

---

## IPCMUSIC 2023

---

Elmer Wilson Miguel Juan 201931295	(Externo)
Moisés Aldair Vásquez González 201930195	(Externo)
Engelbert Emanuel Estrada Estrada 201930200	(Externo)
Selvyn Estuardo Ixtabalan Tistoj 201930524	(Externo)
Jerson Emanuel Estrada Estrada 201930199	(Externo)

### Resumen

El objetivo principal del proyecto fue la elaboración de un reproductor de música en el periodo de diciembre del 2023 que fuera totalmente funcional utilizando el lenguaje de Python y herramientas como Graphviz enfocándose en el concepto de "Listas Enlazadas". En el cual los datos de entrada del reproductor de música son leídos a través de una estructura XML y estos son almacenados, manipulados y utilizados junto con la interfaz gráfica y métodos para lograr su reproducción.

El conocimiento y técnicas principales de utilización para la elaboración de dicho proyecto fue de listas doblemente enlazadas, estructuras HTML, POO, estructuras cíclicas, condicionales, listas circulares, el uso de Tkinter y Pygame para el completo funcionamiento. El reproductor puede ser utilizado para múltiples propósitos de ocio y posee diferentes funcionalidades como reproducir la música que se desee, el mostrar un resumen de reproducciones, reproducir música aleatoriamente, mostrar información de los artistas, entre otros.

### Palabras clave

Listas, POO, Reproductor, XML, Graphviz

### Summary

*The main objective of the project was the development of a music player in the period of December 2023 that was fully functional using the Python language and tools such as Graphviz focusing on the concept of "Linked Lists". In which the input data of the music player is read through an XML structure and these are stored, manipulated and used together with the graphical interface and methods to achieve its playback.*

*The main knowledge and techniques used for the elaboration of this project were doubly linked lists, HTML structures, OOP, cyclic structures, conditionals, circular lists, the use of Tkinter and Pygame for the complete functioning. The player can be used for multiple leisure purposes and has different functionalities such as playing the desired music, showing a summary of reproductions, playing music randomly, displaying artist information, among others.*

### Keywords

*Listas OOP, Player, XML, Graphviz.*

## **Abstracción**

*Se basa en poder describir un objeto con propiedades y métodos principales sin pensar en detalle. La abstracción separa el comportamiento específico de un objeto, a esta división que se realiza se le conoce como la barrera de abstracción. (Lara, 2017)*

## **Clases**

*Una clase (en programación) es una agrupación de datos (variables o campos) y de funciones (métodos) que operan sobre esos datos. A estos datos y funciones pertenecientes a una clase se les denominan variables y métodos o funciones miembro. (Anonimo, s.f.)*

## **Nodo**

*Es un componente que forma parte de una red. En otras palabras, tanto si se trata de Internet como de Intranet (utilizada en ámbitos cerrados, con acceso limitado a los usuarios autorizados), cada servidor u ordenador constituye un nodo y se encuentra conectado a otro u otros nodos. (Gardey, 2009)*

## **XML**

*proviene de extensible Markup Language ("Lenguaje de Marcas Extensible"). Se trata de un metalenguaje (un lenguaje que se utiliza para decir algo acerca de otro) extensible de etiquetas que fue desarrollado por el Word Wide Web Consortium (W3C), una sociedad mercantil internacional que elabora recomendaciones para la World Wide Web. (Gardey, Definicion.de, 2010)*

## **Abstraction**

*It is based on being able to describe an object with main properties and methods without thinking in detail. Abstraction separates the specific behavior of an object, this division that is performed is known as the abstraction barrier.*

## **Classes**

*A class (in programming) is a grouping of data (variables or fields) and functions (methods) that operate on that data. This data and functions belonging to a class are called variables and member methods or functions. (Anonymous, s.f.)*

## **Node**

*is a component that is part of a network. In other words, whether it is the Internet or Intranet (used in closed scopes, with limited access to authorized users), each server or computer is a node and is connected to another or other nodes.(Gardey, 2009)*

## **XML**

*comes from extensible Markup Language. It is a label-extendable metalanguage (a language used to say something about another) that was developed by the Word Wide Web Consortium (W3C), an international trading company that develops recommendations for the World Wide Web. (Gardey, Definicion.de, 2010)*

## Desarrollo del tema

Para poder entrar en contexto se deberá de conocer como temas fundamentales los siguientes:

1. Programación Orientada a objetos
2. Tipos de Datos Abstractos
3. Listas Enlazadas
  - a. Listas Enlazadas Simples
  - b. Lista Doblemente Enlazada
  - c. Listas Circulares
4. Que es XML
  - a. Lectura y Escritura de archivos XML

### 1). PROGRAMACION ORIENTADA A OBJETOS

Para hablar de la Programación Orientada a Objetos se debe de hablar de primero que es un Paradigma, lo cual se puede mencionar que un Paradigma es un modelo o patrón el cual se refiere a la forma que un programador pueda leer e interpretar el código de distintos programadores, dándole solución a los problemas que se le plantean al mismo.

