es necesario, interactúa con la base de datos a través de la capa de acceso a datos. Una vez obtenida la respuesta, esta se propaga de vuelta a la interfaz de usuario, proporcionando la información oportuna al usuario.



Esta arquitectura en capas no solo mejora la organización del código, sino que también facilita la implementación de futuras mejoras o cambios en el sistema, ya que cada componente puede ser modificado de manera independiente, siempre que se mantengan las interfaces de comunicación establecidas.

## 2.6 CONSIDERACIONES DE DISEÑO

Durante el desarrollo del sistema, se han tenido en cuenta diversas consideraciones para asegurar su calidad y usabilidad:

- **Modularidad:** El código se ha organizado en paquetes y clases que representan claramente las diferentes funcionalidades del sistema, facilitando su mantenimiento y comprensión.
- **Escalabilidad:** La arquitectura permite la incorporación de nuevas funcionalidades, como recomendaciones personalizadas o notificaciones de devolución, sin requerir una reestructuración significativa del sistema.
- **Usabilidad:** Se ha priorizado la creación de una interfaz intuitiva y fácil de usar, permitiendo que usuarios sin conocimientos técnicos puedan interactuar con el sistema de manera efectiva.
- **Compatibilidad:** La aplicación está diseñada para funcionar en sistemas operativos Windows, asegurando una amplia compatibilidad y facilidad de instalación.

## CAPÍTULO 3: DESARROLLO

La aplicación se construyó siguiendo el patrón de arquitectura **Modelo-Vista-Controlador (MVC)**, que permite una separación clara entre la lógica de negocio, la interfaz gráfica y el control del flujo de la aplicación. A continuación, explico para que sirve cada uno de ellos:

### **Modelo-Vista-Controlador (MVC)**

**Modelo:** Contiene las clases que representan los datos y su lógica (Libro, Usuario, Préstamo), así como la conexión con la base de datos mediante JDBC, una librería de Java para conectar con bases de datos.

**Vista:** Incluye las interfaces desarrolladas con el framework JavaFX, diseñadas para ser intuitivas y fáciles de usar. Aquí se incluyen todas las ventanas de la aplicación además de los estilos (CSS) utilizados para darles color y claridad.

**Controlador:** Gestiona los eventos de la vista, validando entradas, ejecutando acciones en el modelo y actualizando la vista según sea necesario.

### 3.1 INICIO DEL DESARROLLO

#### 3.1.1 FUNCIONALIDAD SOBRE DISEÑO

Comencé el desarrollo de la aplicación hace aproximadamente seis meses. En esta primera etapa, mi prioridad era implementar la funcionalidad básica, por lo que diseñé una interfaz muy sencilla, sin colores ni elementos visuales destacados. Utilicé pocas clases, con bastante código repetido, y no presté atención a la optimización, ya que consideraba que lo más importante era que todo funcionara correctamente. Además, intenté crear un servidor web utilizando Node.js junto con Ngrok, con el objetivo de acceder remotamente a los datos almacenados. Quería alojar la base de datos en un servidor propio, lo cual me obligó a aprender a utilizar ambas herramientas desde cero.

Después de unos 35 días, noté que la interfaz de usuario no era lo mejor que podía hacer. Dejar el diseño para el final me impedía ver cómo iba avanzando hacia el producto final. Además, para almacenar los datos y permitir conexiones, el servidor requería que mi ordenador estuviera encendido durante todas las horas del día, lo que tenía muchos inconvenientes: cortes de luz, fallos de internet, falta de almacenamiento y alto consumo de recursos. Eso era poco eficiente y difícil de mantener. Por eso, decidí empezar de nuevo.

## 3.2 DISEÑO ESTRUCTURADO Y ESCALABLE

En este capítulo solo muestro como fueron mis diferentes etapas de desarrollo, las ilustraciones sobre el resultado de la interfaz se ven en el [capítulo 4](#).

### 3.2.1 REDISEÑO DE LA INTERFAZ

En esta segunda etapa, enfoqué el proyecto de forma diferente. Mi nuevo objetivo fue crear una interfaz limpia, intuitiva y fácil de mantener, que facilitara la incorporación de nuevas funcionalidades en el futuro. Comencé con la ventana principal, diseñando cuadros para cada función con el fin de modularizar el trabajo y centrarme en completar cada apartado por separado.

El siguiente paso fue desarrollar el formulario de registro e inicio de sesión. Para ello, era necesario definir la primera tabla de la base de datos, que me indicaría cuántos campos de texto serían necesarios.

### 3.2.2 ELECCIÓN DEL TIPO DE BASE DE DATOS, PROVEEDOR Y CONFIGURACIÓN

Inicialmente pensé en usar una base de datos local, pero finalmente opté por una solución en la nube, ya que esto reducía el tamaño de la aplicación y permitía el acceso remoto desde cualquier lugar, sin necesidad de crear un servidor propio como planteé inicialmente en el proyecto. Esta opción también facilitaba la expansión, permitiendo que varias bibliotecas pudieran compartir la misma base de datos si fuera necesario.

Para alojar la base de datos en la nube busqué varias opciones y la que más me gustó fue Clever Cloud, un hosting con bases de datos tanto gratuitas como de pago. Podía elegir entre MySQL, PostgreSQL o MongoDB, y me decidí por MySQL, ya que es una de las más usadas y había trabajado con ella anteriormente. La opción gratuita permitía solo 5 conexiones simultáneas. Esto era un reto ya que no podía hacer demasiadas consultas a la base de datos al mismo tiempo.

Finalmente, creé la tabla de **Usuarios** con las siguientes columnas: DNI, nombre, apellidos, sexo, email, contraseña, teléfono, dirección, fecha de nacimiento, fecha de registro y permiso. Y después añadí los campos de texto necesarios para la ventana de registro e inicio de sesión.

### 3.3 DESARROLLO DEL SISTEMA DE GESTIÓN DE BIBLIOTECAS

#### 3.3.1 ESTRUCTURA DE DATOS Y RELACIONES

La siguiente fase fue la creación de la tabla **Libros**, que incluía los campos de ISBN, título, autor, categoría, editorial, número de páginas, idioma, año de publicación y estado. Pronto observé que ciertos datos como autor, categoría, editorial e idioma se repetían frecuentemente. Para mejorar la eficiencia y la normalización, decidí crear tablas independientes para cada uno de ellos, estableciendo relaciones mediante claves foráneas.

Una vez finalizado, continué con la tabla **Libros**, que incluía los siguientes campos: ISBN, título, autor, categoría, editorial, número de páginas, idioma, año de publicación y estado. En este punto noté que varios valores como autor, categoría, editorial e idioma podían repetirse con frecuencia. Entonces, decidí crear una tabla aparte para cada uno de ellos estableciendo claves foráneas en la tabla **Libros**.

La tabla de **Autores** era la más compleja de estas cuatro, con los siguientes campos: ID, nombre, apellido, país y fecha de nacimiento. Las tablas de **Categorías**, **Editoriales** e **Idiomas** eran las más sencillas de todas, cada una con solo dos campos: ID y nombre.

