IPCmusic – PROYECTO1

202200110– Evelyn Maricely Balcarcel Rivera
Ruiz 202111530 Emily Flor de Maria Cordon
202000648 Helmouts David Gomez Chiquin
202201457 Pakal B'alam Rodriguez Espantzay
201709035 Rocío Samaí López Vásquez

Resumen

Las listas enlazadas son estructuras de datos esenciales compuestas por nodos. Estos nodos forman una secuencia que almacena datos de forma flexible y se accede a ellos mediante el uso de punteros. Estos punteros pueden apuntar hacia el nodo anterior o siguiente, dependiendo de la naturaleza de la lista enlazada.

Una lista de listas, también conocida como lista anidada, es una estructura de datos donde cada elemento de una lista es, a su vez, otra lista. Esto permite crear estructuras de datos multidimensionales o representar relaciones entre elementos de manera jerárquica. Cada elemento de la lista principal es una lista en sí misma. Puedes acceder a elementos individuales utilizando índices tanto en la lista principal como en las listas anidadas.

Palabras clave

- Listas enlazadas
- POO
- Graphiz
- Tkinter
-PySimpleGUI

Abstract

Linked lists are essential data structures composed of nodes. These nodes form a sequence that stores data flexibly and is accessed using pointers. These pointers can point to the previous or next node, depending on the nature of the linked list.

A list of lists, also known as a nested list, is a data structure where each element of a list is, in turn, another list. This allows for the creation of multidimensional data structures or representing relationships between elements hierarchically. Each element of the main list is a list itself. You can access individual elements using indices both in the main list and in the nested lists.

Keywords

Linked Lists

-OOP (Object-Oriented Programming)

- Graphviz

- Tkinter

-PySimpleGUI

Introducción

Conociendo la importancia de diversas herramientas para la solución de problemas y aplicaciones en el lenguaje de programación Python, podemos adentrarnos a un mundo nuevo y lleno de incógnitas que mediante investigaciones y sobre todo mucha práctica pueden ser respondidas de una manera muy dócil, basándonos en el problema general, nos encontramos en una situación donde se necesita realizar una aplicación de escritorio IPCmusic que consiste en un reproductor de música el cual debe contar con una interfaz de usuario amigable e intuitiva que permitirá al usuario ordenar la música que encuentra en su ordenador, así como mostrar estadísticas con relación a sus canciones.

Desarrollo del tema

Al inicio puede ser algo confuso por el amplio campo dentro de un mundo nuevo lleno de incógnitas, pero mediante el adentramiento a este “nuevo mundo” llamado programación podemos darnos cuenta que siempre se aprenden nuevas cosas y nuevas habilidades, que seguramente antes no teníamos. Hablando un poco del problema central de la aplicación, necesitamos tener 2 cosas en la mente, la primera es que tiene que ser una aplicación amigable para el usuario, ya que la aplicación o reproductor de música debe contar con una interfaz interactiva y fácil de utilizar, porque se debe de poder visualizar la canción que se está reproduciendo con su respectivo artista y al álbum que pertenece en conjunto con su imagen. Así también, el reproductor debe de ser capaz de pausar una canción o detenerla por completo o bien adelantar o regresar, mientras más fácil sea de utilizar para el usuario, será mucho mejor la repercusión y aceptación que pueda tener el proyecto. Al implementar librerías de Python como los son Tkinter, Graphiz, PySimpleGUI, PIL; debemos saber que son y cómo funcionan. Tkinter es una biblioteca estándar de Python que proporciona herramientas para crear interfaces gráficas de usuario. Con Tkinter, puedes diseñar ventanas, botones, cuadros de texto y otros elementos de la interfaz gráfica de usuario para tus aplicaciones. Tkinter utiliza una interfaz con el kit de herramientas de ventanas Tcl/Tk (de ahí el nombre "Tkinter"). Permite que los desarrolladores creen ventanas y widgets, definan eventos y manejen interacciones de usuario en aplicaciones GUI. Por otro lado, Graphviz es una colección de herramientas y bibliotecas utilizadas para visualizar gráficos y diagramas. Puede generar gráficos dirigidos y no dirigidos, árboles, diagramas de flujo y más a partir de datos en formato DOT (lenguaje de descripción de gráficos). Graphviz toma una descripción en formato DOT y genera una representación visual del gráfico especificado. Puede generar imágenes estáticas (como PNG, SVG) o incluso representaciones interactivas utilizando la herramienta dot. PySimpleGUI: es una biblioteca de Python que permite a los programadores de Python de todos los niveles crear interfaces gráficas de usuario (GUI). Transforma tkinter, Qt, WxPython y Remi en interfaces Pythonicas amigables para el usuario. Tu código PySimpleGUI es más simple y corto que escribir directamente usando el marco subyacente porque PySimpleGUI implementa gran parte del “código repetitivo” por ti.

