Análisis de Frontend

Daniela Andres Valbuena Rodriguez

Simon Esteban Vargas Mora

Katherin Morcillo Quiroga

Servicio Nacional de Aprendizaje

Analisis y Desarrollo de Software

Isaura Maria Suarez Novoa Bogota D.C., 20 de Junio de 2025

# Planteamiento del problema

CASO DE ESTUDIO

Lorenzo es un melómano y un coleccionista de música. Ha pasado toda su vida coleccionando álbumes musicales en discos, casetes y Cds, y tiene tantos, que tiene problemas para recordar en cuál álbum puede encontrar una canción que quiere escuchar.

Para solucionar este problema crea un sistema para almacenar la información de su colección y tener donde buscar en su colección de manera más fácil las canciones que quiere ubicar.

Su amigo vio su solución y le propuso convertirla en un servicio de almacenamiento de información musical en línea.

Con este propósito, Lorenzo ha decidido construir una aplicación web que le permita administrar sus álbumes y canciones. Este sistema debe ser capaz de permitir el registro de otros melómanos con un nombre y una contraseña. Cada usuario podrá tener un listado de álbumes para guardar por cada uno el título, año de producción, descripción, los artistas que interpretaron las canciones, las canciones que contiene y el medio sobre el cual está grabado.

El sistema le debe permitir a un usuario adicionar álbumes en cualquier momento. El sistema también le debe permitir modificar sus datos, e incluso borrar los álbumes si ingresa alguno repetido o que se dañó y ya no se puede escuchar.

Cuando un usuario elige un álbum de la lista, debe ver su información y las canciones que tiene. Para las canciones del álbum, Lorenzo necesita almacenar su título, duración en minutos y segundos y su intérprete. La idea que tiene es que en cualquier momento pueda editar la información o borrar las canciones, e incluso añadir canciones al registro

de canciones para que otros usuarios puedan buscar y añadirlas a sus propios álbumes.

# Requerimientos

Inicio de sesión

1. Crear Usuario
2. Gestionar usuario
3. Recuperar contraseña

Administración de Álbumes

1. Crear álbumes
2. Crear lista de álbumes
3. Eliminar álbumes
4. Editar álbumes
5. Visualizar canciones
6. Gestionar descripción
7. -Añadir titulo
8. -Agregar el año de producción
9. -Añadir la descripción
10. -Agregar artista
11. -Añadir canciones
12. -Añadir medio de grabación

Gestionar de canciones

1. Añadir canciones
2. Editar canciones
3. Eliminar canciones
4. Buscar canción
5. Añadir a álbum
6. Gestionar descripción
7. -Añadir titulo
8. -Agregar el año de producción
9. -Añadir la descripción
10. -Agregar artista
11. -Añadir canciones
12. -Añadir medio de grabación
13. -registro de duración de canción (en minutos/ segundos)

# UX/IU DESIGN

Color primario: #1c3b36 Color secundario: #476762

Boton de alerta: #72938d

Color primario: #1c3b36 (Verde oscuro / Verde bosque)

Psicología: Este tono transmite serenidad, estabilidad y profundidad emocional. Es un color que recuerda a la naturaleza, evocando sensaciones de calma y confianza.

Aplicación en una web de canciones: Ideal como fondo o base principal porque no distrae, permite que los elementos visuales destaquen y genera un entorno donde el usuario se siente cómodo para explorar música con calma y atención.

Color secundario: #476762 (Verde grisáceo / Verde apagado)

Psicología: Este verde más suave y neutral representa armonía, sofisticación y equilibrio. Es menos intenso que el primario, lo cual permite usarlo para resaltar contenido sin romper con la atmósfera de calma.

Aplicación: Perfecto para menús, secciones destacadas o detalles gráficos. Complementa el color principal manteniendo una coherencia visual elegante.

Botón de alerta: #72938d (Verde azulado suave)

Psicología: Aunque no es un rojo típico de alerta, este verde-azulado claro llama la atención de manera sutil y no agresiva. Sugiere una llamada a la acción con confianza y tranquilidad, ideal para una web que busca una experiencia más relajada.

Aplicación: Bueno para botones como "Reproducir ahora", "Descargar canción", o "Agregar a playlist", ya que destaca sin romper el tono general del sitio.

# Nombres de las paginas

Para la pagina, habrá un total de 6 vistas, las cuales son: Inicio de sesión.html- es la pagina el usuario inicia sesión

Recuperación de contraseña.html- pagina donde el usuario recupera la contraseña Home.html- pagina principal donde se visualizara los álbumes

Canciones.html-pagina para visualizar las canciones de los álbumes edicionCancion.html-pagina para editar la información relacionada a las canciones ediciónAlbum- pagina para editar la información relacionada a los álbumes edicionUsuario- pagina para gestionar la información del usuario

# Capacitación

Se realizo una capacitación sobre Bootstrap, sobre lo mas básico hasta lo mas complejo.