También creé la tabla **Prestamos** para guardar todos los préstamos y devoluciones. Esta tabla incluye las siguientes columnas: ID, usuario, libro, fecha de préstamo, fecha de devolución, fecha de devolución real y multa. Al igual que en otras tablas, añadí claves foráneas para enlazarla con usuarios y libros que son las tablas principales.

### 3.4 INTERFAZ Y CONTROLADORES

En este momento, decidí que no solo el administrador de la aplicación pudiese ver información relevante de los libros, sino que también quería que los usuarios desde sus casas tuvieran acceso para hacer consultas sobre los libros y pudieran modificar datos sobre su usuario.

#### 3.4.1 DISEÑO MODULAR

Hice una interfaz similar para todas las ventanas en cada apartado: Libros, Autores, Editoriales, Categorías, Idiomas, Usuarios, Prestamos y Devoluciones. Incluye un controlador general para cada una de las ventanas y un sub controlador para cada una de las funciones como registrar, modificar y eliminar. El apartado de libros también tiene una función extra que sirve para consultar.

### **3.4.2 DESARROLLO PROGRESIVO DE LAS SECCIONES**

En este punto del desarrollo, empecé creando los apartados de autores, categorías, editoriales e idiomas. Estos apartados son exclusivos para los administradores. Como fueron los más sencillos, todos ellos los hice en un intervalo de dos semanas, incluyendo la interfaz, los controladores y las consultas necesarias para registrar, modificar y eliminar cualquier valor guardado en las correspondientes tablas de la base de datos.

Después seguí con el apartado de usuarios, que solo para el estuve más de una semana. Esto se debe a que los administradores pueden registrar, modificar y eliminar usuarios, pero los usuarios tienen permisos para poder modificar únicamente sus propios datos.

Lo siguiente fue desarrollar el apartado de libros que era el más complejo de todos debido a la cantidad de campos y relaciones con otras tablas de la base de datos, como los autores, categorías, editoriales e idiomas. Era la interfaz con más campos de texto y con las consultas más complejas. Además, los administradores no solo tenían que poder registrar, modificar y eliminar datos, también era necesario poder consultar los libros. Sin olvidarnos de los usuarios que también deben poder consultarlos.

Esto hizo que tuviera que crear consultas complejas con posibilidad de añadir condiciones para poder filtrar los libros según ciertos criterios seleccionados en la interfaz. Estuve más de tres semanas para conseguir que funcionase todo correctamente.

Finalmente, terminé esta fase con el desarrollo de los apartados de préstamos y devoluciones, los cuales también requerían el uso de varias tablas. Tras tres semanas, conseguí terminarlo. Lo más difícil fue tener que modificar el estado del libro cuando estaba prestado y volver a ponerlo disponible después de la devolución.

## **3.5 PRUEBAS Y MEJORAS FINALES**

### **3.5.1 OPTIMIZACIÓN DE LAS CONEXIONES A LA BASE DE DATOS**

Durante las pruebas finales, detecté que algunas secciones no cargaban correctamente debido al límite de cinco conexiones simultáneas. Para solucionarlo, modifiqué la lógica de las consultas, haciendo que se ejecutaran secuencialmente.

Noté que la interfaz cargaba con lentitud al esperar respuestas de la base de datos. Recordando lo aprendido en clase sobre hilos, decidí ejecutar las consultas en hilos secundarios para que la interfaz principal no se bloqueara, y dejar el renderizado de la aplicación en el hilo principal.

### 3.5.2 ÚLTIMAS CORRECCIONES Y AÑADIDOS

Quedando apenas una semana para la entrega, detecté errores al eliminar registros que estaban relacionados con claves foráneas. Esto ocurría, por ejemplo, al eliminar autores o categorías que todavía estaban referenciados por libros. Lo resolví permitiendo valores nulos en esos campos para evitar conflictos.

Para finalizar con la aplicación añadí un archivo modificable llamado **“database.properties”**, este permite cambiar los datos de acceso de la base de datos para permitir al usuario conectarse a una base de datos propia si así lo desea.

A falta de pocos días de terminar el proceso, decidí diseñar y desarrollar una página web de apoyo, en la que los usuarios pudiesen consultar documentación sobre la aplicación, como, por ejemplo, como instalarla o como utilizar los diferentes apartados que vienen en ella. Finalmente, la página fue subida a un nuevo repositorio de GitHub y lo vinculé con **Netlify** para publicar la web en internet. Además, le di un dominio propio que ya tenía en mi poder.

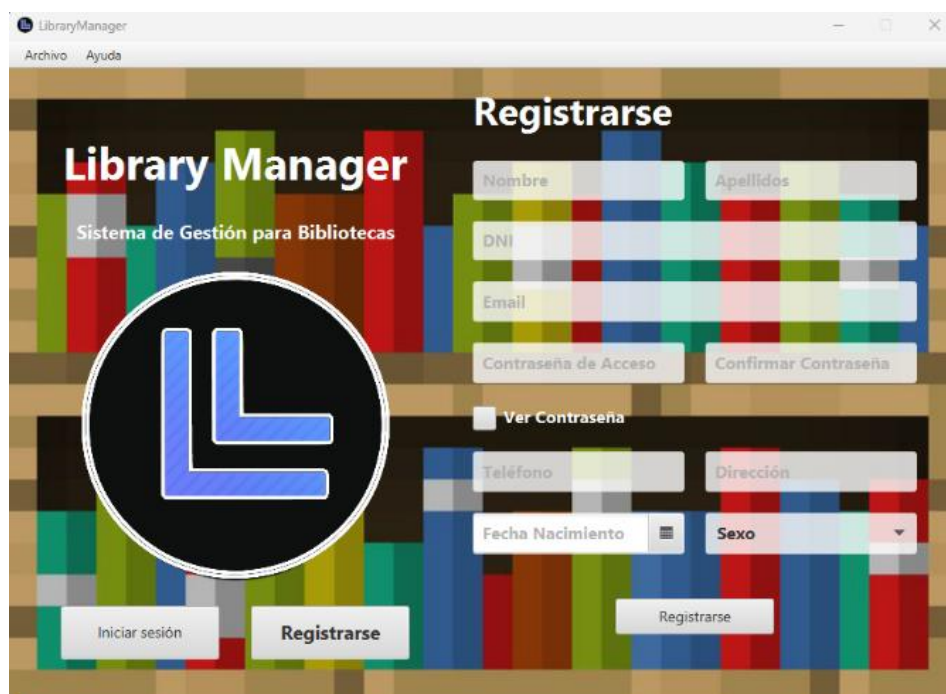
## CAPÍTULO 4: INTERFAZ

La interfaz gráfica de usuario (GUI) es un componente esencial en el desarrollo de aplicaciones, ya que constituye el principal medio de interacción entre el usuario y el sistema. Una interfaz bien diseñada no solo facilita la navegación y el uso eficiente de la aplicación, sino que también mejora la experiencia del usuario al presentar la información de manera clara y ordenada.