PIL (Python Imaging Library): Es una biblioteca adicional gratuita y de código abierto para el lenguaje de programación Python que agrega soporte para abrir, manipular y guardar muchos formatos de archivos de imagen diferentes. Esta biblioteca proporciona un soporte de formato de archivo extenso

Tabla I.

Librería para la escritura "XML".

IMPORT	AS
Xml.etree.ElementTree	ET

Tabla II.

Librería para la interfaz gráfica.

IMPORT	AS
PySimpleGUI	sg

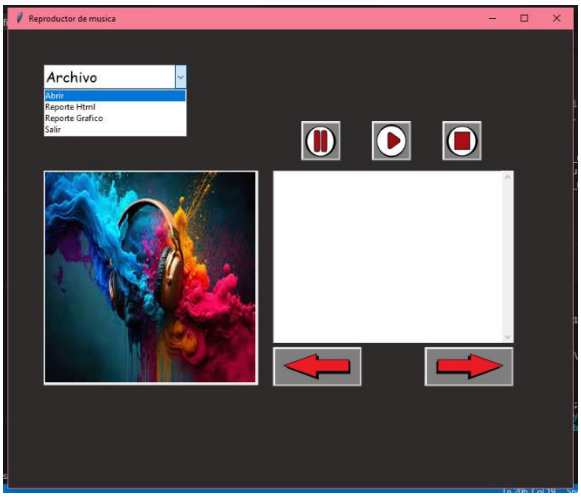
Tabla III.

Librería para manipular imágenes.

IMPORT	Import
Pil	Image, ImageTk

Para la estructura de la aplicación se muestra un posible prototipo de la interfaz y posteriormente la explicación de cada una de sus funcionalidades.

Figura 1. Interfaz - Archivo



Fuente: Elaboración propia

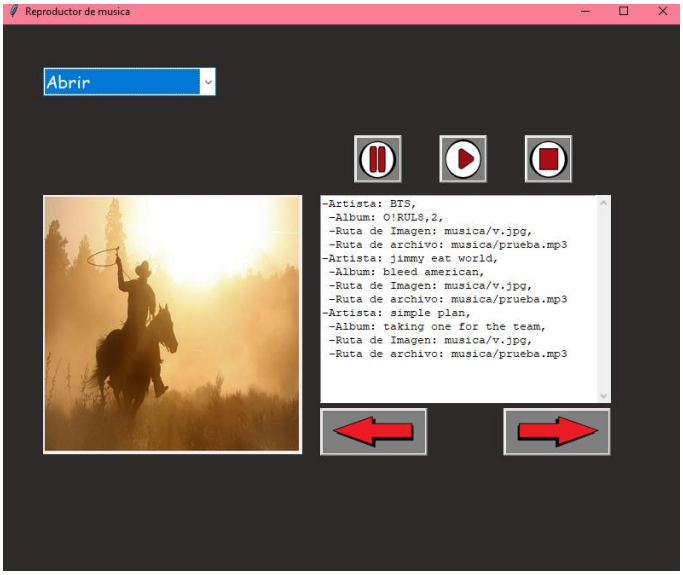
Abrir: Este botón permite al usuario seleccionar un archivo XML utilizando el cuadro de diálogo de apertura de archivos. Después de seleccionar el archivo, el programa carga el contenido del archivo y lo muestra en un cuadro de texto en la interfaz gráfica.

Reporte Html: Este botón genera un reporte HTML utilizando la biblioteca reporteHTML. El reporte contiene información sobre las canciones cargadas desde el archivo XML y se guarda en un archivo llamado "reporte_HTML.html".

Reporte Grafico: Este botón genera un gráfico utilizando la biblioteca graphviz. El gráfico representa la lista enlazada doble de canciones cargadas desde el archivo XML.

Salir: Este botón cierra la aplicación.

