

---

## EMULADOR DE MISIONES CHAPIN WARRIORS S. A

---

201902888 – Ivan Alessandro Hilario

### Resumen

El presente proyecto se centra en la implementación de tipos de datos abstractos, así mismo con el fin de poder visualizar tipo de dato utilizando archivos de entrada con extensión XML. Un tipo de dato abstracto (TDA) es un conjunto de datos u objetos al cual se le asocian operaciones. El manejo de TDA se utiliza cuando se requiere de un manejo mas adecuado de una cantidad elevada de datos, los cuales precisan de un acceso eficaz y ordenado.

Un TDA puede ser implementado utilizando distintas estructuras de datos y que proveen la misma funcionalidad. Así mismo mostrado el presente proyecto creando e instanciando clase como Lista la cual fue heredada a la clase Ciudad, Unidades Civiles, Unidades Militares, Casillas, Robots, así mismo con la clase Nodo, demostrando y representando cada uno de las Ciudades y Robots implementando el manejo de reutilización de código dentro del programa así mismo demostrando el uso del paradigma de programación orientado a objetos, así mismo usando la herramienta Graphviz para demostrar gráficamente cada ciudad almacenada en Lista.

### Palabras clave

Tipos de datos abstracto, programación orientada a objetos, archivos XML, librería Graphviz, estructuras de programación.

### Abstract

*This project is focused on the implementation of abstract data types, in order to be able to visualize data types using input files with XML extension. An abstract data type (ADT) is a set of data or objects to which operations are associated. The handling of TDA is used when a more adequate handling of a large amount of data is required, which need an efficient and orderly access.*

*A TDA can be implemented using different data structures that provide the same functionality. Likewise shown the present project creating and instantiating class as List which was inherited to the class City, Civil Units, Military Units, Boxes, Robots, likewise with the class Node, demonstrating and representing each one of the Cities and Robots implementing the management of code reuse within the program as well as demonstrating the use of the object oriented programming paradigm, likewise using the Graphviz tool to graphically demonstrate each city stored in List.*

### Keywords

*Abstract data types, Object-oriented programming, XML files, Graphviz library, programming structures.*

## Introducción

En ciencias de la computación un tipo de dato abstracto (TDA) se define como un modelo matemático compuesto por una colección de operaciones definidas sobre un conjunto de datos para el modelo. Por lo cual, el uso de TDA para la realización de un analizador de datos es lo más ideal, ya que se cuenta con una manera más eficiente de acceder a los datos con el fin de poder operarlos a conveniencia del desarrollador.

El presente ensayo explica de mejor manera la forma en la que se implementaron las estructuras de datos utilizadas, asimismo expandir la teoría que no se puede plasmar en un proyecto de programación y explicar las razones por las que se optó por escoger ciertas tecnologías utilizadas en el desarrollo del proyecto de misma manera implementado ciertas librerías para el uso grafico dentro de la aplicación

## Desarrollo del tema

Como todo proyecto el software tiene un ciclo para desarrollarse y consta de una serie de pasos que se van completando en diferentes tiempos; este ciclo de desarrollo de software depende directamente de la metodología que utilizamos para este desarrollo, y no es más que una serie de pasos/tareas que tenemos que seguir como en cualquier otro proyecto. Los pasos utilizados fueron:

- Leer y entender el proyecto
- aprender lenguajes a utilizar
- investigar temas de los cuales surgen dudas
- implementar todo en el ámbito de la programación

Así mismo usando el paradigma de programación orientado a objetos implementando las clases necesarias para el manejo de información.

La (poo) fue de suma importancia en este proyecto ya que con ella se implementaron clases con las cuales se pudieron crear las listas enlazadas, aparte también se crearon objetos con propiedades para cada nodo esto con el fin de almacenar la información de la matriz

dentro de otro nodo y así poder agrupar en cada nodo la información de una sola matriz del archivo xml. Con lo mencionado anteriormente se da a entender que cada nodo contiene un objeto dentro de él.

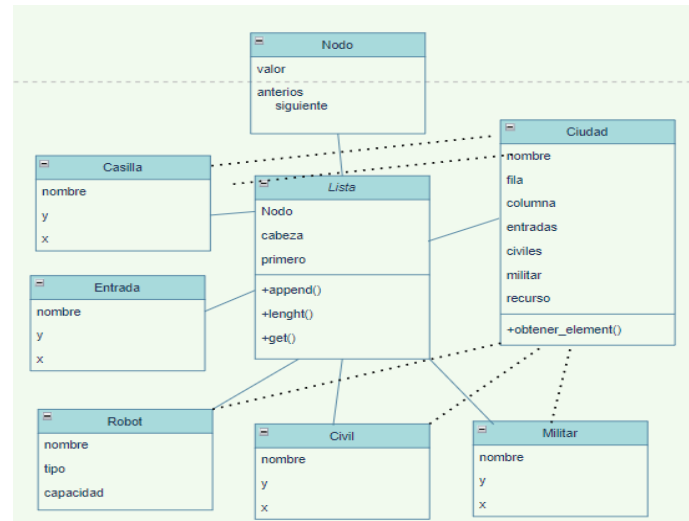


Figura 1. Diagrama de clases.

Fuente: elaboración propia.