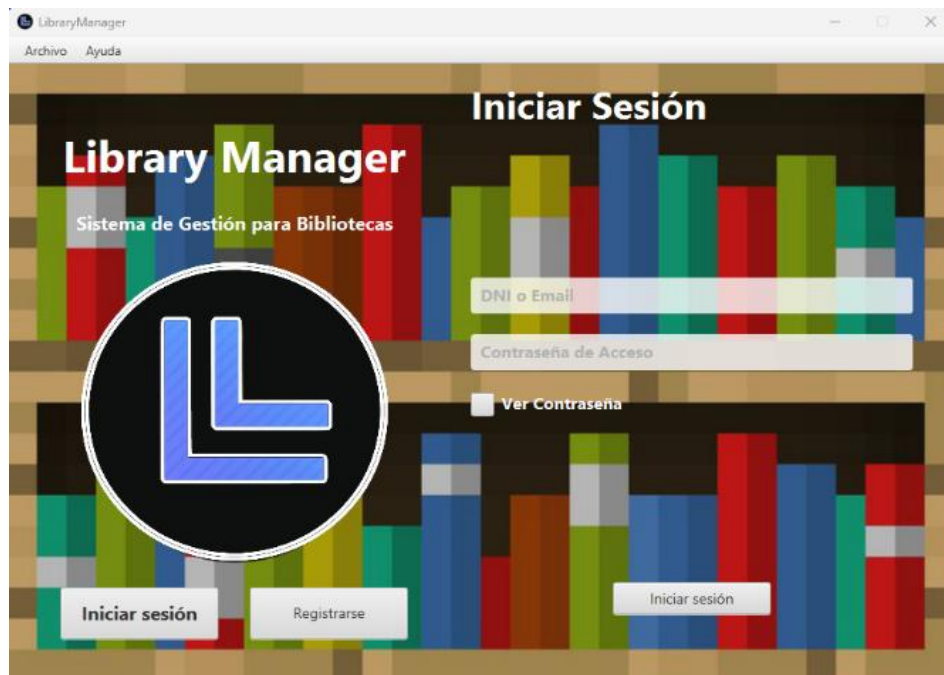
En el desarrollo de esta aplicación, se ha priorizado la creación de una interfaz intuitiva, coherente y estéticamente agradable. Se ha prestado especial atención a la distribución de los elementos visuales, la jerarquía de la información, el uso adecuado de colores y tipografías, así como al espaciado, con el objetivo de optimizar la usabilidad y accesibilidad del sistema.

### 4.1 VENTANA DE ACCESO

La ventana de acceso es la primera pantalla que se presenta al iniciar la aplicación. En ella, los usuarios pueden autenticarse mediante sus credenciales o registrarse para obtener acceso al sistema. Este mecanismo de autenticación garantiza la seguridad y privacidad de los datos, asegurando que solo usuarios autorizados puedan interactuar con la aplicación.

The image shows a screenshot of a web application window titled "LibraryManager". The window has a menu bar with "Archivo" and "Ayuda". The main content area has a dark background with a colorful, abstract pattern of vertical bars in red, green, blue, and yellow. On the left, the text "Library Manager" is displayed in a large, bold, white font, with "Sistema de Gestión para Bibliotecas" in a smaller font below it. A large, circular logo with a stylized "L" is centered on the left. On the right, there is a "Registrarse" section with several input fields: "Nombre", "Apellidos", "DNI", "Email", "Contraseña de Acceso", "Confirmar Contraseña", "Teléfono", "Dirección", "Fecha Nacimiento", and "Sexo". There is also a "Ver Contraseña" checkbox. At the bottom, there are two buttons: "Iniciar sesión" and "Registrarse".

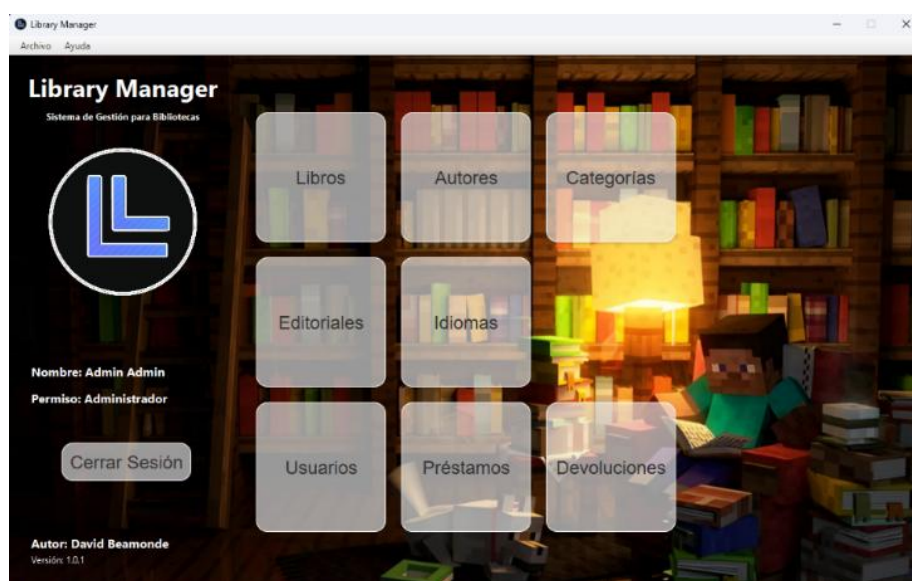
1 Ventana de acceso. Registro de usuarios.



2 Ventana de acceso. Inicio de sesión.

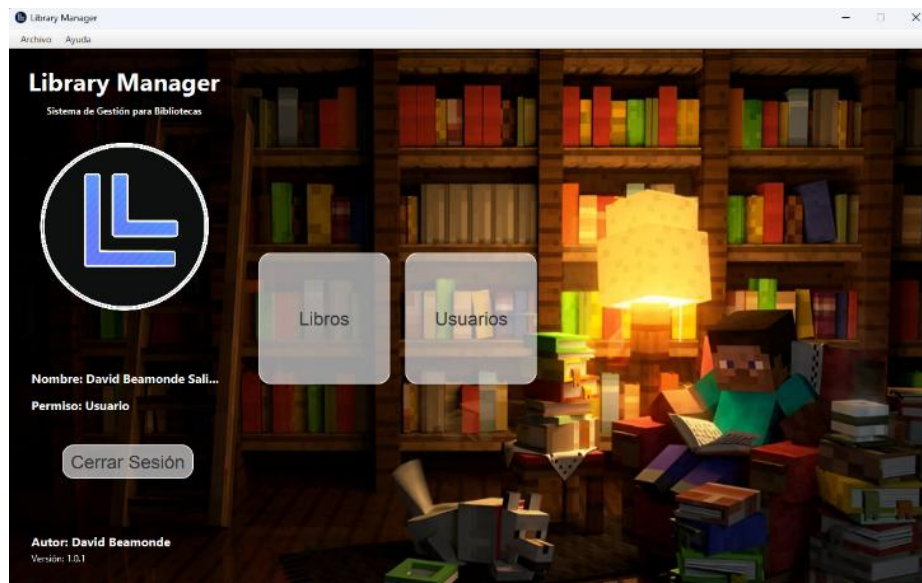
## 4.2 VENTANA PRINCIPAL

Una vez autenticado, el usuario accede a la ventana principal, que actúa como panel de control del sistema. Desde esta interfaz, se pueden realizar operaciones de registro, modificación y eliminación de información en la base de datos. La ventana principal presenta diferentes secciones representadas por cuadros interactivos; al seleccionar uno de ellos, se despliega una nueva ventana correspondiente al módulo elegido.



