| **PROYECTO** |
| --- |
| **201801539 – Brenda Paola Gramajo Paniagua**  **202100692- Josue Ricardo Carias Ordoñez**  **201900081 - José Luis Saloj Julajuj** |

**Resumen**

El proyecto realizado se basó en la búsqueda de una solución al problema que se planteó, mediante la lógica y conocimientos que se han adquirido durante la clase y que han sido aplicados en el laboratorio. Este problema consistió en la elaboración de un reproductor de música. El programa fue desarrollado en el lenguaje de programación de Python y se utilizaron Listas Enlazadas para la creación del algoritmo y almacenamiento de datos. El problema planteado representa funciones que se pueden observar en el medio en el que vivimos, y la solución se puede utilizar como recurso de optimización en los casos similares, al tener un programa que realice tareas como estas, se mejora el entorno de trabajo y, por lo tanto, la producción. La solución planteada es la muestra de conocimientos adquiridos y es preparación para los problemas que se puedan presentar en un futuro.

**Palabras clave**

Listas, Nodo, Música, Canciones, Circular

***Abstract***

*The project carried out was based on the search for a solution to the problem that was raised, through the logic and knowledge that have been acquired during the class and that have been applied in the laboratory. This problem consisted of the development of a console program that will consist of implementing a grouping methodology for the solution to achieve compressing audio signals. The program was developed in the Python programming language and linked lists were used for algorithm creation and data storage. The problem posed represents functions that can be observed in the environment in which we live, and the solution can be used as a resource for optimization in similar cases, having a program that performs tasks like these, improves the working environment and therefore, the production. The proposed solution is a sample of acquired knowledge and a preparation for future problems.*

***Keywords***

*Lists, Node, Music, Songs, Time*

**Introducción**

El proyecto por realizar describe la solución para construir un programa en lenguaje Python que consiste en un reproductor de música el cual cuenta 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. El programa realizado podrá realizar varias operaciones como crear las listas de reproducción que desee, podrá seleccionar las canciones por agregar para luego darle un nombre a la lista, cargar un archivo XML que contenga la información de la biblioteca, entre otras operaciones. Una de las principales funciones es la carga de archivos XML, así como el uso de estructuras Lineales creadas personalmente para el almacenamiento de la información, esto implementando nodos y listas simples enlazadas y listas circularmente enlazadas. También se implementaron soluciones de forma gráfica con la herramienta de graphviz y por último se implementaron reportes en formato HTML. Al combinar todas estas herramientas se logró implementar y dar una solución al problema planteado.

**Desarrollo del tema**

El desarrollo del proyecto consistió en dos partes importantes, la primera parte se desarrolló la lectura del archivo de entrada XML, y la extracción de los datos contenidos en el documento. Toda la segunda parte se centró en la creación de las respectivas clases para trabajar con la POO y los TDA´s necesarios para brindarle una solución al problema.

Se creó una clase “Canciones” donde se almacenarían los atributos de cada una de los canciones cargados en el sistema, como la lista de instrucciones para emitir el puntero, que almacenaba los atributos respectivos de cada Sistema individual, así como su lista de caracteres.

También se implementó para la solución la creación de una clase Nodo, en la que se tenían los atributos necesarios para la creación de varios Nodos que se incluirían en la clase de Lista Enlazada y una lista Doblemente Enlazada. La clase de Lista enlazada contenía las funciones necesarias para la creación de las listas, como puede ser la función insertar, buscar, recorrer, etc. Se crearon los respectivos objetos ‘Mensaje’ extrayendo los datos del archivo XML brindado por el usuario y se almacenaron en el respectivo Nodo de la Lista Enlazada. Con esto se pudieron crear funciones como lo son el insertar nodos y el recorrer. Para la función de graficar, se utilizó la librería de Python llamada Graphviz. Esta se implementó como una solución practica para graficar los sistemas de drones cargados y la serie de instrucciones para emitir los mensajes. El proyecto incluye un menú en consola agradable al usuario. Este tiene funciones como lo son:

1. Cargar Archivo:

Función que permite cargar al usuario el archivo XML solicitando la ruta del archivo.

2. Generar XML de salida:

Genera un archivo XML de salida con las lista de cancion escuchas

3. Gestionar canciones:

Muestra gráficamente el listado de canciones.

4. Gestión de ListaDeCanciones:

Esta opción permite ver el listado de canciones

5. Inicializar Sistema:

Esta opción permite que el sistema pueda inicializarse sin información previa.

