

---

## Analizador de Muestras

---

201801027 – Abner Martín Noj Hernández

### Resumen

Este ensayo presenta el desarrollo de una aplicación que brinda los servicios de análisis de muestras recolectadas en diferentes lugares, para identificar el tipo de organismos que existen y que puedan existir en diferentes muestras. Para el manejo de la información se utiliza archivos XML debido a que se puede describir los datos de forma estructurada permitiendo así una mayor legibilidad y también porque se puede integrar fácilmente con otras tecnologías como bases de datos, aplicaciones web, servicios web, y otros lenguajes de programación. La aplicación fue desarrollada en el lenguaje de programación Python y se utilizaron los tipos de datos abstractos (TDA) para el almacenamiento y procesamiento de toda la información. Además, se utilizó Graphviz para la creación de imágenes para mostrar gráficamente cada muestra ingresada y poder visualizar el comportamiento de cada organismo. El manejo de esta aplicación es por medio de consola para que el usuario pueda interactuar de la forma más agradable e intuitiva.

### Palabras clave

Python, Lenguaje de Marcado extensible (XML), Tipos de Datos abstractos (TDA), Programación Orientada a Objetos (POO), Graphviz

### Abstract

. This essay presents the development of an application that provides the services of analysis of samples collected in different places, to identify the type of organisms that exist and that may exist in different samples. For information management, XML files are used because the data can be described in a structured way, thus allowing greater readability and also because it can be easily integrated with other technologies such as databases, web applications, web services, and other languages. The application was developed in the Python programming language and abstract data types (ADT) were used to store and process all the information. In addition, Graphviz was achieved for the creation of images to graphically display each sample entered and to be able to visualize the behavior of each organism. The management of this application is through a console so that the user can interact in the most pleasant and intuitive way.

### Keywords

Python, Extensible Markup Language (XML), Abstract Data Types (ADT), Object Oriented Programming (OOP), Graphviz

## Introducción

Existen diferentes tipos de organismos que viven a nuestro alrededor algunos pueden ser benignos otros malignos, es por ello la importancia de identificar cual es cada uno. Es por esto que el laboratorio nacional de Guatemala (LNG) ha tomado diferentes muestras y ha solicitado un programa para conocer el comportamiento de cada organismo dependiendo del lugar donde se encuentre. Para ello realizó el desarrollo de una aplicación de consola que brinda los servicios de lectura, análisis, proyección de vida y creación de grafico de las muestras. Para el ingreso y la escritura de información se optó por el uso de archivos XML debido a la amplia compatibilidad que tiene con diferentes aplicaciones, lenguajes de programación, además de su facilidad de lectura y escritura. Para el almacenamiento y procesamiento de información se utilizaron los tipos de datos Abstractos, en específico: Listas Simples, debido a la facilidad del manejo de información y también la facilidad que se puede obtener para el mantenimiento de la aplicación, así como de futuras actualizaciones.

## Desarrollo del tema

Los tipos de datos abstractos (TDA) es una forma en que se puede definir una estructura de datos donde se especifica cómo se puede usar los datos y que operaciones se pueden realizar entre ellos. Todo esto por medio de la programación orientada a objetos.

Este programa de Analizador de muestras permite el recibir la información, procesarla y devolver un resultado que se muestra en forma gráfica además de poder escribir un archivo XML en donde contenga toda la información que fue procesada. Al iniciar esta aplicación se le mostrará un menú con diferentes opciones. Primero se debe seleccionar la opción 1

para poder cargar la información y así acceder a todas las funcionalidades que contiene este programa. El archivo XML tiene que tener la siguiente forma:

```
<?xml version="1.0"?>
<datosMarte>
  <listaOrganismos>
    <organismo>
      <codigo>A001df</codigo>
      <nombre>Org01</nombre>
    </organismo>
  </listaOrganismos>
  <listadoMuestras>
    <muestra>
      <codigo>M01Marte</codigo>
      <descripcion>Primera muestra</descripcion>
      <filas>15</filas>
      <columnas>20</columnas>
      <listadoCeldasVivas>
        <celdaViva>
          <fila>3</fila>
          <columna>4</columna>

          <codigoOrganismo>A001df</codigoOrganismo>
          </celdaViva>
          <celdaViva>
            <fila>3</fila>
            <columna>5</columna>

          <codigoOrganismo>A001df</codigoOrganismo>
          </celdaViva>
          </celdaViva>
        </listadoCeldasVivas>
      </muestra>
    </listadoMuestras>
  </datosMarte>
```