## Estructura de Datos

En ciencias de la computación, una estructura de datos es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas. Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indexación de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes. Las estructuras de datos se basan generalmente en la capacidad de un ordenador para recuperar y almacenar datos en cualquier lugar de su memoria

Existen numerosos tipos de estructuras de datos, generalmente construidas sobre otras más simples:

utilizarse para representar redes, dado que los nodos pueden referenciarse entre ellos. Las conexiones entre

nodos pueden tener dirección, es decir un nodo de partida y uno de llegada.

a) Un árbol es un caso particular de grafo dirigido en el que no se admiten ciclos y existe un camino desde un nodo llamado raíz hasta cada uno de los otros nodos. Una colección de árboles es llamada un bosque.

b) Una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan como representación abstracta de conceptos, incluyen campos como los registros y operaciones que pueden consultar el valor de los campos o cambiar sus valores.

### Características de XML

- El XML separa datos de HTML: Si necesita mostrar datos dinámicos en su documento HTML, tendrá que dedicarle mucho trabajo a editarlos cada vez que los datos cambien. Con el XML, los datos se pueden almacenar en archivos XML separados. De esa manera, puedes usar HTML para la visualización y el diseño. Con algunas líneas de código JavaScript, puedes leer un archivo XML externo y actualizar el contenido de los datos de tu página web.

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web. Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que ayudan en el procesamiento de datos XML – que no deben confundirse con HTML

```
1 <persona>
2   <nombres>Elsa</nombres>
3   <apellidos>Zambrano</apellidos>
4   <fecha-de-nacimiento>
5     <día>18</día>
6     <mes>6</mes>
7     <año>1996</año>
8   </fecha-de-nacimiento>
9   <ciudad>Pamplona</ciudad>
10</persona>
```

Figura 2. Estructura

XML. Fuente: ResearchGate.

### Interfaz Grafica (Tkinter)

Tkinter es el paquete más utilizado para crear interfaces gráficas en Python. Es una capa orientada a objetos basada en Tcl (sencillo y versátil lenguaje de programación open- source) y Tk (la herramienta GUI estándar para Tcl).

Se considera un estándar para la interfaz gráfica de usuario para Python y es el que viene por defecto con la instalación para Microsoft Windows

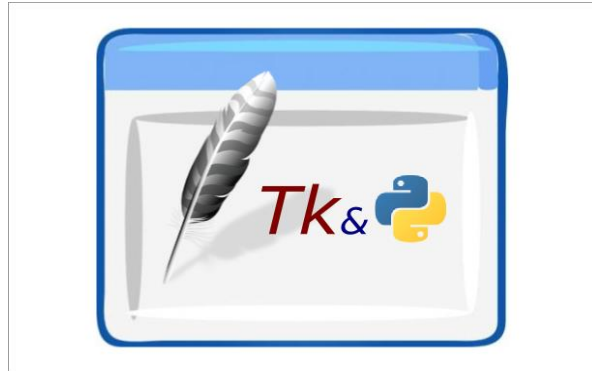


Figura 3. Librería.

Fuente: Documentación Python

### Utilización de graphviz:

Esta librería tuvo una gran importancia al final en nuestro proyecto ya que con ella podemos representar de una manera gráfica diferentes datos en este caso fueron las ciudades que se nos proporcionan y con esto hacer que el usuario pueda entender de una mejor manera lo que se está realizando, esto conlleva colocar en cada una de las ciudades graficadas el nombre, valor de las filas, valor de las columnas y por último colocar el valor que representa cada posición dada en la ciudades entre casillas.

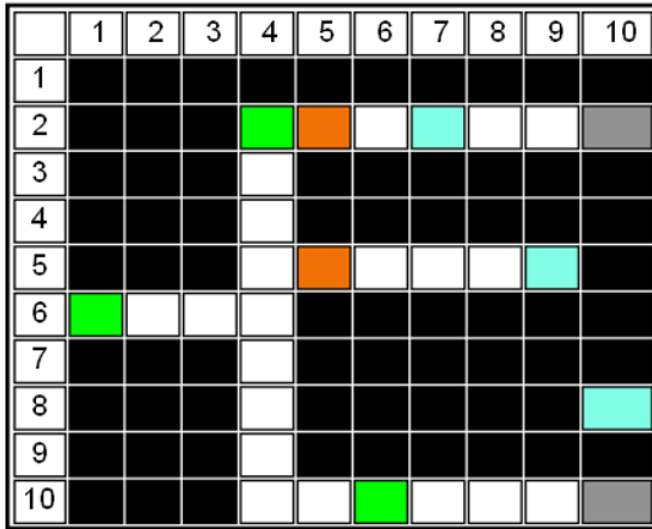


Figura 4. Imagen de ciudad

Fuente: Elaboración propia.

## Conclusiones

Implementar un tipo de dato abstracto permite la comprensión de la correcta utilización del paradigma de programación orientado a objetos.

Manejar datos matricialmente y convertirlos visualmente en un grafo favorece la precepción y facilita el aprendizaje para futuros eventos en los cuales se vean involucrados grandes cantidades de datos.

La implementación de nodos es una gran herramienta para poder guardar datos ya que dentro de cada uno de ellos puede haber más nodos que pertenecen a este mismo objeto

## Referencias bibliográficas

Charles R. Severance, (2009), Python para todos

Jonathan Bona, Michael Prentice, PyRovio: Python API for WowWee Rovio, 2009.

Python Software Foundation, <http://www.python.org>, [Online; accedido el 6- Marzo-2011]