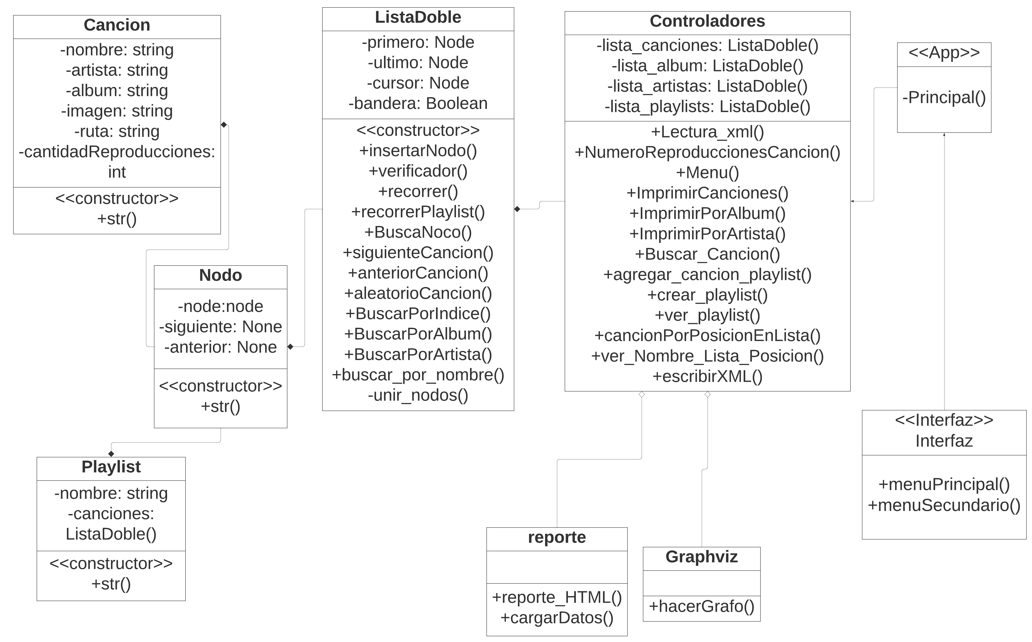
6. Salir:

Termina la ejecución del programa

*Figura 1. Menú en consola.*

*Fuente: elaboración propia.*

El programa se visualiza a través de una interfaz creada con Tkinter además se desarrolló con base de Programación Orientada a Objetos y la aplicación de las Listas Enlazadas Dobles y Circulares.



*Figura 2. Diagrama de Clases.*

*Fuente: elaboración propia.*

**Conclusiones**

· La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Python es un lenguaje orientado a objetos.

· Las listas enlazadas funcionan mediante la conexión de nodos en una secuencia lineal, donde cada nodo contiene un valor y una referencia (puntero) al siguiente nodo en la secuencia. Estas listas son dinámicas y eficiente.

La memoria dinámica se refiere a la asignación y liberación de memoria en tiempo de ejecución según sea necesario esto genera que sea eficiente y efectiva para el almacenamiento.

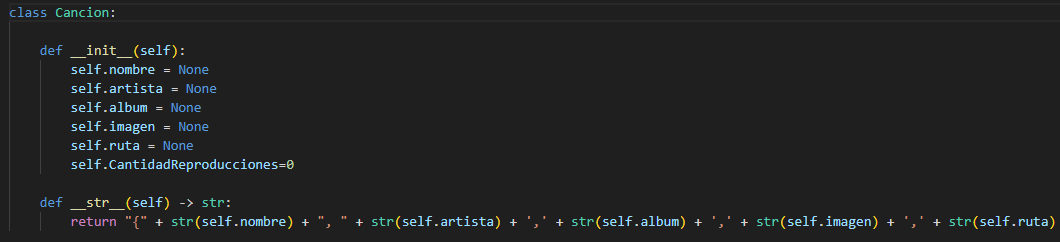
· Los archivos XML implementan una forma de almacenar datos de manera legible y jerárquica y pueden albergar diferentes tipos de datos.

