

# DOCUMENTACIÓN BEATFLOW



## 1. Introducción

Descripción Breve del Proyecto:

"BeatFlow" es un reproductor de música inspirado en Spotify, desarrollado en Java por un equipo de cuatro personas como proyecto para el instituto.

Objetivos:

- Proporcionar una experiencia de reproducción de música similar a Spotify.
- Aprender y aplicar tecnologías como JavaFX, Scene Builder, Maven, Git, etc.

## 2. Instrucciones de Uso

Descripción de las principales características del reproductor:

- Reproducción
- Creación de listas de reproducción
- Suscripción
- Búsqueda
- Publicación de Discos

## 2. Configuración y Entorno

Configuración del Entorno de Desarrollo:

Dependencias de Maven:

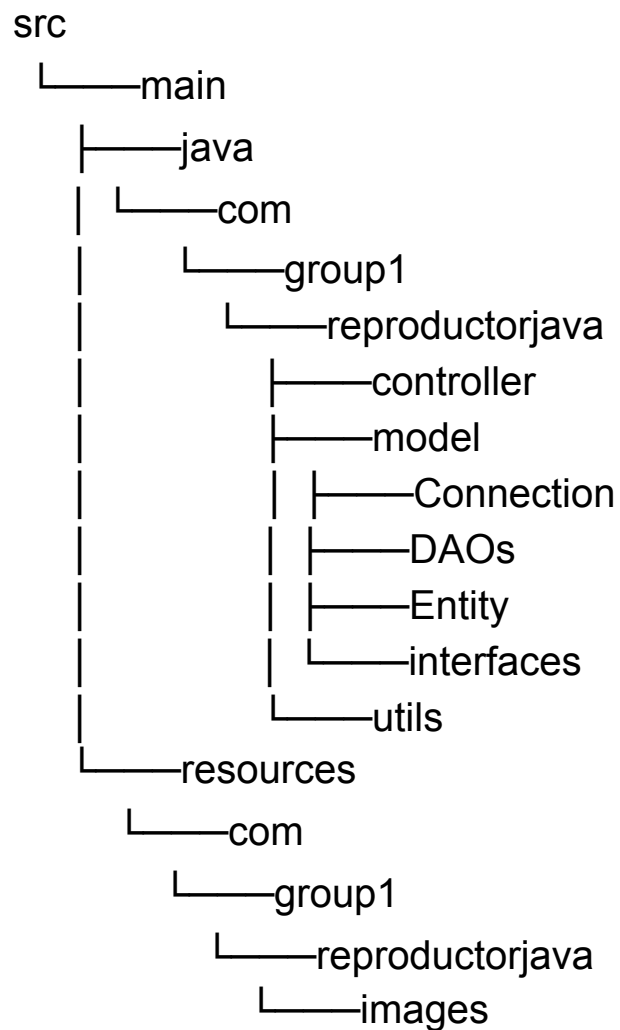
- javafx-fxml (v. 19)
- javafx-controls (v. 19)
- controlsfx (v. 11.1.2)
- junit-jupiter-api (v. latest)
- junit-jupiter-engine (v. latest)
- mariadb-java-client (v. 3.2.0)
- hibernate-core (v. 5.4.24 Final)
- hibernate-entitymanager (v. 5.4.25 Final)
- javax.persistence-api (v. 2.2)
- javafx-media (v. 21)

## 3. Diseño y Arquitectura

### 3.1. Organización de Directorios

Estructura de carpetas y archivos en el proyecto.

A continuación se encuentra una imagen con el diagrama de clases del proyecto donde se puede ver la estructura de directorios.