Figura 2. Interfaz – Cuadro de texto



Fuente: Elaboración propia

Un cuadro de texto es un elemento utilizado para mostrar y recopilar datos en un formato de texto. Puede ser utilizado en diferentes contextos, como en aplicaciones web, formularios electrónicos o documentos de texto.

En el caso de mostrar datos de un archivo XML en un cuadro de texto, es necesario realizar una serie de pasos. Primero, se debe acceder al archivo XML y extraer la información deseada. Luego, se puede utilizar un lenguaje de programación o una herramienta específica para mostrar los datos en el cuadro de texto.

Figura 3 . Interfaz – HTML

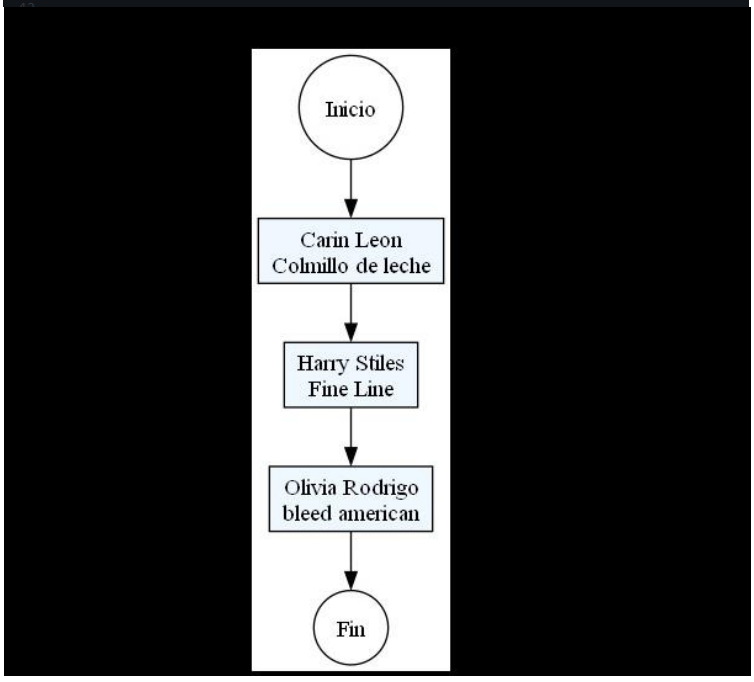
```python
#reporte html
def generar_reporte_html():
    reporte.html('reporte_HTML.html',lista_externa_temp)
    messagebox.showinfo("Generar Reporte", "Reporte de html generado.")
```

Fuente: Elaboración propia

La función generar_reporte_html utiliza la biblioteca reporte para crear un informe HTML. Se llama a la función html de la biblioteca reporte, pasando como argumentos el nombre del archivo de salida ('reporte_HTML.html') y la lista lista_externa_temp. Este método probablemente crea un informe HTML basado en la información contenida en la lista lista_externa_temp. Además, muestra un cuadro de información con el mensaje "Reporte de HTML generado" utilizando messagebox.showinfo.

Figura 4. Interfaz – Graphviz

```python
def generar_grafico_lista(lista):
    dot = Digraph(comment='Lista Enlazada Doble', format='png')

    # Configuración de nodos y bordes
    dot.node('Inicio', label='Inicio', shape='circle')
    dot.node('Fin', label='Fin', shape='circle')

    # Recorrido de la lista para agregar nodos y enlaces
    nodo_actual = lista.primero
    while nodo_actual is not None: ...

    dot.edge(f'Nodo_{lista.ultimo.dato.artista}_{lista.ultimo.dato.album}',
             'Fin') if lista.ultimo is not None else None

    # Guardar el gráfico como imagen
    dot.render('lista_grafico', format='png', cleanup=True)


    generar_grafico_lista(lista_externa_temp)
```



Fuente: Elaboración propia

La función generar_grafico_lista crea un gráfico utilizando la biblioteca Graphviz. Se inicializa un objeto Digraph llamado dot con el comentario 'Lista Enlazada Doble' y el formato de imagen 'png'. Luego, se configuran nodos de inicio y fin con etiquetas 'Inicio' y 'Fin', respectivamente.

Se recorre la lista doblemente enlazada lista y se agregan nodos con etiquetas basadas en la información de cada elemento en la lista. Los nodos se conectan con bordes que representan la relación anterior-siguiente en la lista. Finalmente, se guarda el gráfico como una imagen PNG.