**Anexos**

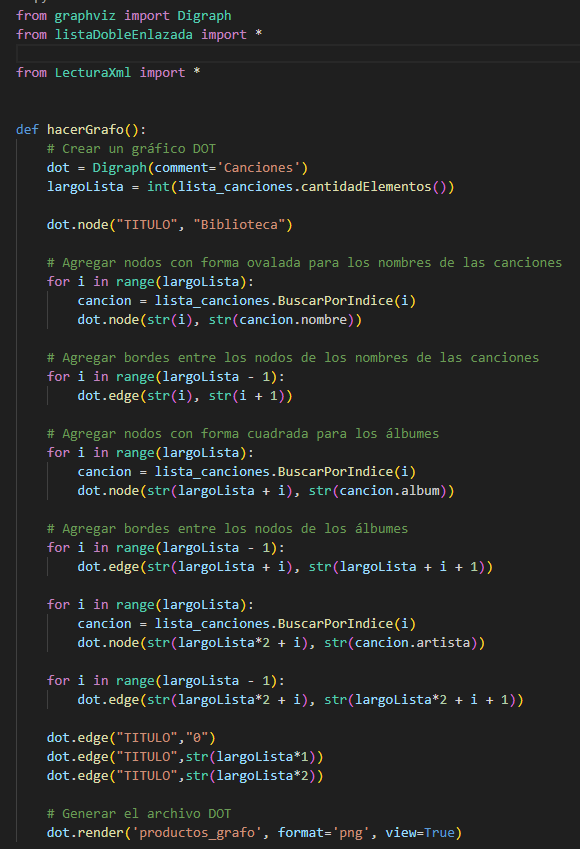
**Clase app.py:**

****

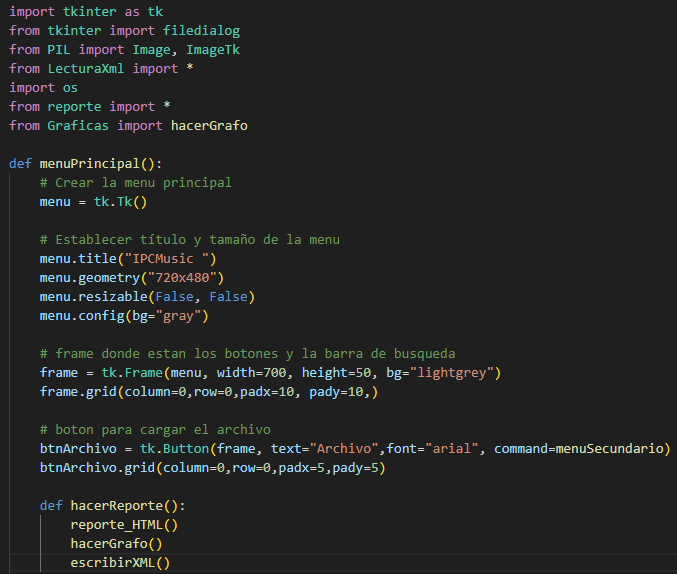
**Clase cancion.py:**

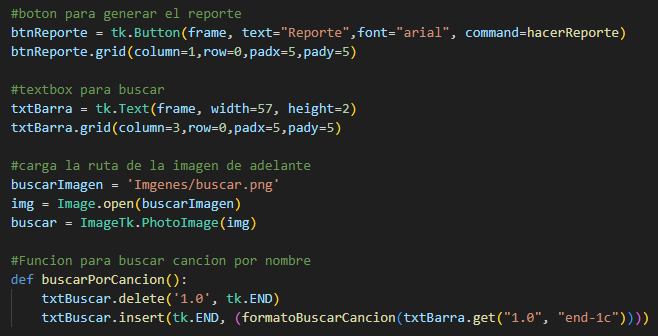
