
PROYECTO 2 DE IPC 2, CREACION DE UN COMPUESTO

202100119 – Samuel Isaí Muñoz Pereira

Resumen

El presente proyecto tiene como objetivo la creación de un programa el cual pueda mostrar los pasos necesarios para crear un nuevo compuesto en base a ciertos elementos.

Así mismo mejorar y comprender los temas de listas simples, listas dobles, colas y pilas, lo cual fue muy necesario para esta práctica.

En este caso se realizó una interfaz de usuario o también conocida como GUI, para que la interacción entre el usuario y el programa sea de manera agradable e interactiva. Una vez se cargan todos los datos con un simple clic se crean todos los reportes que el usuario necesita para conocer cuáles son los pasos necesarios.

Para este proyecto se implementaron múltiples algoritmos de reconocimiento de datos, búsqueda, resolución, ordenamiento, entre otros.

Palabras clave

GUI: Interfaz gráfica.

Algoritmo: pasos para realizar determinada acción

Abstract

The present project aims to create a program that can show the necessary steps to create a new compound based on certain elements.

Likewise, it aims to improve and understand the topics of single lists, double lists, queues, and stacks, which were very necessary for this practice.

In this case, a user interface or GUI was created so that the interaction between the user and the program is pleasant and interactive. Once all the data is loaded with a simple click, all the reports that the user needs to know what the necessary steps are, are created.

Multiple algorithms for data recognition, search, resolution, sorting, among others, were implemented for this project.

Keywords

GUI: Graphical User Interface.

Algorithm: Steps to perform a certain action.

Introducción

El proyecto número 2 de introducción a la programación consiste en crear un programa el cual pueda leer un archivo xml con maquinas y compuestos, en el cual se pueda seleccionar la maquina y el compuesto y el programa pueda generar la lista de pasos necesarios para crear el compuesto.

En este caso, únicamente se puede funcionar un elemento a la vez y la función de elementos tiene que ir en orden en base al compuesto.

Una de las grandes complicaciones para este proyecto es la utilización de listas enlazadas y una pila o cola, lo cual hace que sea un poco mas complicado recorrer cada una de las listas.

Fue sumamente necesario la utilización de clases para guardar cada uno de los elementos en las listas.

Desarrollo del tema

Listas Simplemente Enlazadas:

Las listas simplemente enlazadas son una estructura de datos en la que cada nodo contiene una referencia al siguiente nodo en la lista. Cada nodo contiene un elemento y un puntero al siguiente nodo. La cabeza de la lista es el primer nodo de la lista, y el último nodo apunta a un nodo nulo.

Las listas enlazadas son la forma básica de una lista nativa en Python, pero para este proyecto fue necesario la utilización de múltiples listas enlazadas.

Listas Doblemente Enlazadas:

Las listas doblemente enlazadas son una estructura de datos en la que cada nodo tiene una referencia tanto

al nodo siguiente como al nodo anterior. Cada nodo contiene un elemento y dos punteros, uno que apunta al siguiente nodo y otro que apunta al nodo anterior. La cabeza de la lista es el primer nodo de la lista, y el último nodo apunta a un nodo nulo.

La utilización de listas doblemente enlazadas no fue algo tan utilizado en la forma en la que realice el algoritmo para la generación de pasos, pero en un caso distinto hubiera sido muy importante debido a que una de las grandes ventajas de las listas doblemente enlazadas es la posibilidad de moverse hacia adelante y hacia atrás, así como añadir datos desde adelante y o desde atrás

Cola:

También conocidas como FIFO, “primero en entrar y primero en salir”, las colas son un tipo de opción de guardado la cual va guardando los datos en un orden tal cual que al irlos sacando el primer dato que saldrá será el que entre de primero.

Las colas fueron utilizadas en el proyecto para guardar cada una de las direcciones donde se encuentra cada elemento del compuesto, debido a la posibilidad de sacar los elementos en base al primero que entre fue una gran opción para guardar las direcciones, ya que cuando se fusiona un elemento este se borra de la cola haciendo que la cola se reduzca hasta que ya no exista ningún otro elemento por fusionar

POO:

La Programación Orientada a Objetos (POO) es un paradigma de programación que se enfoca en la

creación de objetos que interactúan entre sí para realizar tareas. Cada objeto es una instancia de una clase, que define sus atributos (variables) y métodos (funciones). Los objetos interactúan entre sí mediante el envío de mensajes, lo que significa que un objeto llama a un método de otro objeto.

