

---

## Reproductor de Música - Implementación en Python

---

202204578 – Estiben Yair Lopez Leveron  
202201405 – Johan Moises Cardona Rosales  
202200133 – Cesar Augusto Francisco Vicente  
202100039 – Nombre completo del estudiante  
202004762 – Angel Guillermo Arreaga Barrientos

### Resumen

IPCmusic, una innovadora aplicación de música, fue meticulosamente diseñada con el propósito principal de proporcionar una organización eficiente y una categorización precisa de las pistas de audio. Esta organización y categorización se basan en criterios esenciales como el artista y el álbum al que pertenecen las pistas.

Para alcanzar este objetivo ambicioso, se recurrió al uso de estructuras de datos dinámicas, específicamente Listas Doblemente Enlazadas y su variante Lista Circular Doblemente Enlazada. Estas estructuras de datos se desarrollaron tomando inspiración en la teoría de grafos, utilizando de manera efectiva las propiedades intrínsecas de los nodos y las aristas para emular un grafo dirigido.

Se implementaron los Tipos de Datos Abstractos (TDA) para gestionar los datos de manera eficiente, optimizando así el rendimiento del software en relación con el uso de memoria. La Programación Orientada a Objetos (POO), una metodología de programación moderna, desempeñó un papel fundamental en la optimización de la organización de los datos. Esta metodología facilitó enormemente la gestión de los datos.

### Palabras clave

Estructura  
HTML  
XML  
Nodo  
Poo

### Abstract

IPCmusic, an innovative music application, was meticulously designed with the primary purpose of providing efficient organization and precise categorization of audio tracks. This organization and categorization are based on essential criteria such as the artist and the album to which the tracks belong.

To achieve this ambitious goal, dynamic data structures were employed, specifically Doubly Linked Lists and their variant, the Doubly Linked Circular List. These data structures were developed drawing inspiration from graph theory, effectively utilizing the intrinsic properties of nodes and edges to emulate a directed graph.

Abstract Data Types (ADT) were implemented to manage the data efficiently, thereby optimizing the software performance concerning memory usage. Object-Oriented Programming (OOP), a modern programming methodology, played a fundamental role in optimizing the data organization. This methodology greatly facilitated data management.

### **Keywords**

Structure

HTML

XML

Node

OOP

### **Introducción**

En el desarrollo de IPCmusic, un innovador reproductor de música, se emplean conceptos fundamentales, como la teoría de grafos. Esta teoría se centra en los nodos y se utiliza para crear estructuras de datos específicas, como las Listas Doblemente Enlazadas y su variante circular.

Estas estructuras de datos se diseñaron para optimizar el uso de la memoria del sistema, solicitando y liberando memoria de manera dinámica durante la ejecución del software. Esto evita la monopolización de la memoria que podría no utilizarse.

La Programación Orientada a Objetos (POO), una metodología de programación moderna, juega un papel crucial en este proceso. Permite representar los grafos en código y facilita la gestión de los datos. Esta introducción proporciona una visión general de los conceptos y técnicas de programación utilizados en IPCmusic.

## **Desarrollo del tema**

### **1. ¿Qué es Python?**

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

#### **1.1) ¿Cómo se utiliza Python?**

El lenguaje Python se aplica a varios casos de uso en el desarrollo de aplicaciones, incluidos los ejemplos siguientes:

Desarrollo web del lado del servidor

El desarrollo web del lado del servidor incluye las funciones complejas de backend que los sitios web llevan a cabo para mostrar información al usuario. Por ejemplo, los sitios web deben interactuar con las bases de datos, comunicarse con otros sitios web y proteger los datos cuando se los envía a través de la red.

Python es útil para escribir código del lado del servidor debido a que ofrece muchas bibliotecas que constan de código prescrito para crear funciones de backend complejas. Los desarrolladores también utilizan un amplio rango de marcos de Python que proporcionan todas las herramientas necesarias para crear aplicaciones web con mayor rapidez y facilidad. Por ejemplo, los desarrolladores pueden crear la aplicación web esqueleto en segundos porque no deben escribirla desde cero. Pueden probarla por

medio de las herramientas de prueba del marco, sin depender de herramientas de prueba externas.

