

Universidad Rafael Landívar

Facultad Ingeniería

Ingeniería en Informática y Sistemas

Ingeniero Joseph Abraham Soto Gutiérrez

Proyecto No. 1

“Reproductor de Música”

Harriett Alexandra Guzmán López

1112423

David Alconero Lepe

1216123

Guatemala, 7 de octubre del 2023

Tabla de contenido

INTRODUCCIÓN	III
ANALISIS.....	4
PARTE 1: CARGA DE RESPALDOS	4
<i>Entradas:</i>	4
<i>Proceso:</i>	4
<i>Salidas:</i>	4
PARTE 2: REPRODUCTOR DE CANCIONES	5
<i>Entradas:</i>	5
<i>Proceso (para cada acción):</i>	5
<i>Salidas (para cada acción):</i>	6
PREGUNTAS.....	7
DISEÑO.....	9
DIAGRAMAS DE FLUJO	9
DIAGRAMAS DE CLASES	10
CONCLUSIONES	12
RECOMENDACIONES.....	13
BIBLIOGRAFÍA.....	14

Introducción

La música es una parte fundamental de nuestras vidas, y en la era digital actual, tenemos la ventaja de almacenar y disfrutar de nuestras canciones favoritas de manera conveniente en nuestros dispositivos. Sin embargo, a medida que acumulamos una gran cantidad de música en nuestros discos duros, organizarla y acceder a ella de manera eficiente se vuelve un desafío.

Es en este contexto que surge la necesidad de una herramienta que simplifique la gestión y reproducción de nuestras colecciones de música digital. En este proyecto de programación, he desarrollado un programa que permite cargar copias digitales de CDs en formato TXT y proporciona una variedad de funciones para facilitar la gestión y reproducción de canciones.

Este programa se basa en el paradigma de programación orientada a objetos y ofrece características como la carga de respaldos, la creación de colas de reproducción personalizadas, la ordenación de canciones y la visualización de la información detallada de cada pista. A través de una interfaz amigable, los usuarios pueden interactuar con sus bibliotecas de música de manera eficiente y disfrutar de sus canciones favoritas en cualquier momento.

A lo largo de este proyecto, exploraremos en detalle las funciones y el funcionamiento del programa, así como su capacidad para manejar y organizar nuestra música digital de manera efectiva.

Analysis

Parte 1: Carga de Respaldos

Entradas:

Ruta de la carpeta donde están almacenados los archivos TXT que contienen las copias digitales de los CDs.

Proceso:

- ☐ El programa solicita al usuario que ingrese la ruta de la carpeta.
- ☐ El programa accede a la carpeta especificada y lee cada archivo TXT que representa una copia digital de un CD.

Para cada archivo:

- ☐ Se verifica que no esté vacío.
- ☐ Se procesa cada línea del archivo, verificando que cumple con el formato especificado ({nombre} | | {artista} | | {duración}).
- ☐ Si todas las líneas del archivo cumplen con el formato, se crea un objeto CD y se almacena en una lista de CDs. Se registra el nombre del CD y la cantidad de canciones leídas.
- ☐ Si el archivo contiene errores, se muestra un mensaje de error que indica el nombre del archivo con el problema y el tipo de problema (archivo vacío o formato de línea incorrecto).
- ☐ Después de cargar todos los archivos correctamente, se muestra un mensaje de confirmación.

Salidas:

- ☐ Para cada archivo cargado correctamente:
- ☐ Nombre del CD.
- ☐ Cantidad de canciones leídas.

- ☐ Para cada archivo con errores:
- ☐ Mensaje de error que indica el nombre del archivo con el problema y el tipo de problema identificado (archivo vacío o formato de línea incorrecto).
- ☐ Mensaje de confirmación de carga exitosa.

Parte 2: Reproductor de Canciones

Entradas:

El usuario puede seleccionar una de las siguientes acciones principales:

- ☐ "Agregar Canción": elige un CD y una canción.
- ☐ "Ver Cola de Reproducción": decide si ver la cola tal como está o seleccionar una opción de ordenación.
- ☐ "Reproducción Actual": muestra la canción que se está reproduciendo.
- ☐ "Reproducir Siguiente": avanza a la siguiente canción en la cola.
- ☐ "Ordenar": selecciona un criterio de ordenación para la cola.

Proceso (para cada acción):

"Agregar Canción":

- ☐ Muestra la lista de CDs para que el usuario elija uno.
- ☐ Si se selecciona un CD, muestra las canciones del CD seleccionado y solicita al usuario que elija una canción.
- ☐ La canción seleccionada se agrega a la cola de reproducción.

