
PROYECTO 2

201602734 – Wilfred Alejandro Barrios Ola

Resumen

El presente proyecto académico presenta, mediante el uso del lenguaje de programación Python para crear un aplicativo de escritorio, la implementación de TDA aplicados a memoria dinámica y uso de programación orientada a objetos y procesamiento de archivos XML.

Adicional a esto se acompaña de material gráfico para que sea más sencilla la comprensión de los temas a tratar a continuación. De esta manera facilitar el entendimiento del uso de TDA, programación orientada a objetos, archivos XML y el uso de grafos, para este caso específico una cola de secuencia.

Palabras clave

Python, TDA, Grafo, XML, POO

Abstract

The present academic project was created using Python as the programming language used to create a desktop app. The main idea of the project is giving an example of how to implement an ADT applicated to dynamic memory, Oriented Object Programming and XML files processing.

Also, the present inform comes with graphic material to make easier the comprehension of the next topics. In that way make easier the understating of how to use an ADT, OOP, XML Files and how to use graphs, for this specific case a sequency queue.

Keywords

Python, ADT, Graph, XML, OOP

Introducción

El desarrollo de un sistema está compuesto por una gran diversidad de factores, desde el lenguaje de programación a utilizar hasta la interacción correcta del usuario con estos mismos.

Factores principales en el proceso de desarrollo son los paradigmas a utilizar para facilitar el desarrollo y su mantenibilidad posterior a la finalización, los tipos de datos tanto primitivos como definidos por el programador. La correcta utilización de la memoria, ya sea estática o dinámica pues está tendrá un impacto significativo en la ejecución del programa dependiendo de las limitaciones del hardware y la cantidad de recursos destinados a la ejecución del programa desarrollado.

También vemos, en menor medida, como de importante es la teoría de grafos para las ciencias de la computación pues de esta teoría obtenemos las listas (Enlazadas, circulares, doblemente enlazadas, etc) y el uso que se le puede dar a documentos XML en el desarrollo de un sistema, independientemente de su propósito.

Marco Teórico

Programación Orientada a objetos

La programación orientada a objetos es uno de los paradigmas más conocidos y utilizados en el desarrollo de software.

Este paradigma está basado en el concepto de clases y objetos, facilitando el estructurar un proyecto en piezas simples y reutilizables de planos de código para crear instancias individuales de un objeto.

En palabras sencillas, una clase es una plantilla para crear objetos. Esta plantilla tendrá propiedades y métodos, que definiran los atributos del objeto y las acciones que este mismo puede realizar.

La siguiente imagen se incluye con fines ilustrativos.

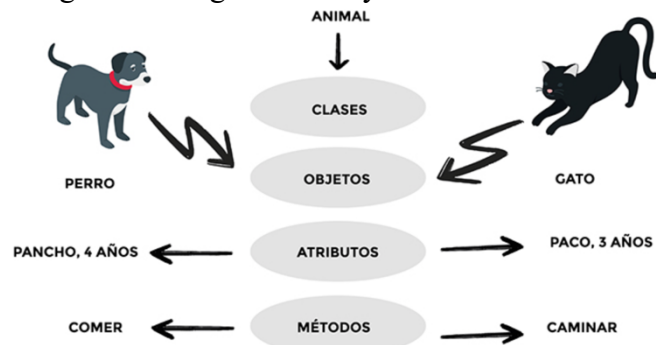


Figura 1. Clase y objeto.

Fuente: <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/> 2 de noviembre de 2020.

Además de esto, se compone por 4 grandes pilares: abstracción, encapsulamiento, herencia y polimorfismo.

La abstracción se define como separar por medio de una operación intelectual una cualidad de algo y considerarla aislada de esta misma.

El encapsulamiento es, en pocas palabras, el ocultamiento del estado de manera que estos sólo pueden ser modificados por medio de los métodos definidas para este.

La herencia se utiliza para crear una clase a partir de una existente, de manera que la subclase obtendrá todo lo existente en la primera, siempre y cuando la primera lo permita.

El polimorfismo se define por la capacidad que tiene un objeto de tomar muchas formas, de acuerdo a los parámetros que se le envían.

Tipos de dato Abstracto (TDA)

Un tipo de dato abstracto es, por definición, un modelo matemático compuesto por una colección de

operaciones definidas sobre un conjunto de datos para el modelo.

Este tipo de dato es uno propio definido por el programador luego de abstraer la información completamente necesaria para el desarrollo de un sistema y limitando el modelo únicamente a información que será necesaria para el correcto funcionamiento del programa.

Esto después puede definirse como clase y por consiguiente objeto, cuando se trabaja con el paradigma de programación orientada a objetos.