## **2. Programación Orientada a Objetos (POO)**

Es un paradigma de programación que se basa en la organización de datos y comportamientos en estructuras llamadas “objetos”. Este enfoque revoluciona la forma en que creamos y estructuramos software al alinear más estrechamente nuestro código con cómo vemos y categorizamos el mundo real.

## **3. Estructuras de Datos**

Las estructuras de datos son una forma especializada de organizar, procesar y almacenar datos en la memoria de una computadora. Permiten el acceso y la manipulación eficiente de los datos, lo que facilita a los programadores realizar tareas específicas de manera más efectiva. Las estructuras de datos sirven como base para crear algoritmos que resuelvan problemas complejos y desempeñan un papel clave en la configuración del rendimiento y la funcionalidad del software. Las estructuras de datos comunes incluyen matrices, listas vinculadas, pilas, colas, tablas hash, árboles y gráficos. Cada tipo tiene un propósito específico y tiene sus propias fortalezas y debilidades, según los requisitos de la aplicación.

### **3.1) Lista Enlazada**

Es una de las estructuras de datos fundamentales, que consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. Las listas enlazadas simples son estructuras de datos lineales, dinámicas y fáciles de gestionar, que se componen de nodos con punteros y datos.

### **3.2) Lista Doblemente Enlazada**

Es una estructura de datos que consiste en un conjunto de nodos enlazados secuencialmente. Cada nodo contiene tres campos, dos para los llamados enlaces, que son referencias al nodo siguiente y al anterior en la secuencia de nodos, y otro más para el almacenamiento de la información. El doble enlace de los nodos permite recorrer la lista en cualquier dirección.

### **3.3) Lista Circular**

Es una lista lineal en la que el último nodo apunta al primero. Las listas circulares evitan excepciones en las operaciones que se realicen sobre ellas. No existen casos especiales, cada nodo siempre tiene uno anterior y uno siguiente. En las listas circulares, nunca se llega a una posición en la que ya no sea posible desplazarse.

## **4. XML tree and elements**

XML es un formato de datos inherentemente jerárquico, y la forma más natural de representarlo es con un árbol. ET tiene dos clases para este propósito: ElementTree representa todo el documento XML como un árbol, y Element representa un solo nodo en este árbol. Las interacciones con todo el documento (lectura y escritura desde/hacia archivos) generalmente se realizan en el nivel de ElementTree. Las interacciones con un solo elemento XML y sus subelementos se realizan en el nivel de Element.

Implementaciones para resolución del proyecto:

- a) Clases con métodos y atributos.
- b) Clases del tipo nodo.

### **a) Clases con métodos y atributos**

Una clase define una plantilla o molde para crear objeto, los cuales son instancias de esa clase. Los objetos creados a partir de una clase tienen las mismas propiedades y comportamientos definidos

por la clase, pero pueden tener valores diferentes para los atributos que se definen en la clase.

En Python, una clase se define mediante la palabra clave «class», seguida del nombre de la clase y dos puntos (:) y luego el cuerpo de la clase. El cuerpo de la clase contiene definiciones de métodos y atributos, que pueden ser públicos o privados según su acceso.

Un atributo es una variable que se define dentro de una clase, la cual almacena datos que pertenecen a un objeto de esa clase. Los atributos se utilizan para representar características o propiedades de un objeto, como su estado actual, su identificador, su tamaño, su color, etc.

En la definición de una clase, los atributos se definen como variables que se inicializan en el método especial `__init__`. Por ejemplo, en la clase `Persona` que definimos anteriormente, los atributos «nombre» y «edad» se definen de la siguiente manera:

#### b) Clases del tipo nodo.

Que es un nodo:

El nodo es un elemento de la lista el cual contiene: el elemento que queremos guardar, una referencia hacia el nodo anterior y una referencia al nodo siguiente.

Nodo de una lista.

