

## Proyecto Intermodular: Aplicación en JAVA

Sandra Biviana Vega Dussán

Desarrollo de aplicaciones web

Proyecto Intermodular

Ciclo Formativo a distancia 1er año 2024 – 2025

<b>Introducción</b>	<b>3</b>
<hr/>	
Introducción	4
Objetivos	5
<b>Temporalización</b>	<b>6</b>
<hr/>	
Diagrama de Gantt	6
<b>Tecnologías usadas</b>	<b>7</b>
<hr/>	
Tecnologías usadas	7
<b>Descripción del problema</b>	<b>9</b>
<hr/>	
Descripción del problema	9
<b>Análisis</b>	<b>11</b>
<hr/>	
Diagrama de Entidad / Relación	11
Diagrama de clases	12
Diagrama de caso de uso	13
<b>Implementación</b>	<b>14</b>
<hr/>	
Implementación	14

<b>Futuras implementaciones y conclusiones</b>	<b>16</b>
<hr/>	
Futuras implementaciones	16
Conclusiones	16
<b>Bibliografía</b>	<b>17</b>
<hr/>	
Bibliografía	17

## Introducción

Este proyecto implica el desarrollo de una aplicación web en Java, estructurada sobre el modelo CRUD (Create, Read, Update and, Delete), que facilita la gestión de un catálogo de canciones. La aplicación no solo permite administrar una lista de canciones, sino que además incorpora la funcionalidad de registrar y autenticar usuarios. Brindando así una experiencia tanto personalizada como segura para cada usuario que ingrese al sistema.

La elección de este tema proviene de mi experiencia como productora musical y la necesidad de disponer de una herramienta sencilla para organizar y gestionar colecciones de música. Esta actividad es común para usuarios individuales, así como para expertos en música, bibliotecas digitales o programadores que buscan incluir bases de datos de canciones en sus aplicaciones.

A través de este proyecto, no solo pretendo consolidar los conocimientos técnicos adquiridos durante el curso, sino que, al mismo tiempo deseo promover buenas prácticas de desarrollo en situaciones reales, mostrando la fusión de diversas tecnologías para satisfacer necesidades específicas.

## Objetivos

### Objetivo general:

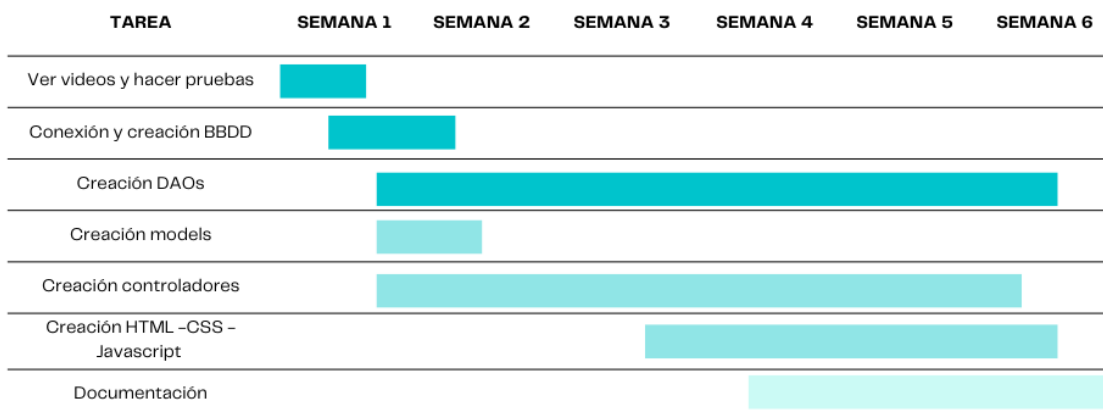
Desarrollar una aplicación web CRUD utilizando Java, que permita la gestión de un inventario de canciones, incluyendo funciones para agregar, modificar y eliminar canciones, así como registrar usuarios y su autenticación a través de un el sistema de inicio de sesión seguro.

### Objetivos específicos:

- Diseñar y llevar a cabo una base de datos relacional que almacene de forma efectiva los datos de canciones y usuarios.
- Desarrollar una interfaz web que sea sencilla y fácil de usar para mejorar la experiencia del usuario con la aplicación.
- Implementar el modelo Singleton para la conexión con la base de datos y utilizar el modelo MVC para asegurar que el código sea bien estructurado y fácil de mantener.
- Incorporar un sistema de autenticación de usuarios que proteja los datos.
- Aplicar buenas prácticas en el desarrollo web, que incluyan la validación de datos, el manejo de errores y la optimización de consultas SQL.