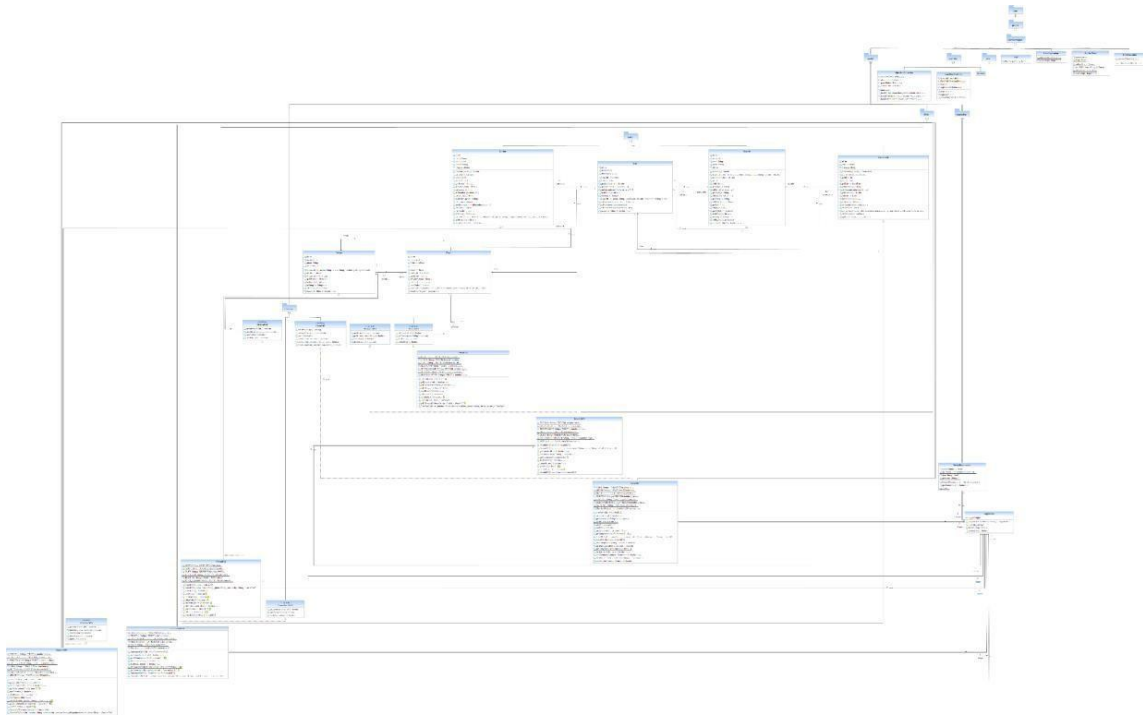


### 3.2. Organización de Directorios

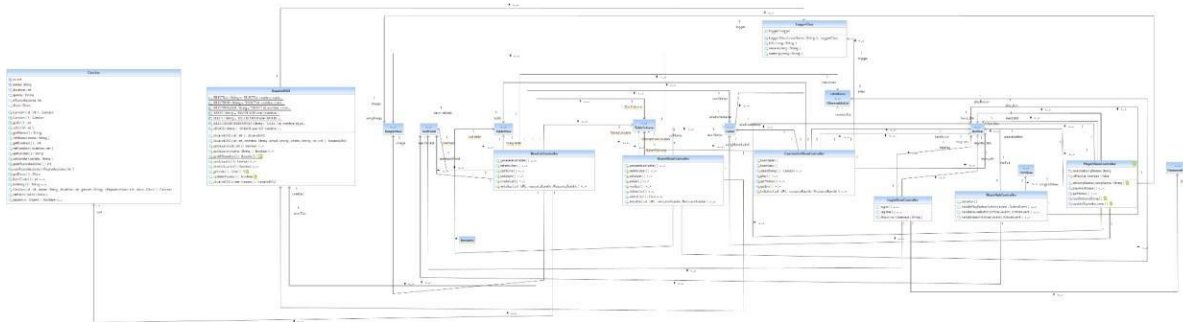
Enlace para verlo más grande y descargarlo:

<https://i.imgur.com/JNRPXfj.png>

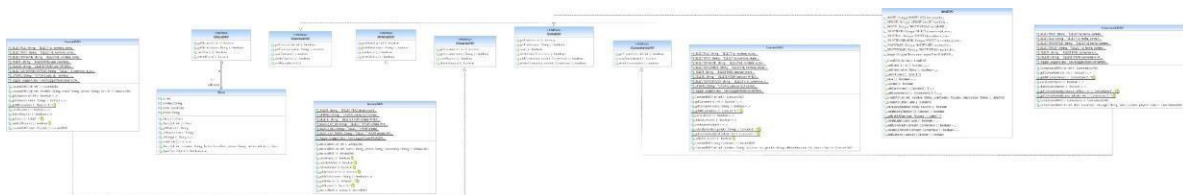
-Completo



-Por partes



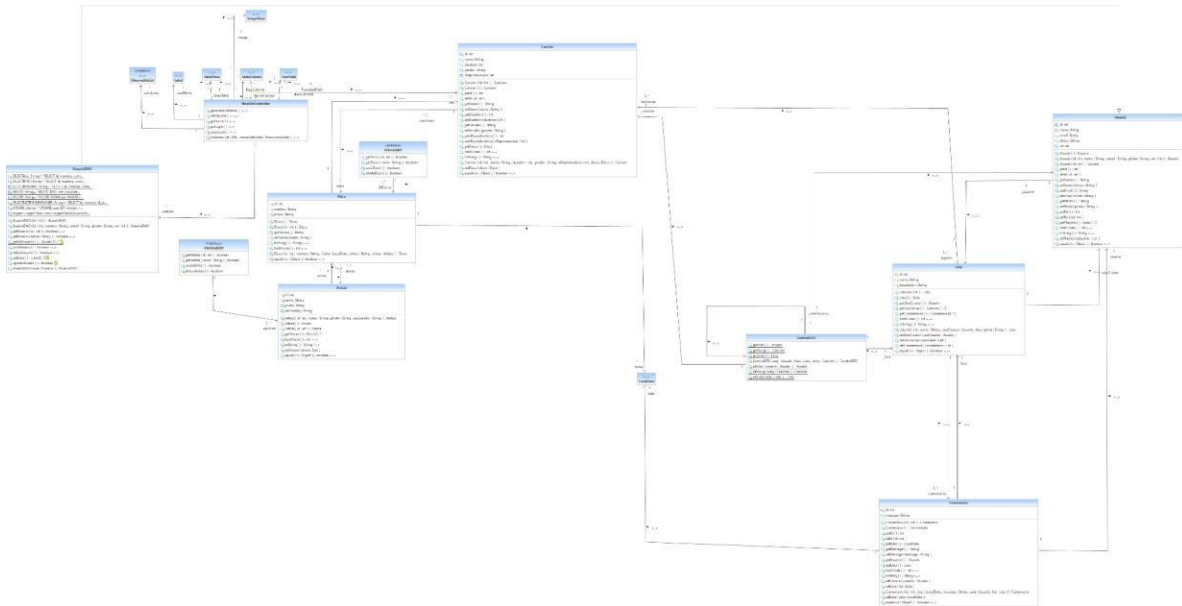
↑ Controladores vistas



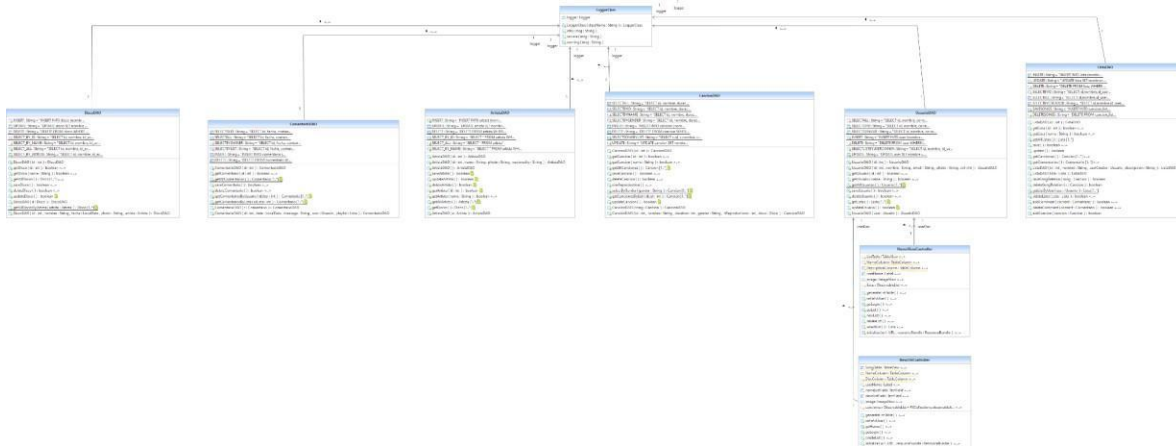
↑ Interfaces



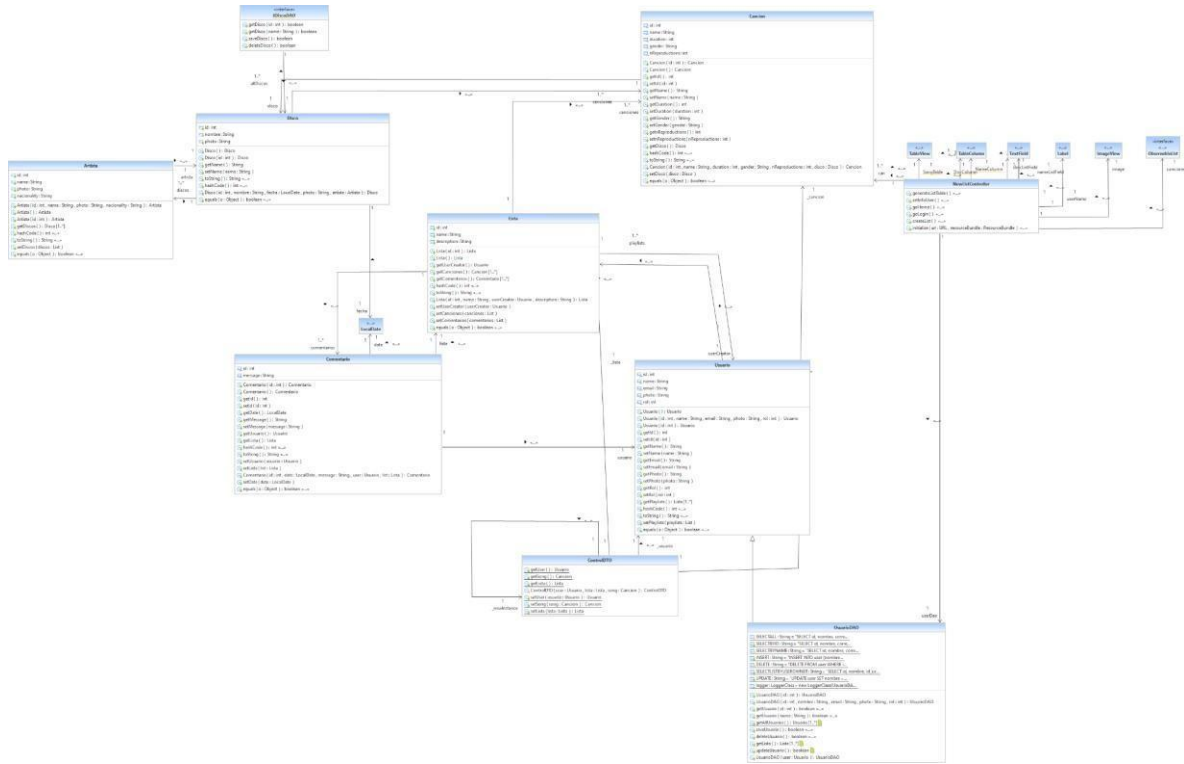
↑ Conexión MDB y log



↑ Entidades