Para este proyecto fue muy necesaria la programación orientada a objetos, ya que se utilizaron muchas clases para guardar cada uno de los elementos en sus respectivas listas.

Normalmente había muchos datos para guardar en cada una de las listas, por ejemplo en la lista de máquinas, existe los datos del nombre de la máquina, número de pines, número de elementos, y la lista de elementos, debió a la naturaleza de las listas enlazadas se puede guardar un único dato en un nodo, por lo que la solución fue utilizar múltiples clases con sus respectivos métodos constructores para asignar cada uno de los valores en sus atributos, para al final poder cargar el objeto clase en la lista enlazada.

También fue muy necesarios para acceder a cada uno de los atributos de los objetos, para luego utilizarlo en cada una de las aplicaciones.

Librerías utilizadas

Tkinter:

es un módulo de Python que permite crear interfaces gráficas de usuario (GUI, por sus siglas en inglés) de una manera sencilla y rápida. Es una biblioteca estándar que viene con Python, lo que significa que no es necesario instalar nada adicional.

Tkinter es compatible con múltiples sistemas operativos, lo que significa que se puede crear una GUI en Windows, Linux o MacOS, entre otros. Proporciona un conjunto de widgets, como botones, cuadros de texto, menús, entre otros, que se pueden colocar en una ventana para crear una interfaz de usuario.

Para crear una GUI con Tkinter, es necesario crear una ventana principal y agregar widgets a ella utilizando el lenguaje de programación Python. Tkinter también permite vincular funciones a los eventos de los widgets, lo que significa que se puede controlar la funcionalidad de la GUI a través de las acciones del usuario.

Para la interfaz de usuario fue necesario crear múltiples ventanas que pudieran spawnear para diferentes situaciones, como las opciones de visualizar cada uno de los elementos, agregar elementos nuevos, o la opción de correr el experimento

Para el apartado gráfico fue un gran reto ya que era necesario programar cada uno de los botones, labels, tablas, etc. de manera manual, por lo que fue necesario repetir ciertas partes del código para ahorrar un poco de tiempo.

La programación de las tablas fue algo totalmente nuevo ya que era necesario añadir cada uno de los elementos mediante el uso de las listas enlazadas, también fue un reto el hecho de tener que colocar las tablas de manera precisa y correr la GUI múltiples veces para verificar que todo quedara en orden

xml.etree.ElementTree:

es una biblioteca de Python que permite analizar, manipular y crear archivos XML (Lenguaje de Marcado Extensible). Permite leer archivos XML y crear un árbol de elementos que puede ser recorrido para acceder a la información y modificarla. Además, la biblioteca también permite la creación de archivos XML desde cero, lo que es muy útil para generar archivos de configuración, archivos de datos estructurados, etc.

La biblioteca es fácil de usar y tiene una sintaxis simple y clara. Es parte de la biblioteca estándar de

Python, lo que significa que está disponible en todas las instalaciones de Python por defecto, sin necesidad de instalar ninguna biblioteca adicional.

Para este proyecto fue necesario la utilización de esta librería ya que el archivo de entrada es de la extensión xml, con esta librería la lectura del archivo es mucho mas simple ya que funciona con un árbol que tiene múltiples ramas. En cada una de las ramas fue necesario utilizar una lista simple, o doble según sea el caso para guardar cada uno de los datos, como elementos, numero atómico, nombre del elemento etc.