"Ver Cola de Reproducción":

Consulta al usuario si desea ver la cola de reproducción actual o si desea que se imprima ordenada por nombre del artista, nombre de la canción o duración de la canción.

"Reproducción Actual":

- ☐ Muestra si hay una canción siendo reproducida en este momento o si la reproducción está en pausa.
- ☐ Si hay una canción en reproducción, muestra información completa de la canción.

"Reproducir Siguiente":

- ☐ La canción en "Reproducción Actual" se inserta nuevamente al final de la cola.
- ☐ La canción que está al principio de la cola se remueve de la cola y se muestra su información como la nueva canción en reproducción.

"Ordenar":

- ☐ Permite al usuario cambiar el orden de la cola de reproducción de forma permanente.
- ☐ Las opciones de ordenación incluyen: nombre del artista, nombre de la canción y duración de la canción.

Salidas (para cada acción):

Dependiendo de la acción seleccionada por el usuario, se realizará una de las siguientes salidas, que incluirán información sobre el estado de la cola de reproducción y detalles sobre las canciones en reproducción o las acciones específicas realizadas.

Preguntas

¿Qué acciones debe poder hacer su programa? Enumérelas.

- ☐ Cargar Respaldos: Permitir al usuario cargar respaldos de CDs desde una carpeta especificada.
- ☐ Agregar Canción a la Cola de Reproducción: Permite al usuario seleccionar un CD y una canción para agregar a la cola de reproducción.
- ☐ Ver Cola de Reproducción: Mostrar la cola de reproducción actual o permitir al usuario ordenarla por diferentes criterios.
- ☐ Reproducción Actual: Mostrar la canción que se está reproduciendo actualmente o indicar que la reproducción está en pausa.
- ☐ Reproducir Siguiente: Avanzar a la siguiente canción en la cola de reproducción.
- ☐ Ordenar la Cola de Reproducción: Permitir al usuario cambiar el orden de la cola de reproducción por nombre del artista, nombre de la canción o duración.

¿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.

- ☐ Archivos TXT que representan copias digitales de CDs.
- ☐ Información sobre CDs, que incluye el nombre del CD y una lista de canciones.
- ☐ Información sobre canciones, que incluye el nombre de la canción, el nombre del artista y la duración de la canción.

Datos de Entrada y Tipos de Datos:

- Ruta de la carpeta que contiene los archivos de respaldo de CDs (cadena de texto).
- Selecciones del usuario al realizar acciones en el reproductor de canciones (entero o cadena de texto según el caso), como la elección de un CD, una canción o un criterio de ordenación.

¿Qué estructuras de datos trabajará para almacenar la información?

- Lista de CDs: Una lista que contiene objetos CD, donde cada CD tiene un nombre y una lista de canciones.
- Lista de Canciones: Cada CD contiene una lista de objetos de canción, donde cada canción tiene un nombre, el nombre del artista y la duración.

