Logotipo

El contenido generado por IA puede ser incorrecto.UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

DEPARTAMENTO DE INGENIERÍA EN SOFTWARE

LICENCIATURA EN INGENIERÍA EN SOFTWARE

INGENIERÍA WEB

PROYECTO SEMESTRAL: FASE 1-2

DIEGO CORRALES 8-1001-1890

JUAN COPRI 8-976-1088

DANIEL PEREZ 20-14-7935

JOSUÉ PINO 8-1012-688

PROFESORA: DRA. ELBA VALDERRAMA

GRUPO 1SF134

II SEMESTRE

2025

**Índice**

[**Fase 1: Análisis y Modelado** 3](#_Toc212577759)

[**Descripción del Proyecto** 3](#_Toc212577760)

[**Escenarios de uso** 3](#_Toc212577761)

[**Requisitos Funcionales** 5](#_Toc212577762)

[**Requisitos No Funcionales** 5](#_Toc212577763)

[**Descripción de Caso de Uso CRUD** 6](#_Toc212577764)

[**Modelo de requisitos** 10](#_Toc212577765)

[**Modelo de contenido** 11](#_Toc212577766)

[**Modelo de Flujo de Proceso** 12](#_Toc212577767)

[**Modelo de navegación** 13](#_Toc212577768)

[**Fase 2: Arquitectura del contenido y diseño visual** 14](#_Toc212577769)

[**Ordenamiento de Tarjetas con 2 Usuarios** 14](#_Toc212577770)

[**Mapa de sitio web** 15](#_Toc212577771)

[**Prueba de Tree-Testing con 2 Usuarios** 16](#_Toc212577772)

[**Tipo de organización del contenido y el por qué** 19](#_Toc212577773)

[**Tipo de navegación a utilizar y explicación del por qué** 20](#_Toc212577774)

[**Prototipos de baja fidelidad con los labels escritos** 21](#_Toc212577775)

# **Fase 1: Análisis y Modelado**

## **Descripción del Proyecto**

El presente proyecto tiene como objetivo el desarrollo de una página web de música construida desde cero, aplicando los principios de la Ingeniería de Software en todas sus etapas: análisis, diseño, desarrollo, pruebas y documentación. El propósito principal es diseñar e implementar una plataforma digital que permita al usuario gestionar su propia biblioteca musical, brindándole la posibilidad de agregar, editar, eliminar y organizar canciones de manera personalizada mediante listas de reproducción o playlists.

El sistema busca ofrecer una experiencia intuitiva, visualmente atractiva y funcional, en la que el usuario pueda acceder a una lista general de canciones, realizar búsquedas filtradas (por género, artista o nombre) y crear playlists según sus preferencias. Además, permitirá modificar los atributos de cada canción (como título, género o artista), editar los nombres de las playlists, eliminar canciones específicas o listas completas, y visualizar tanto las canciones como las playlists existentes.

Desde el punto de vista técnico, el proyecto se desarrollará utilizando HTML5, CSS3, JavaScript y PHP, integrando principios de diseño responsivo para garantizar una correcta visualización en distintos dispositivos. Asimismo, el proyecto incluirá toda la documentación de análisis y diseño necesaria: diagramas de casos de uso, diagramas de clases, modelo entidad-relación, flujos de interacción y especificaciones funcionales, siguiendo los estándares académicos de la carrera de Ingeniería de Software.

En esencia, esta aplicación web busca simular un entorno musical interactivo que combine una estructura de gestión de datos bien definida con una interfaz práctica y moderna, reflejando un proceso completo de desarrollo de software orientado a la experiencia del usuario y al cumplimiento de requerimientos funcionales claramente establecidos.

## **Escenarios de uso**

Caso de uso: Agregar Canción

Narrativa:  
María inicia sesión y entra a Canciones. Presiona “Agregar canción” y se abre el formulario con Título, Artista, Género y Año. Escribe *“Midnight”* en Título, *“Coldplay”* en Artista, selecciona Alternative Rock como Género y deja Año vacío por descuido. Pulsa Guardar.  
El sistema valida y resalta Año con el mensaje “Este campo es obligatorio”. María ingresa 2014 y vuelve a pulsar Guardar. El sistema almacena la canción y muestra “Canción creada exitosamente”. Al regresar a la lista, *“Midnight – Coldplay”* aparece en la primera posición con las acciones Editar y Eliminar disponibles.