Conclusiones

Con la realización de este proyecto se ha demostrado cómo diversas disciplinas, como la POO, Listas, XML, tkinter y Graphviz, colaboran de manera efectiva al momento de la creación de la app de música. Cada una de estas disciplinas aporta su experiencia y herramientas específicas para crear una nueva tecnología que permita el manejo de archivos xml para generar musica.

la interfaz de usuario se crea utilizando Tkinter, PySimpleGUI y bibliotecas adicionales que se utilizan para procesar archivos XML y que el usuario pueda reproducir canciones.

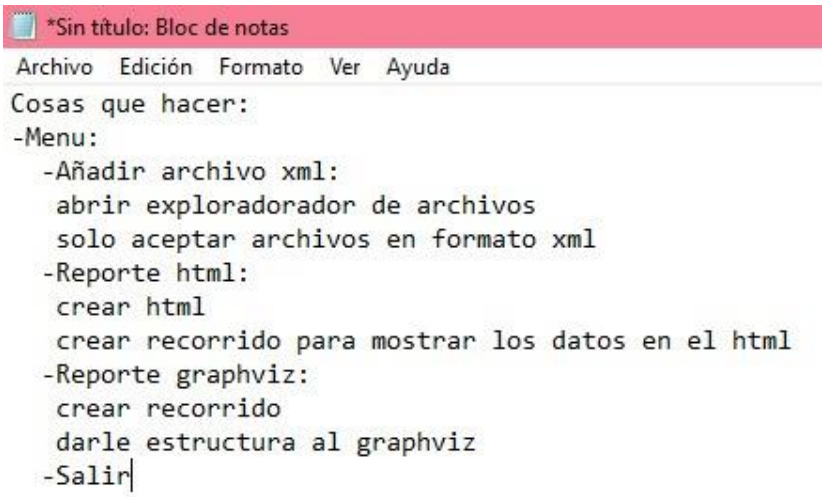
Referencias bibliográficas

https://www.youtube.com/watch?v=hTUJC8HsC2I
how/extension-de-archivo-xml-
https://graphviz.org/download/

Anexos

Forma en la que planteamos en proyecto1 para su respectiva solución.

```
interfaz grafica

listas doblemente enlazadas (pausar, detener la canción, regresar o adelantarla)

cargar biblioteca a traves de un archivo XML

Crear listas de reproducciónn seleccionando los datos de la biblioteca y reproducirlas utilizando estructuras de datos lineales

Reporte de las canciones y artistas más reproducidos en HTMK

Reporte de la biblioteca a traves de Graphviz

Guardar las listas de reproducción

Documentación
```