Comenzaremos con construir un nodo utilizando una clase `class` `Nodo`: el constructor toma de valores a `self` y a elemento el cual será el elemento que se encuentre dentro del nodo, además, podemos ver que se encuentran otros dos atributos los cuales nos van a indicar las conexiones entre los nodos. Inicializamos siguiente y anterior a `None` ya al crear un `Nodo` queremos que no apunten a ningún otro nodo. Y por

último tenemos al atributo elemento el cual será el dato que vamos a guardar en el nodo.

### 5. ¿Qué es Graphviz?

Graphviz es un conjunto de herramientas de software de código abierto diseñado para la creación de diagramas gráficos. Fue desarrollado por AT&T Labs y es ampliamente utilizado en una variedad de aplicaciones, incluyendo visualización de redes, visualización de software, visualización de bases de datos y visualización web.

#### 5.1) Grafos

Un grafo es un conjunto de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas (arcs en inglés) que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).

### Conclusiones

1. **Eficiencia y organización a través de la POO:** La Programación Orientada a Objetos (POO) en Python ha demostrado ser una herramienta eficaz para la organización y estructuración del código en el proyecto. Al modelar conceptos del mundo real como objetos, se ha facilitado la implementación de funciones complejas y se ha mejorado la legibilidad y mantenibilidad del código.
- Manejo de datos con estructuras avanzadas: El uso de listas doblemente enlazadas para almacenar y acceder a las canciones, artistas y álbumes ha permitido un acceso rápido a los datos y una gestión eficiente de la

memoria. Esto demuestra el poder de las estructuras de datos avanzadas en la resolución de problemas complejos en la programación.

1. **Integración de tecnologías:** La integración exitosa de varias tecnologías, como XML para la carga de la biblioteca de música y Graphviz para la visualización de las estructuras de datos, ha permitido la creación de una aplicación robusta y funcional. Esto subraya la importancia de la versatilidad en la programación y la capacidad de trabajar con diversas tecnologías y formatos de datos.

## Referencias bibliográficas

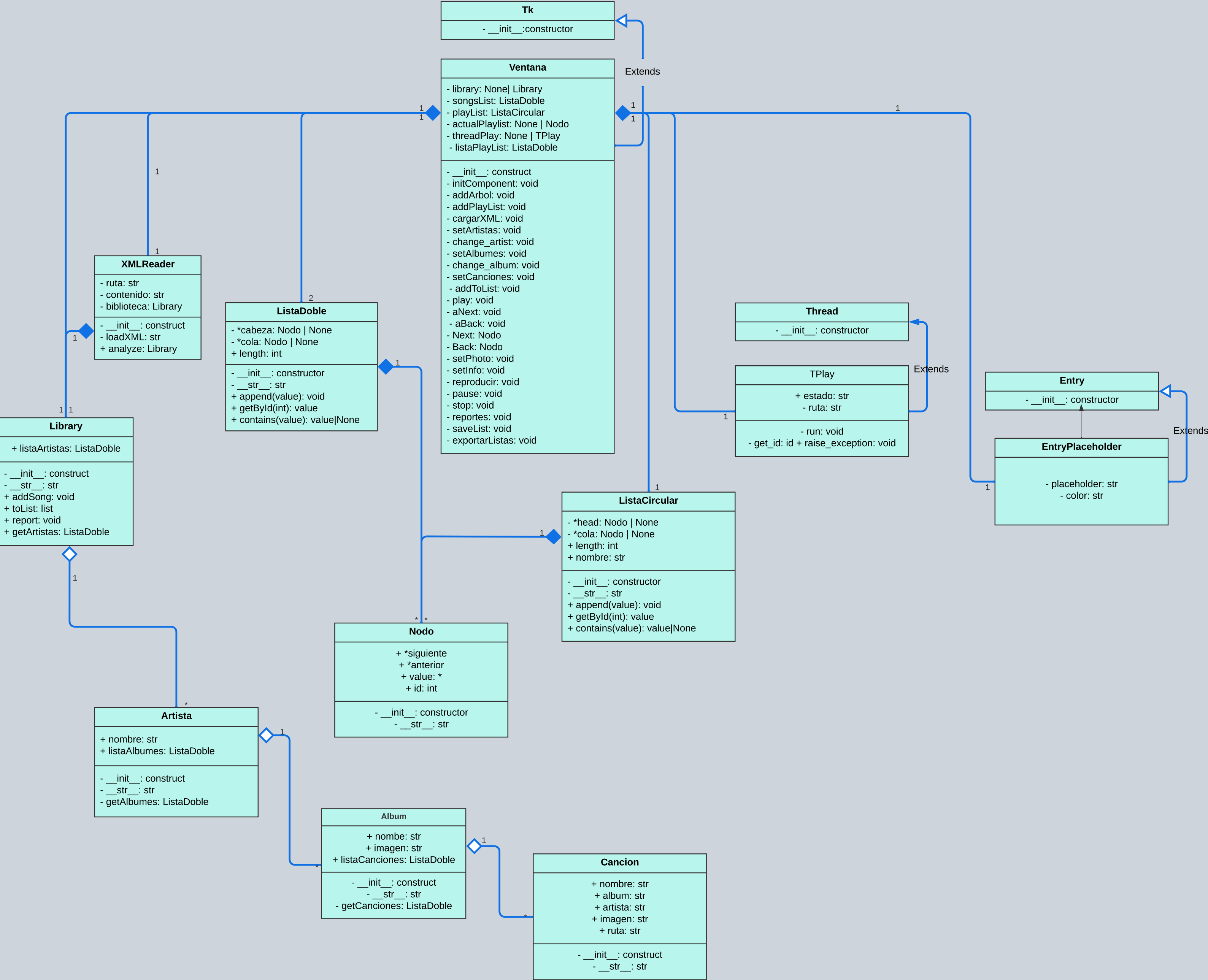
Gomez, O. A. (s/f). Visualizando Grafos usando Graphviz. OSiUX. Recuperado el 15 de diciembre de 2023, de <https://osiux.com/visualizando-grafos-graphviz.org>