****

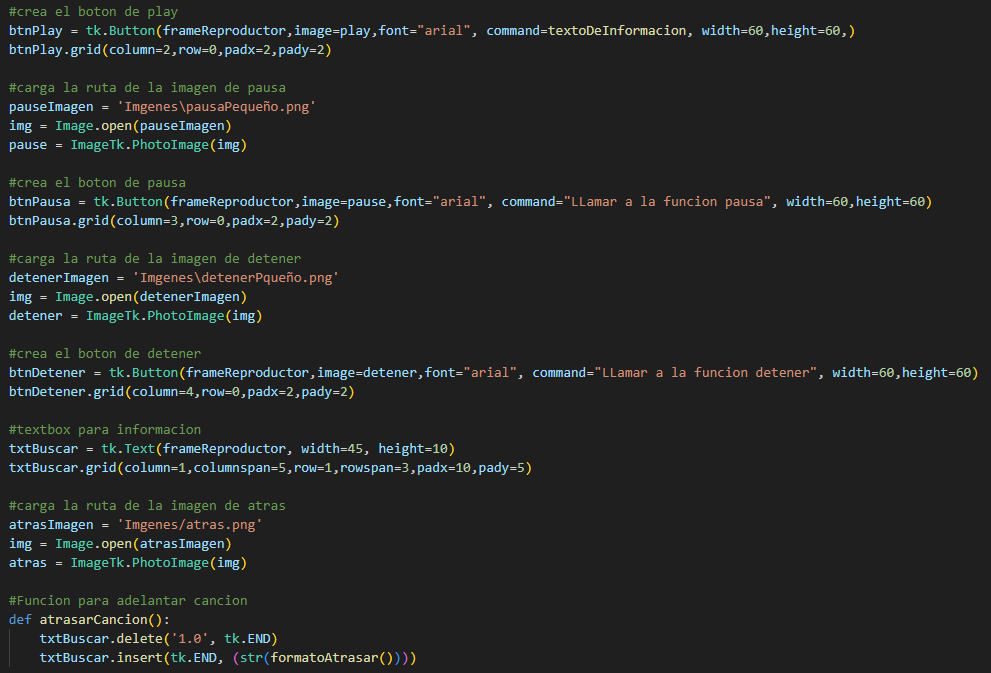
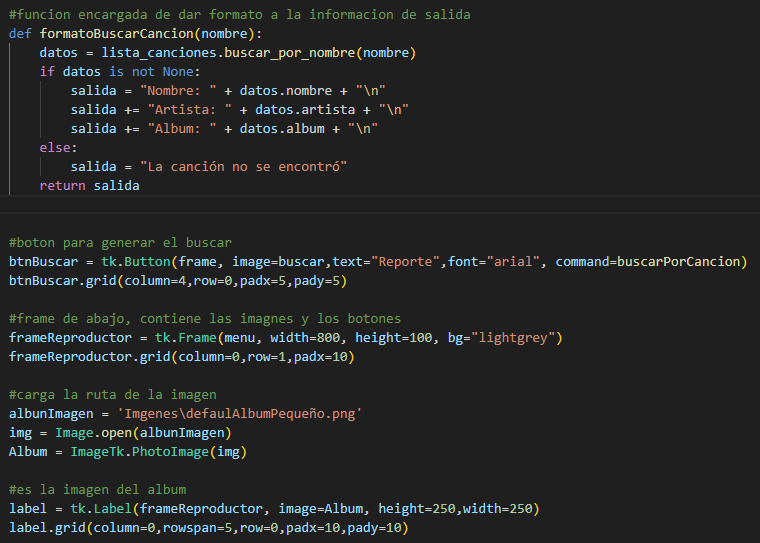
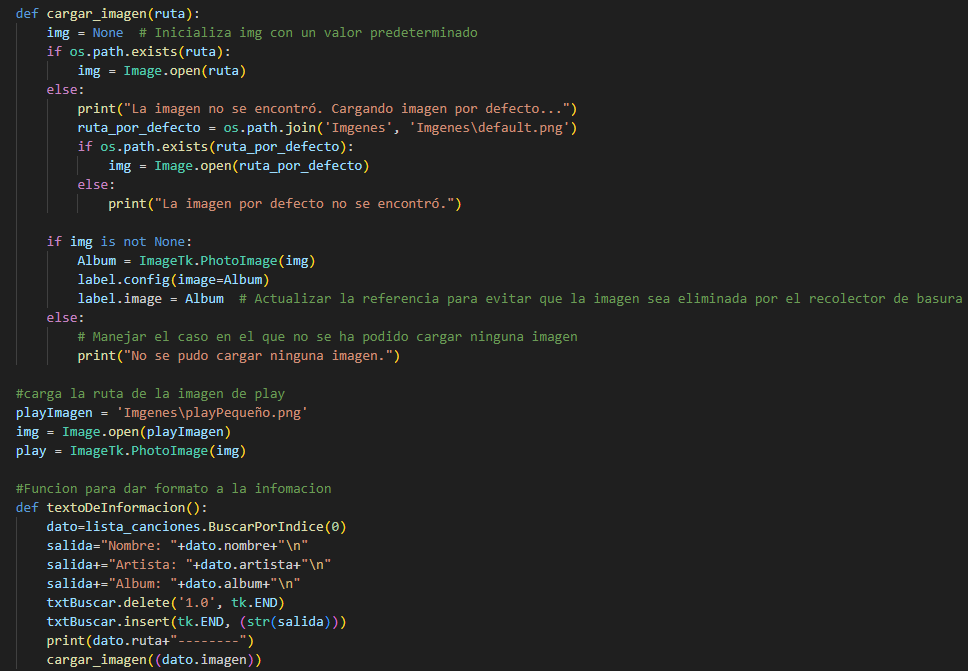
**Clase Graficas.py:**