Caso de uso: Editar Canción

Narrativa:  
Carlos inicia sesión, abre Canciones y localiza *“Midnight – Coldplay”*. Selecciona Editar. El sistema muestra el formulario prellenado. Carlos cambia el Género de Alternative Rock a Pop Rock y ajusta el Artista a *“Coldplay (UK)”* para estandarizar el nombre. Antes de guardar, presiona Cancelar; el sistema lo regresa a la lista sin cambios.  
Minutos después, vuelve a Editar, realiza las mismas modificaciones y pulsa Guardar cambios. El sistema intenta actualizar, pero ocurre un error intermitente y muestra “No se pudo actualizar la canción. Intente nuevamente.”. Carlos permanece en el formulario con sus cambios; presiona Guardar cambios por segunda vez. El sistema ahora actualiza con éxito y despliega “Cambios guardados correctamente”. Al volver a la lista, la tarjeta muestra Género: Pop Rock y Artista: Coldplay (UK).

Caso de uso: Borrar Canción

Narrativa:  
Ana entra a Canciones, abre el menú de *“Midnight – Coldplay (UK)”* y elige Eliminar. El sistema muestra un diálogo de confirmación con Cancelar y Eliminar. Ana se arrepiente y presiona Cancelar, por lo que no hay cambios y la lista sigue igual.  
Más tarde, decide realmente quitarla. Vuelve a Eliminar, confirma con Eliminar y el sistema intenta procesar la solicitud. Aparece “No se pudo eliminar la canción. Intente nuevamente.” por un problema temporal; la canción permanece en la lista. Ana repite la acción (Eliminar → Confirmar). Esta vez, el sistema elimina la canción y muestra “Canción eliminada”; la lista se actualiza y *“Midnight – Coldplay (UK)”* ya no aparece.

Caso de uso: Ver Canciones (y Filtrar/Buscar)

Narrativa:  
Julián inicia sesión y navega a Canciones. El sistema despliega la lista completa con barra de Búsqueda y filtros de Género, Artista y Título. Julián primero escribe *“The 1975”* en Búsqueda; la lista se actualiza mostrando coincidencias del artista (por ejemplo, *“Somebody Else – The 1975”*, *“Robbers – The 1975”*). Decide afinar: activa el filtro Género: Alternative/Indie. La lista se reduce a canciones de *The 1975* dentro de ese género.  
Curioso, borra el texto de búsqueda y deja solo Género: Alternative/Indie, pero agrega Título: “Believer” esperando ver *Imagine Dragons*. El sistema no encuentra coincidencias y muestra “No se encontraron resultados” junto a la opción Limpiar filtros. Julián pulsa Limpiar filtros, se restablece la lista completa y luego filtra por Artista: Imagine Dragons para revisar detalles (p. ej., *“Believer – Imagine Dragons”*). Abre una tarjeta para ver información y desde el menú contextual observa la opción Agregar a playlist (sin ejecutarla en este caso).

## **Requisitos Funcionales**

|  |  |  |
| --- | --- | --- |
| RF-1 | Agregar canciones | El sistema permitirá al usuario agregar nuevas canciones a la base de datos. |
| RF-2 | Ver lista general de canciones | El sistema debe mostrar una lista completa de canciones disponibles. |
| RF-3 | Crear playlist | Los usuarios podrán crear nuevas playlists. |
| RF-4 | Agregar canción a playlist | El sistema permitirá agregar canciones a playlists existentes. |
| RF-5 | Editar playlist | Los usuarios podrán modificar los nombres de sus playlists. |
| RF-6 | Editar canciones | Los usuarios podrán modificar los atributos de las canciones (género, artista, título, etc.). |
| RF-7 | Borrar canciones de una playlist | Los usuarios podrán eliminar canciones de sus playlists. |
| RF-8 | Borrar playlist completa | Los usuarios podrán eliminar completamente sus playlists. |
| RF-9 | Ver playlist | Los usuarios podrán ver las playlists que han creado. |
| RF-10 | Ver canciones | El sistema permitirá al usuario visualizar la lista completa de canciones. |
| RF-11 | Filtro de búsqueda | El sistema debe permitir buscar canciones por género, artista, nombre, etc. |

## **Requisitos No Funcionales**

|  |  |  |
| --- | --- | --- |
| RNF-1 | Responsividad | El sistema debe ser accesible desde dispositivos móviles y desktop. |
| RNF-2 | Seguridad | El sistema debe garantizar la protección de los datos del usuario mediante cifrado y autenticación adecuada. |
| RNF-3 | Rendimiento | El sistema debe manejar grandes volúmenes de canciones y playlists sin afectar la experiencia del usuario. |
| RNF-4 | Rendimiento | El sistema debe manejar grandes volúmenes de canciones y playlists sin afectar la experiencia del usuario. |
| RNF-5 | Escalabilidad | El sistema debe ser capaz de expandirse fácilmente para añadir más funcionalidades en el futuro. |
| RNF-6 | Usabilidad | El sistema debe tener una interfaz intuitiva y fácil de usar. |

## **Descripción de Caso de Uso CRUD**

1. **Agregar Canción**

**Actor:** Usuario

**Precondiciones:**

* El usuario debe estar autenticado.

