

---

## PROYECTO 2 IPC2 PRIMER SEMESTRE 2021

---

20906051 – Juan Francisco Urbina Silva

### Resumen

El Proyecto 2 que se desarrolló mediante el lenguaje de programación Python, se llevó a cabo con el objetivo de ampliar los conceptos relacionados al manejo de archivos XML e interfaces, los tipos de datos abstractos, el manejo de la herramienta Graphviz para graficar las matrices ortogonales de las operaciones.

Los tipos de datos abstractos (TDA) son un modelo el cual define los valores y operaciones que pueden ser realizadas sobre ellos; denominadas abstractas ya que la intención es que quien lo utiliza, no requiere de conocer los detalles de la representación interna o el cómo están implementadas las operaciones.

Los archivos XML es un lenguaje que define un conjunto de reglas en la codificación de documentos, aplicando en el análisis de datos o lectura de textos creados por computadoras o personas; proporcionando una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

### Palabras clave

Listas  
XML  
TDA  
Interfaz

### Abstract

*The project 2 that developed using the Python programming language, it was carried out with the objective of expanding the concepts related to handling of XML files and interfaces, the abstract data types, the use of the Graphviz tool to graph the orthogonal matrix of the operations.*

*The abstract data types (TAD) are a model which defines the values and operations that can be performed on them; called abstract since the intention is that whoever use it, it does not require knowing the details of the internal representation or how the operations are implemented.*

*The XML files is a language that defines a set of rules in the encoding of documents, applying in data analysis or reading texts created by computers or people; providing a platform to define elements to create a format and generate a custom language.*

### Keywords

*Lists  
XML  
TDA  
Interface*

## Introducción

Dentro del primer proyecto realizado se aplicaron diversos conocimientos y conceptos aprendidos del lenguaje de programación Python basada en listas enlazadas, utilizando funciones para la obtención y manipulación de datos en las listas, funciones para el agregado, búsqueda y obtención de estos; los cuales fueron aplicados a las listas.

Se empleo el lenguaje de programación Python para aplicarla como herramienta de desarrollo de la aplicación, obtención y manipulación de los datos de un archivo XML, usando funciones para enviar los datos en las listas enlazadas y doblemente enlazadas, enviando instrucciones de posicionamiento y búsqueda en las mismas. Cabe de resaltar la implementación de listas enlazadas dentro del programa mediante ciclos y condicionales para la obtención de datos de un archivo XML.

Se pudo determinar que en la implementación del desarrollo de escritura, obtención y manipulación de los datos en las listas no mostro ningún inconveniente durante la ejecución del programa.

## Desarrollo del tema

El Proyecto que se realizó mediante el lenguaje de programación Python, consta de un menú el cual se encarga de gestionar y manipular de los datos obtenidos de un archivo XML, obteniendo en el menú las siguientes opciones:

1. **Cargar Archivo:** Dicha opción se encarga de obtener la dirección del archivo XML.
2. **Procesar Archivo:** Se encarga de procesar la información del archivo XML para luego enviarla a las listas enlazadas.

3. **Crear Archivo de Salida:** Se encarga de escribir el archivo de salida nuevo como archivo XML.
4. **Mostrar datos del Estudiante:** Muestra la información del desarrollador (Estudiante).
5. **Generar Gráfica:** Genera la gráfica de una matriz específica.
6. **Salir:** Sale del programa.

## ¿Qué es XML?

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. Es un conjunto de códigos que se aplican al análisis de datos o a la lectura de textos creados ya sea por computadoras o por personas; proporcionando una plataforma para definir elementos para crear un formato, generando un lenguaje personalizado.

Un archivo XML se divide en dos partes: Prolog y Body; donde prolog consiste en los metadatos administrativos, como la declaración XML, instrucciones de procesamiento opcional, declaración de tipos de documentos y comentarios; por otro lado, la parte del body se compone en la parte estructural y la parte de contenido.

El diseño de XML se centra en la simplicidad, generalidad y facilidad de uso, por lo que se utiliza en varios servicios web; donde hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, como las APIs que ayudan en el procesamiento de datos XML los cuales no se deben confundir con HTML.