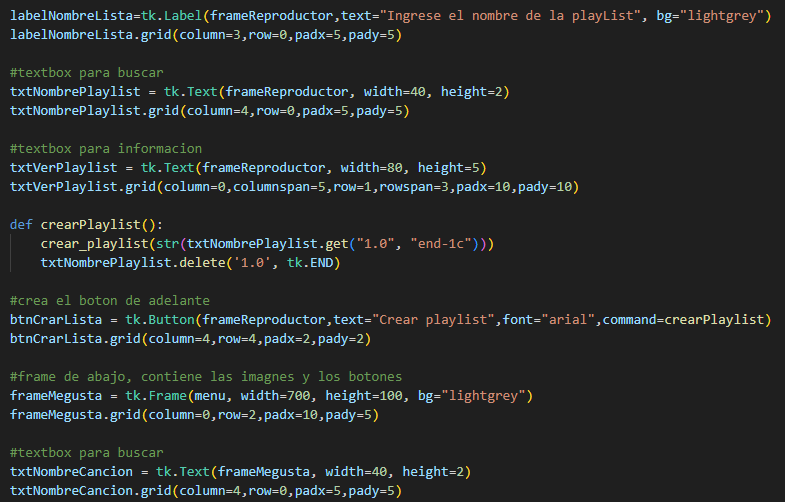
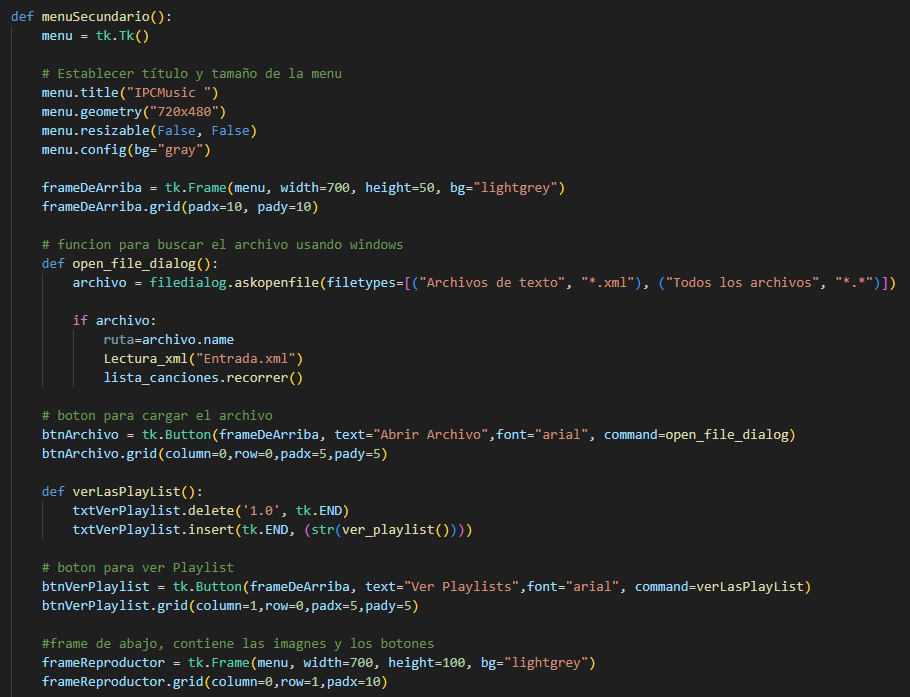
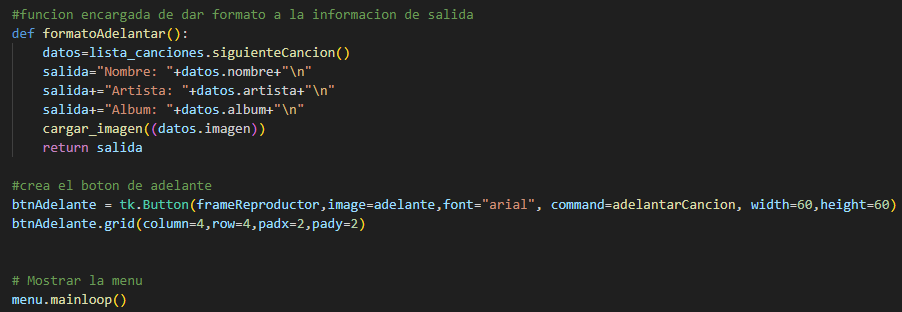
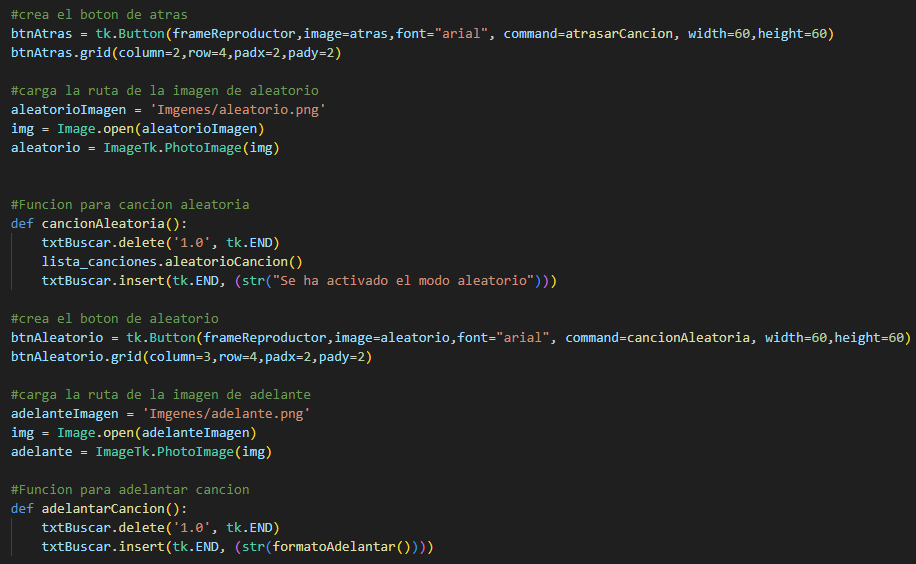
****