Londoño, P. (2023, abril 26). Qué son las clases en Python, para qué sirven y cómo funcionan. Hubspot.es. <https://blog.hubspot.es/website/clases-python>

xml.etree.ElementTree — The ElementTree XML API. (s/f). Python documentation. Recuperado el 15 de diciembre de 2023, de <https://docs.python.org/3/library/xml.etree.elementtree.html>

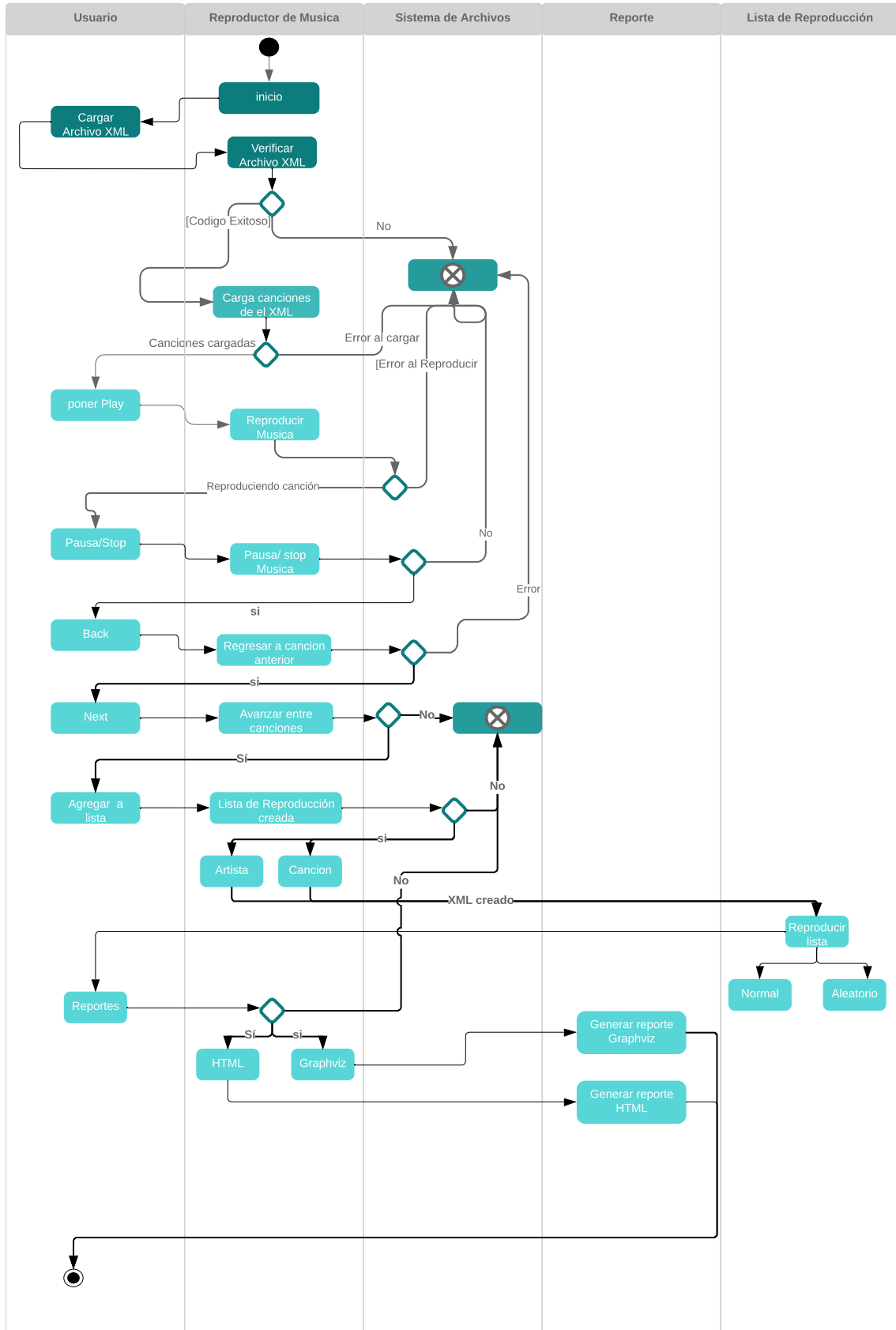
# Diagrama de Clases

GRUPO 7 | Diciembre 2023



# Diagrama de Actividades

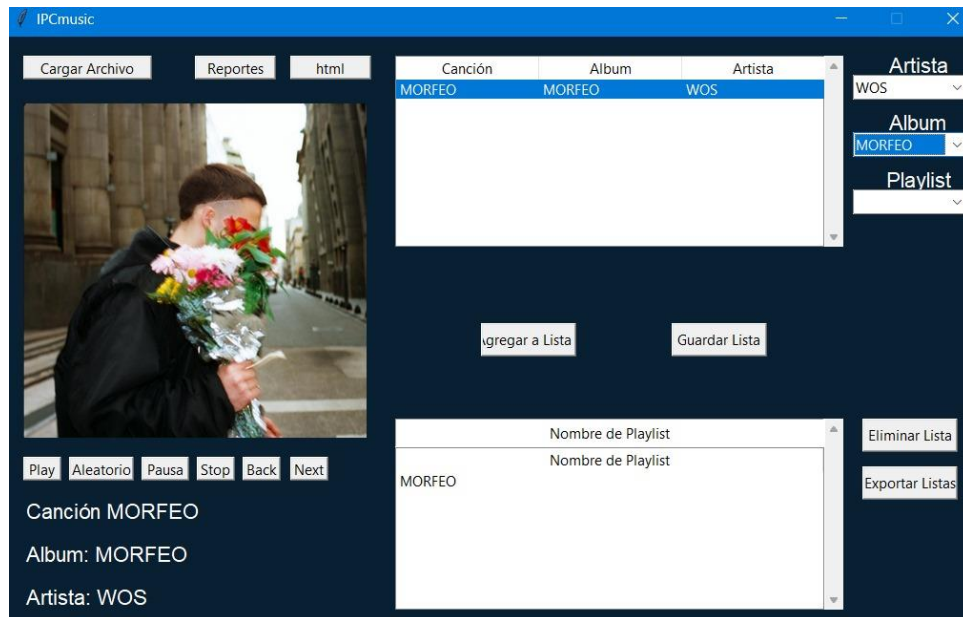
