

SERVICIO NACIONAL DE APRENDIZAJE SENA

Paula Andrea Cassiani Castillo

Duvier Olmos Cometa

Sara Ruiz López

20/06/2025

Bogotá Colombia

Isaura Suarez

ADSO IV Trimestre

Introducción

Este informe presenta el desarrollo de una aplicación web llamada Melomanos.net, la cual permite a usuarios melómanos gestionar su colección de música, agregando álbumes y canciones, editándolos, buscándolos y reutilizando canciones registradas por otros usuarios.

La aplicación fue diseñada utilizando el patrón arquitectónico MVC y tecnologías como HTML, CSS, JavaScript y Bootstrap. A continuación, se describe su estructura técnica, requerimientos funcionales y sentencias SQL correspondientes.

Planteamiento del problema:

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

RF001. Registro e inicio de sesión

- El sistema debe permitir que los usuarios se registren con un nombre de usuario y contraseña.
- El sistema debe permitir que los usuarios inicien sesión para acceder a su cuenta.

RF002. Gestión de álbumes

- RF002.1. El usuario debe poder agregar álbumes a su colección en cualquier momento.
- RF002.2. El usuario debe poder editar los datos de un álbum existente.

- RF002.3. El usuario debe poder eliminar álbumes (por estar repetidos o dañados).
- RF002.4. El usuario debe poder ver la información completa de un álbum.

RF003. Información del álbum

El sistema debe permitir almacenar por cada álbum:

- RF003.1. Título del álbum.
- RF003.2. Año de producción.
- RF003.3. Descripción.
- RF003.4. Medio en el que está grabado (CD, vinilo, casete, etc.).
- RF003.5. Artistas que interpretan las canciones.
- RF003.6. Canciones incluidas en el álbum.

RF004. Gestión de canciones

- RF004.1. El usuario debe poder agregar canciones a sus álbumes.
- RF004.2. El usuario debe poder editar la información de una canción.
- RF004.3. El usuario debe poder eliminar canciones de un álbum.
- RF004.4. Cada canción debe tener:

-Título

-Duración (minutos y segundos)

-Intérprete

RF005. Búsqueda y reutilización de canciones

- El sistema debe permitir que las canciones registradas por un usuario estén disponibles para que otros usuarios puedan buscarlas y agregarlas a sus propios álbumes

Estructura del proyecto

```

melomanos.net/
|
├─ app.py                # Archivo principal que ejecuta Flask
├─ requirements.txt      # Dependencias del entorno virtual
|
├─ venv/                 # Entorno virtual (NO se sube a GitHub)
|
├─ css/                  # Estilos personalizados
|   └─ estilos.css
|
├─ img/                  # Imágenes del proyecto
|   └─ login-fondo.jpg
|   └─ registro-fondo.jpg
|
├─ vistas/               # Archivos HTML de las vistas
|   └─ login.html
|   └─ registro.html
|   └─ albumes.html
|   └─ canciones.html
|   └─ busqueda.html
|
└─ database/             # Base de datos y scripts SQL
    └─ script.sql

```

Sentencias SQL de la base de datos

SENTENCIAS:

crear tablas

```

CREATE TABLE usuarios (...) -- Tabla de usuarios del sistema
CREATE TABLE artistas (...) -- Tabla de artistas musicales
CREATE TABLE albumes (...) -- Tabla principal de álbumes
CREATE TABLE canciones (...) -- Tabla de canciones
CREATE TABLE album_canciones (...) -- Relación álbumes-canciones
CREATE TABLE cancion_artistas (...) -- Relación canciones-artistas

```

Procedimientos Almacenados

```

CREATE PROCEDURE registrar_usuario(...) -- Registro de nuevos usuarios
CREATE PROCEDURE agregar_cancion_a_album(...) -- Agrega canciones a álbumes

```

Operaciones CRUD

```
INSERT INTO usuarios (...) -- Crear usuario  
SELECT... FROM usuarios -- Consultar usuario (login)
```

Tecnologías utilizadas

- HTML5, CSS3 y Bootstrap 5 para la interfaz.
- JavaScript para la lógica de interacción.
- Patrón MVC para organización del código.
- MySQL para la base de datos (modelo lógico).

Repositorio del proyecto

https://github.com/SaraRuizL/Caso_de_estudio