También se utilizó esta librería para la creación del reporte de pasos, la cual tiene una extensión xml, muy bien organizada en forma de árbol, con los respectivos segundos o pasos, su movimiento y el pin que se está ejecutando en ese momento

```
IPC2_Proyecto2_202100119 > Reportes > reportePasos.xml
1  <?xml version="1.0" ?>
2  <RESPUESTA>
3    <listaCompuestos>
4      <compuesto>
5        <nombre>Energon</nombre>
6        <maquina>prueba</maquina>
7        <tiempoOptimo>15</tiempoOptimo>
8        <instrucciones>
9          <tiempo>
10           <numeroSegundo>1</numeroSegundo>
11           <acciones>
12             <accionPin>
13               <numeroPin>3</numeroPin>
14               <accion>Mover Adelante</accion>
15             </accionPin>
16             <accionPin>
17               <numeroPin>4</numeroPin>
18               <accion>Mover Adelante</accion>
19             </accionPin>
20             <accionPin>
21               <numeroPin>5</numeroPin>
22               <accion>Mover Adelante</accion>
23             </accionPin>
24           </acciones>
25         </tiempo>
26         <tiempo>
27           <numeroSegundo>2</numeroSegundo>
28           <acciones>
29             <accionPin>
30               <numeroPin>3</numeroPin>
31               <accion>Fusionar Elemento: In</accion>
32             </accionPin>
33             <accionPin>
34               <numeroPin>4</numeroPin>
35               <accion>Esperar</accion>
```

Librerías OS:

Integra opciones básicas de manejo del sistema, como las de abrir archivos, utilizar comandos en consola, etc.

Fue utilizada para la generación del archivo pdf de graphviz

Graphviz:

Es una biblioteca de visualización de grafos de código abierto y gratuita, que permite crear diagramas y representaciones gráficas de redes, relaciones y estructuras de datos complejas. La biblioteca se basa en la descripción de los grafos mediante lenguaje de descripción de gráficos DOT

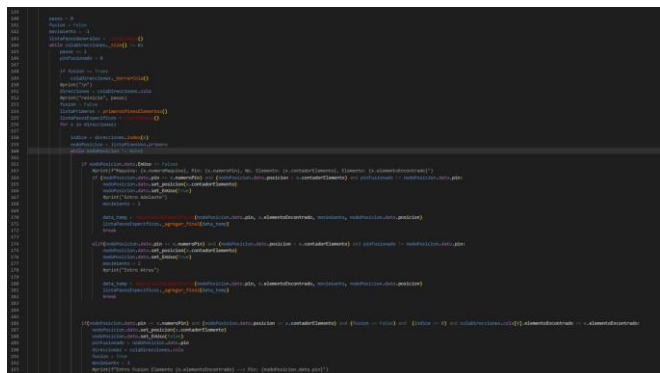
La biblioteca Graphviz proporciona una variedad de herramientas y algoritmos para generar gráficos automatizados y personalizados. Estas herramientas incluyen la capacidad de definir nodos y aristas, etiquetas, formas, colores y estilos. También es posible ajustar la posición de los nodos y aristas, y aplicar diseños automáticos de grafos para obtener un mejor aspecto visual.

Graphviz es una herramienta popular en áreas como la informática, matemáticas, ciencias sociales, biología, ingeniería y muchas otras. Es compatible con varios formatos de salida, incluyendo PNG, PDF, SVG, entre otros.

Graphviz es una biblioteca ya utilizada en proyectos y practicas anteriores, por lo que es sumamente indispensable en la creación de reportes y de grafos, para esta practica se utilizo en la parte del reporte de pasos para la creación del compuesto.

También se utilizo para la muestra de maquinas en el sistema, la cual muestra una tabla con cada una de las maquinas y su respectiva información

Algoritmos importantes

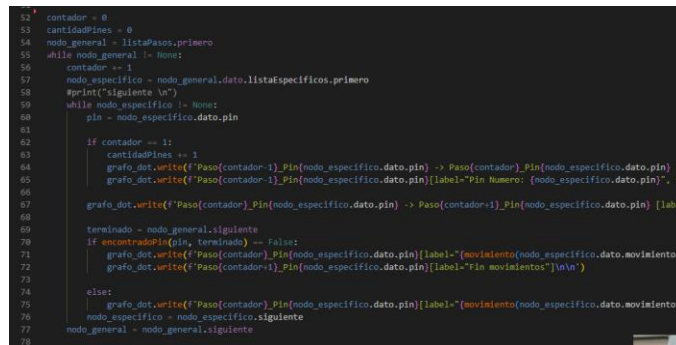


El algoritmo de reconocimiento de pasos es el algoritmo mas importante ya que es el encargado de generar cada uno de los pasos necesarios para fusionar cada uno de los elementos del compuesto.

