

## Informe de responsabilidades

La subida de este documento firmado al repositorio es obligatoria para acceder a la sustentación y debe hacerse a más tardar el día anterior a la misma. Es obligatorio solo para entregas en pareja.

Nombre del componente desarrollado (clase , métodos o funciones)	Descripción breve de dicho componente	Porcentaje de complejidad respecto al desarrollo global	Desarrollador principal (nombre de sólo una persona)
AdManager.cpp ; AdManager.h	Gestiona una colección de anuncios y un generador de números aleatorios. Su método principal, selecciona un anuncio basándose en su peso ponderado, implementando una lógica de no repetición inmediata del último anuncio elegido	6%	Jeisson Martinez
Advertisement.h	Define la estructura de un anuncio, que incluye un ID, una categoría y un texto. Nos Proporciona el método para obtener el peso numérico utilizado en la selección ponderada.	3%	Jeisson Martinez
DynArray.h	Implementa una clase genérica tipo contenedor que funciona como un arreglo dinámico. Maneja la reasignación de memoria, los constructores de copia y el operador de asignación, utilizando asignación de memoria manual y destructores explícitos.	14%	Jeisson Martinez
FavoriteList.h	Clase que contiene una colección de punteros a canciones. Impide la adición de canciones duplicadas e impone un límite de 10,000 canciones.	7%	Jeisson Martinez
FileStore.cpp ; FileStore.h	Contiene métodos estáticos para manejar la carga y el guardado de todas las estructuras de datos principales del sistema (Canciones,	9%	Jeisson Martinez

	Anuncios, Usuarios, Favoritos, Seguidos) desde y hacia archivos de texto plano, realizando la validación de los datos.		
Menu.cpp ; Menu.h	Es el módulo que contiene las colecciones de datos. Gestiona el flujo de la aplicación, incluyendo el Login y los bucles de interacción tanto para los estándar como para los premiums e Implementa la funcionalidad de pausa/reanudación global.	8%	Jeisson Martinez
Player.cpp ; Player.h	Controla la reproducción de canciones. Mantiene la lista de canciones global y dos historiales: un buffer circular (Ring) y un historial lineal navegable (_histList) para permitir pasar y devolverse. El método maybeAd() inserta anuncios para usuarios Standard).	11%	Jeisson Martinez
main.cpp	Es responsable de mostrar el resumen de recursos (Memoria e Iteraciones) al finalizar, gracias a los trackers globales.	2%	Jeisson Martinez

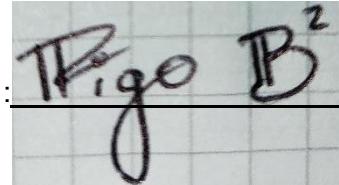
Song.h, Song.cpp	<p>Declara la clase Song, que modela cada canción del sistema: ID de 9 dígitos, nombre, duración, número de reproducciones, ruta base del archivo de audio, artista, álbum y portada.</p> <p>Inicializa correctamente todos los campos de la canción, configura rutas por defecto, actualiza el contador de reproducciones y construye las rutas de los archivos de audio según la calidad.</p>	12%	Rigoberto Berrio
Iteration.h	Define el pequeño módulo de instrumentación de complejidad. Proporciona el contador global de iteraciones y las macros para medir cuántas operaciones realiza cada algoritmo.	2%	Rigoberto Berrio
RandomLCG.h	Implementa un generador de números pseudoaleatorios tipo (LCG) de 64 bits. Ofrece métodos para obtener valores aleatorios acotados, usados por el sistema para seleccionar elementos sin depender de la STL.	3%	Rigoberto Berrio
Ring.h	Implementa un búfer circular de capacidad fija gestionado manualmente. Proporciona operaciones de inserción, consulta de tamaño/capacidad y acceso a elementos recientes.	3%	Rigoberto Berrio
StringUtil.h , StringUtil.cpp	Declara las utilidades de manejo de cadenas dentro del StringUtil.cpp. Implementa las funciones declaradas en StringUtil.h. Elimina espacios en blanco al inicio y al final de una cadena, separa una línea de texto por un delimitador almacenando los campos. Es clave para parsear correctamente los .txt del dataset.	5%	Rigoberto Berrio

User.h	Representa a cada usuario del sistema: nick, ciudad, país, tipo, fecha de registro, lista de favoritos y el usuario Premium al que sigue. Incluye la lógica para configurar el tipo, gestionar favoritos y una operación, que restringe el seguimiento solo a usuarios Premium y evita casos inválidos	5%	Rigoberto Berrio
MemoryTracker.h, MemoryTracker.cpp	Declara la interfaz del módulo MemTrack, encargado de llevar el conteo de bytes asignados, liberados y de la memoria actualmente en uso. Mantiene contadores globales de bytes asignados y liberados y sobrecarga los operadores globales.	10%	

Tabla resumen

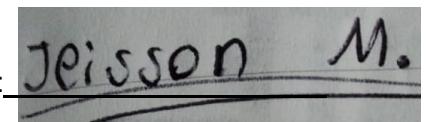
Nombre del integrante (Una fila por integrante del equipo)	Nombre de todos los componentes desarrollados	Porcentaje total desarrollado (la suma de los ítems es 100)
Jeisson Martinez	AdManager.cpp, AdManager.h, Advertisement.h, DynArray.h, FavoriteList.h, FileStore.cpp, FileStore.h, Menu.cpp, Menu.h, Player.cpp, Player.h, main.cpp	60%
Rigoberto Berrio	Iteration.h, RandomLCG.h, Ring.h, StringUtil.h, StringUtil.cpp, User.h, Song.h, Song.cpp, MemoryTracker.h, MemoryTracker.cpp	40%
	Total	100%

Nombre y firma autógrafa del integrante 1:



Rigo B<sup>2</sup>

Nombre y firma autógrafa del integrante 2:



Jeisson M.

**Nota:** La repartición de responsabilidades especificada en este formato no exime a ninguno de los miembros del equipo de la responsabilidad de conocer y explicar el análisis y diseño de las estrategias que fundamentan toda la solución entregada.