

Universidad Rafael Landívar
Facultad de Ingeniería
Ingeniería en Informática y Sistemas
Proyecto 01, programación avanzada, Sección 01
Catedrático: Ing. Méndez Soto, José Luis

PROYECTO 1

José Ricardo Guerra Morales - 1234123
Mauricio Enrique Cabrera Girón- 1073323

Guatemala, 01 de noviembre del 2023

INDICE

I.	INTRODUCCIÓN.....	1
II.	¿QUÉ ACCIONES DEBE REALIZAR EL PROGRAMA?.....	2
III.	¿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	3
IV.	¿QUÉ ESTRUCTURAS DE DATOS TRABAJARÁ PARA ALMACENAR LA INFORMACIÓN?.....	4
V.	¿QUÉ CONDICIONES O RESTRICCIONES DEBE TOMAR EN CUENTA? ¿QUÉ CÁLCULOS DEBE HACER?.....	5
VI.	ALGORITMO PARA MOSTRAR LAS FUNCIONES PRINCIPALES QUE DEBE REALIZAR EL PROGRAMA.....	6

I. INTRODUCCIÓN

El proyecto consiste en la creación de un reproductor de CDs que pueda reproducir música almacenada en archivos de texto con formato especial. Estos archivos contienen la información de las pistas. El reproductor brinda al usuario una opción para cargar todos los respaldos desde una carpeta que el mismo indica. Esta opción inicia consultando al usuario la ruta hacia una carpeta donde están todas las copias digitales y debe leer el contenido de cada archivo, cumpliendo las siguientes consideraciones:

- El reproductor reconoce los archivos con extensión .txt que se encuentren en la carpeta indicada por el usuario.
- El reproductor valida que el formato de los archivos sea el correcto, es decir, que contengan la información del CD, las pistas y las notas musicales.
- El reproductor almacena en memoria la información leída de cada archivo, para poder reproducir la música correspondiente.
- El reproductor muestra al usuario un menú con las opciones disponibles, tales como seleccionar un CD, reproducir una pista, pausar, reanudar, detener, etc.

El objetivo de este reporte es describir el proceso de diseño, implementación y prueba del reproductor de CDs, así como los resultados obtenidos y las conclusiones del proyecto. Se utilizó el lenguaje de programación c++ y las bibliotecas iostream, fstream, string, sstream, vector, algorithm y ctime para realizar el proyecto. En este documento se explicarán los algoritmos, las estructuras de datos y las funciones utilizadas para desarrollar el reproductor.

II. ¿QUÉ ACCIONES DEBE REALIZAR EL PROGRAMA?

Las acciones que realiza el programa son las siguientes:

1. Leer el documento: Esta acción lee un archivo de texto llamado "archivo.txt" y extrae la información de las canciones de cada línea. La información de la canción incluye el nombre, el artista y la duración de la canción. La información extraída se almacena en un vector de estructuras `Cancion`.
2. Ordenar el documento: Esta acción permite al usuario ordenar las canciones en el documento según diferentes criterios. El submenú de opciones de ordenamiento incluye ordenar por nombre, artista o duración. Las canciones se ordenan utilizando la función `std::sort` y la función de comparación adecuada.
3. Guardar el documento: Esta acción guarda las canciones ordenadas en un nuevo archivo de texto llamado "archivo_ordenado.txt". Las canciones se escriben en el mismo formato que el archivo de entrada, con cada canción encerrada entre llaves.
4. Reproducir música: Esta acción permite al usuario reproducir las canciones. El submenú de opciones de reproducción incluye reproducir aleatoriamente, seleccionar una canción específica, reproducir la siguiente canción, reproducir la canción anterior y pausar la reproducción.
5. Cerrar el programa: Esta acción termina el programa.

III. ¿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

El código trabajará con los siguientes datos:

1. **Archivo de entrada:** El código leerá un archivo de texto llamado "archivo.txt" que contiene información sobre canciones. Cada línea del archivo representa una canción y contiene el nombre de la canción, el artista y la duración de la canción.
2. **Datos de entrada del usuario:** El código solicitará al usuario que seleccione una opción del menú principal y del submenú de ordenamiento. También pedirá al usuario que ingrese el número de la canción seleccionada para reproducir o seleccionar una canción específica.
3. **Tipo de dato utilizado para los datos principales:** El código utiliza una estructura llamada Cancion para almacenar la información de cada canción. La estructura Cancion tiene tres miembros de tipo std::string: nombre, artista y duracion. Estos miembros almacenan el nombre de la canción, el nombre del artista y la duración de la canción, respectivamente.

