



## **1. REQUISITOS DEL SISTEMA**

### **1.1. FASE INICIAL**

A modo de introducción, explicaré de manera general en lo que se basa este proyecto. La idea es agrupar todos los posibles conocimientos del primer año de DAM (Desarrollo de Aplicaciones Multiplataforma), en un único proyecto, abarcando conocimientos como conceptos básicos de bases de datos relacionales, fundamentos de la programación orientada a objetos en Java, estructura básica de páginas web con HTML, estilización de páginas web con CSS e interactividad en la página con JavaScript, conceptos básicos de pruebas de software, y la correspondiente documentación del código, entre otros.

El proyecto se centra en desarrollar una aplicación, la cual, será una plataforma para gestionar información sobre artistas, canciones, álbumes y profesionales relacionados en la industria musical. Permitirá a los usuarios almacenar, editar y visualizar datos detallados, ofreciendo funcionalidades avanzadas como búsquedas específicas, análisis de estadísticas y comparaciones entre artistas. La aplicación estará optimizada para un alto rendimiento, seguridad y usabilidad, asegurando una experiencia fluida y accesible. Además, facilitará la integración con otros sistemas y proporcionará una interfaz intuitiva y adaptable a diferentes dispositivos. En resumen, será una herramienta robusta y versátil para organizar y analizar la información musical de manera eficiente. A continuación, detallaremos cada una de estas funcionalidades.

### **1.2. ANÁLISIS**

A continuación, especificaré los requisitos funcionales y no funcionales, proporcionando una breve descripción de cada uno para entender sus diferencias. Los requisitos funcionales detallan las funciones y características específicas que debe cumplir el sistema, mientras que los requisitos no funcionales describen los criterios de calidad y restricciones operativas que debe seguir el sistema para garantizar su rendimiento y usabilidad.

#### **1.2.1 REQUISITOS FUNCIONALES**

##### *1) Gestión de Artistas*

- Agregar, editar y eliminar artistas: Permitir la gestión completa de los registros de artistas.
- Visualizar detalles de un artista: Mostrar toda la información de un artista, incluyendo canciones, discos, premios, conciertos, etc.
- Buscar artistas por diferentes criterios: Nombre artístico, país, ciudad, género, estado, etc.
- Listar artistas activos, retirados y fallecidos.

##### *2) Gestión de Representantes*

- Agregar, editar y eliminar representantes.
- Visualizar detalles de un representante, incluyendo la lista de artistas representados.
- Buscar representantes por nombre, nacionalidad, etc.

3) *Gestión de Productores*

- Agregar, editar y eliminar productores.
- Visualizar detalles de un productor, incluyendo canciones producidas.
- Buscar productores por nombre, nacionalidad, etc.

4) *Gestión de Redes Sociales*

- Agregar, editar y eliminar redes sociales de los artistas.
- Visualizar detalles de las redes sociales de un artista, incluyendo número de seguidores.

5) *Gestión de Premios*

- Agregar, editar y eliminar premios.
- Visualizar detalles de un premio, incluyendo el ganador y la organización que lo entrega.
- Buscar premios por nombre, fecha, etc.

6) *Gestión de Canciones*

- Agregar, editar y eliminar canciones.
- Visualizar detalles de una canción, incluyendo artistas, álbum, productor, etc.
- Buscar canciones por título, artista, género, etc.
- Mostrar el número total de reproducciones de una canción.

7) *Gestión de Conciertos*

- Agregar, editar y eliminar conciertos.
- Visualizar detalles de un concierto, incluyendo artistas colaboradores, lugar, duración, etc.
- Buscar conciertos por artista, país, ciudad, fecha, etc.

8) *Gestión de Géneros*

- Agregar, editar y eliminar géneros.
- Visualizar detalles de un género, incluyendo su país de origen.
- Buscar géneros por nombre, país de origen, etc.

9) *Gestión de Álbumes*

- Agregar, editar y eliminar álbumes.
- Visualizar detalles de un álbum, incluyendo canciones, artistas colaboradores, géneros, etc.
- Buscar álbumes por título, artista, género, etc.
- Mostrar el número total de reproducciones de un álbum.