La lógica del algoritmo es la siguiente:

- Recorrer la cola con las direcciones de cada uno de los elementos del compuesto dentro de cada pin
- Recorrer una lista con los pines que se están utilizando en ese momento, con su respectiva posición y último movimiento
- Verificar si el pin por recorrer no está en uso, en caso que no lo este, puede optar por el movimiento de hacia adelante o hacia atrás
- En caso de que el pin ya esta en uso, optar por las opciones de fusionar u esperar,
- Si en caso puede esperar, es necesario que verifique si el pin por esperar es uno de los elementos que ya esta disponible para fusionar pero que aún no puede
- Todos los pasos se van guardando en una lista que organiza cada uno de los pasos de cada pin, y guarda una lista en cada uno de los segundos o pasos

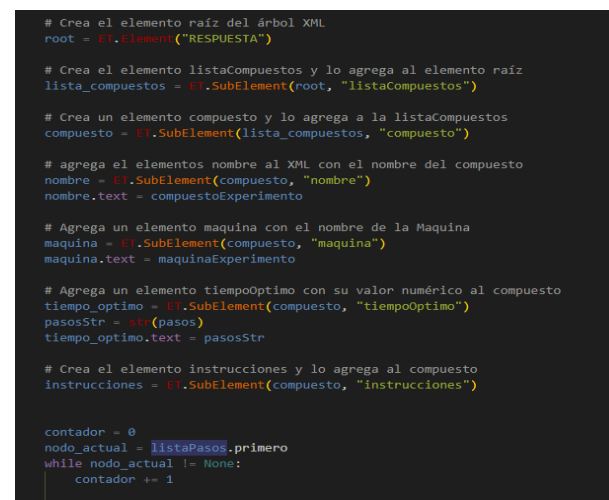
Generación del archivo .dot



La generación del archivo de pasos es muy importante, ya que indica cada uno de los pasos y movimientos de una manera agradable al usuario:

- Primero entra en la lista general que guarda las listas específicas para cada uno de los segundos.
- Entra la lista específica, y escribe de manera logística cada uno de los nodos con sus respectivos movimientos, pasos o segundos
- Se mueve de manera lineal en búsqueda de una segunda, tercera, o n lista.

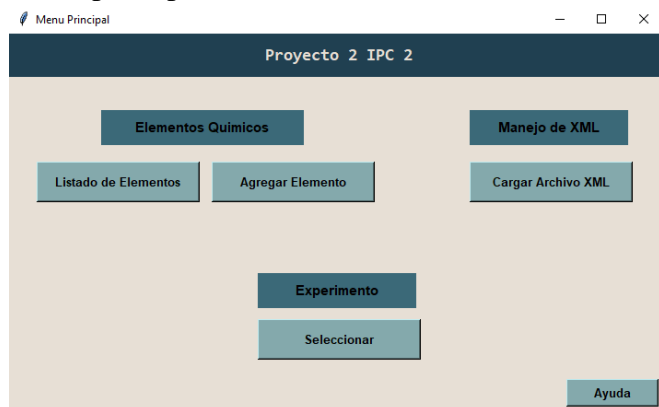
Creación de XML



Este algoritmo es uno de los mas sencillos ya que solo se encarga de ir generando cada uno de las ramas del árbol y en las partes donde es necesario añadir mas de un elemento, se utiliza un bucle para ir ejecutando o añadiendo cada uno de los elementos adicionales.
Como en la parte de los tiempos, el algoritmo debe ser capaz de añadir mas o menos tiempo según sean los pasos para crear el compuesto