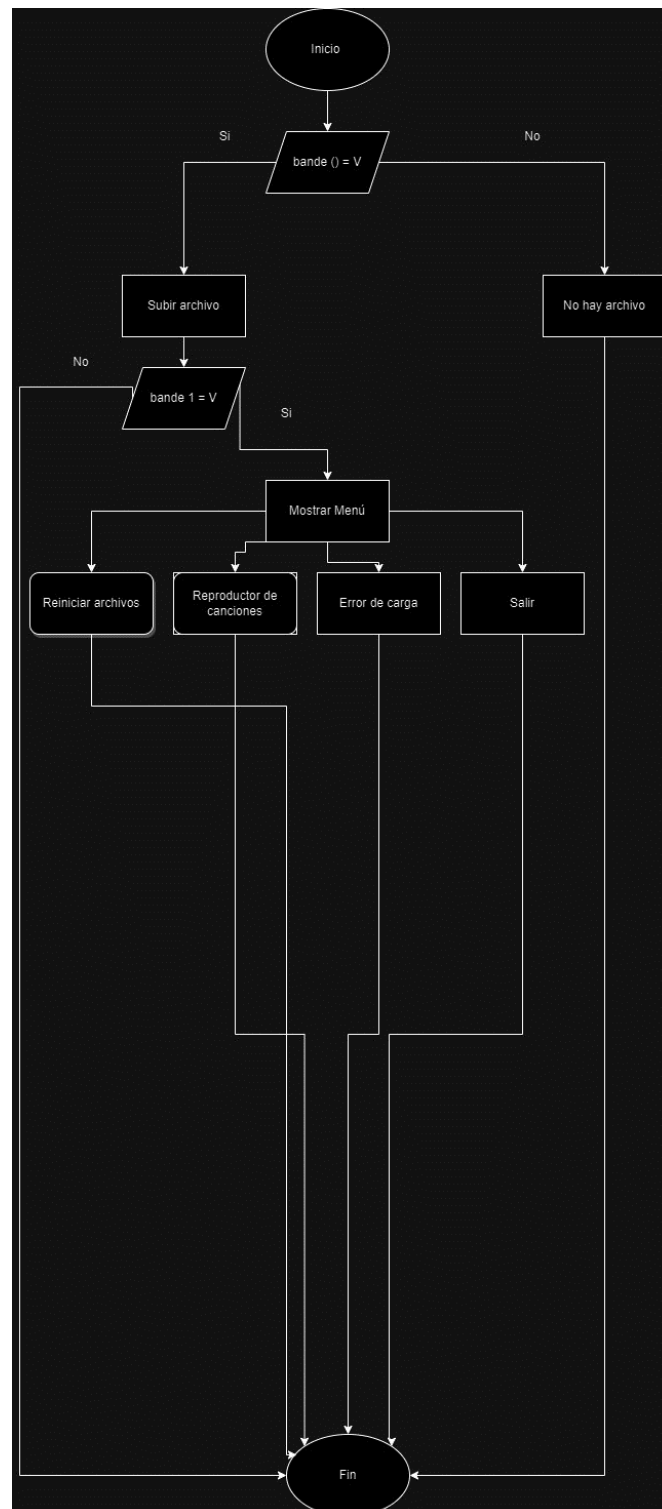
¿Qué condiciones o restricciones debe tomar en cuenta? ¿Qué cálculos debe hacer?

- Validación de formato: El programa debe verificar que los archivos de respaldo tengan el formato especificado (nombre | | artista | | duración).
- Errores en archivos: El programa debe manejar archivos vacíos o líneas con un formato incorrecto.
- Ordenación ascendente o descendente: El programa debe permitir al usuario especificar si desea ordenar la cola de reproducción en orden ascendente o descendente según el criterio seleccionado.
- El programa no realiza cálculos complejos, pero reorganiza y muestra la información de las canciones y los CDs según las acciones del usuario.