3 Ventana principal. Administrador.





4 Ventana principal. Usuario.

La funcionalidad y el acceso a los distintos módulos varían según el rol del usuario:

- **Administradores:** Tienen acceso completo a todos los módulos del sistema, incluyendo la gestión de autores, categorías, editoriales, idiomas, libros, usuarios, préstamos y devoluciones.
- **Usuarios:** Su acceso está limitado a los módulos de consulta y modificación de sus datos personales, así como a la búsqueda y visualización de libros disponibles.

### 4.3 VENTANAS INTERIORES

Las ventanas internas mantienen una estructura uniforme para facilitar la navegación y comprensión por parte del usuario. Cada una de estas ventanas incluye subapartados específicos para las operaciones de registrar, modificar y eliminar datos, proporcionando un control completo sobre la información gestionada.

En el caso de los administradores, además de las operaciones mencionadas, se les permite realizar consultas avanzadas sobre los libros, utilizando múltiples filtros como autor, editorial, número de páginas, año de publicación, entre otros, para refinar los resultados de búsqueda.

Por otro lado, los usuarios tienen la capacidad de modificar únicamente sus datos personales y realizar consultas sobre los libros disponibles, aplicando los mismos filtros que los administradores para facilitar la localización de títulos específicos.

4.3.1 VENTANA DE LIBROS Y CONSULTAS

En esta ventana se pueden registrar, modificar, eliminar y consultar libros.

Libros

Registrar

Modificar

Eliminar

Consultar

Volver

ISBN

Título

Autor

Categoría

Editorial

Idioma

Número páginas

Año publicación

Estado

Consultar

Limpiar

5 Ventana de Libros. Ventana para consultar libros.

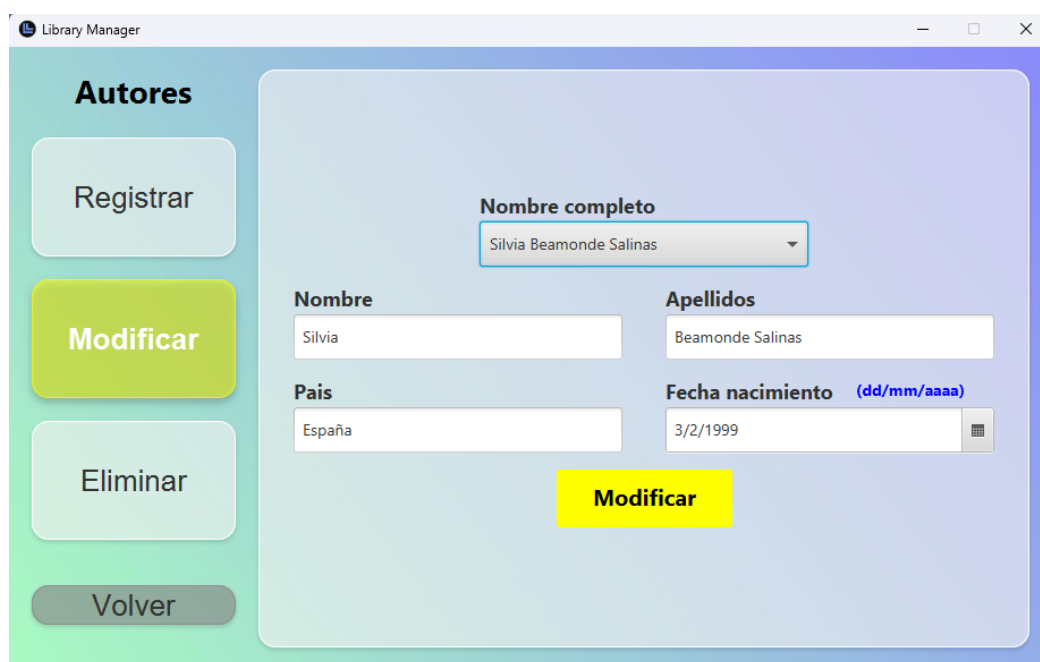
En esta ventana se muestran el resultado de la consulta de la ventana libros.

ISBN	Título	Autor	Categoría	Editorial	Páginas	Idioma	Año	Estado
978-84-1144-159-9	Pirata A Bordo	Silvia Beamonde Salinas	Autobiografía	Letrame	113	Español	2022	Disponible

6 Ventana de consultas.

### 4.3.2 VENTANA DE AUTORES

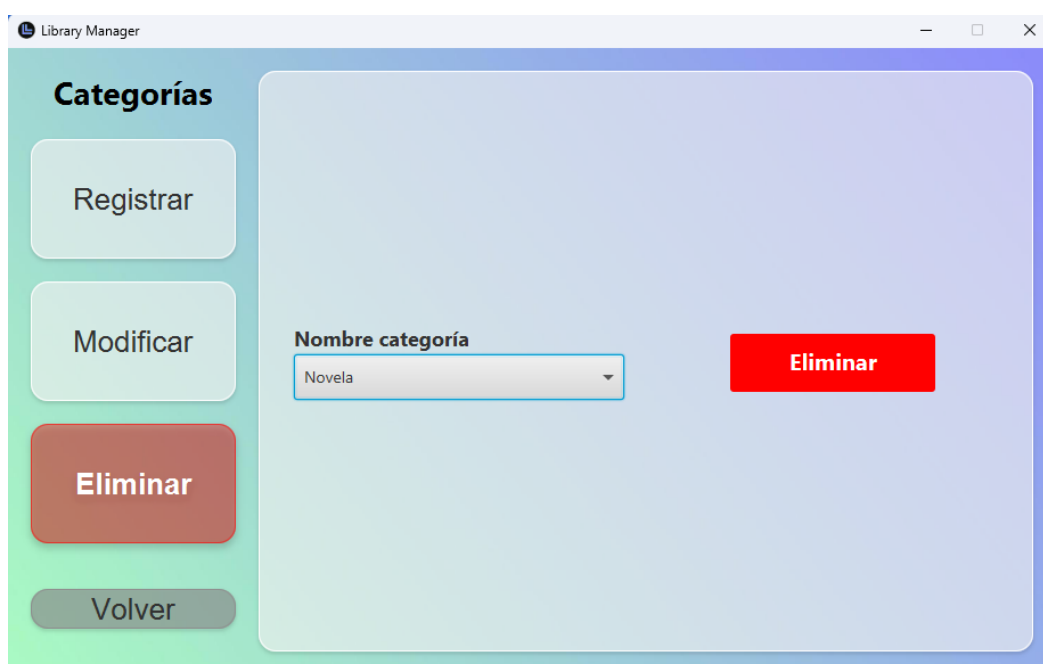
En esta ventana se pueden registrar, modificar y eliminar autores.