IV. ¿QUÉ ESTRUCTURAS DE DATOS TRABAJARÁ PARA ALMACENAR LA INFORMACIÓN?

Para almacenar la información, el programa utilizará las siguientes estructuras de datos:

- Vector o array de estructuras Cancion para almacenar la información básica de cada canción: nombre, artista y duración. Esta estructura permitirá acceder y organizar fácilmente las canciones.
- Mapas o diccionarios asociativos para almacenar información adicional como la canción actualmente reproduciéndose, el número de la canción seleccionada, etc.
- Strings para almacenar el nombre de la canción, artista y duración. Utilizará esto como los miembros de la estructura Cancion.

El programa pedirá al usuario lo siguiente:

- Seleccionar una opción numérica del menú principal.
- Seleccionar una opción numérica del submenú de ordenamiento.
- Ingresar el número de la canción seleccionada para reproducir.

V. ¿QUÉ CONDICIONES O RESTRICCIONES DEBE TOMAR EN CUENTA? ¿QUÉ CÁLCULOS DEBE HACER?

El programa contiene las siguientes condiciones y restricciones:

- El programa requiere un archivo llamado "archivo.txt" en una ubicación específica. Este archivo contiene las canciones en un formato específico, donde cada línea representa una canción en el formato "{nombre||artista||duracion}".
- El programa muestra un menú con diferentes opciones para que el usuario elija. El usuario puede seleccionar una opción ingresando el número correspondiente.
- El programa utiliza las clases ifstream y ofstream para leer y escribir archivos.
- El programa utiliza un vector de estructuras Cancion para almacenar las canciones leídas del archivo.
- El programa incluye funciones para eliminar los corchetes de los nombres de las canciones, artistas y duraciones.
- El programa incluye un generador de números aleatorios para reproducir canciones de forma aleatoria.
- El programa permite al usuario seleccionar una canción para reproducir, reproducir la siguiente o anterior canción, pausar la canción actual y volver al menú principal.

Los cálculos realizados por el programa incluyen:

- Leer las canciones del archivo y almacenarlas en un vector.
- Ordenar las canciones según la elección del usuario de criterios (nombre, artista o duración).
- Escribir las canciones ordenadas en un nuevo archivo.
- Generar un número aleatorio para seleccionar una canción para la reproducción aleatoria.
- En general, el programa proporciona funcionalidad básica para administrar y reproducir canciones.

VI. ALGORITMO PARA MOSTRAR LAS FUNCIONES PRINCIPALES QUE DEBE REALIZAR EL PROGRAMA.

El algoritmo para mostrar las funciones principales que debe realizar el programa sería el siguiente:

- Leer el archivo de texto "archivo.txt" que contiene las canciones en formato {nombre||artista||duracion}.
- Al leer cada línea del archivo:
 - Extraer el nombre, artista y duración de la canción.
 - Eliminar los corchetes y comillas.
 - Almacenar la información de la canción en una estructura Canción.
 - Agregar la canción a un vector de canciones.
 - Mostrar la canción con su número, nombre, artista y duración.
 - Mostrar un menú de opciones al usuario y leer su selección.
- Si el usuario selecciona ordenar:
 - Mostrar un submenú para ordenar por nombre, artista o duración.
 - Ordenar el vector de canciones según la opción seleccionada.
 - Mostrar las canciones ordenadas.
- Si el usuario selecciona guardar:
 - Escribir todas las canciones en el vector en un nuevo archivo "archivo_ordenado.txt".
 - Mostrar un mensaje de que el archivo fue guardado.
- Si el usuario selecciona reproducir música:
 - Reproducir una canción de forma aleatoria.
 - Permitir seleccionar una canción
 - Reproducir la siguiente/anterior canción.
 - Poner en pausa la reproducción.