## ¿Qué es HTML?

HTML (Hypertext Markup Language) es el lenguaje de marcado de documentos para construir páginas web. Todos los comandos del formato son utilizados en los contenidos para web refiriéndose a la estructura de este y al diseño que se mostrará en el navegador; donde las instrucciones incorporadas se conocen como elementos que muestran la estructura

y presentación del documento en el navegador, componiéndose de los tags que definen el formato del texto. Los tags suelen estar dos veces: tag inicial y tag final.

### Diferencias entre XML y HTML

- El XML es un lenguaje de marcado basado en el texto que tiene una estructura de autodescripción, pudiendo definir de forma efectiva otro lenguaje de marcado, mientras el HTML es un lenguaje de marcado predefinido con una capacidad limitada.
- El XML proporciona la estructura lógica del documento, mientras que la estructura del HTML esta predefinida.
- El HTML es insensible a mayúsculas y minúsculas, mientras que el XML las distingue.
- El XML no permite ningún error de código, mientras que, en el HTML, pequeños errores pueden ser descuidados.
- Los espacios en blanco en el XML se utilizan para un fin específico ya que considera todos los caracteres el HTML, en cambio puede ignorar espacios en blanco.
- Los tags de XML están necesariamente cerrados, mientras que en HTML una etiqueta abierta también funciona.
- En XML la sintaxis es de gran importancia, mientras que en HTML no se preocupa mucho por ese aspecto.

### XML Simplifica el Intercambio de Datos

Los datos XML se almacenan en un formato de texto simple, lo cual posibilita una forma independiente de almacenar datos; facilitando la creación de datos que pueden ser compartidos por diferentes aplicaciones. El intercambio de datos como XML reduce en gran medida la complejidad de intercambiar los datos entre sistemas incompatibles a través de internet, ya que los datos pueden ser leídos por diferentes aplicaciones incompatibles.

### Disponibilidad de Datos en XML

Diferentes aplicaciones pueden acceder a los datos, no necesario las páginas HTML, sino también en fuentes de datos XML; los datos pueden estar

disponibles para todos los tipos de máquinas de lectura, facilitando la accesibilidad para personas con capacidades diferentes.

### Tipo de Dato Abstracto (TDA)

Es un modelo que define valores y operaciones que se pueden realizar sobre ellos; se denominan abstractos ya que no requiere conocer los detalles de la representación interna o más bien el cómo están implementadas las operaciones; proveyendo un grado de abstracción que permite desacoplar al código que usa un TDA de aquel código que lo implementa.

### TDA Diccionario

Un diccionario o mapa es una abstracción que vincula un dato con otro; permitiendo tener muchas entradas y cada una de estas tendrá una clave y valor asociado. Este concepto no define el cómo se asociarán dichos datos, por decir que se guardarán internamente los valores; obteniendo una implementación que usa listas, arrays de tuplas, entre otras. Para el código que utiliza el diccionario o mapa es indiferente.

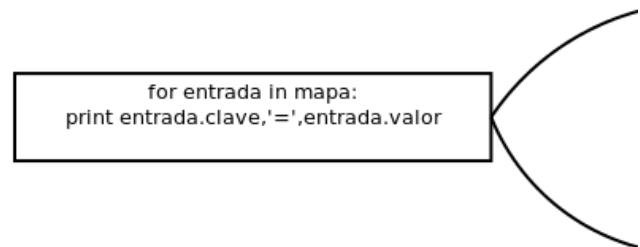


Figura 1. Diccionario TDA.

Fuente: 2021, Tipo de Dato Abstracto – Programación II.

### Listas

Es una estructura dinámica de datos que contiene una colección de elementos homogéneos de manera que se establece entre ellos un orden. Es decir, cada elemento, menos el primero, tiene predecesor y cada elemento, a excepción del ultimo tiene un sucesor. Se puede distinguir a la organización de los nodos, entre:

- Listas Simplemente Enlazadas:** Cada nodo tiene un campo que apunta al siguiente nodo.
- Listas Doblemente Enlazadas:** Cada nodo dispone de un puntero que apunta al siguiente nodo, y otro que apunta al nodo anterior.
- Listas Lineales:** Son listas que tienen un comienzo y un final.
- Listas Circulares:** El último elemento apunta al primero, por lo tanto, se puede recorrer siempre, ya que no tienen final.
- Listas Ortogonales:** En este tipo de lista se utiliza para representar matrices. Los nodos contienen cuatro apuntadores. Uno para apuntar al nodo izquierdo, otro para apuntar al derecho, otro al nodo inferior y por último un apuntador al nodo superior.