7 Ventana de Autores. Ventana para modificar autores.

### 4.3.3 VENTANA DE CATEGORÍAS

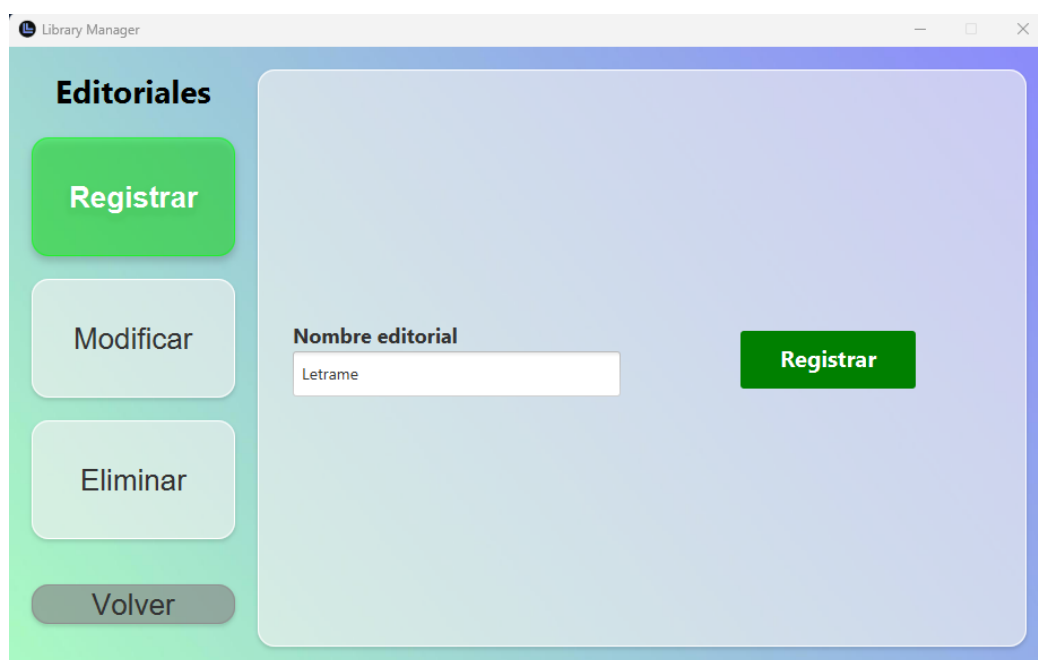
En esta ventana se pueden registrar, modificar y eliminar categorías.



8 Ventana de Categorías. Ventana para eliminar categorías.

#### 4.3.4 VENTANA DE EDITORIALES

En esta ventana se pueden registrar, modificar y eliminar editoriales.



The screenshot shows a window titled 'Library Manager' with a sidebar on the left and a main content area. The sidebar is titled 'Editoriales' and contains four buttons: 'Registrar' (green), 'Modificar' (light blue), 'Eliminar' (light blue), and 'Volver' (grey). The main content area has a light blue background and contains a form with a label 'Nombre editorial' above a text input field containing 'Letrame'. To the right of the input field is a green 'Registrar' button.

9 Ventana de Editoriales. Ventana para registrar editoriales.

#### 4.3.5 VENTANA DE IDIOMAS

En esta ventana se pueden registrar, modificar y eliminar idiomas.



The screenshot shows a window titled 'Library Manager' with a sidebar on the left and a main content area. The sidebar is titled 'Idiomas' and contains four buttons: 'Registrar' (light blue), 'Modificar' (yellow-green), 'Eliminar' (light blue), and 'Volver' (grey). The main content area has a light blue background and contains a form with two labels: 'Idioma' above a dropdown menu showing 'Inglés', and 'Nuevo idioma' above a text input field containing 'English'. Below these fields is a yellow 'Modificar' button.

10 Ventana de Idiomas. Ventana para modificar editoriales.

#### 4.3.6 VENTANA DE USUARIOS

En esta ventana se pueden registrar, modificar y eliminar usuarios.

**Usuarios**

**Registrar**

**Modificar**

**Eliminar**

**Volver**

**DNI**  
16857384V

**Fecha nacimiento** (d/m/aaaa)  
16/1/2004

**Nombre**  
David

**Apellidos**  
Beamonde Salinas

**Correo electrónico**  
davidbeamonde2004@gmail.com

**Teléfono**  
619516832

**Contraseña** ☐ Ver  
.....

**Confirmar contraseña**  
.....

**Dirección**  
Calle Aternos 18

**Sexo**  
Hombre

☒ Admin

**Registrar**

11 Ventana de Usuarios. Ventana para registrar usuarios.

#### 4.3.7 VENTANA DE PRÉSTAMOS

En esta ventana se pueden registrar, modificar y eliminar préstamos.

**Préstamos**

**Registrar**

**Modificar**

**Eliminar**

**Volver**

**Usuario**  
Admin Admin

**Libro**  
Rayuela

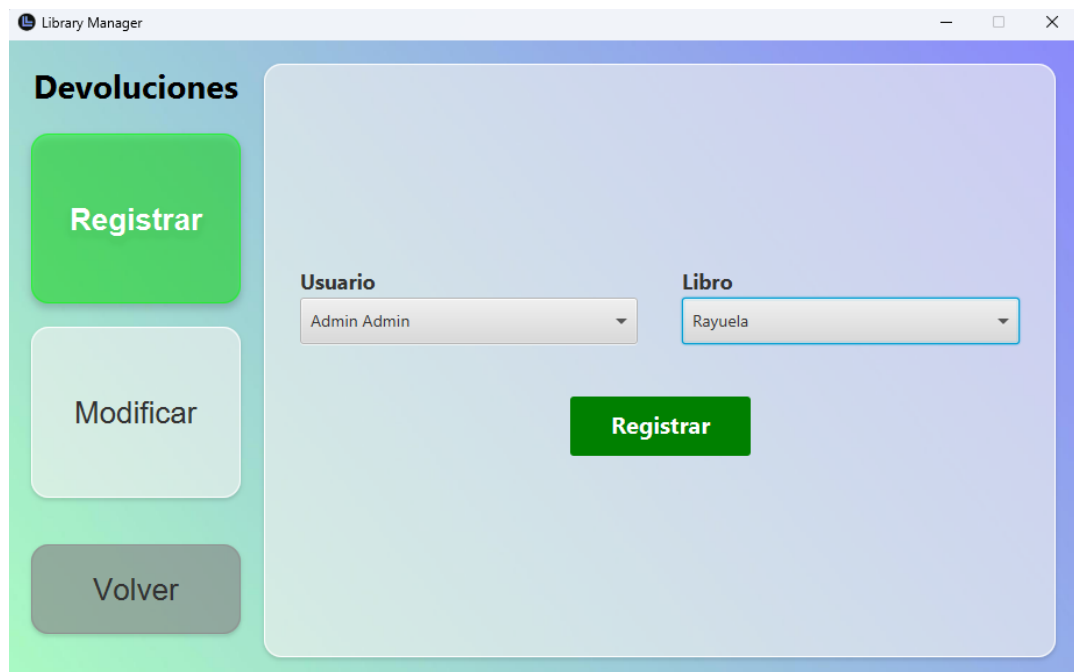
**Dias**  
25

**Modificar**

12 Ventana de Préstamos. Ventana para modificar préstamos.

#### 4.3.8 VENTANA DE DEVOLUCIONES

En esta ventana se pueden registrar y modificar devoluciones.