Figura 1. Estructura del XML

Fuente: Elaboración Propia

Al cargar este archivo se obtiene el contenido de las etiquetas y estas a su vez se almacenan en las listas simples.

EL diagrama de la estructura del funcionamiento de la lista simple es de la siguiente manera:

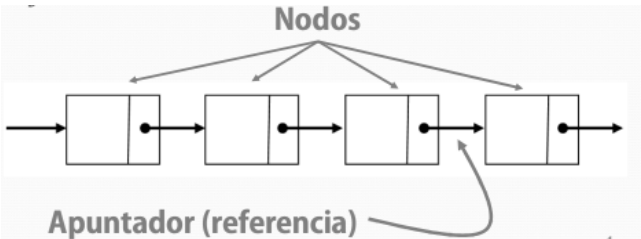


Figura 2. Modelo tipo dato Celda Viva  
Fuente: Elaboración Propia

En donde cada nodo contiene dos atributos uno de ellos es el valor del objeto creado de acuerdo al tipo de etiqueta y el otro es el apuntador que hace referencia al siguiente nodo.

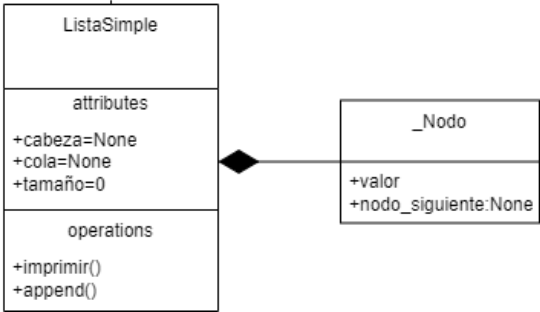


Figura 3. Modelo tipo dato Celda Viva  
Fuente: Elaboración Propia

Luego se puede administrar la información ingresando nuevos organismos dentro de la muestra cargada y el programa le mostrara el comportamiento de cada célula si vive o muere en el entorno que se encuentra.

Se utilizó la programación orientada a objetos para crear diferentes clases para poder almacenar y procesar la información que se obtuvo del archivo XML

Estas son las clases utilizadas:

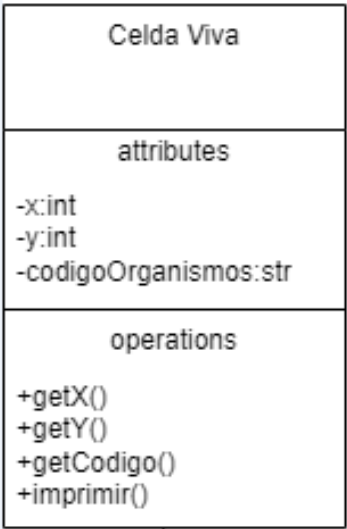


Figura 4. Modelo tipo dato Celda Viva  
Fuente: Elaboración Propia

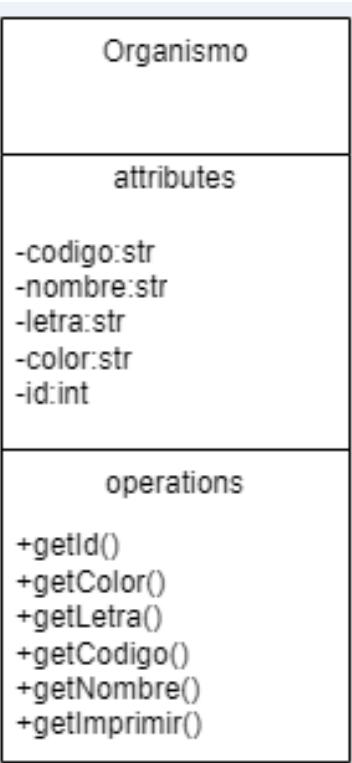


Figura 5. Modelo tipo dato Organismo  
Fuente: Elaboración Propia



Figura 6. Modelo tipo dato Muestra

Fuente: Elaboración Propia

Luego de ser almacenada la información se procede a realizar la gráfica de la muestra.