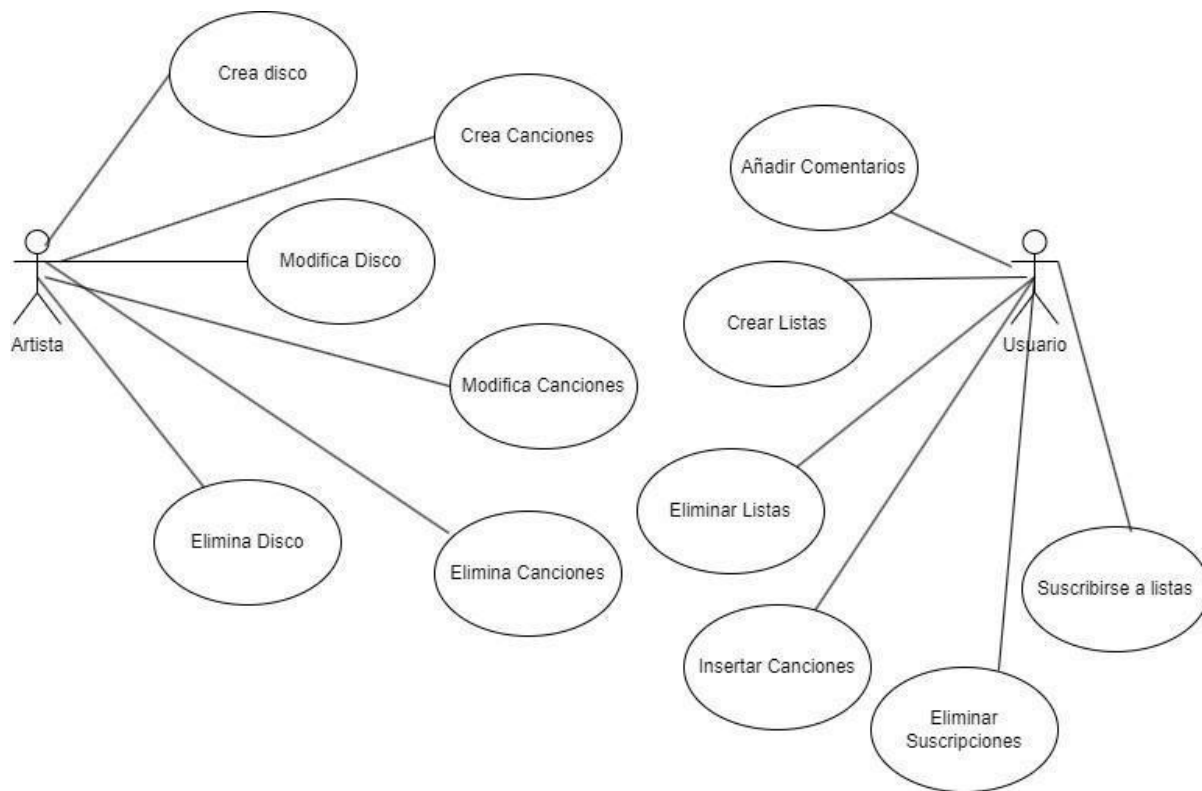
↑ DAOs



↑ Control Principal

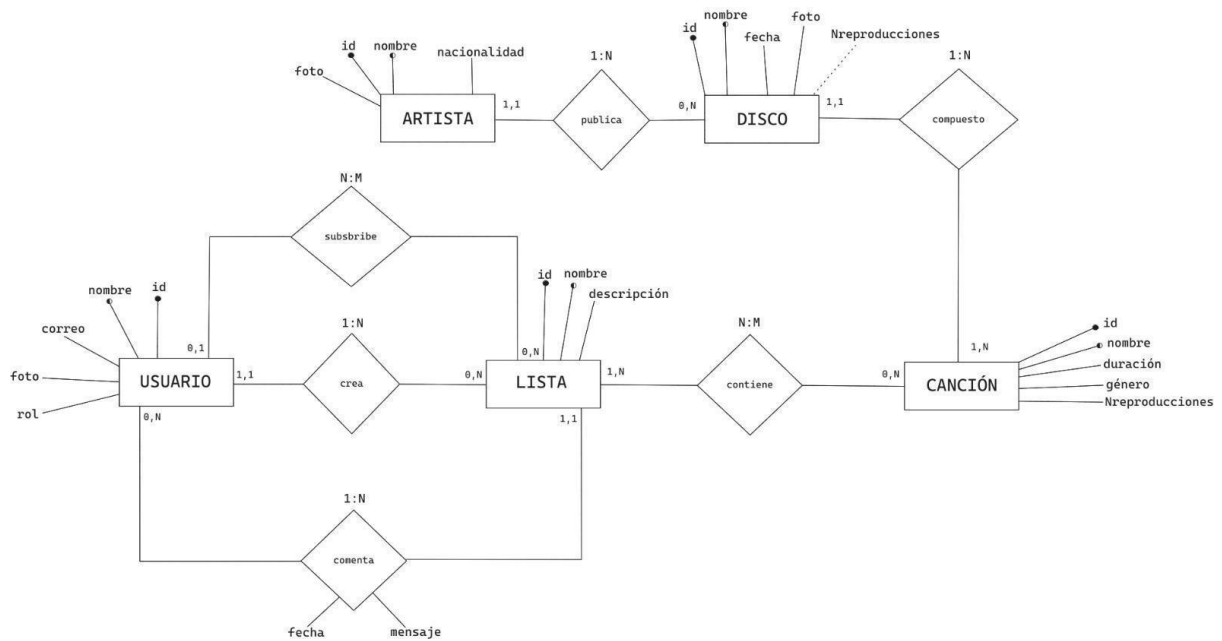


### 3.2. Diagrama de Casos de Uso

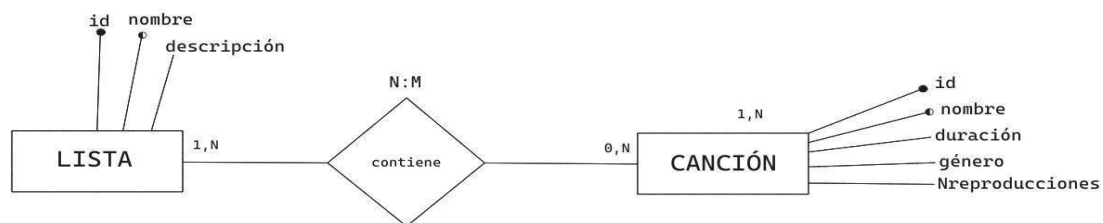


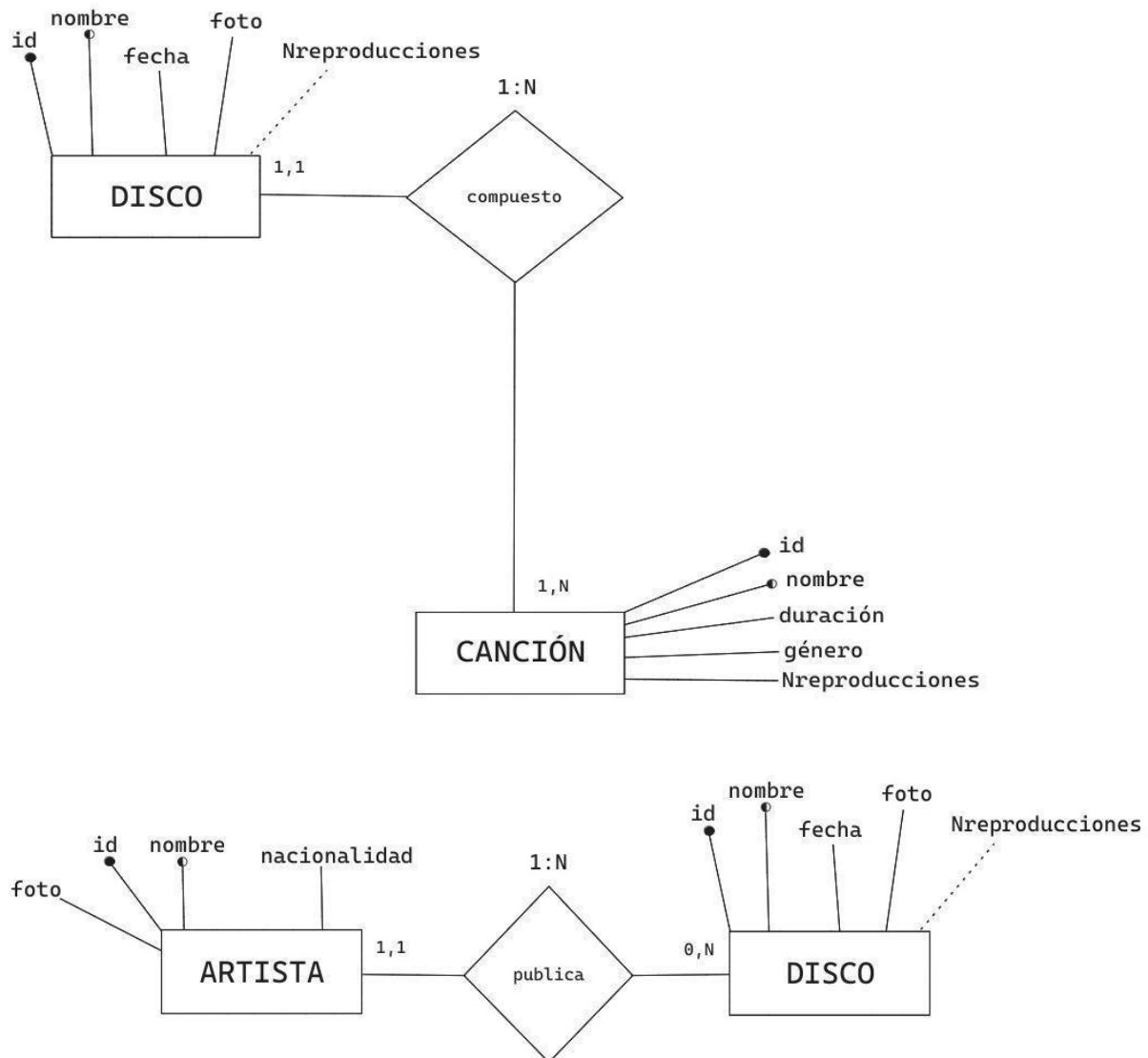
### 3.3. Modelo Entidad Relación

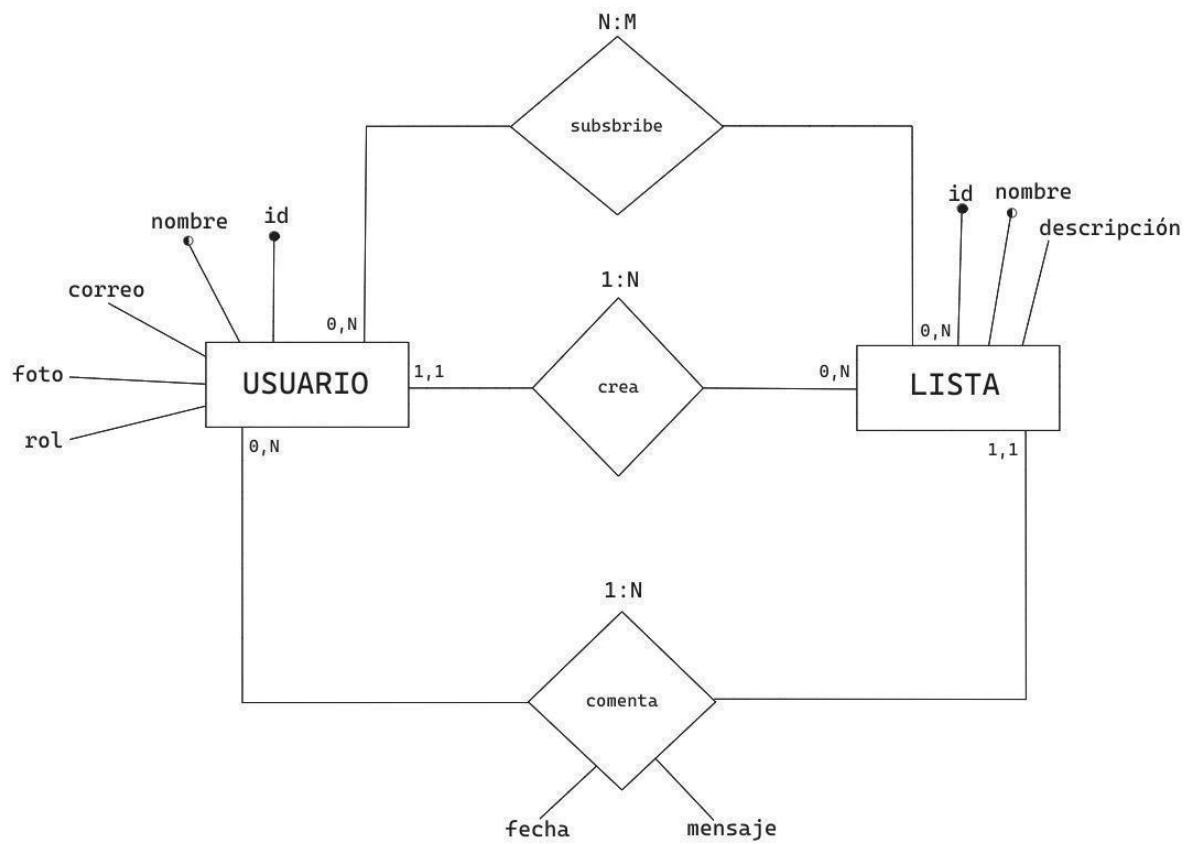
-Completo



