

PROYECTO FINAL

Motor de Recomendaciones Musicales 'SyncUp'
Curso: Estructura de Datos
FACULTAD DE INGENIERÍA

DESCRIPCIÓN DEL PROYECTO FINAL - CONTEXTO

Desarrollar el motor de backend para "SyncUp", una plataforma de streaming y descubrimiento social de música. La plataforma permite a los usuarios gestionar su perfil, buscar canciones en un catálogo masivo, recibir recomendaciones inteligentes generadas por algoritmos y conectar con otros usuarios según sus gustos musicales. Existen dos perfiles: **Usuario** y **Administrador**.

1. Requerimientos del Usuario

Requisito	Especificación
RF-001	Registrarse y/o iniciar sesión en la plataforma.
RF-002	Gestionar su perfil (nombre, contraseña) y su lista de canciones favoritas.
RF-003	Buscar canciones por autocompletado de título.
RF-004	Realizar búsquedas avanzadas de canciones por múltiples atributos (artista, género, año) con lógica AND y OR.
RF-005	Generar una playlist "Descubrimiento Semanal" basada en sus gustos.
RF-006	Iniciar una "Radio" a partir de una canción, generando una cola de reproducción con temas similares.
RF-007	Conectar con otros usuarios (seguir/dejar de seguir).
RF-008	Recibir sugerencias de usuarios a quienes seguir.
RF-009	Descargar un reporte de sus canciones favoritas en formato CSV.

2. Requerimientos del Administrador

Requisito	Especificación
RF-010	Gestionar el catálogo de canciones (agregar/actualizar/eliminar).
RF-011	Gestionar usuarios (listar/eliminar).
RF-012	Cargar canciones masivamente desde un archivo de texto plano.
RF-013	Visualizar un panel de métricas del sistema.
RF-014	Visualización de métricas con JavaFX Charts (ej: Pie Chart de géneros, Bar Chart de artistas más populares).

3. Requerimientos sobre Entidades y Módulos del Sistema

3.1. Usuario

Entidad que representa a los usuarios de la plataforma.

Requisito	Especificación
RF-015	Almacenar username (único), password, nombre, listaFavoritos (LinkedList<Cancion>).
RF-016	Ser indexado en un HashMap<String, Usuario> para acceso $O(1)$.
RF-017	Implementar hashCode() y equals() basado en username.

3.2. Cancion

Entidad que representa una pista musical en el catálogo.

Requisito	Especificación
RF-018	Almacenar id (único), titulo, artista, genero, año, duracion.
RF-019	Funcionar como nodo en el Grafo De Similitud.
RF-020	Implementar hashCode() y equals() basado en id.

3.3. GrafoDeSimilitud

Módulo que conecta canciones basadas en su similitud.

Requisito	Especificación
RF-021	Implementarse como un Grafo Ponderado No Dirigido .
RF-022	Soportar algoritmos de recorrido como Dijkstra para encontrar rutas de menor costo (mayor similitud).

3.4. GrafoSocial

Módulo que modela las conexiones entre usuarios.

Requisito	Especificación
RF-023	Implementarse como un Grafo No Dirigido .
RF-024	Soportar recorridos BFS para encontrar "amigos de amigos".

3.5. TrieAutocompletado

Módulo para búsquedas de texto eficientes.

Requisito	Especificación
RF-025	Implementarse como un Árbol de Prefijos (Trie) .
RF-026	Devolver todas las palabras que comiencen con un prefijo dado.

4. Requerimientos Técnicos

Requisito	Especificación
RF-027	Diagrama de Clases: Completo, mostrando relaciones, multiplicidad y jerarquías entre las entidades y los módulos del sistema.
RF-028	Implementación Técnica: Proyecto funcional en Java con interfaz gráfica en JavaFX para

	los perfiles de Usuario y Administrador.
RF-029	Generador de Reportes: Exportación de datos a CSV (ej: lista de favoritos del usuario).
RF-030	Concurrencia: La búsqueda avanzada por atributos (RF-004) debe ser implementada usando Hilos de ejecución para optimizar el rendimiento.
RF-031	Pruebas Unitarias: Cobertura de al menos 7 métodos clave del sistema con pruebas unitarias.
RF-032	JavaDoc: Generación de la documentación completa del proyecto.

5. Evaluación

La evaluación se realizará de la siguiente forma:

- Sustentación del proyecto de forma presencial. Deben de estar los integrantes del grupo, si algún integrante no se presenta solo se les califica a los integrantes que estén.
- Cumplimiento de los requisitos de la aplicación.
- Funcionamiento de la aplicación.

Importante:

La nota del proyecto está compuesta por dos partes:

1. La **codificación**, que se califica de 0 a 5.
2. Un **factor multiplicador** basado en la **sustentación**, que varía de 0 a 1.

Este resultado, obtenido mediante la multiplicación de las dos notas (partes), nos proporciona la nota final del proyecto.

Recuerden:

El profesor elige a la persona que realiza la sustentación, debido a que, todos los integrantes del grupo deben estar en condiciones de sustentar. La persona elegida, de acuerdo con su nivel de sustentación, determina el valor de la nota de la sustentación grupal, que será multiplicada por el valor de la codificación.