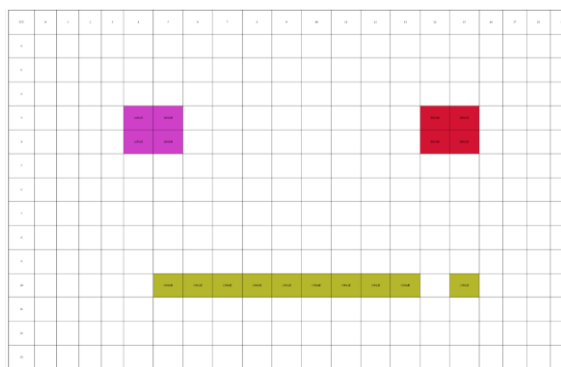


Figura 7. Grafica de una muestra

Fuente: Elaboración Propia

Entre las librerías utilizadas en el programa fueron:

- elementTree: Librería para la lectura y escritura del XML
- tkinter: librería para abrir una ventana emergente para seleccionar el archivo
- os: Librería para la escritura de archivos, y para abrir el archivo
- random: librería para obtener utilizada para obtener números aleatorios y así obtener un color hexadecimal para asignarlos a cada organismo

## Conclusiones

Las listas simples son muy útiles ya que nos ayudan a tener un mejor manejo de memoria, debido a que estas listas son dinámicas, eso significa que el tamaño de nuestra lista puede ir aumentando o disminuyendo en tiempo de ejecución.

El conocer más tipos de TDA para el manejo de información permite un mejor rendimiento, por ejemplo, si se hubiera utilizado una matriz dispersa en este proyecto en lugar de la lista simple tiempo de ejecución para las funciones sería menor que usando las listas simples. Pero esto requiere un mayor conocimiento de una matriz dispersa porque su funcionamiento es más complejo que una de una lista simple

Graphviz es una gran herramienta para poder representar gráficamente cualquier información, debido a que se puede personalizar cada nodo en el color forma, tamaño y esto permite que sea más atractivo para el usuario.

## Referencias bibliográficas

Castell, R.F. (no date) *Tipos de Datos, estructuras de Datos y Tipos abstractos de datos - UV*. Available at: [http://informatica.uv.es/iiguia/AED/oldwww/2001\\_02/Teoria/Tema\\_10.pdf](http://informatica.uv.es/iiguia/AED/oldwww/2001_02/Teoria/Tema_10.pdf) (Accessed: February 31, 2023).

Takeyas, B.L. (no date) *Listas Simples, Instituto Tecnol.* Instituto Tecnológico de Nuevo Laredo . Available at: <http://www.itnuevolaredo.edu.mx/takeyas/> (Accessed: February 31, 2023).

Arimerics (2021) *Qué Es xml - definición, Significado y Ejemplos, Arimerics*. Available at: <https://www.arimerics.com/glosario-digital/xml> (Accessed: February 31, 2023).

Lahtela, M. and Kaplan, P.(P. (1966) *ES, Amazon*. Oberbaumppresse. Available at: <https://aws.amazon.com/es/what-is/xml/#:~:text=El%20lenguaje%20de%20marcado%20extensible,datos%20y%20aplicacion es%20de%20terceros>. (Accessed: February 31, 2023).

Gomez, O.A. (2014) *Visualizando Grafos , OSiUX*. Available at: <https://osiux.com/visualizando-grafos-graphviz.html> (Accessed: February 31, 2023).

