

Voy a organizar el programa en varias clases que representen los elementos más importantes del sistema
(usuario, usuariopremium, artista, album, cancion, publicidad, listadefavoritos)
y una clase principal que voy a llamar plataforma

La clase plataforma seria la que maneja todo cargar los datos
iniciar sesión, controlar la reproducción y medir los recursos
de esta manera mantengo el código bien por secciones y fácil de probar porque cada clase se encarga de su parte

como no puedo usar `std::vector` para las listas que serviría mucho para las las canciones dentro de un album
mi idea es usar un puntero doble y variables que indiquen cuántos elementos tengo y cuál es la capacidad máxima
tipo, `|Cancion** canciones;, int numcanciones, int capacidad|`
lo que planeo hacer probar con generarlas de manera automática en memoria asignándoles nombres identificadores y duraciones aleatorias
Eso me permite probar la capacidad del sistema sin depender de datos reales ni de archivos externos y principalmente medir el rendimiento del programa y como pide el punto de medición de consumo de recursos
el manejo se puede hacer con punteros y arreglos

para manejar nombres y rutas realmente prefiero usar string porque es más práctico que trabajar con arreglos de char
las rutas de los audios se pueden armar fácilmente concatenando cadenas y pues eso facilitara mucho mostrar en pantalla la ubicación completa de los archivos cuando se muestra una canción

en cuanto a la simulación de la reproduccion aun no tengo una buena base pero podría usar librerías que ayuden

Los usuarios estándar tienen que ver anuncios cada dos canciones por lo que puedo tener una lista de mensajes publicitarios con una categoría, como cada categoría tiene distinta prioridad se puede simular la probabilidad repitiendo más veces los mensajes que tienen más peso, cuando elija un mensaje aleatoriamente con `rand()`, habrá más chances de que salgan los que tienen prioridad alta esto para no tener que hacer una estructura complicada

después de cada funcionalidad voy a mostrar dos cosas, cantidad de iteraciones y cuanta memoria se esta usando
El número de iteraciones se puede contar con un contador global que aumente en cada bucle o función relevante

Y la memoria la puedo estimar con el operador sizeof sumando el tamaño de los objetos creados y de los arreglos dinámicos para mayor facilidad

El inicio de sesión es bastante sencillo por que pienso que principalmente son menus, principalmente los datos de los usuarios estarán guardados en un archivo de texto usuarios.txt y al ingresa el sistema busca si existe
no se puede registrar un usuario nuevo por lo que solo se puede entrar con uno ya existente
si el usuario es premium se le muestra un menú con más opciones como editar o reproducir su lista de favoritos
si es estándar tendrá un menú más limitado

las listas de favoritos solo las tendrán los usuarios premium
voy a manejarlas como un arreglo dinámico de punteros a canciones sin permitir repeticiones
Cuando un usuario siga la lista de otro puedo usar la sobrecarga del operador + para unir ambas listas
como lo puede ser, mi_lista + listadelotro_usuario
eso demuestra el uso de sobrecarga de operadores y me ayuda en simplificar el código

EN PRUEBAS

lo que pienso es ir por partes
primero probar las clases individualmente tipo, crear una canción y mostrar su información
Luego probar cómo se relaciona un álbum con sus canciones, un artista con sus álbumes
despues, probar el login y el menu
probar la reproducción con el temporizador y la publicidad
(claro resalto que que tienen mucho material pero principalmente, asi que se pueden hacer cambios a conveniencia para el desarrollo)