#### Lista Ortogonal

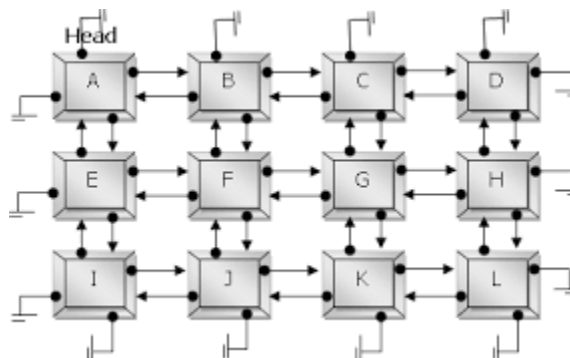


Figura 11. Tipos de Listas.

Fuente: 2021, Listas | Programación avanzada: Estructuras de Datos y Funciones.

#### **Operaciones Básicas sobre una Lista**

Las operaciones básicas que se pueden realizar en cualquier lista, independiente del tipo que sea:

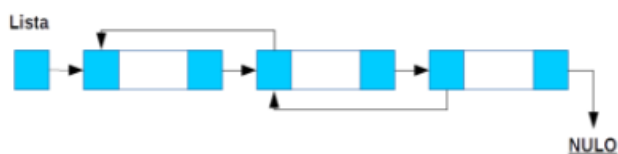
- Crear:** Se genera todo lo necesario para trabajar con una lista.
- Insertar:** Permite añadir un elemento a la lista; indicando al programa si se va a añadir el elemento nuevo al comienzo de la lista o al final de la misma.
- Eliminar:** Borra un elemento de la lista; también se puede indicar si se quiere borrar el primero o el último.
- vaciar:** Devolverá cierto si la lista está vacía.

También se podrán realizar otras operaciones como destruir una lista completamente, contar el número de elementos de la misma, añadir un elemento en una determinada posición, borrar un elemento de una determinada posición, comprobar si hay un dato en la lista, entre otros.

#### Lista simplemente enlazada



#### Lista doblemente enlazada



#### Lista circular



#### **Interfaz**

Es un medio común para que los objetos no relacionados se comuniquen entre sí. Estas son definiciones de métodos y valores sobre los cuales los objetos están de acuerdo para cooperar.

El protocolo es una descripción de:

- Los mensajes que el objeto entiende.
- Los argumentos con los que estos mensajes pueden ser suministrados.
- Los tipos de resultados que devuelven estos mensajes.
- Los invariantes que se conservan a pesar de las modificaciones en el estado de un objeto.
- Las situaciones excepcionales que serán requeridas para ser manejadas por los clientes al objeto.

Si los objetos están totalmente encapsulados, el protocolo describirá la única manera en la cual los objetos pueden ser accedidos por otros objetos.

Algunos lenguajes de programación proporcionan soporte de lenguaje explícito para protocolos o interfaces (Ada, C, D, Dart, Delphi, Go, Java, Object Pascal, Objective-C, PHP, Racket, Seed7, Swift).

En C++ las interfaces son conocidas como clases base abstractas e implementadas usando funciones virtuales puras. Las funciones orientadas a objetos de Perl también admiten interfaces.