A su vez se puede mencionar que hay distintos Paradigmas de Lenguajes de programación entre ellos:

1. Paradigma Funcional
2. Paradigma Estructural
3. Paradigma Modular
4. Paradigma Lógico
5. Paradigma Procedimental
6. Paradigma Orientado a Eventos
7. Paradigma Orientado a Objetos

Mas en nuestro caso se hablará del Paradigma Orientado a Objetos o también llamado Programación Orientada a Objetos (POO).

La programación orientada a objetos se puede mencionar que se refiere a una manera específica de poder organizar el código por medio de pequeños fragmentos del mismo llamadas clases, de las cuales se puede obtener *Objetos*, los cuales podrán interactuar con el resto del código para obtener el funcionamiento adecuado de la aplicación (APP) y/o software (Programa).

La intención de este paradigma es el poder programar de una forma especial, y que se observe como objetos de la vida real; con lo cual se puede aprender a resolver los problemas de maneras

Para poder hablar de dicho paradigma se debe de conocer de igual manera los conceptos de clases y objetos los cuales se describirán a continuación:

a) **Clases:** *es una agrupación de datos (variables o campos) y de funciones (métodos) que operan sobre esos datos. A estos datos y funciones pertenecientes a una clase se les denominan variables y métodos o funciones miembro. (Anónimo, s.f.)*

b) **Objetos:** *consta de un ente perteneciente al POO la cual consta de datos y tareas que puede realizar en el transcurso del programa; los cuales se puede instanciar a partir de una clase.*

## 2). TIPOS DE DATOS ABSTRACTOS(TDA)

Para poder hablar de los tipos de datos abstractos debemos de hacer referencia al concepto de Abstracción o Encapsulamiento, el cual es un tema fundamental para dicho de tipos de datos.

**Abstracción:** se puede mencionar que la abstracción o separación de la especificación de un objeto o algoritmo de su implementación, en base a la utilización que se le dará en el programa y solo depende de una interfaz explícitamente definida (la especificación) y no de los detalles de su representación física (la implementación) ya que están ocultos.

Conociendo el concepto de abstracción se puede mencionar que los Tipos de Datos Abstractos son Tipos de Datos que puede crear el usuario (programador) los cuales se realizan a partir del análisis y la abstracción de los datos y recursos

que se manejaran en las mismas, así mismo se puede mencionar varios tipos de TDA'S entre ellas se mencionan:

1. Colas
2. Pilas
3. Listas enlazadas

Se menciona que cada TDA constara de su especificación la cual será independiente de su implementación y a su vez nos proporcionara las operaciones que podrá realizar las cuales nos determinaran su utilización.

### COLAS:

Colección de elementos homogéneos (del mismo tipo: Tipo Elemento) ordenados cronológicamente (por orden de inserción) y en el que sólo se pueden

añadir elementos por un extremo (final) y sacarlos sólo por el otro (frente). Es una estructura FIFO (First In First Out):

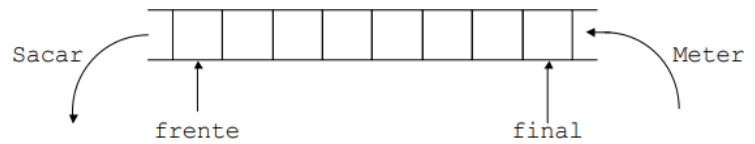


Ilustración I. TDA Cola

### PILAS:

Colección de elementos homogéneos (del mismo tipo: Tipo Elemento) ordenados cronológicamente (por orden de inserción) y en el que sólo se pueden añadir y extraer elementos por el mismo extremo, la cabeza. Es una estructura LIFO (Last In First Out):

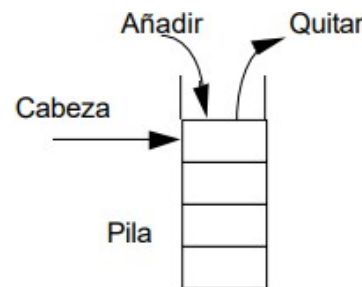


Ilustración II. TDA Pila

## 3). LISTAS ENLAZADAS:

Colección de elementos homogéneos (del mismo tipo: Tipo Elemento) con una relación LINEAL establecida entre ellos. Pueden estar ordenadas o no con respecto a algún valor de los elementos y se puede acceder a cualquier elemento de la lista.

- Hay que tener en cuenta que la posición de la inserción no se especifica, por lo que dependerá de la implementación. No obstante, cuando la posición de inserción es la misma que la de eliminación,

tenemos una subclase de lista denominada pila, y cuando insertamos siempre por un extremo de la lista y eliminamos por el otro tenemos otra subclase de lista denominada cola.

- En el procedimiento Eliminar el argumento el podría ser una clave, un campo de un registro Tipo Elemento.

A continuación, se mostrarán unas graficas para entender la representación gráfica las listas simple.