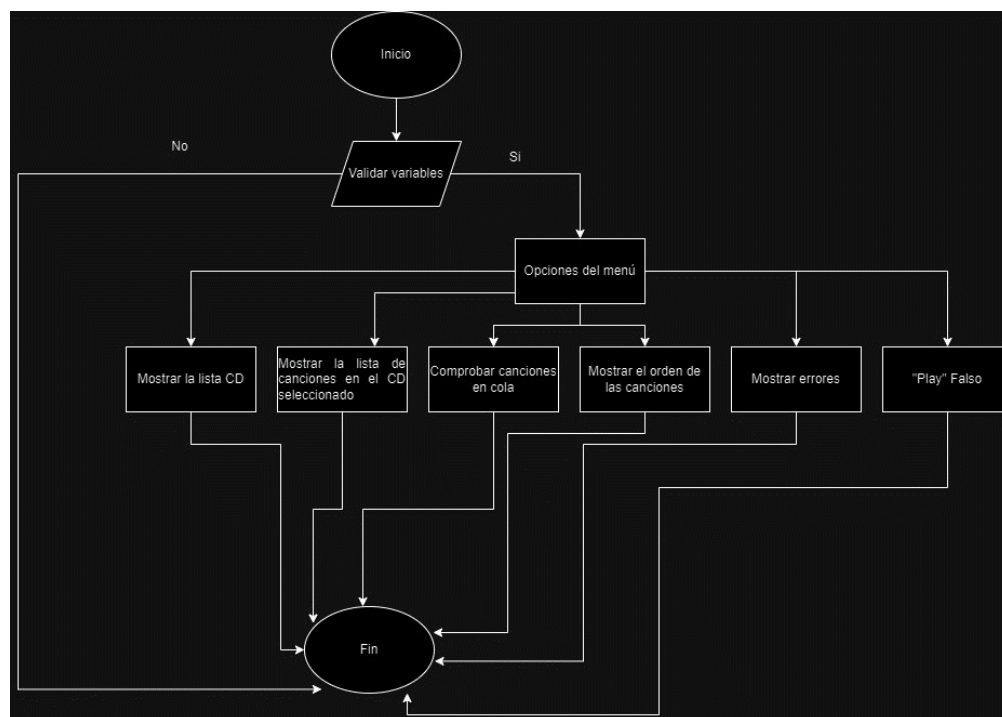
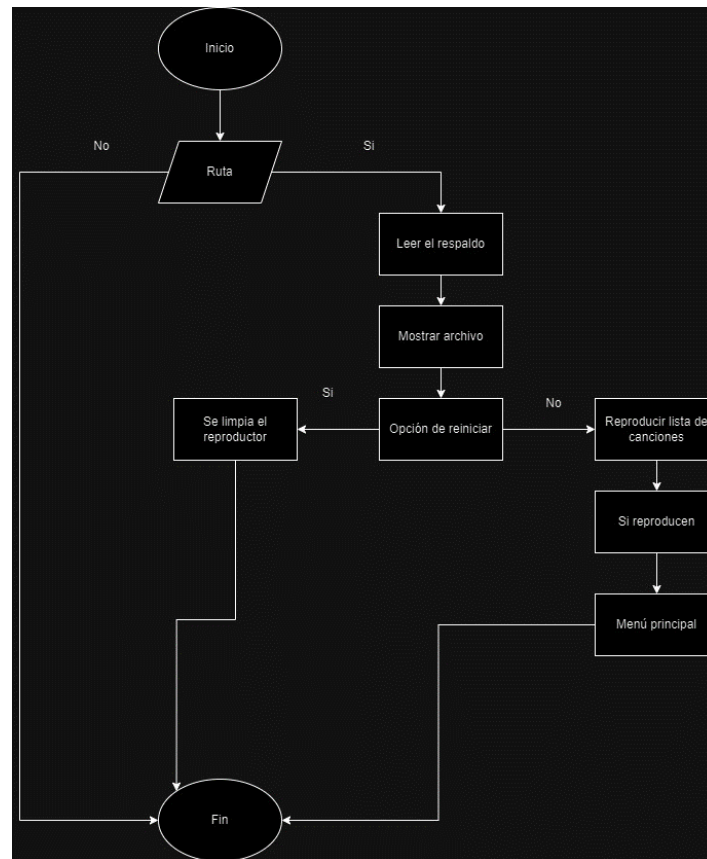
Algoritmo para mostrar las funciones principales que debe realizar el programa.