## Diagrama Gantt

# PROYECTO INTERMODULAR



## Tecnologías

Para llevar a cabo el desarrollo de la aplicación web destinada a la gestión de un inventario musical, se ha decidido adoptar un enfoque de trabajo que consiste en varias fases: planificación, desarrollo, diseño, pruebas y documentación. Cada una de estas fases ha estado respaldada por herramientas y entornos de desarrollo específicos, lo que ha facilitado tanto la organización como la implementación de las distintas funcionalidades.

### Herramientas y tecnologías empleadas:

- **IntelliJ IDEA:**

Este es el entorno de desarrollo principal para la escritura y depuración del código en Java y JSP. Se utilizó a lo largo de todas las fases de programación y de pruebas.

- **Visual Studio Code:**

Este IDE se utilizó principalmente para editar archivos estáticos, tales como HTML, CSS y JavaScript. Se aplicó en las tareas de maquetación y en el desarrollo de la parte visual de la aplicación.

- **Java + Jakarta EE:**

Este es el lenguaje de programación principal y el estándar para el desarrollo de la aplicación web. Su uso se concentró en la fase de desarrollo del servidor, así como en los controladores y servicios.

- **MySQL Workbench:**

Se empleó para el diseño, la creación y la gestión de la base de datos relacional. Fue fundamental durante las etapas de diseño y pruebas de la base de datos.

- **Apache Tomcat:**

Este servidor de aplicaciones que actúa como un puente entre el cliente y el servidor. Se implementó durante las pruebas locales y para el despliegue de la aplicación.

- **Maven:**

Esta herramienta de automatización de compilación y gestión de dependencias. Permitió organizar el proyecto y facilitó la inclusión de las librerías.

- **HTML, CSS y JavaScript:**

Estas son las tecnologías estándar para desarrollar interfaces web dinámicas y responsivas. Responsables de la presentación y la experiencia del usuario dentro de la aplicación.



## Descripción del problema

La propuesta de esta aplicación radica en la necesidad de contar con una herramienta centralizada, simple y eficiente para gestionar de manera segura y personalizada los registros de canciones. Aunque hay diversas plataformas comerciales que ofrecen gestión musical, su enfoque está más dirigido al consumo y reproducción de música en lugar de a la organización personal de catálogos y colecciones privadas.

El objetivo de la aplicación es proporcionar una solución sencilla que permita a los usuarios crear, editar y eliminar registros de canciones, así como gestionar sus propios usuarios y salvaguardar la información a través de autenticación.

## Análisis del mercado y competencias:

Entre las aplicaciones disponible actualmente en el mercado que cumplen funciones similares se encuentran:

- Spotify y YouTube Music: estas aplicaciones permiten la gestión de listas de reproducción, pero están limitadas a catálogos determinados y no ofrecen opciones para personalizar la información de las canciones o guardar registros propios.
- iTunes / MusicBee / Clementine: estas plataformas permiten organizar colecciones locales, pero no disponen de funciones accesibles desde cualquier dispositivo en línea y su enfoque principal es la reproducción más que la gestión ordenada.
- Bases de datos especializadas (Discogs, MusicBrainz): proporcionan catálogos detallados, pero no permiten una gestión personal con CRUD libre y login de usuarios.

## Análisis DAFO:

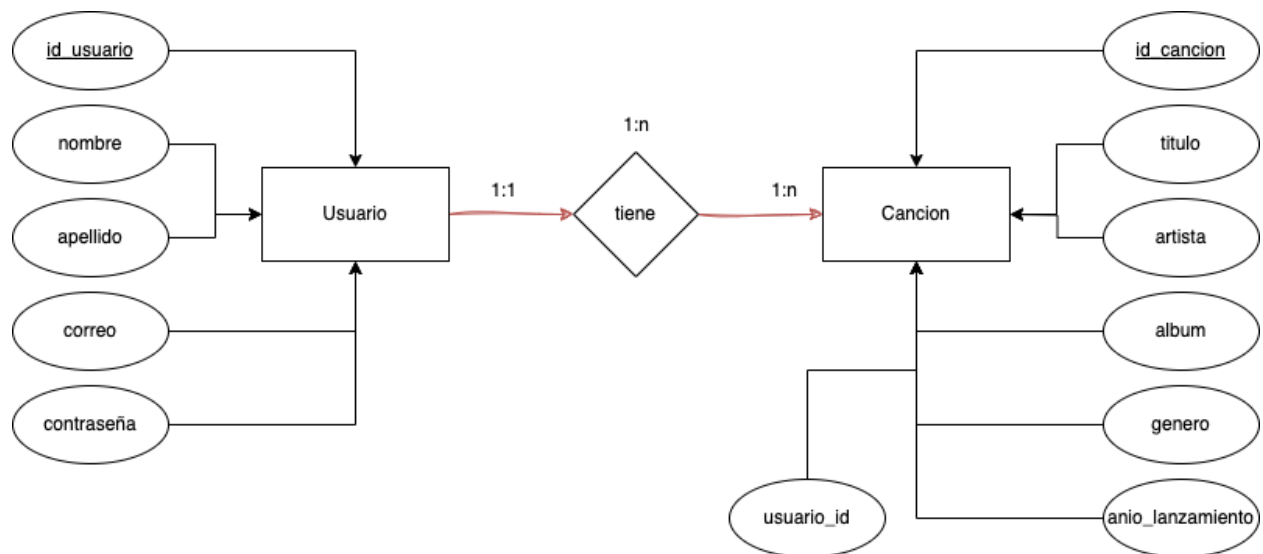
- **Fortalezas:**
  - Uso de tecnologías comunes y escalables (Java, MySQL)