**Flujo Básico:**

* El usuario selecciona la opción para agregar una nueva canción.
* El sistema muestra un formulario para ingresar los detalles de la canción (título, artista, género, etc.).
* El usuario completa los campos y selecciona "Guardar".
* El sistema almacena los datos en la base de datos y confirma la creación de la canción.

**Flujo Alternativo:**

* Si los campos obligatorios están vacíos, el sistema solicita que se complete la información antes de guardar.

**Flujos de Excepción:**

* Si ocurre un error en el servidor, el sistema muestra un mensaje de error y no guarda la canción.

**Flujo Postcondicional:**

* La canción está disponible en la lista general y puede ser editada o eliminada.

**2. Editar Canción**

**Actor:** Usuario

**Precondiciones:**

* El usuario debe estar autenticado.

**Flujo Básico:**

* El usuario selecciona la canción que desea editar.
* El sistema muestra un formulario con los detalles actuales de la canción.
* El usuario modifica los campos que desea cambiar.
* El sistema actualiza la base de datos con los nuevos detalles de la canción.

**Flujo Alternativo:**

* Si el usuario decide no realizar cambios, puede cancelar la edición y regresar a la lista de canciones.

**Flujos de Excepción:**

* Si hay un error al intentar actualizar la canción, el sistema muestra un mensaje de error.

**Flujo Postcondicional:**

* La canción modificada aparece con los cambios reflejados en la lista general.

1. **Borrar Canción**

**Actor:** Usuario

**Precondiciones:**

* El usuario debe estar autenticado y haber seleccionado una canción existente.

**Flujo Básico:**

* El usuario selecciona la opción para eliminar una canción.
* El sistema muestra una confirmación de eliminación.
* El usuario confirma la eliminación.
* El sistema elimina la canción de la base de datos y la confirma visualmente.

**Flujo Alternativo:**

* Si el usuario decide cancelar la eliminación, el sistema regresa sin hacer cambios.

**Flujos de Excepción:**

* Si hay un error al eliminar la canción, el sistema muestra un mensaje de error y no la elimina.

**Flujo Postcondicional:**

* La canción ya no aparece en la lista general.

**4. Ver Canciones**

**Actor**: Usuario

**Precondiciones:**

* El usuario debe estar autenticado.
* El sistema debe tener canciones registradas en la base de datos.

**Flujo Básico:**

1. El usuario ingresa al sistema.
2. El usuario navega a la sección de canciones.
3. El sistema muestra una lista completa de todas las canciones registradas en la base de datos.
4. El usuario puede interactuar con la lista (ver detalles de la canción, añadir canciones a playlists, etc.).

**Flujo Alternativo - Filtro de Búsqueda:**

1. Si el usuario desea buscar canciones específicas, puede utilizar la barra de búsqueda o filtros.
2. El sistema ofrece opciones de filtro por género, artista, nombre de la canción, etc.
3. El usuario selecciona un filtro o introduce un término de búsqueda.
4. El sistema filtra las canciones según el criterio seleccionado y muestra la lista filtrada.

**Flujos Alternativos del Filtro**:

1. **Filtro por género**: Si el usuario selecciona un género específico, el sistema solo muestra canciones de ese género.
2. **Filtro por artista**: Si el usuario ingresa un nombre de artista, el sistema muestra solo las canciones de ese artista.
3. **Filtro por nombre de canción**: Si el usuario busca por nombre de canción, el sistema muestra las canciones que coincidan con ese nombre.

**Flujo Excepcional:**

1. Si no hay coincidencias para los filtros seleccionados o la búsqueda es vacía, el sistema muestra un mensaje indicando que no se encontraron resultados.
2. Si el sistema no puede aplicar el filtro debido a un error técnico (como problemas en la base de datos), el sistema muestra un mensaje de error informando al usuario de la falla.

**Flujo Postcondicional:**

* El sistema presenta la lista filtrada correctamente.
* Si no hay canciones que coincidan con el filtro, la lista estará vacía con un mensaje que informe al usuario de la ausencia de resultados.

## **Modelo de requisitos**

Diagrama

El contenido generado por IA puede ser incorrecto.

## **Modelo de contenido**

Diagrama

El contenido generado por IA puede ser incorrecto.

## **Modelo de Flujo de Proceso**

## **Modelo de navegación**

# Interfaz de usuario gráfica, Aplicación El contenido generado por IA puede ser incorrecto.**Fase 2: Arquitectura del contenido y diseño visual**

## **Ordenamiento de Tarjetas con 2 Usuarios**

Tabla

El contenido generado por IA puede ser incorrecto.

## **Mapa de sitio web**

Diagrama

El contenido generado por IA puede ser incorrecto.

## **Prueba de Tree-Testing con 2 Usuarios**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.**