The screenshot shows a web application window titled "Library Manager". The main heading is "Devoluciones". On the left side, there is a vertical sidebar with three buttons: "Registrar" (green), "Modificar" (light blue), and "Volver" (grey). The main content area has a light blue background and contains two dropdown menus. The first dropdown is labeled "Usuario" and has "Admin Admin" selected. The second dropdown is labeled "Libro" and has "Rayuela" selected. Below these dropdowns, there is a green "Registrar" button.

13 Ventana de Devoluciones. Ventana para registrar devoluciones.

## CAPÍTULO 5: BASE DE DATOS

Para el desarrollo de esta aplicación se ha optado por utilizar **MySQL** como sistema gestor de bases de datos, debido a su ligereza, modularidad, **facilidad de uso** y amplia compatibilidad con entornos de desarrollo en Java. Además, al tratarse de un SGBD relacional, permite una estructuración eficiente y escalable de los datos.

Con el objetivo de facilitar la gestión centralizada de los datos, se ha decidido alojar la base de datos en la nube. Esta decisión permite a la biblioteca, o a una red de bibliotecas, acceder a una única instancia de datos compartida, garantizando la coherencia y disponibilidad de la información desde diferentes ubicaciones físicas.

### 5.1 TABLAS

La base de datos ha sido diseñada siguiendo los principios de la normalización, con el fin de evitar redundancias y asegurar la integridad referencial. Está compuesta por siete tablas principales que se relacionan entre sí mediante claves primarias y foráneas:

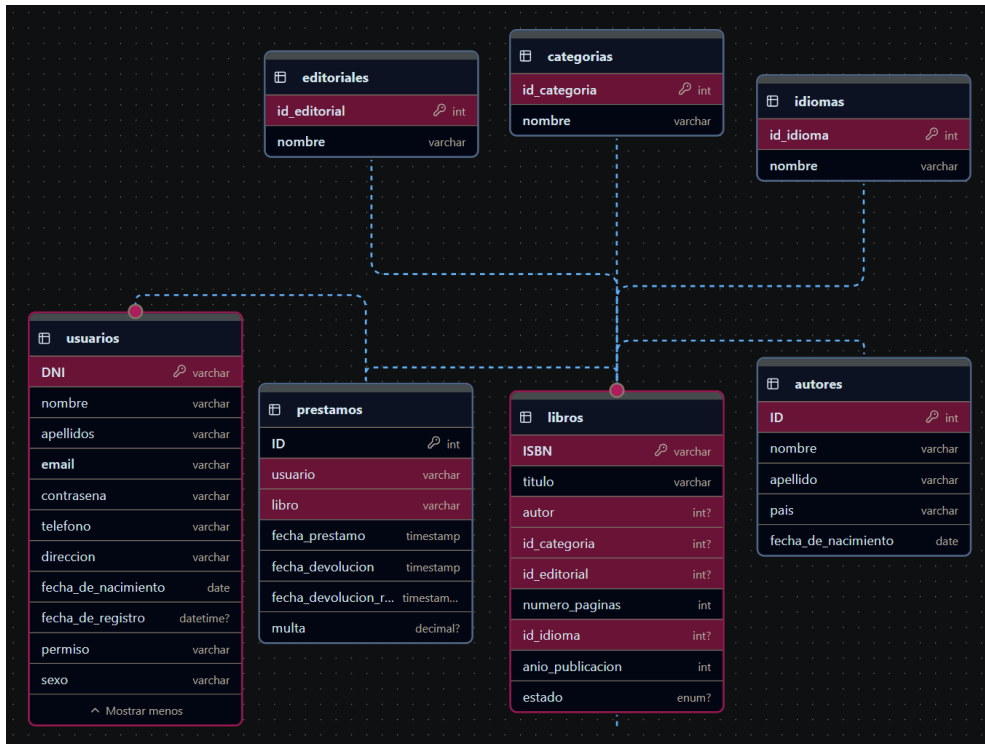
Empezamos con la tabla de **Usuarios**, que es la principal, esta es la que permite iniciar sesión y registrarse a los usuarios y administradores de mi sistema. Esta tabla cuenta con los siguientes campos: DNI, nombre, apellidos, sexo, email, contraseña, teléfono, dirección, fecha de nacimiento, fecha de registro y permiso.

Seguimos con la tabla de **Libros**, que está compuesta por: ISBN, título, autor, categoría, editorial, número de páginas, idioma, año de publicación y estado.

Continuamos con la de **Autores**, que tiene: ID, nombre, apellido, país y fecha de nacimiento.

Después, tenemos tres tablas que cuentan con los mismos datos, estas son la tabla de **Categorías**, **Editoriales** e **Idiomas**, todas compuestas por: ID, nombre.

Finalmente, la tabla **Prestamos**, creada para guardar todos los préstamos y devoluciones, esta tabla tiene estos campos: ID, usuario, libro, fecha de préstamo, fecha de devolución, fecha de devolución real y multa.



14 Diagrama de flujo de la base de datos.

## 5.2 FUNCIONALIDADES PRINCIPALES

El sistema de base de datos permite llevar a cabo una serie de funcionalidades clave para el funcionamiento de la aplicación:

1. Registro, edición y eliminación de libros, autores, categorías, editoriales, idiomas, usuarios y préstamos.
2. Registro y edición de devoluciones.
3. Consultar libros con posibilidad de añadir varios filtros para acortar los resultados a lo buscado.
4. Registro de préstamos y devoluciones, con actualización automática del estado del libro.
5. Filtros y búsquedas por campos clave (título, autor, usuario, estado).
6. Validación de datos y manejo de errores



### **5.3 MODIFICACIÓN DE BASE DE DATOS**

La aplicación cuenta con un archivo llamado "database.properties" que se puede modificar. Permite cambiar los datos de acceso a la base de datos, para permitir al usuario conectarse a una base de datos propia si así lo desea. Permite acceder a bases de datos en la nube y también el local.

## CAPÍTULO 6: REQUISITOS DEL SISTEMA

Para garantizar el correcto funcionamiento de la aplicación, se establecen los siguientes requisitos mínimos y recomendados en términos de hardware, software y conectividad.

### 6.1 REQUISITOS DE HARDWARE

1. **Arquitectura:** Sistema con arquitectura de 64 bits.
2. **Procesador:** Mínimo de 2 núcleos; se recomienda 4 núcleos o más para un rendimiento óptimo.
3. **Memoria RAM:** Al menos 4 GB; se recomienda 8 GB o más para una experiencia fluida.
4. **Almacenamiento:** Espacio disponible de al menos 500 MB para la instalación y funcionamiento de la aplicación.
5. **Resolución de pantalla:** Mínima de 1200x700 píxeles; se recomienda 1920x1080 píxeles para una visualización óptima de la interfaz gráfica.

### 6.2 REQUISITOS DE SOFTWARE

1. Sistema Operativo: Windows 10 o Windows 11, ya sea como sistema operativo principal o ejecutado en una máquina virtual.
2. Java Development Kit (JDK): Versión 22 o superior, correctamente instalada y configurada en el sistema.
3. Herramienta de compresión/descompresión: Software como WinRAR, 7-Zip o equivalente, para la gestión de archivos comprimidos.
4. Navegador web: Cualquier navegador actualizado para acceder a funcionalidades web adicionales, si aplica.