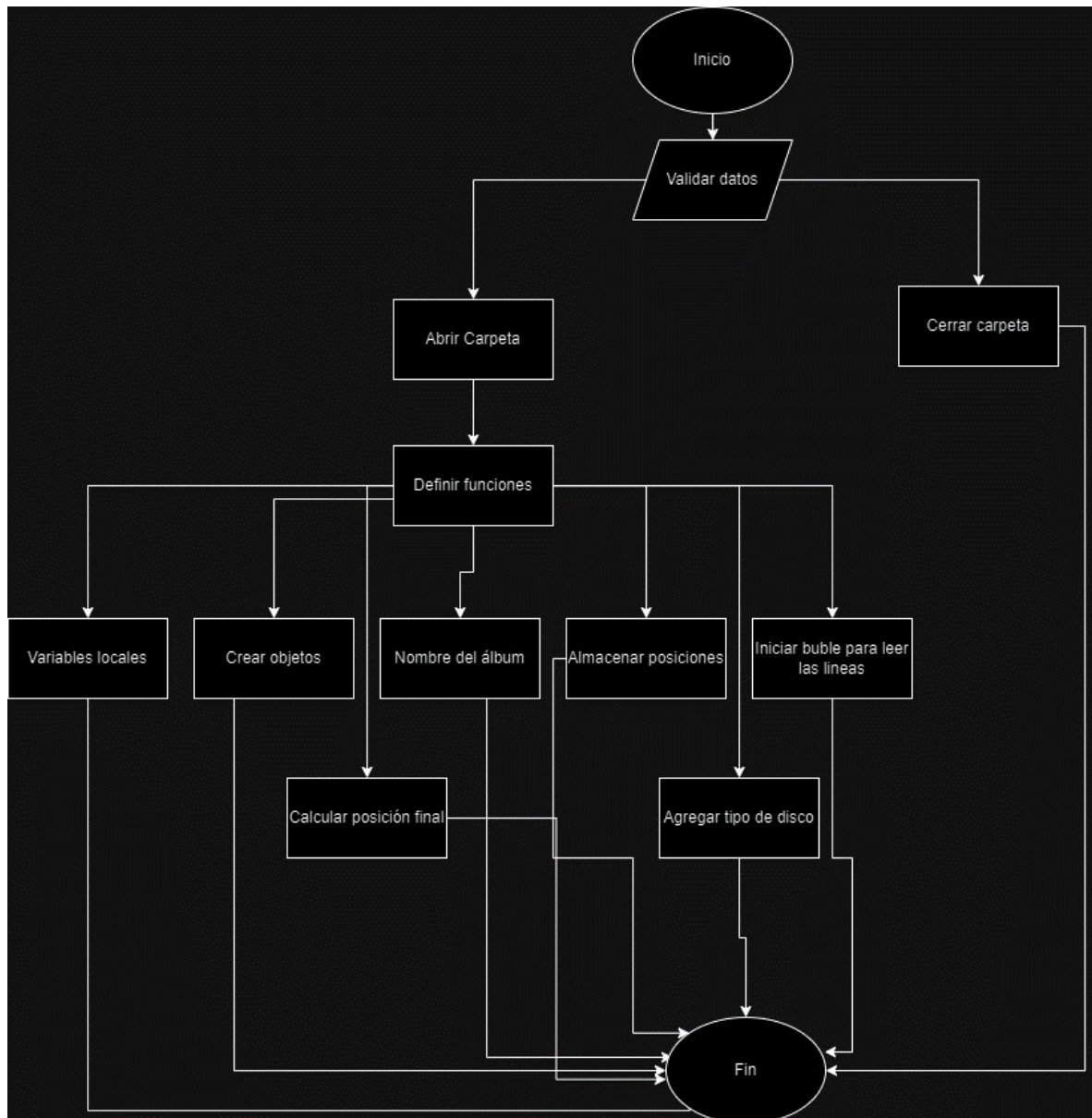
Diseño

Diagramas de flujo



Diagramas de clases





Conclusiones

- Desarrollar un programa de gestión y reproducción de CDs ha sido un proyecto educativo valioso. A lo largo de este proceso, he adquirido una comprensión más profunda de la programación y he aplicado los conceptos aprendidos en mis estudios.
- Algunas de las lecciones clave que he aprendido incluyen la importancia de la planificación y el diseño previo, la necesidad de mantener el código limpio y documentado, y la importancia de las pruebas rigurosas para garantizar un software de calidad. También he experimentado la satisfacción de ver cómo un proyecto evoluciona desde una idea hasta una aplicación funcional.
- Las pruebas exhaustivas y la depuración son partes esenciales del desarrollo de software. He aprendido la importancia de realizar pruebas sistemáticas para identificar y corregir errores. Esto garantiza que el software funcione de manera confiable y cumpla con las expectativas del usuario.

Recomendaciones

- Antes de comenzar a programar, es fundamental planificar y diseñar detenidamente el programa. Esto incluye la definición de los objetivos, las funcionalidades clave y la estructura general del programa. Un buen diseño previo puede ahorrar mucho tiempo y evitar problemas más adelante.
- La programación es un campo en constante evolución. Recomiendo mantenerse actualizado con las nuevas tecnologías, lenguajes de programación y mejores prácticas. La educación continua es esencial para crecer como programador.
- Es importante comentar y documentar tu código de manera adecuada. Esto no solo facilita la comprensión de tu código por parte de otros, sino que también te ayudará a ti mismo a recordar cómo funciona cuando vuelvas a él más adelante.
- Antes de dar por terminado un proyecto, realiza pruebas exhaustivas para identificar y solucionar errores. Las pruebas de unidades y las pruebas de integración son esenciales para garantizar que el programa funcione como se espera.
- Aprovecha la comunidad de programadores en línea y los foros de ayuda. La colaboración con otros estudiantes y programadores puede brindarte nuevas perspectivas y soluciones a los desafíos que enfrentas.

Bibliografía

Codes Cracker. (s.f.). Obtenido de
 <https://codescracker.com/cpp/program/cpp-program-list-files-in-directory.htm>

Stroustrup (1985) describió en su libro "The C++ Programming Language"

"Effective C++" (Meyers, 1992)

Sutter y Alexandrescu (2004)

Josuttis (2012) presentó un tutorial y referencia exhaustivos sobre la Biblioteca Estándar de C++ en su libro "The C++ Standard Library"