Un claro ejemplo podemos verlo en la figura 1, que presenta un TDA de un animal, definiendo atributos y métodos necesarios para el correcto funcionamiento del sistema.

Memoria dinámica

La memoria dinámica se refiere a aquella memoria que no puede ser definida ya que se desconoce la cantidad necesaria para el funcionamiento, por lo tanto esta misma puede incrementarse o decrementarse de acuerdo a lo que se agrega.

Claros ejemplos de la memoria dinámica son pilas, colas, listas enlazadas, listas circulares, y demás tipos de listas que presentan un crecimiento y decrecimiento de acuerdo a las necesidades requeridas por el software.

Grafo

En el ámbito de las matemáticas y las ciencias de la computación, un grafo es un conjunto de objetos llamados nodos que se unen por medio de aristas, estas permiten representar relaciones binarias entre elementos de un conjunto.

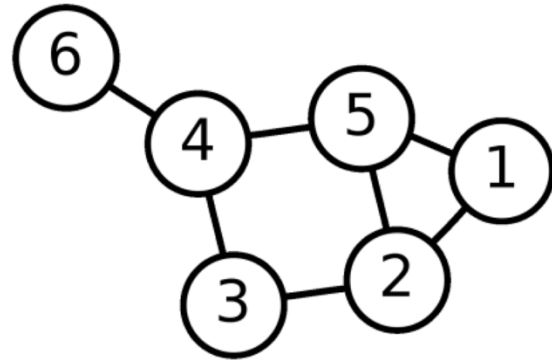


Figura 2. Diagrama de un grafo con 6 nodos y 7 aristas
Fuente: https://www.unipamplona.edu.co/unipamplona/portallG/home_23/recursos/general/11072012/graf03.pdf 11 de Julio de 2012

XML

Es un lenguaje de marcado similar a HTML. Es una especificación de W3C como lenguaje de marcado de propósito general. Esto significa que XML no está predefinido, por lo que es necesario definir las propias etiquetas dependiendo del propósito del XML a utilizar.

Uno de los principales propósitos es compartir datos a través de diferentes sistemas.

Para que un documento XML sea correcto, debe ser un documento bien formado, cumpliendo con todas las reglas de sintaxis de XML, y válidas, acordes a las reglas de un lenguaje específico.

```
<message>
  <warning>
    Hola, mundo
  </warning>
</message>
```

Figura 3. Documento XML correcto

Fuente: https://developer.mozilla.org/es/docs/Web/XML/XML_introduccion 25 de septiembre de 2021.

Conclusiones

De acuerdo a lo anteriormente presentado, podemos concluir que:

La programación orientada a objetos es uno de los paradigmas más utilizados debido a su facilidad de integración y que el nivel de abstracción que el programador debe tener no es tan alto como utilizando otros paradigmas, además brinda herramientas robustas para el desarrollo completo de un software.

Los tipos de datos abstractos son tipos de datos definidos por el programador de acuerdo a las necesidades del modelo, realizados luego de abstraer lo estrictamente necesario para el correcto funcionamiento del software. No tienen una estructura definida pues esta será definida por el programador de acuerdo a las necesidades del proyecto.

La memoria dinámica es utilizada cuando la memoria estática se ve limitada, pues esta puede incrementar o decrementar de acuerdo a lo que se necesita, pero es importante tomar en cuenta que afectará el funcionamiento del sistema si no se realiza de manera óptima.

Los grafos son una base fuerte para la creación de listas, pues los conceptos de nodos y aristas son los mismos aplicados en listas de datos abstractos.

El lenguaje XML puede ser utilizado independientemente del propósito, pues no tiene una estructura definida y se acopla a las necesidades del proyecto. Dependerá del programador brindarle la estructura necesaria para la finalidad correspondiente del mismo.

Referencias bibliográficas

Universidad de Pamplona (2012). Teoría de grafos.

https://www.unipamplona.edu.co/unipamplona/portallG/home_23/recursos/general/11072012/grafos3.pdf

Mozilla Contributors (2021) Introducción a XML.

https://developer.mozilla.org/es/docs/Web/XML/XML_introduction

Profile (2020) ¿Qué es la programación orientada a objetos?

<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

Desconocido (2021) Tipo de Dato Abstracto

<https://sites.google.com/site/programacioniiuno/temario/unidad-2---tipo-abstracto-de-dato/tipo-de-dato-abstracto>

Universidad Politécnica de Cataluña (2019 – 2020)
Gestión de Memoria

<https://docencia.ac.upc.edu/FIB/grau/SO2/documents/tema6.pdf>

Diagrama de clases - Proyecto 2

Alejandro Barrios | September 26, 2021