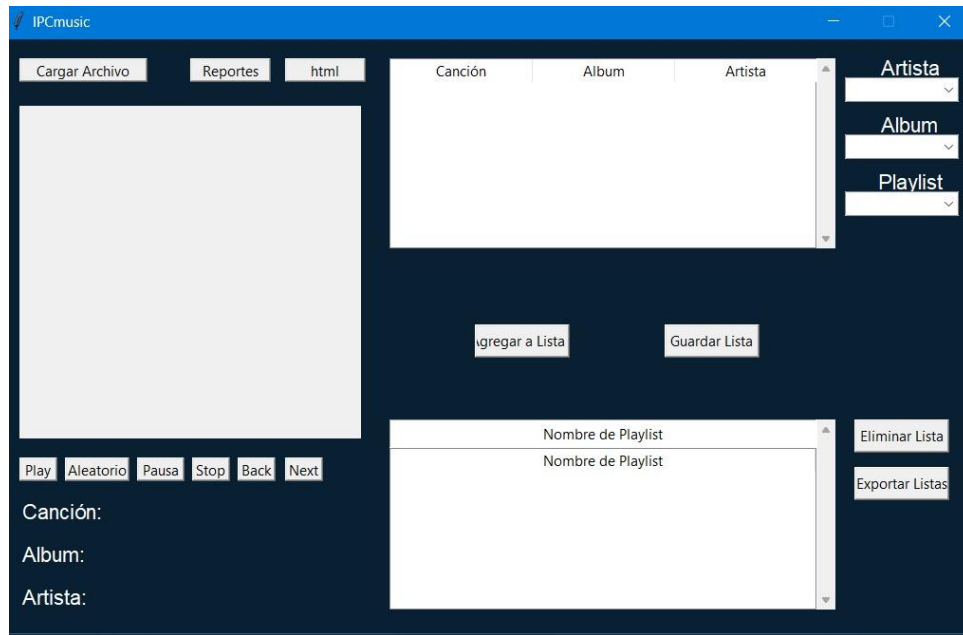
Grupo 7 | Diciembre 2023



----- Link de Repositorio -----

[https://github.com/YairCarrito45/IPC2\\_Proyecto1Diciembre.git](https://github.com/YairCarrito45/IPC2_Proyecto1Diciembre.git)

----- Interfaz -----





## ---- HTML --

### Listas de Reproducción

#### Yair

Nombre: MORFEO

Artista: WOS

Veces Reproducida: 1



Portada:

Nombre: ARRANCARMELO

Artista: WOS

Veces Reproducida: 1



Portada:

Nombre: MIDDLE

Artista: Dj snake, Bipolar Sunshine

Veces Reproducida: 1



Portada:

### Gráfico de Reproducciones



## ----- XML DE ENTRADA -----

```
libreria.xml
<biblioteca>
  <cancion nombre="MORFEO">
    <artista>WOS</artista>
    <album>MORFEO</album>
    <imagen>musica\MORFEO.jpg</imagen>
    <ruta>musica\WOS - MORFEO (Audio).mp3</ruta>
  </cancion>

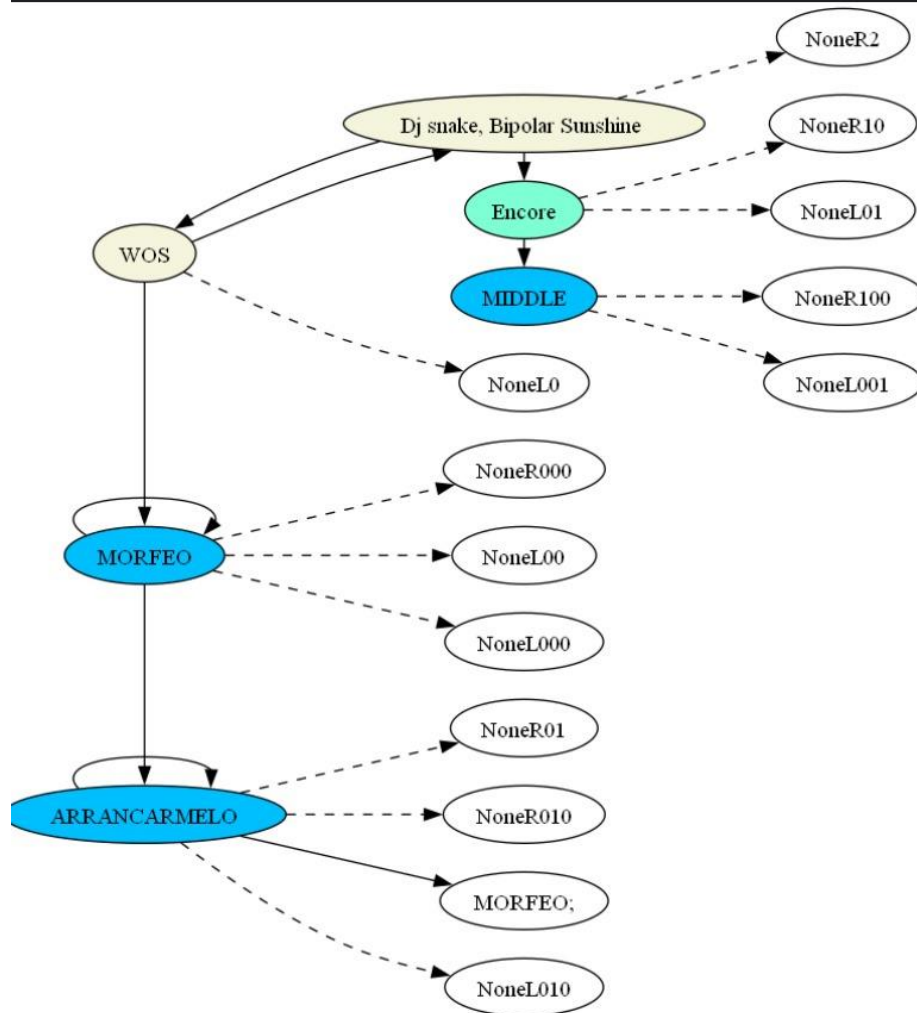
  <cancion nombre="MIDDLE">
    <artista>Dj snake, Bipolar Sunshine </artista>
    <album>Encore</album>
    <imagen>musica\MIDDLE .jpg</imagen>
    <ruta>musica\Dj snake - Middle.mp3</ruta>
  </cancion>

  <cancion nombre="ARRANCARMELO">
    <artista>WOS</artista>
    <album>ARRANCARMELO</album>
    <imagen>musica\ARRANCARMELO.jpg</imagen>
    <ruta>musica\Wos - ARRANCARMELO.mp3</ruta>
  </cancion>
</biblioteca>
```

## ----- XML DE SALIDA -----

```
listas_de_reproducción.xml
<?xml version = "1.0" encoding="UTF-8"?>
<ListasReproduccion>
  <Lista nombre = "Yair">
    <cancion nombre = "MORFEO">
      <artista>WOS</artista>
      <album>MORFEO</album>
      <vecesReproducida>0</vecesReproducida>
      <imagen>musica\MORFEO.jpg</imagen>
      <ruta>musica\WOS - MORFEO (Audio).mp3</ruta>
    </cancion>
    <cancion nombre = "ARRANCARMELO">
      <artista>WOS</artista>
      <album>ARRANCARMELO</album>
      <vecesReproducida>0</vecesReproducida>
      <imagen>musica\ARRANCARMELO.jpg</imagen>
      <ruta>musica\Wos - ARRANCARMELO.mp3</ruta>
    </cancion>
    <cancion nombre = "MIDDLE">
      <artista>Dj snake, Bipolar Sunshine </artista>
      <album>Encore</album>
      <vecesReproducida>0</vecesReproducida>
      <imagen>musica\MIDDLE .jpg</imagen>
      <ruta>musica\Dj snake - Middle.mp3</ruta>
    </cancion>
  </Lista>
</ListasReproduccion>
```

## ---- Graphviz ----



## ---- Archivos ----