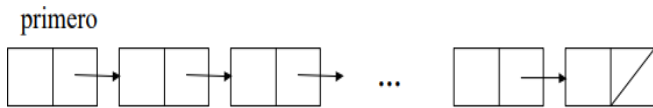


Figura 1. Representación grafica de una lista enlazada

#### b). LISTA DOBLEMENTE ENLAZADA:

Son listas en las que cada nodo, además de contener los datos información propios del nodo, contiene un enlace al nodo anterior y otro al nodo siguiente:

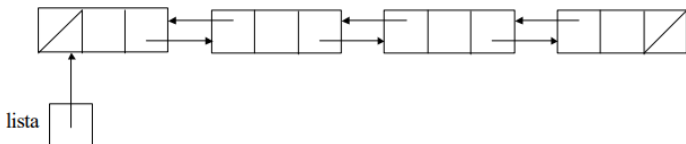


Ilustración VI. Lista Doblemente Enlazada 1

Aparte de que ocupan más memoria (el tamaño de un puntero por cada nodo más), los algoritmos para implementar operaciones para listas dobles son más complicados que para las listas simples porque requieren manejar más punteros. Para insertar un nodo, por ejemplo, habría que cambiar

cuatro punteros (o índices si se simulan con un array):

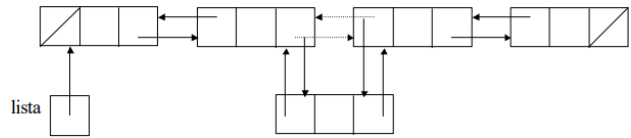


Ilustración VII. Lista Doblemente Enlazada 2

#### c). LISTAS CIRCULARES:

Son listas en las que el último elemento está enlazado con el primero, en lugar de contener el valor NULL o NULO. Evidentemente se implementarán con una representación enlazada (con punteros o variables estáticas). Con punteros sería:

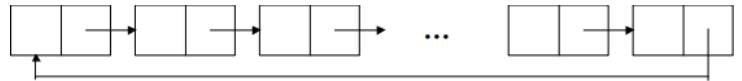


Ilustración VIII. Lista Circular

#### 4). QUE ES XML

XML es un subconjunto de SGML (Estándar Generalised Mark-up Language), simplificado y adaptado a Internet

- XML no es, como su nombre puede sugerir, un lenguaje de marcado.

- XML es un metalenguaje que nos permite definir lenguajes de marcado adecuados a usos determinados.

## CARACTERÍSTICAS

- XML es un subconjunto de SGML que incorpora las tres características más importantes de este:
  - o Extensibilidad
  - o Estructura
  - o Validación
- Basado en texto.
- Orientado a los contenidos no presentación.
- Las etiquetas se definen para crear los documentos, no tienen un significado preestablecido.
- No es sustituto de HTML.
- No existe un visor genérico de XML.

## ESTRUCTURA DE UN DOCUMENTO XML

Un documento XML está formado por datos de caracteres y marcado, el marcado lo forman las etiquetas:

Prologo	{	<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
		<! DOCTYPE persona SYSTEM "persona.dtd"
Cuerpo	{	<persona>
		<nombre>Luis</nombre>
		<apellidos>Pérez</apellidos>
		</persona>

Ilustración XII. Estructura de un Archivo XML

## ESTRUCTURA DE ETIQUETAS



Ilustración XIII. Estructura de Etiquetas en XML

## Conclusiones

Para concluir sobre el tema de un reproductor elaborado con listas enlazadas se reconoce que la aplicación del tema proporciona ventajas que a la larga facilitan la programación y brinda múltiples técnicas que aportan una consistencia y cumplen con los propósitos planteados, que de otra manera no son posibles. El uso de herramientas externas como Graphviz, XML y HTML agregan diversidad y concisión al proyecto realizado y son necesarias para crear soluciones reales a problemáticas reales.

Las listas enlazadas son posiblemente las estructuras de datos más eficientes, rápidas de aprender y de aplicar.

La resolución de un problema cotidiano que se resuelve con listas enlazadas suelen ser soluciones eficaces.

## Referencias bibliográficas

de la Vega, R. (2021, 1 febrero). ► Lista doblemente enlazada con ejemplos de Python. Pharos. Recuperado 15 de diciembre de 2023, de <https://pharos.sh/lista-doblemente-enlazada-con-ejemplos-de-python/>

D. (2015, 1 julio). Evolucion De Los Reproductores

De Musica. Ensayos para estudiantes - DAANYB. Recuperado 16 de diciembre de 2023, de <https://www.clubensayos.com/M%C3%BAsica-y-Cine/Evolucion-De-Los-Reproductores-De-Musica/2618194.html>

Eli Ramos. (2020, 4 diciembre). Reproductor de Música en Python (Tkinter y Pygame) [Vídeo]. YouTube. <https://www.youtube.com/watch?v=nKisggy-jnA&t=1s>

freeCodeCamp.org. (2019, 19 noviembre). TkinterCourse - CreateGraphicUser Interfaces in Python Tutorial [Vídeo]. YouTube.<https://www.youtube.com/watch?v=YXPyB4XeYLA>

Turmero, P. (s. f.). Introducción a la ingeniería de software. Arquitectura de software. Monografias.com. Recuperado 16 de diciembre de 2023, de <https://www.monografias.com/trabajos107/introduccion-ingenieria-software-arquitectura-software/introduccion-ingenieria-software-arquitectura-software2.shtml>