# Patrón de diseño arquitectónico

Nuestro equipo de trabajo a elegido trabajar el patrón arquitectónico MVC (Modelo-Vista- Controlador), ya que nos brinda mayor mantenibilidad facilita el desarrollo en equipo, se puede reutilizar el código, nos brinda mayor flexibilidad y escalabilidad al proyecto. Junto a ello facilita las pruebas y su curva de aprendizaje es óptima para principiantes.

# Script y sentencias SQL

CREATE DATABASE IF NOT EXISTS discografia;

USE discografia;

CREATE TABLE Usuarios (

id\_usuario INT AUTO\_INCREMENT PRIMARY KEY,

nombre\_usuario VARCHAR(100) NOT NULL UNIQUE,

contraseña\_hash VARCHAR(255) NOT NULL, -- Almacenar el hash de la contraseña

email VARCHAR(255) UNIQUE,

fecha\_registro DATETIME DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE Artistas (

id\_artista INT AUTO\_INCREMENT PRIMARY KEY,

nombre\_artista VARCHAR(255) NOT NULL

);

-- 5. Crear la tabla de Álbumes

CREATE TABLE Albumes (

id\_album INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR(255) NOT NULL,

año\_produccion INT,

descripcion TEXT,

id\_artista INT,

medio\_grabacion VARCHAR(100),

FOREIGN KEY (id\_artista) REFERENCES Artistas(id\_artista)

);

CREATE TABLE Canciones (

id\_cancion INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR(255) NOT NULL,

año\_produccion INT,

descripcion TEXT,

id\_artista INT,

id\_album INT,

medio\_grabacion VARCHAR(100),

duracion\_segundos INT NOT NULL, -- Duración en segundos

FOREIGN KEY (id\_artista) REFERENCES Artistas(id\_artista),

FOREIGN KEY (id\_album) REFERENCES Albumes(id\_album)

);

INSERT INTO Artistas (nombre\_artista) VALUES

('Queen'),

('The Beatles'),

('Michael Jackson'),

('Adele'),

('Shakira');

INSERT INTO Usuarios (nombre\_usuario, contraseña\_hash, email) VALUES

('admin', 'password\_segura\_hash\_del\_admin', 'admin@ejemplo.com'),

('usuario1', 'hash\_de\_contraseña\_usuario1', 'usuario1@ejemplo.com'),

('juanperez', 'hash\_de\_contraseña\_juanperez', 'juan.perez@email.com');

INSERT INTO Albumes (titulo, año\_produccion, descripcion, id\_artista, medio\_grabacion) VALUES

('A Night at the Opera', 1975, 'Cuarto álbum de estudio de Queen, conocido por Bohemian Rhapsody.', 1, 'Vinilo'),

('Abbey Road', 1969, 'Undécimo álbum de estudio de The Beatles.', 2, 'Digital'),

('Thriller', 1982, 'El álbum más vendido de todos los tiempos.', 3, 'CD'),

('21', 2011, 'Segundo álbum de estudio de Adele, con éxitos como Rolling in the Deep.', 4, 'CD'),

('El Dorado', 2017, 'Undécimo álbum de estudio de Shakira, con canciones en español e inglés.', 5, 'Digital');

SELECT A.titulo AS Album, A.año\_produccion, AR.nombre\_artista

FROM Albumes A

JOIN Artistas AR ON A.id\_artista = AR.id\_artista;

SELECT C.titulo AS Cancion, C.año\_produccion, AL.titulo AS Album, AR.nombre\_artista

FROM Canciones C

JOIN Albumes AL ON C.id\_album = AL.id\_album

JOIN Artistas AR ON C.id\_artista = AR.id\_artista;

SELECT nombre\_usuario, email, fecha\_registro

FROM Usuarios;

SELECT \*

FROM Artistas;

SELECT

a.id\_album,

a.titulo AS album,

a.año\_produccion,

ar.nombre\_artista

FROM Albumes AS a

JOIN Artistas AS ar USING (id\_artista);

SELECT nombre\_usuario, fecha\_registro

FROM Usuarios

WHERE fecha\_registro > '2024-01-01';

SELECT titulo, año\_produccion

FROM Albumes

WHERE año\_produccion = (SELECT MAX(año\_produccion) FROM Albumes);

SELECT id\_usuario, nombre\_usuario, email, fecha\_registro

FROM Usuarios;

# Link Repositorio

[Valbu07/Biblioteca---Isaura](https://github.com/Valbu07/Biblioteca---Isaura/tree/main)