### 6.3 REQUISITOS DE CONECTIVIDAD

Se requiere conexión estable sin restricciones que impidan el acceso a páginas web necesarias para el funcionamiento de la aplicación, incluyendo configuraciones de firewall o proxies que puedan bloquear conexiones salientes.

## CAPÍTULO 7: REALIZACIÓN DE PÁGINA WEB

La [página web](#) se ha construido de forma progresiva, intentando respetar el diseño responsivo para adaptarse lo mejor posible a todo tipo de pantallas. Se ha utilizado HTML, CSS y JavaScript para desarrollar el contenido y navegar por los diferentes apartados de la página.

### 7.1 PÁGINA PRINCIPAL

Comenzando por la página principal, se muestra el logo de la aplicación a modo de presentación. También se muestra una breve explicación sobre el uso de la aplicación. Además, hay botón para poder descargar la aplicación dirigido para aquellos usuarios que directamente les interesa descargar la aplicación sin consultar ningún tipo de información extra. En la parte superior tenemos la navegación de la web, donde podremos ir accediendo a los distintos apartados que la componen.

En la parte inferior se pueden ver algunas de las herramientas utilizadas para este proyecto, las cuales son: Java, MySQL, Clever Cloud, Maven y JavaFX.

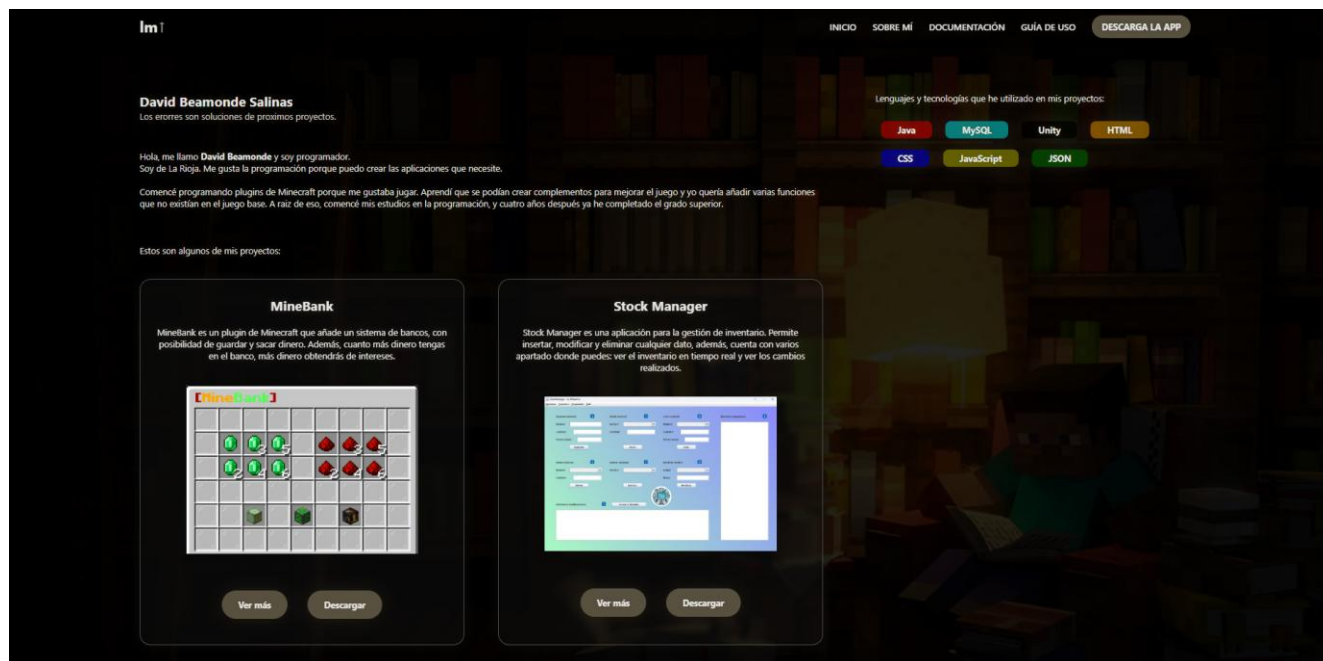


15 Página principal de la página web.

## 7.2 SOBRE MÍ

En este apartado he aprovechado para colocar información sobre mí y sobre otros proyectos que he realizado, además de los lenguajes y las tecnologías que he utilizado para crearlos.

A mí el proyecto que más me gusta es el plugin de Minecraft, a día de hoy, todavía lo estoy desarrollando. Me gusta mucho porque es con lo que empecé en el mundo de la programación.



16 Información personal y proyectos realizados.

## 7.3 DOCUMENTACIÓN

En este apartado se ve toda la documentación del proyecto, incluyendo este documento. También se puede ver una página para reportar posibles errores de la aplicación, para solucionarlos en futuras versiones.

## 7.4 GUÍA DE USO

En este apartado se puede ver varios apartados donde se explica (usando imágenes y textos) cómo usar la aplicación de forma correcta. En la web, hay un apartado por cada apartado que hay en la aplicación.

## **CAPÍTULO 8: DISCUSIÓN**

### **8.1 VENTAJAS**

La aplicación Library Manager ofrece diversas ventajas que optimizan la gestión bibliotecaria. Permite llevar un control detallado de los préstamos y devoluciones de cada libro, facilitando la administración y reduciendo la posibilidad de duplicación o pérdida de información.

Una de las características destacadas de la aplicación es su flexibilidad en la conexión a bases de datos, ya que ofrece la posibilidad de utilizar bases de datos tanto en la nube como locales, adaptándose a diferentes necesidades y entornos de uso. La interfaz ha sido diseñada para ser clara y fácil de usar, lo que mejora la experiencia del usuario y reduce la curva de aprendizaje.

Por último, la estructura relacional de la base de datos permite futuras ampliaciones y adaptaciones según las necesidades específicas de cada entorno.

### **8.2 LIMITACIONES**

A pesar de sus ventajas, la aplicación presenta algunas limitaciones que podrían ser abordadas en futuras mejoras. Actualmente, no permite modificar los nombres de las tablas ni los campos existentes, lo que limita la personalización de la base de datos según necesidades específicas. Además, está diseñada principalmente para funcionar en entornos Windows, lo que podría presentar inconvenientes para bibliotecas que operan con sistemas Linux u otros.

El uso de bases de datos alojadas en la nube requiere una conexión a internet estable, lo que podría ser una limitación en entornos con conectividad limitada.

### **8.3 LÍNEAS DE TRABAJO FUTURO**

Tras la finalización del proyecto "Library Manager", se han identificado varias áreas de mejora y expansión que permitirán evolucionar tanto la aplicación como su página web.

#### **8.3.1 MEJORAS PARA LA APLICACIÓN**

Una mejora significativa sería la implementación de un sistema de alertas automatizadas que notifique a los usuarios sobre las fechas próximas de devolución de libros. Estas notificaciones podrían enviarse por correo electrónico o a través de la propia aplicación, recordando a los usuarios sus compromisos y ayudando a reducir los retrasos en las devoluciones.