**Clase Interfaz.py:**

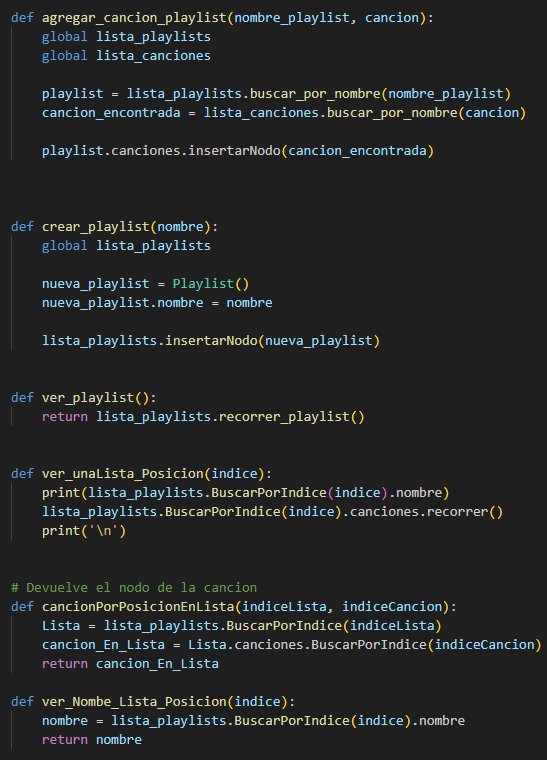
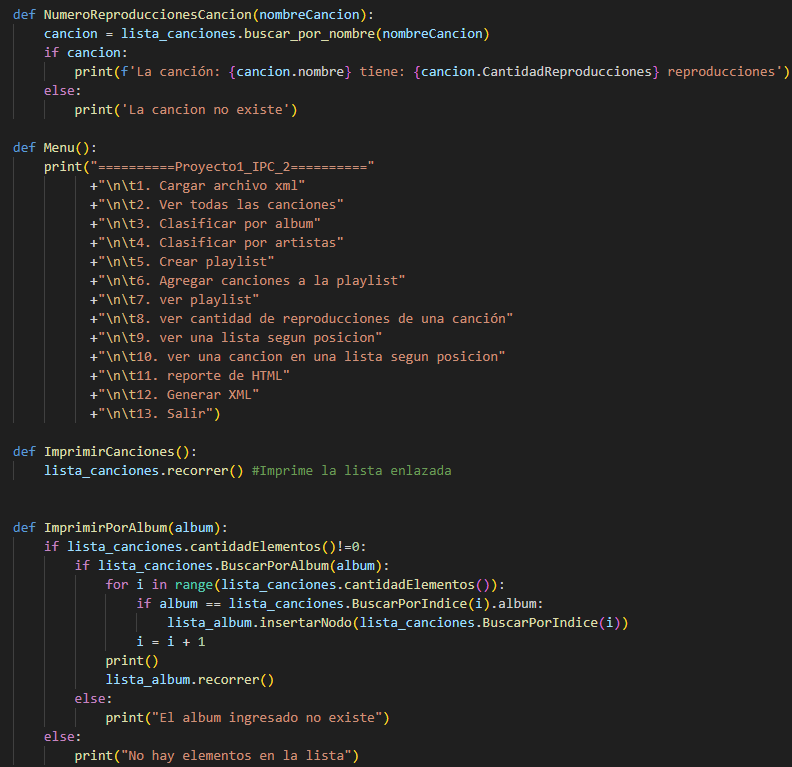
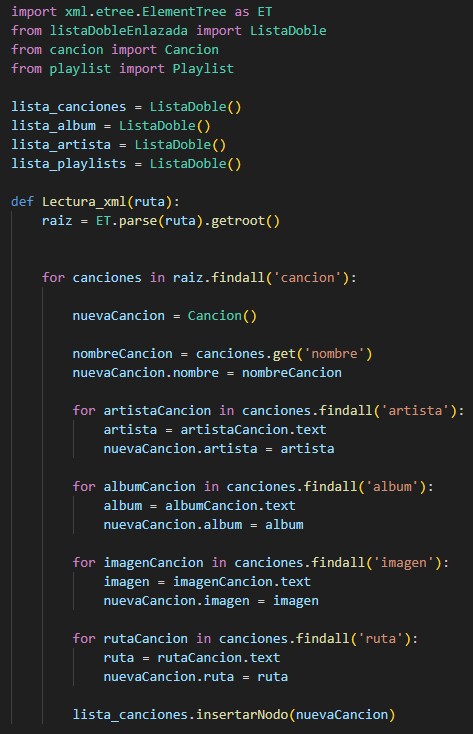
****

