

Proyecto de Aplicación No. 1 - Programación Avanzada

Francisco Andrés Rodríguez Orozco - 1060820

October 9, 2023

Contents

1 Documentación General

1.1 ¿Qué acciones debe poder hacer su programa? Enumérelas.

El programa tiene seis acciones principales.

1. Cargar respaldos de una carpeta. El programa siempre empieza pidiéndolas antes de cualquier otra acción, pero se puede volver a repetir después de iniciado.
2. Agregar canciones. Añade una canción deseada desde los CDs hacia la cola de reproducción.
3. Ver la cola de reproducción. Permite ver la cola, tanto en su estado actual como en hipotéticos estados ordenados.
4. Reproducción actual. Permite verificar qué canción se está reproduciendo actualmente.
5. Ordenar. Permite ordenar la cola de reproducción en sí en base a lo que el usuario desee.
6. Salir. Sale del programa.

1.2 ¿Con qué datos va a trabajar? ¿Qué información debe pedir al usuario?, defina sus datos de entrada y el tipo de dato que utilizará para los datos principales.

Con archivos de texto que sigan una estructura ya delimitada anterior a ser pasados al programa. Pueden tener muchas o pocas líneas, pero siempre bajo reglas entendibles que hacen que el programa siempre pueda leer los archivos con un formato. La información que se le pedirá al usuario consiste en la ubicación de los archivos de texto a leer, además de todas las opciones para las diferentes acciones que existen. El único dato de entrada explícito es el directorio donde se encuentran los archivos a leer. El tipo de dato principal es el string, ya que la mayor parte de la información se almacenará en forma de texto.

1.3 ¿Qué estructuras de datos trabajará para almacenar la información?

Se eligieron la lista y la cola tanto por sus propiedades como por sus comportamientos. No se sabe por adelantado cuantos archivos el usuario leerá ni cuantas canciones contienen cada uno, por lo que es necesario una estructura de datos que no tenga tope o tamaño definido, haciendo a la lista ideal en esa situación. Por otro lado, la cola de reproducción necesariamente implica trabajar con colas por su comportamiento especial, por lo que naturalmente se requiere trabajar con ellas.

1.4 ¿Qué condiciones o restricciones debe tomar en cuenta? ¿Qué cálculos debe hacer?

Las condiciones y restricciones que se deben tomar en cuenta tienen que ver con el directorio, los archivos y su contenido. El directorio debe de existir, en primer lugar. Los archivos deben de ser de texto plano y tienen que seguir un formato determinado para asegurar su buen funcionamiento y entendimiento. El contenido de los archivos debe seguir ciertas reglas como que no se repitan las líneas o que incluso haya contenido en los archivos y no sean únicamente archivos vacíos. Los cálculos son pocos, pero es necesario hacer bastantes cálculos para el ordenamiento de las colas de reproducción, mayormente. También hay consideraciones en cuanto a velocidad y espacio que pueden ocupar las estructuras de datos.

1.5 Algoritmo para mostrar las funciones principales que debe realizar el programa.

El programa empieza por pedir al usuario la dirección del directorio donde se tienen guardados los archivos. Si existe, procede a procesar los archivos. Si todos son procesados sin errores, se procede a la parte principal del programa, donde el usuario tiene varias opciones para elegir y le permite acceder a las seis funciones principales descritas en el primer punto de esta documentación, cada una teniendo sus propias eventualidades. Finalmente, el programa es interactivo, por lo que persiste hasta que el usuario decide salir de él.

2 Documentación Específica

2.1 Estructura del Directorio

```
.
ArchivosDePrueba
  100 Greatest Hits of All Time.txt
  Abbey Road.txt
  At Night at the Opera.txt
  Back in Black.txt
  Dark Side of The Moon.txt
  ejemplo.txt
  Kid A.txt
  My Beautiful Dark Twisted Fantasy.txt
  Parachutes.txt
  Romours.txt
  Stankonia.txt
  The Marshall Matters LP.txt
  Thriller.txt
  Vicente Fernandez Para Siempre.txt
ClassDiagramProject1.dia
CMakeLists.txt
include
  CD.h
  List.h
  Queue.h
  Song.h
README.org
src
```

```
CD.cpp
main.cpp
main.cpp~
Queue.cpp
test
  CDTest.cpp
  CMakeLists.txt
  ListTest.cpp
  QueueTest.cpp
  SongT.cpp
```

2.2 Compilación

Utiliza sistema CMake. Tiene dos opciones: Release y Debug. El binario que genera tiene el nombre **repcd**. El directorio con archivos de prueba tiene varios archivos con errores. Es intencional, por lo que habría que generar un propio directorio solo con archivos funcionales.

2.3 Pruebas

Tiene cuatro unit tests que funcionan con CTest:

LsT Test de la Lista.

SnT Test de la Canción (Song).

CDT Test del CD.

QuT Test de la Cola (Queue).

Los cuatro son simples unit tests que deben de retornar 0 para asegurar su buen comportamiento ante cualquier cambio.