Además, se podría añadir un sistema de gestión de multas para aquellos usuarios que realicen devoluciones tardías, o también se podría añadir un ranking de los libros más leídos, o con un nuevo sistema de reseñas que permita a los usuarios dejar su opinión sobre un libro para que el resto de los usuarios puedan consultar, con lo que se fomentaría la lectura.

Otra posible línea de trabajo futuro sería añadir la generación de reportes periódicos (diarios, semanales y mensuales) tanto para los usuarios como administradores. De esta forma, los usuarios podrían ver sus estadísticas, como por ejemplo el número de libros leídos, y los administradores a su vez podrían ver estadísticas referidas al número de usuarios que han reservado libros, número de retrasos en las devoluciones, etc.

Otra línea de trabajo es la expansión de la compatibilidad multiplataforma, permitiendo que la aplicación funcione no solo en Windows, sino también en sistemas operativos como Linux y macOS, o para dispositivos móviles. Esto ampliará el alcance de la herramienta y la hará más accesible para diferentes entornos bibliotecarios.

Y, por último, se me ocurre que podría trabajar en que los usuarios puedan realizar reservas de libros que todavía no han sido publicados, como podemos ver en otros ámbitos como los videojuegos.

### **8.3.2 IMPLEMENTACIONES PARA LA PÁGINA WEB**

Para poder trabajar con la base de datos en la página web se necesita un API al que pedir información para mostrarla en la web, usar lenguaje PHP para poder conectarse con bases de datos, crear un servidor HTTP para enviar y recibir información o montar un servidor con NodeJS como ya lo había intentado anteriormente.

A la web se podrían añadir las siguientes funciones:

1. Añadir un apartado para poder consultar los libros.
2. Ver el historial de préstamos.
3. Modificar los datos de usuario.

## **CAPÍTULO 9: CONCLUSIÓN**

El desarrollo de este proyecto ha supuesto mucho más que un reto técnico, también ha sido un valioso ejercicio de crecimiento personal y profesional. A lo largo del proceso, he fortalecido habilidades fundamentales como la concentración, la comprensión de problemas complejos y la gestión del tiempo. Este aprendizaje ha resultado clave para completar el proyecto de manera satisfactoria, pese a las dificultades propias de un trabajo de esta magnitud.

En cuanto a la parte técnica, el proyecto me ha permitido consolidar conocimientos previos adquiridos durante mi formación, y adquirir otros nuevos, especialmente relacionados con el diseño de interfaces gráficas, el manejo de bases de datos y el desarrollo de aplicaciones usando tecnologías actuales. Esta experiencia que he adquirido me ayudará a afrontar futuros proyectos de desarrollo de software por iniciativa propia.

## CAPÍTULO 10: BIBLIOGRAFÍA

No Autor (20 de Mayo de 2025). ChartDB. Database schema diagrams visualizer: <https://chartdb.io/>

Oracle. API de Java: Java™ Platform, Standard Edition 8 API Specification: [Overview \(Java Platform SE 8\)](#)

Refsnes Data. W3Schools. Online Web Tutorials: <https://www.w3schools.com/>

Quentin Adam. Clever Cloud. Database storage web service: <https://www.clever-cloud.com/>

Microsoft. Github. Code storage platform service: <https://github.com/>

Matt Biilmann. Netlify. Web hosting service: <https://www.netlify.com/>

Reddit. Foro de preguntas: <https://www.reddit.com/>

Prosus. Stack Overflow. Como poner negrita en JavaFX: <https://stackoverflow.com/questions/12341672/make-portion-of-a-text-bold-in-a-javafx-label-or-text>

Oracle (Septiembre 2013). Aprende con JavaFX: FXML para crear interfaz de usuario: [https://docs.oracle.com/javafx/2/get\\_started/fxml\\_tutorial.htm](https://docs.oracle.com/javafx/2/get_started/fxml_tutorial.htm)

Oracle (21 de Junio de 2012). Oracle: [Introducción a FXML](#)

Hermann DB (10 de Octubre de 2023). Stack Overflow. Como cambiar el color de fondo de un boton en JafaFX: <https://es.stackoverflow.com/questions/605529/c%C3%B3mo-cambiar-el-color-de-un-button-en-javafx-al-momento-de-hacer-hover>

No Autor (17 de Abril de 2025). Geeksforgeeks: [Establecer conexión JDBC en Java](#)

MySQL. Obtención de valor AutoIncremental de columnas usando JDBC: <https://dev.mysql.com/doc/connector-j/en/connector-j-usagenotes-last-insert-id.html>

Rafael Neto (15 de Marzo de 2023). Rafael Neto Blog: Como centrar un div en html: <https://rafaelneto.dev/blog/como-centrar-div/>

No Autor. Koomori: Como poner negrita en un texto de html: <https://koomori.com/como-poner-negrita-cursiva-y-subrayado-en-html/>

Ojaramillo (14 de Marzo de 2021). Platzi: Crear archivo css en aplicación con JavaFX: <https://platzi.com/tutoriales/1631-java-basico/9242-javafx-css-aplicar-estilos-css-en-javafx/>

Manz. LenguajesCSS: Como poner borde con color y redondeado a un div en html: <https://lenguajecss.com/css/bordes/border/>



AQMR (5 de Agosto de 2021). Stack Overflow: Alinear un botón centrado abajo en JavaFX: <https://es.stackoverflow.com/questions/475134/alinear-un-bot%C3%B3n-en-el-bottom-center-en-javafx>

Antonio (11 de Noviembre de 2018). Stack Overflow: Añadir elementos a TableView desde una lista en JavaFX: <https://es.stackoverflow.com/questions/212713/agregar-elementos-a-tableview-desde-una-lista-en-javafx>

DanielDev33 (1 de Febrero de 2022). Stack Overflow: Como poner un archivo FXML dentro de otro en la interfaz de JafaFX: <https://es.stackoverflow.com/questions/512297/responsive-javafx-fxml-dentro-de-otro-fxml-adaptar-pane-interno-seg%C3%BAn-se-actual>

Ramón Maldonado (14 de Enero de 2025). KeepCoding: Como poner objetos verticalmente en JavaFX con FXML: <https://keepcoding.io/blog/que-es-y-como-usar-javafx-scene-layout-vbox/>

Carolina Ochoa (16 de Junio de 2017). Stack Overflow: Como poner una imagen de fondo en una ventana de JavaFX: <https://es.stackoverflow.com/questions/79069/como-agrego-una-imagen-de-fondo-a-mi-scene-en-javafx>

Roberto Palmar (26 de Marzo de 2021). Stack Overflow: Como mostrar y ocultar ventanas dentro de una ventana de JavaFX: <https://es.stackoverflow.com/questions/438546/como-crear-varios-paneles-en-uno-javafx>

Jhonny Luis (21 de Octubre de 2017), Stack Overflow: Como cargar datos a un ComboBox en JavaFX: <https://es.stackoverflow.com/questions/111168/como-mostrar-un-valor-de-un-combobox-en-javafx-pasando-un-valor>