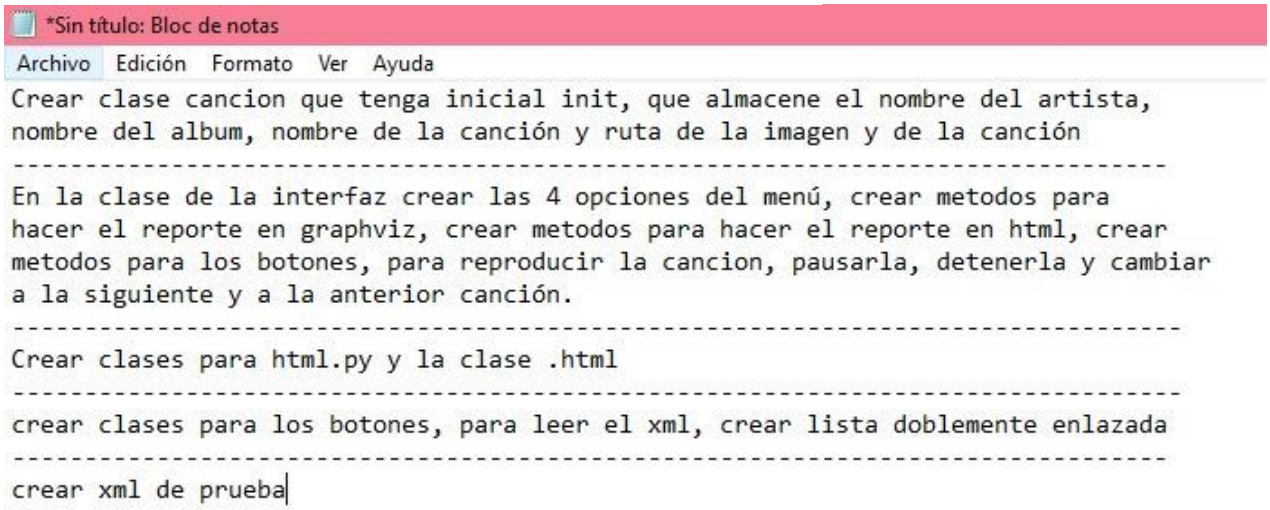
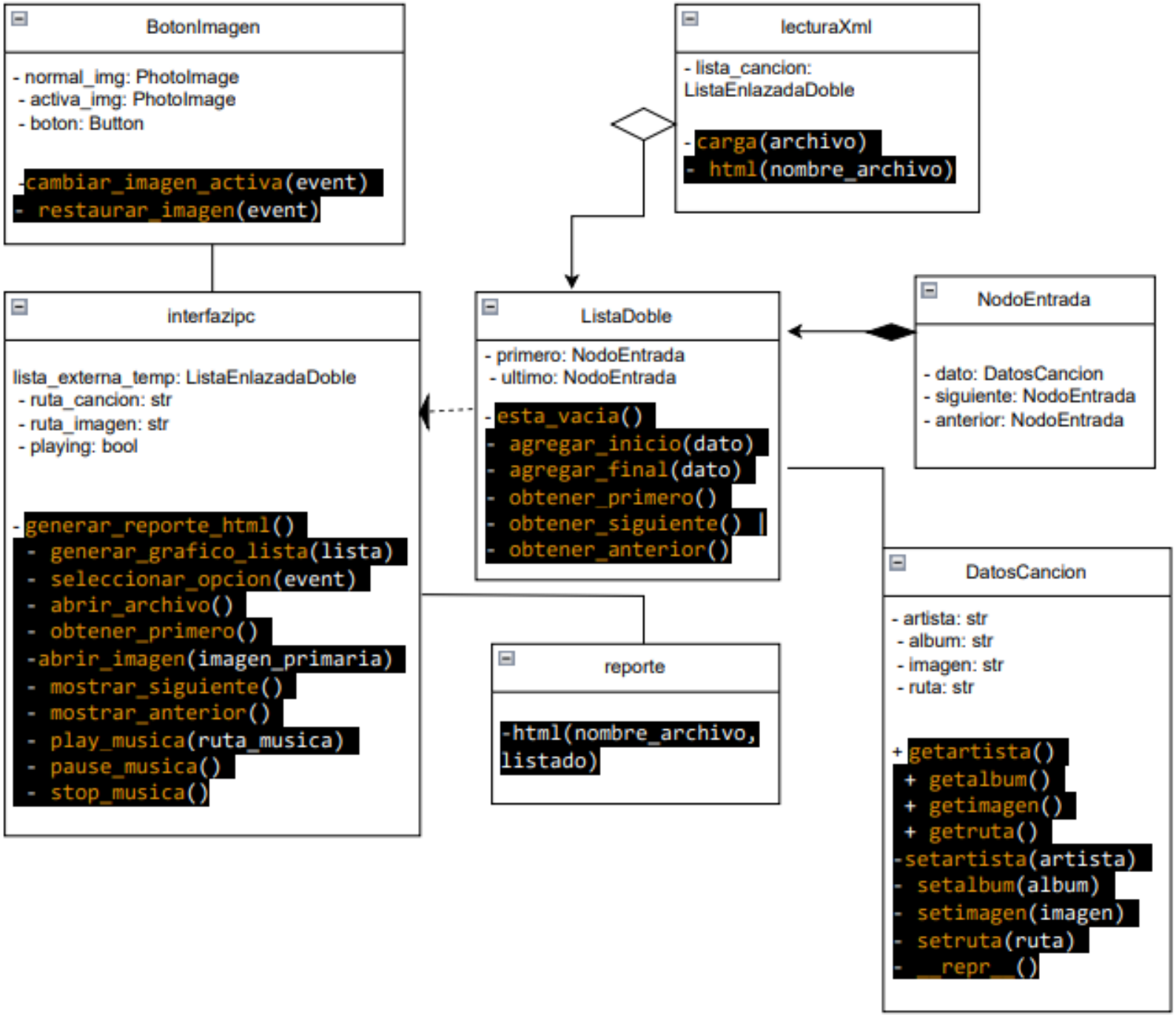


Diagrama de clases



Fuente: Elaboración propia, 2023