- Código estructurado y fácil de mantener gracias al modelo MVC.
- Posibilidad de futuras ampliaciones y personalización.
- **Debilidades:**
  - Funcionalidad visual limitada en comparación con aplicaciones comerciales.
  - Aplicación básica centrada solamente en la gestión de datos y acceso de usuarios.
  - Es necesario contar con conocimientos técnicos para su despliegue.
- **Oportunidades:**
  - Posibilidad de evolucionar hacia una aplicación más compleja que incluyan funciones de reproducción de música o integración de APIs
  - Oportunidad de ser utilizada en contextos educativos o de archivo.
- **Amenazas:**
  - Fuerte competencia con plataformas ya establecidas.
  - Necesidad continua de renovación debido a los avances tecnológicos.

## Análisis

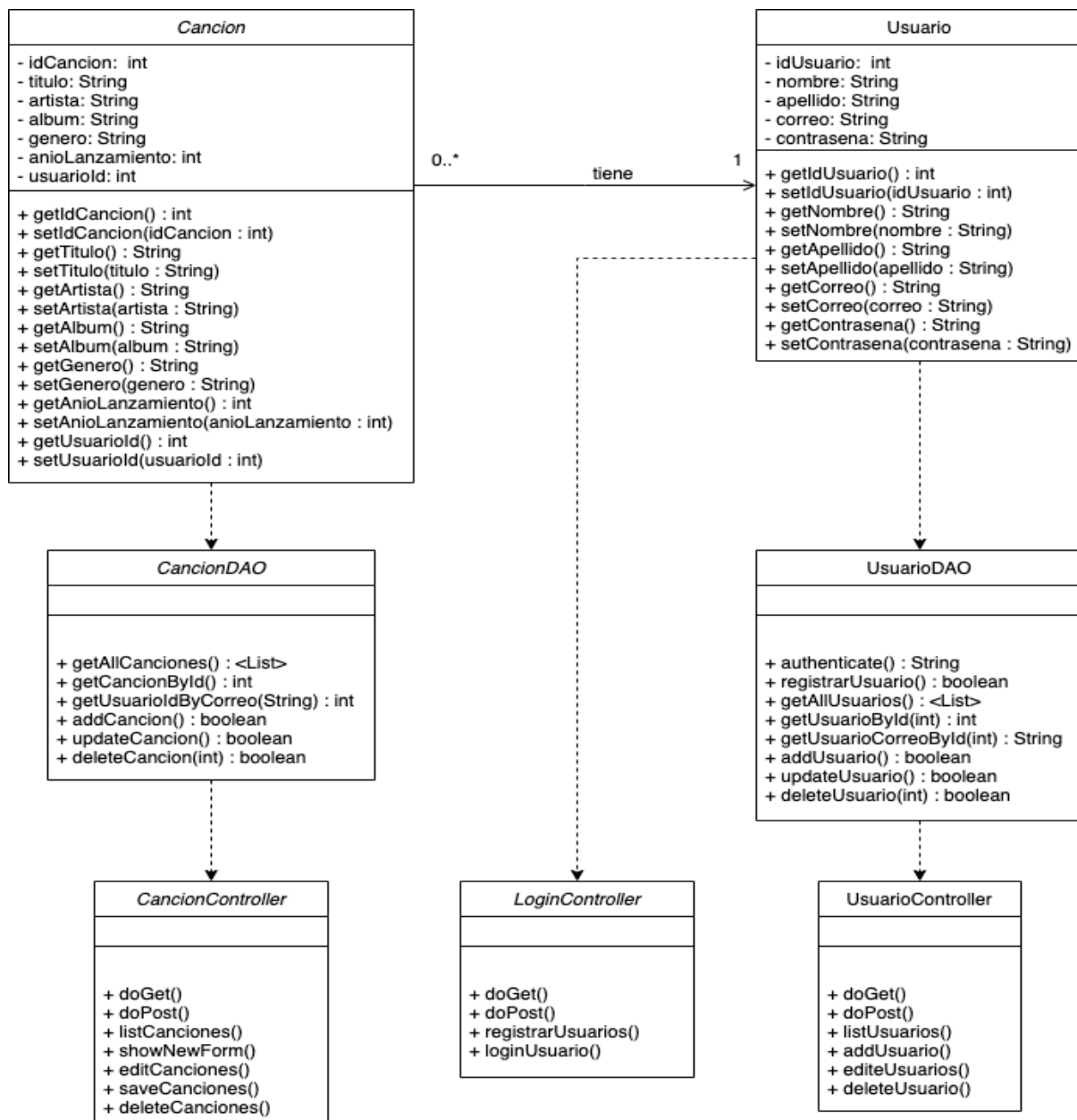
### Diagrama de entidad/relación:

Detalla la configuración de la base de datos, las entidades clave (usuarios, canciones) y sus conexiones, facilitando la comprensión de la disposición lógica de los datos.



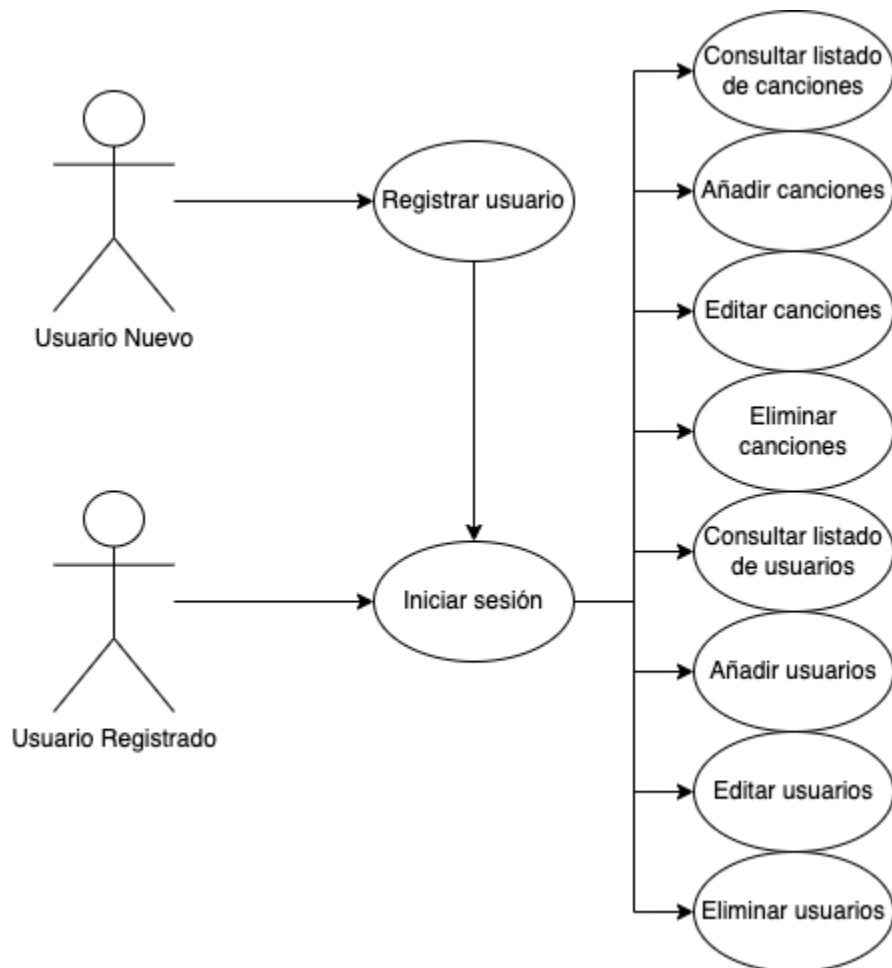
## Diagrama de clases:

Muestra las clases fundamentales de la aplicación, así como sus atributos, métodos y las relaciones entre ellas. Ayuda a comprender la estructura de la programación orientada a objetos utilizada en el proyecto.



## Diagrama de casos de uso:

Facilita la identificación precisa de las acciones que un usuario puede llevar a cabo con la aplicación, como registrarse, iniciar sesión, agregar canciones, editar canciones o eliminar canciones.