Parte Grafica

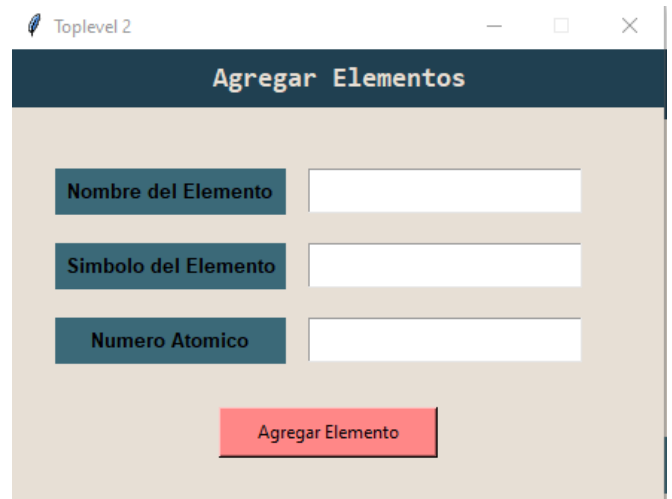
Menu principal



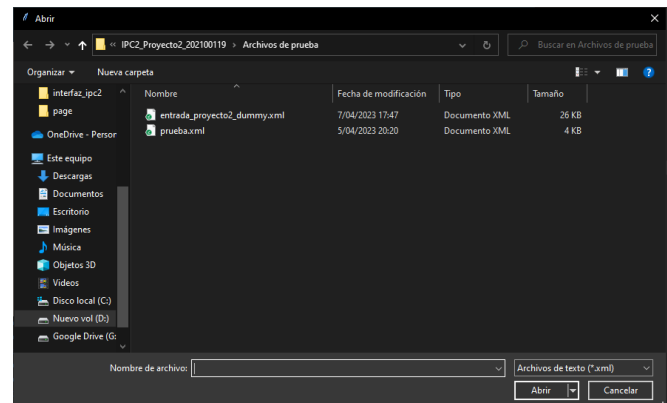
Listado de Elementos

Listado de Elementos		
Numero Atomico	Elemento	Simbolo
1	Hidrógeno	H
2	Helio	He
3	Litio	Li
4	Berilio	Be
5	Boro	B
6	Carbono	C
7	Nitrógeno	N
8	Oxígeno	O
9	Flúor	F
10	Neón	Ne
11	Sodio	Na
12	Magnesio	Mg
13	Aluminio	Al
14	Silicio	Si
15	Fósforo	P

Agregar Elemento



Abrir Archivo



Cargar Experimento:

Experimento

Datos para Experimento

Escriba el nombre de la Maquina

Escriba el nombre del Compuesto

Correr Experimento

Reporte de Maquinas

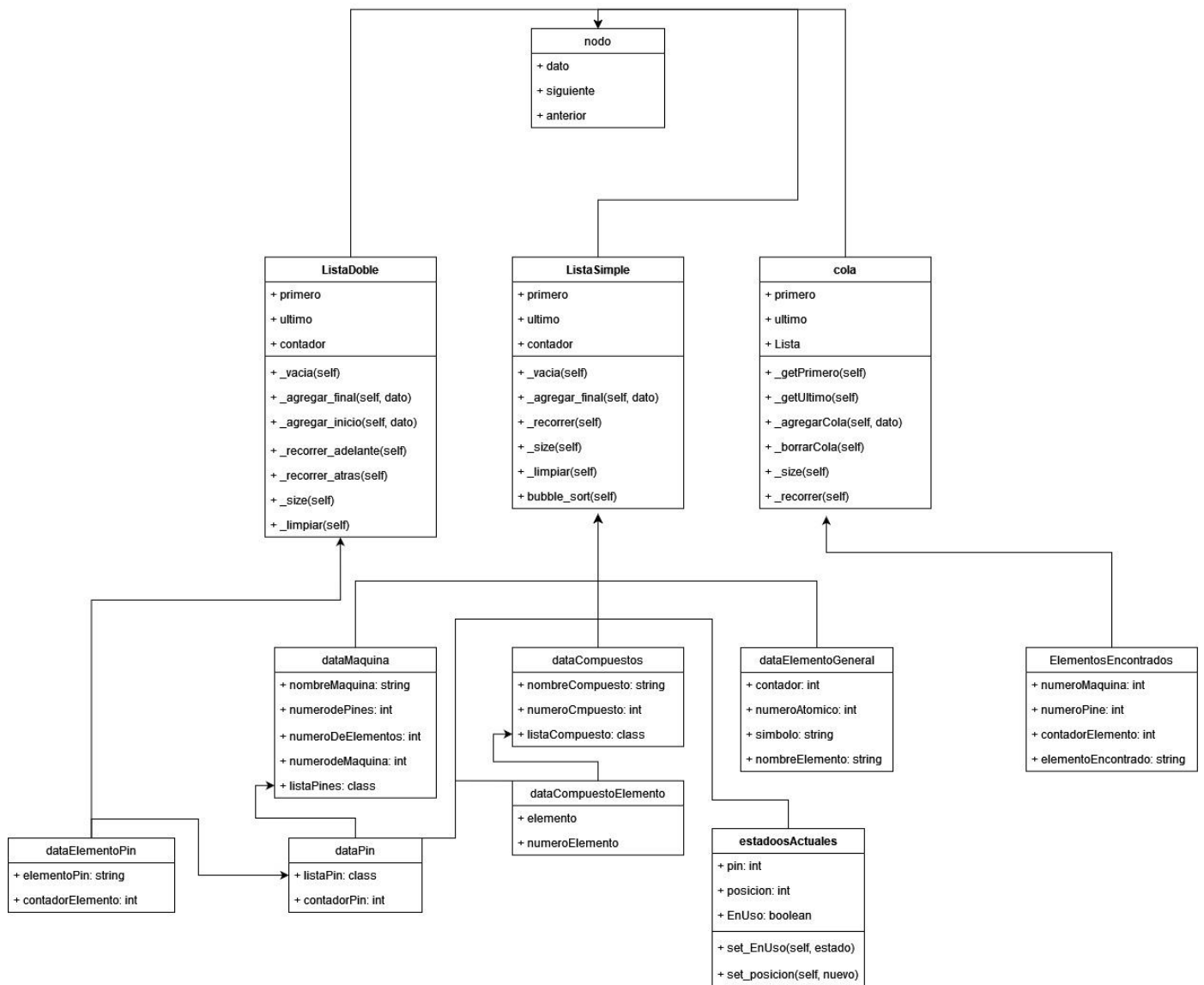
Lista de Maquinas

Maquina	Nombre	No. Pines	No. Elementos

Lista de Compuestos

No. Compuesto	Compuesto	Elementos

Diagrama de Clases



Conclusiones

Se puede concluir que este proyecto fue una experiencia muy importante en mi carrera como estudiante ya que puse en práctica mis conocimientos básicos sobre la programación orientada a objetos además que pude crear nuevas soluciones en base a cada una de las necesidades del programa.

Este proyecto fue de gran ayuda para entender de mejor manera los TDA, como las listas simples o las colas, las cuales fueron de gran ayuda, con esto puedo llegar a decir que soy capaz de utilizarlos en campos donde necesite crear funciones muy específicas para el uso de listas, en casos donde las listas nativas no son lo suficientemente poderosas

El uso de los archivos XML, es muy importante en la programación ya que permite trabajar de una manera ordenada y detallada en cada uno de los procesos a manejar, aunque desde mi punto de vista los archivos json, son mucho más entendibles

Y por último, graphviz es una herramienta poderosa que se puede utilizar en un sinnúmero de proyectos para generar reportes gráficos, detallados y entendibles para los usuarios finales

Referencias bibliográficas

Arcila, D. (2020). Introducción a Tkinter: la interfaz gráfica de Python. Platzi. Recuperado el 7 de abril de 2023, de <https://platzi.com/blog/introduccion-a-tkinter-la-interfaz-grafica-de-python/>

Código Python. (s.f.). Tutorial de Tkinter en Python. Recuperado el 7 de abril de 2023, de <https://codigo-python.com/tkinter-python/>

García, A., & Pina, J. (2016). Tipos de datos abstractos (TDA): conceptos y aplicaciones. Revista de Investigación Académica, 32. Recuperado el 7 de abril de 2023, de

<https://www.redalyc.org/articulo.oa?id=323545506005>

Herrera, J. (2017). Tutorial de XML: qué es, cómo funciona y cómo utilizarlo. Programación Fácil.

Recuperado el 7 de abril de 2023, de

<https://programacion-facil.com/es/xml/tutorial-xml-que-es-como-functiona-como-utilizarlo>

Pagella, M. A. (2018). Introducción a Graphviz: la herramienta para visualización de gráficos.

OpenWebinars. Recuperado el 7 de abril de 2023, de

<https://openwebinars.net/blog/introduccion-a-graphviz-la-herramienta-para-visualizacion-de-graficos/>

Mejía, J. C. (2021). Graphviz: una herramienta para visualizar gráficos. GeeksforGeeks. Recuperado el 7 de abril de 2023, de

<https://www.geeksforgeeks.org/graphviz-a-tool-to-visualize-graphs/>