

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

4. Crear álbumes
5. Crear lista de álbumes
6. Eliminar álbumes
7. Editar álbumes
8. Visualizar canciones
9. Gestionar descripción
10. -Añadir título
11. -Agregar el año de producción
12. -Añadir la descripción
13. -Agregar artista
14. -Añadir canciones
15. -Añadir medio de grabación

Gestionar de canciones

16. Añadir canciones
17. Editar canciones
18. Eliminar canciones
19. Buscar canción
20. Añadir a álbum
21. Gestionar descripción
22. -Añadir título
23. -Agregar el año de producción
24. -Añadir la descripción
25. -Agregar artista
26. -Añadir canciones
27. -Añadir medio de grabación
28. -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](#)