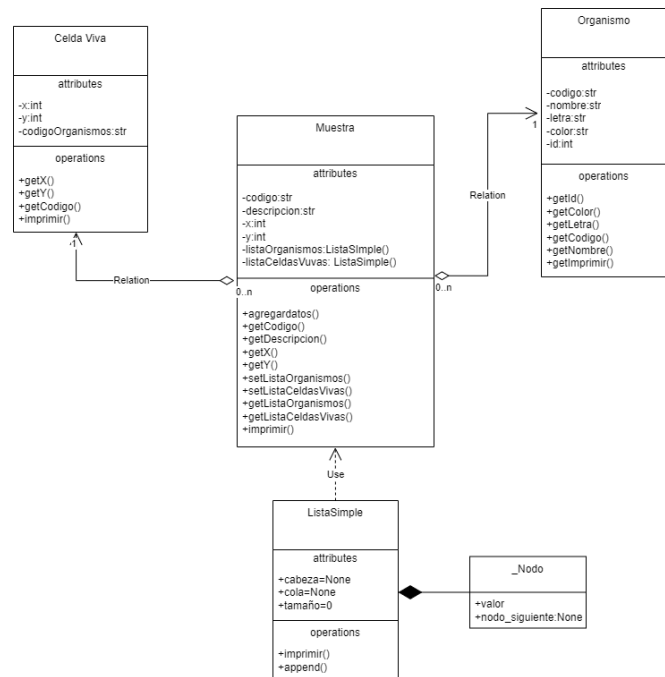


Figura 8. Diagrama de Clases

Fuente: Elaboración Propia

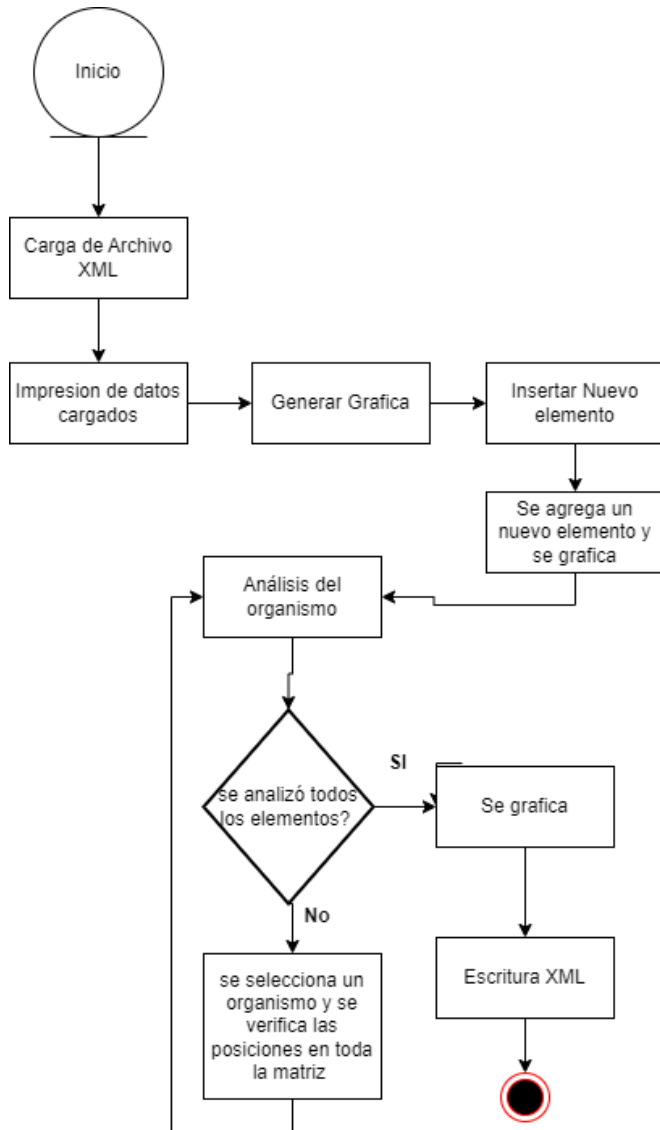


Figura 9. Diagrama de Actividades

Fuente: Elaboración Propia