## Implementación

El desarrollo de la aplicación se llevó a cabo en varias etapas, cada una reflejando un progreso en el desarrollo de las funcionalidades esenciales:

### 1. Diseño de la Base de Datos:

Se establecieron las tablas, junto con sus respectivos datos y sus relaciones utilizando la herramienta de MySQL Workbench, asegurando así su integridad.

### 2. Configuración del entorno de desarrollo:

Se ejecutó la instalación y configuración del servidor Apache Tomcat y la librería de gson, para garantizar un proceso de trabajo coherente y efectivo.

### 3. Implementación del Backend:

Se estableció la conexión con la Base de Datos y posteriormente se desarrollaron las clases Java requeridas para gestionar la lógica del inventario y las operaciones CRUD relacionadas con canciones y usuarios; aplicando el método Singleton y el modelo MVC.

### 4. Diseño del Frontend:

Se elaboraron las visualizaciones a través de JSP, HTML CSS y JavaScript, priorizando la simplicidad y la facilidad de uso de la aplicación.

### 5. Sistema de autenticación y de inicio de sesión:

Se implementó la funcionalidad de inicio de sesión para los usuarios, con un proceso de validación donde se garantiza que únicamente los usuarios autenticados puedan acceder al sistema.

### 6. Prueba y ajustes finales:

Se llevaron a cabo pruebas locales y funcionales para verificar la estabilidad del sistema, corrigiendo fallos y puliendo aspectos de usabilidad.

Durante la fase de implementación, se presentaron varios problemas, tales como dificultades en la conexión a la Base de Datos, así como desincronización de las dependencias de Maven y la falta de datos en las vistas. Cada desafío fue superado mediante la consulta de la documentación

oficial, depuración, preguntas al profesor o a foros especializados; lo que fortaleció tanto mis capacidades técnicas como mi paciencia.

## Futuras implementaciones y conclusiones

### Futuras implementaciones:

- ★ Incluir funciones avanzadas de búsqueda por autor, género o año de publicación.
- ★ Establecer diferentes roles de usuario para distinguir entre admiradores y usuarios generales.
- ★ Desarrollar un sistema para la importación y la exportación de listas de canciones.
- ★ Incorporar paginación y filtros dinámicos para facilitar la gestión de grandes cantidades de información.

### Conclusiones:

La realización de este proyecto ha resultado ser una experiencia muy valiosa tanto a nivel técnico como personal. A lo largo del proceso de desarrollo de la aplicación, he fortalecido y expandido mis competencias en programación, trabajando tanto en la parte del servidor como en el lado del cliente, además de aumentar mi entendimiento sobre cómo funcionan las aplicaciones web.

El uso de bases de datos relacionales y la integración de consultas SQL a través de MySQL me ha permitido profundizar en la optimización de la gestión de datos y en la organización adecuada de la información, que son aspectos esenciales para cualquier aplicación CRUD. Además, la experiencia que obtuve al utilizar las herramientas de este proyecto me ha permitido conocer no solo sus funciones básicas, sino también cómo combinarlas de forma efectiva para crear aplicaciones robustas y funcionales.

A lo largo del proceso, también he aprendido a llevar a cabo la estructuración de proyectos de manera ordenada, planificando tareas de manera lógica y priorizando las necesidades más relevantes en cada etapa de desarrollo. La identificación, resolución de errores y la gestión de imprevistos han sido una parte clave de esta experiencia, dado que cada desafío ha sido una



oportunidad para profundizar en la comprensión de los sistemas y para mejorar mis habilidades de análisis y de resolución de problemas.

Este proyecto también me ha evidenciado la importancia del trabajo constante y la capacidad de adaptarse a nuevas tecnologías y métodos. Aunque he adquirido conocimientos importantes, este proceso ha dejado en claro que siempre hay oportunidad para seguir aprendiendo, evolucionando y perfeccionando tanto mis habilidades técnicas como mis competencias personales, tales como la organización, la autogestión y la perseverancia.

## Bibliografía

*Java Documentation - Get started.* (2023, 31 enero). Oracle Help Center.

<https://docs.oracle.com/en/java/>

Project, A. T. (s. f.). *Apache Tomcat® - Welcome!* <https://tomcat.apache.org/>

*IntelliJ IDEA – the IDE for Pro Java and Kotlin Development.* (2021, 1 junio). JetBrains.

<https://www.jetbrains.com/idea/>

*MDN Web Docs.* (s. f.). MDN Web Docs. <https://developer.mozilla.org/>

Dev, H. C. (2025, 15 abril). *GitKraken Desktop Support Home.* GitKraken Help Center.

<https://help.gitkraken.com/gitkraken-desktop/gitkraken-desktop-home/>