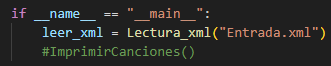
****

****

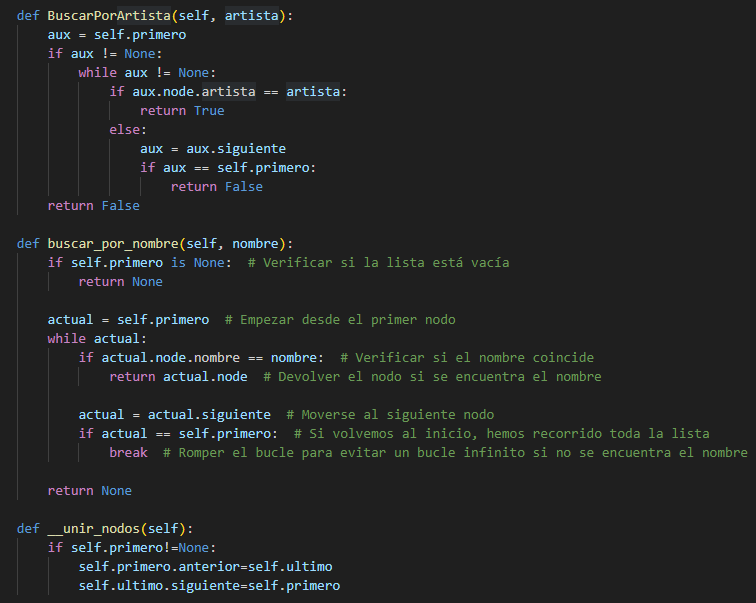
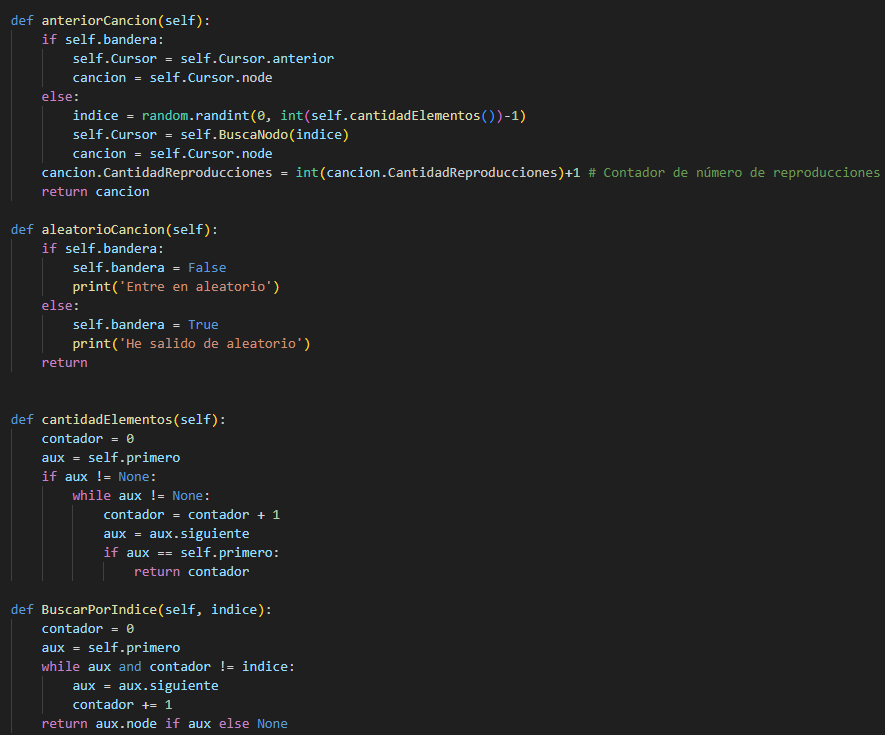
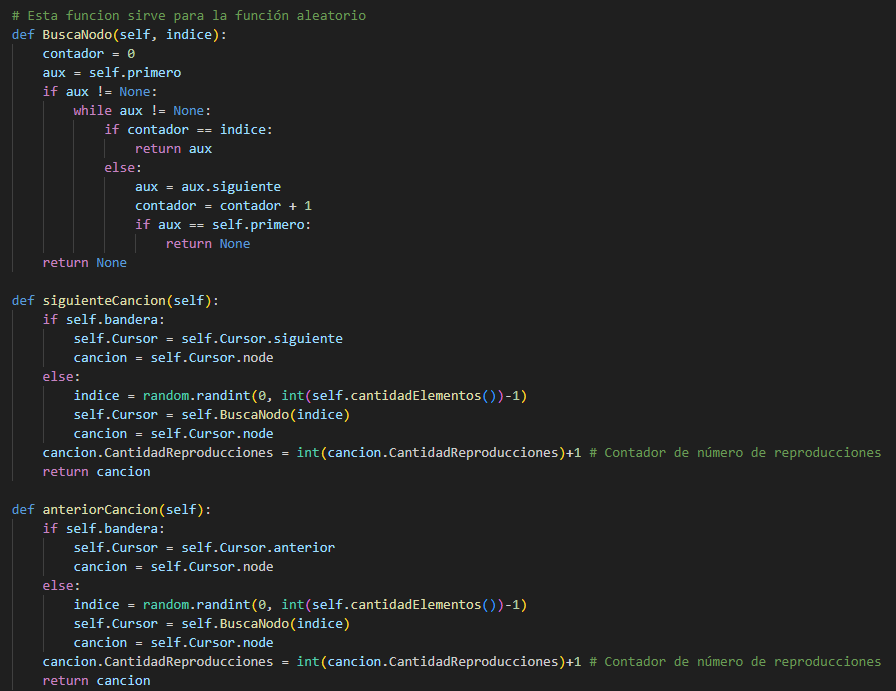
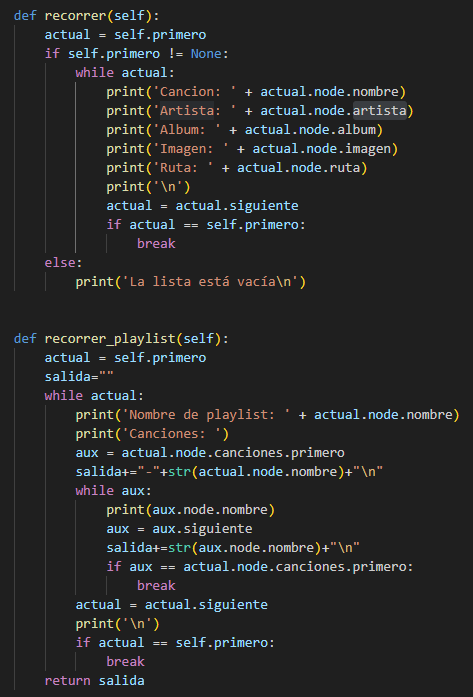
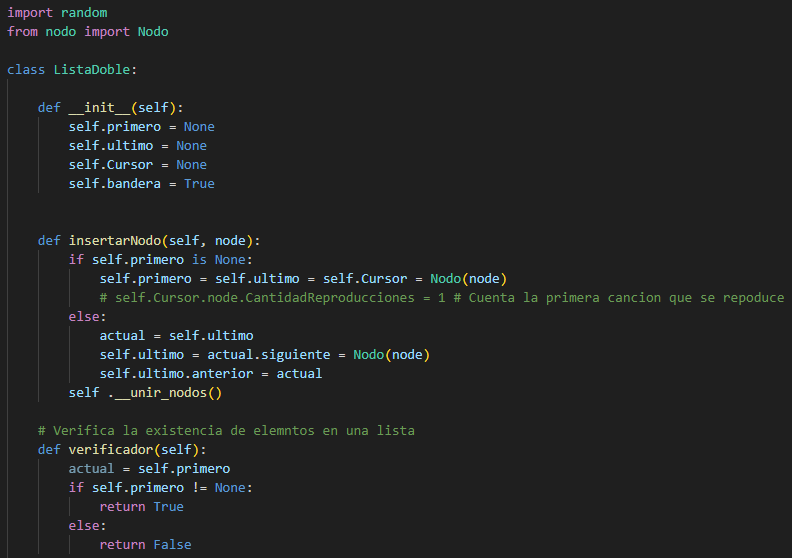
****