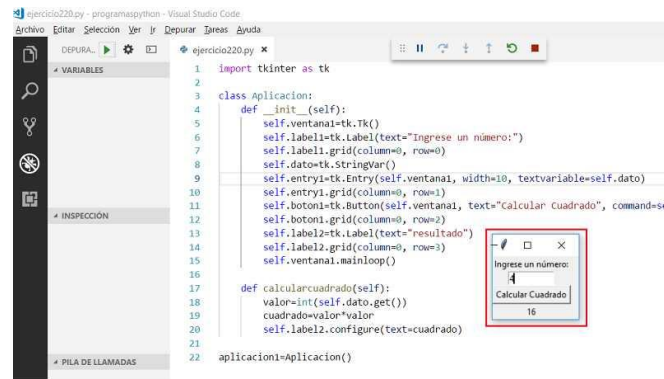
## TKinter

Tk/Tcl ha sido durante mucho tiempo una parte integral de Python. Proporciona un conjunto de herramientas robusto e independiente de la plataforma para administrar ventanas. Disponible para desarrolladores a través del paquete tkinter y sus extensiones, los módulos tkinter.tix y tkinter.ttk.

El paquete tkinter es una capa delgada orientada a objetos encima de Tcl/Tk. Para usar el módulo tkinter, no tiene que escribir ningún código Tcl, pero deberá consultar la documentación de Tk, y ocasionalmente, la documentación de Tcl. El módulo tkinter es un conjunto de funciones que envuelven las implementaciones widgets Tk como

clases de Python. Además, el módulo interno \_tkinter proporciona un mecanismo robusto para que los hilos Python y Tcl interactúen.

Las ventajas de los módulos tkinter son su velocidad y que generalmente se suministra de forma nativa con Python. Aunque su documentación es deficiente, existen otros recursos: referencias, tutoriales, libros y otros. El módulo tkinter también es famoso por su aspecto desactualizado, el cual se ha mejorado mucho desde Tk 8.5. Sin embargo, hay muchas otras bibliotecas GUI que pueden interesarle.



## Conclusiones

Se determinó que, en la implementación de los paradigmas de programación basados en objetos, sirven para la obtención y manipulación de datos dentro de un archivo; facilitando el manejo de los elementos dentro del mismo.

La implementación de los ciclos y condicionales para agregar objetos dentro de una lista ortogonal sirven para la simplificación de código en las mismas, mandando datos por condicionales y funciones llenando estructuras mediante ciclos.

La implementación de archivos XML para su lectura y obtención de sus atributos y elementos mediante ciclos y condicionales, sirven para analizar diferentes tipos de estructuras.

## Referencias bibliográficas

- Anónimo. (2005). *Listas Enlazadas, Pilas y Colas*. Recuperado el 03 de 03 de 2021, de Listas Enlazadas, Pilas y Colas: [http://www.it.uc3m.es/java/2012-13/units/pilas-colas/guides/2/guide\\_es\\_solution.html](http://www.it.uc3m.es/java/2012-13/units/pilas-colas/guides/2/guide_es_solution.html)
- Anónimo. (2008). *Tipo de Dato Abstracto*. Recuperado el 05 de 03 de 2021, de Tipo de Dato Abstracto: <https://sites.google.com/site/programacioniuno/temario/unidad-2---tipo-abstracto-de-dato/tipo-de-dato-abstracto>
- Anónimo. (2012). *Listas enlazadas*. Recuperado el 04 de 03 de 2020, de Listas enlazadas: <https://sites.google.com/site/estdatinfjq/unidad-iii-listas-enlazadas>
- Anónimo. (2014). *Interfaces gráficas de usuario con Tk*. Recuperado el 01 de 04 de 2021, de Interfaces gráficas de usuario con Tk: <https://docs.python.org/es/3/library/tk.html>
- Anónimo. (2014). *tkinter — Python interface to Tcl/Tk*. Recuperado el 01 de 04 de 2021, de tkinter — Python interface to Tcl/Tk: <https://docs.python.org/3/library/tkinter.html>
- Dimeisons, N. (2008). *Listas Enlazadas*. Recuperado el 04 de 03 de 2021, de Listas Enlazadas: <https://www.monografias.com/trabajos101/las-istas-enlazadas/las-istas-enlazadas.shtml>
- Moya, R. (2002). *Estructuras de datos: listas enlazadas, pilas y colas*. Recuperado el 05 de 03 de 2021, de Estructuras de datos: listas enlazadas, pilas y colas.: <https://calcifer.org/documentos/librognome/glib-lists-queues.html>