10) *Funcionalidades Adicionales*

- Artista con más conciertos: Función que permita identificar al artista con más conciertos realizados.
- Álbum con más y menos reproducciones: Función para encontrar los álbumes con más y menos reproducciones totales.

- Artista con más premios ganados: Identificar al artista que ha ganado más premios.
- Canción más reproducida: Función para encontrar la canción con el mayor número de reproducciones.
- Búsqueda avanzada de conciertos: Permitir búsquedas por rango de fechas, país y ciudad.
- Estadísticas de géneros musicales: Mostrar estadísticas de géneros más interpretados por diferentes artistas.
- Conciertos próximos: Listar los próximos conciertos de cada artista.
- Historial de premios de un artista: Mostrar un historial de premios ganados por un artista a lo largo de su carrera.
- Comparación entre artistas: Comparar dos o más artistas en términos de oyentes de Spotify, número de premios, canciones, discos, etc.
- Informe de carrera de un artista: Generar un informe completo sobre la carrera de un artista, incluyendo todos los conciertos dados, premios ganados, álbumes y canciones publicadas.

### **1.2.2 REQUISITOS NO FUNCIONALES**

#### ***1) Rendimiento***

- Tiempo de respuesta: Las consultas y operaciones deben completarse en menos de 2 segundos para mantener una experiencia de usuario fluida.
- Escalabilidad: La aplicación debe poder manejar un aumento en la cantidad de datos (por ejemplo, miles de artistas, canciones, álbumes, etc.) sin degradación significativa del rendimiento.

#### ***2) Seguridad***

- Autenticación y autorización: Implementar un sistema de autenticación para usuarios y un control de acceso basado en roles (administradores, editores, usuarios estándar).
- Protección de datos: Encriptar datos sensibles como contraseñas, correos electrónicos y números de teléfono.
- Auditoría y logs: Mantener registros de acceso y modificaciones de datos para auditorías de seguridad.

#### ***3) Usabilidad***

- Interfaz de usuario intuitiva: La interfaz debe ser fácil de usar y entender, con una navegación clara y coherente.
- Accesibilidad: La aplicación debe cumplir con las pautas de accesibilidad web (WCAG) para ser utilizable por personas con discapacidades.
- Compatibilidad con dispositivos móviles: La interfaz debe ser responsiva, adaptándose correctamente a diferentes tamaños de pantalla y dispositivos.

#### 4) *Mantenimiento*

- Documentación: Proveer documentación detallada y actualizada del código, así como manuales de usuario y guías de instalación.
- Modularidad: El diseño de la aplicación debe ser modular para facilitar el mantenimiento y las futuras actualizaciones.
- Pruebas automatizadas: Implementar un conjunto de pruebas automatizadas para asegurar la funcionalidad del sistema después de cambios y actualizaciones.

#### 5) *Fiabilidad*

- Disponibilidad: La aplicación debe tener una disponibilidad mínima del 99.9%, asegurando que esté accesible en la mayoría del tiempo.
- Recuperación ante fallos: Implementar mecanismos de respaldo y recuperación de datos para minimizar la pérdida de información en caso de fallos del sistema.

#### 6) *Escalabilidad y Rendimiento*

- Carga máxima: La aplicación debe soportar al menos 1000 usuarios concurrentes sin pérdida significativa de rendimiento.
- Optimización de consultas: Optimizar las consultas a la base de datos para asegurar tiempos de respuesta rápidos incluso con grandes volúmenes de datos.

#### 7) *Portabilidad*

- Compatibilidad con múltiples plataformas: La aplicación debe ser compatible con los principales sistemas operativos (Windows, macOS, Linux) y navegadores web (Chrome, Firefox, Edge, Safari).
- Implementación en la nube: Facilitar la implementación en servicios de nube como AWS, Azure o Google Cloud para aprovechar la escalabilidad y disponibilidad que ofrecen estas plataformas.

#### 8) *Integración*

- Importación y exportación de datos: Permitir la importación y exportación de datos en formatos comunes (CSV, JSON, XML) para facilitar la interoperabilidad con otros sistemas.

#### 9) *Localización e Internacionalización*

- Soporte multilingüe: La aplicación debe soportar múltiples idiomas y permitir la fácil adición de nuevas traducciones.
- Formatos regionales: Adaptarse a formatos de fecha, hora, y moneda según la configuración regional del usuario.