**Clase lecturaXML.py**

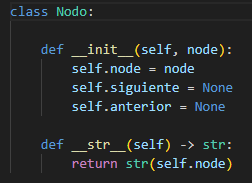
****

****

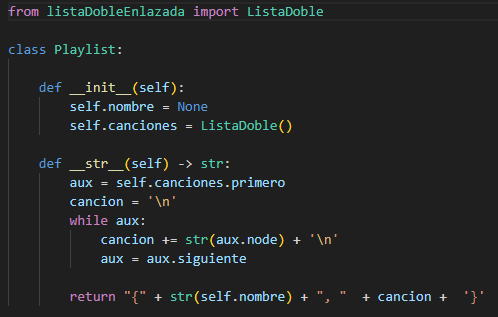
**Clase listaDobleEnlazada.py**

****

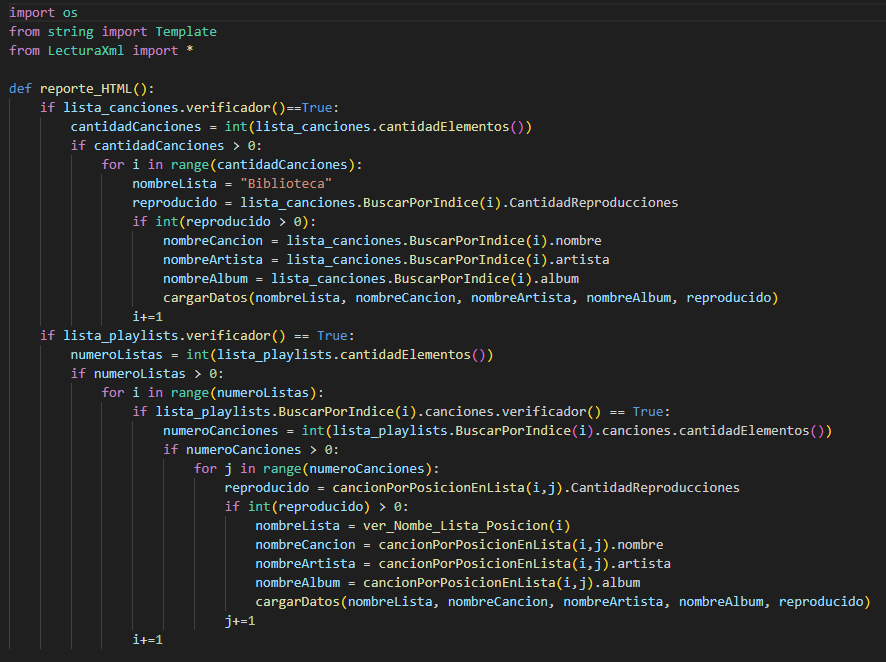
**Clase nodo.py**

****

**Clase playlist.py**

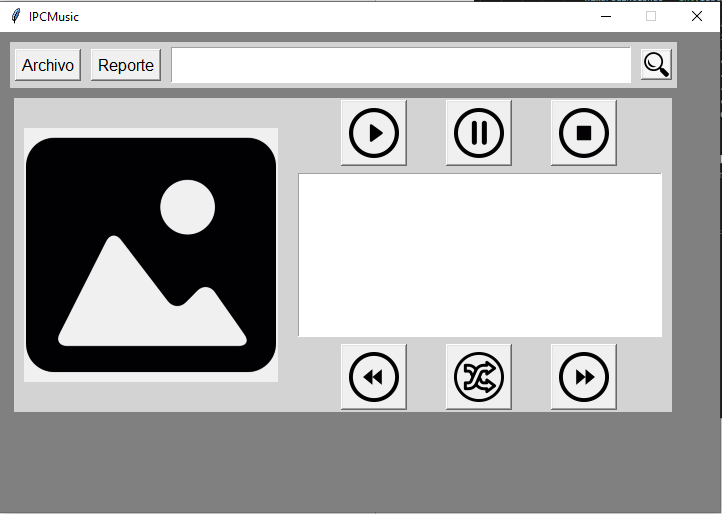
****

**Clase reporte.py**

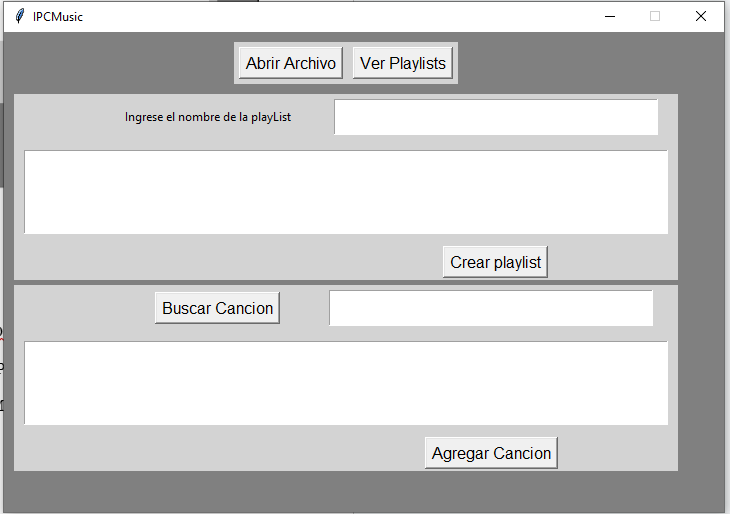
****

**Funcionamiento**

**Ventana principal: nos permite recorrer entre canciones generar el reporte, que incluye la gráfica, el xml de salida y buscar una cancion en especifico**

****

**ventana secundaria: nos permite crear listas de canciones, agregar canciones a las mismas y cargar el xml de entrada**

****

Referencias bibliográficas

W. R. McKinney, (2016). Python for Data Analysis. O’Reailly, Inc.

A. J. Aho, (2005). Estructuras de datos y algoritmos. Prentice-Hall

L. M. Joyanes, (2010). Estructura de Datos en Java. Mc Graw Hill, Inc.

Link repositorio: <https://github.com/JosCarias/IPC2_Proyecto1Diciembre_-Grupo-13.git>