-Por partes





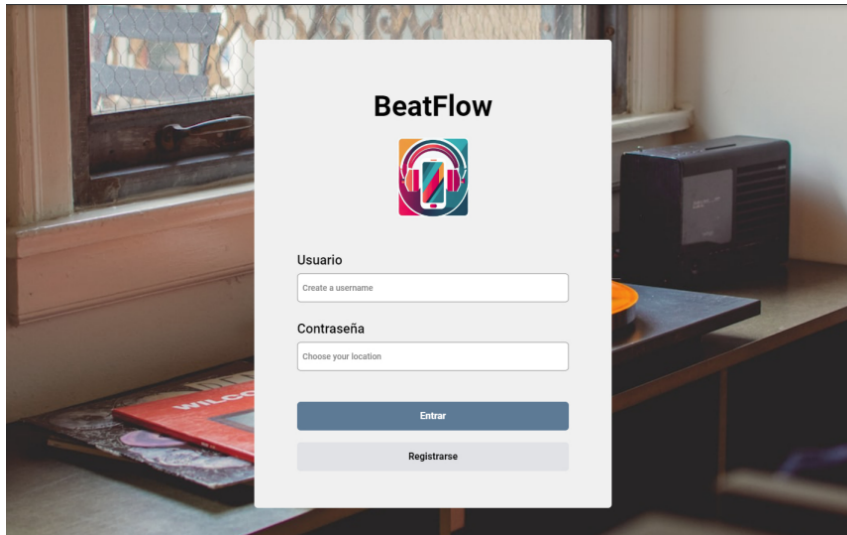


### 3.3. Paso a Tabla

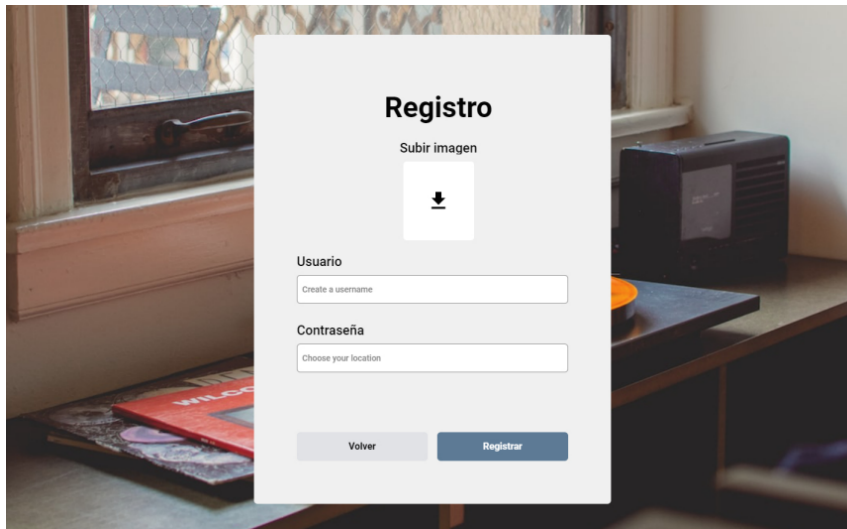


### 3.4 Diagrama de Pantallas

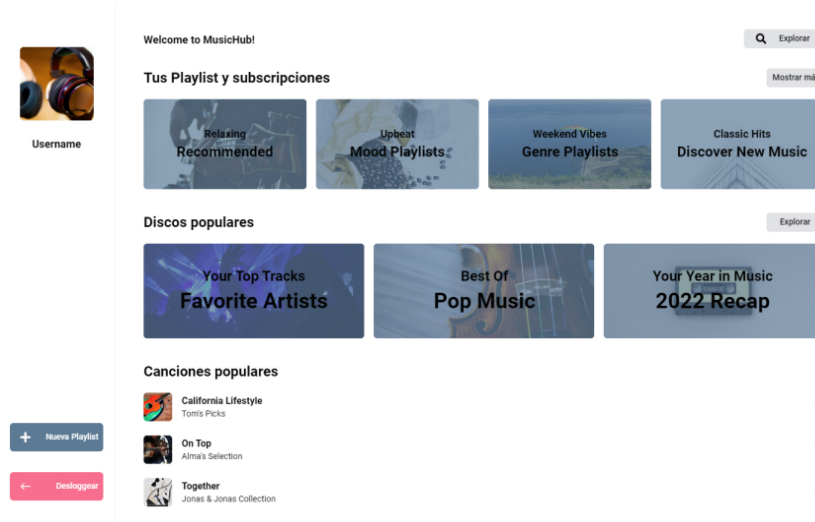
#### - Login



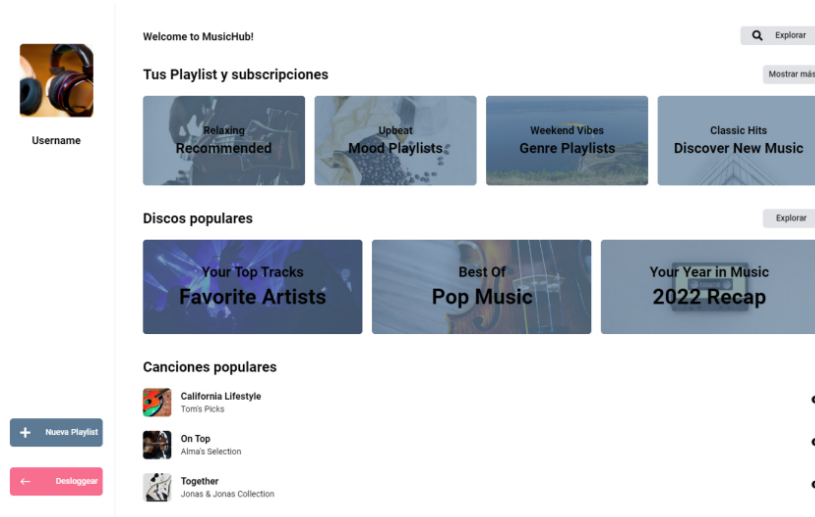
#### - Registro



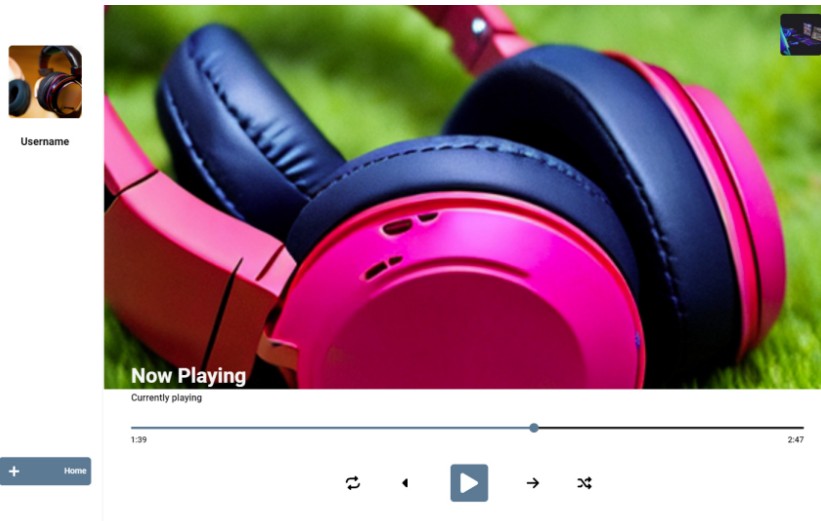
## - Inicio



## - Listas



## - Reproductor



## 4. Pruebas:

Se han realizado exhaustivas pruebas a cada uno de los DAOs para cerciorarse del correcto funcionamiento de las mismas. Gracias a esto obtenemos una sólida estructura sobre la que trabajar en el proyecto.



## **5. Migración a Hibernate**

### **5.1. Implementación de Hibernate:**

Se han reestructurado los DAOs y las clases, para el uso de Hibernate, añadiendo los decoradores necesarios para su correcto funcionamiento y sustituyendo las consultas manuales por las propias de Hibernate.

### **5.2. Actualización de las vistas:**

Se han actualizado el diseño y la lógica de ciertas vistas del proyecto para darle un diseño más complejo.

## 6. Porcentaje de trabajo realizado

-Rubén Castro: 30%

-José Carlos Castro: 30%

-José Javier Galán: 20%

-Ramón Romero: 20%

## 7. Enlaces

Repositorio en GitHub: <https://github.com/najocar/ReproductorHibernate>