

---

## PROYECTO 1 – PISOS ARTESANALES S.A.

---

202004745 – Allen Giankarlo Román Vásquez

### Resumen

El proyecto 1 "Pisos Artesanales" se realiza para poner en práctica los conocimientos adquiridos en el laboratorio y clase magistral del curso "Introducción a la Programación y Computación 2" de la carrera de Ingeniería en Ciencias y Sistemas de la Universidad San Carlos de Guatemala creando un proyecto en el lenguaje de programación "Python" implementando las funciones básicas del lenguaje de programación, también aplicando los temas y conceptos teóricos aprendidos en el laboratorio, por último creando otro tipo de conocimiento para el estudiante al manejar distintos tipos de archivos, ya sean archivos de entrada para cargar en la memoria, como de salida para creación de reportes del proyecto. El proyecto contempla la implementación de TDA en una aplicación que otorgue una solución a una problemática en la vida real. Para finalizar para este proyecto se aplican los conceptos de: Lista, Matriz, Listas enlazadas, Listas doblemente enlazadas, Apuntadores, Nodos, Archivos XML, Graphviz entre otros.

### Palabras clave

Matriz, Lista, Apuntadores, Listas enlazadas, XML.

### Abstract

*The Project 1 "Pisos Artesanales" was performed for performance the knowledge acquired in the laboratory class and the masterly class "Introduction to programming and computation 2" of the Science and systems Engineering career from the University "San Carlos de Guatemala" creating a project with programming language "Python" implementing the basic functions of the programming language, also applying the theoretical topics and concepts learned in the laboratory class, and lastly, creating new knowledge for the student handling different types of files like input files to load into memory, as well as output files for creating project reports. The project contemplates the implementation of ADT in an application that provides a solution to a problem in real life. To finish, this project applies the concepts of: List, Array, Linked Lists, Doubly Linked Links, Pointers, Nodes, XML Files, Graphviz, among others.*

### Keywords

*Array, List, Pointers, Linked lists, XML.*

## Introducción

La aplicación “Pisos Artesanales” consiste en un programa que cuenta con una interfaz gráfica amigable para el usuario, permitiendo el cálculo de precios mínimos para crear nuevos patrones de pisos y como el orden de estas. También permite al usuario visualizar los nombres y códigos de los patrones de los pisos, todo esto con una carga por medio de un archivo XML

## Desarrollo del tema

### Python

Python es un lenguaje de programación interpretado cuya principal filosofía es que sea legible por cualquier persona con conocimientos básicos de programación. Entre las principales características que cuenta este lenguaje de programación están:

- Es totalmente gratuito.
- Esta respaldado por una enorme comunidad.
- Es un lenguaje multiparadigma (combina muchos paradigmas de programación)
- Sus aplicaciones no se limitan a solo un área en concreto
- Es apto para todas las plataformas

Fue desarrollado por el investigador holandés Guido van Rossum, quien en aquellas épocas trabajaba en el centro de investigación CWI (Centrum Wiskunde & Informatica) de Ámsterdam.

A van Rossum le fue asignado un proyecto en el que tenía que desarrollar un sistema operativo distribuido llamado Amoeba, las condiciones en que se encontraba el CWI, es que tenía implementado un lenguaje de programación llamado ABC.

Guido decidió crear un nuevo lenguaje para el proyecto Amoeba, para superar las limitaciones y problemas con los que se encontró. Una vez presentado Python, CWI decidió liberar el intérprete del lenguaje bajo una licencia open source propia.

En tanto, para el inicio del siglo XXI y coincidiendo con la publicación de la versión 1.6, se tomó la decisión de cambiar la licencia por una que sea compatible con GPL (General Public Licence). Esta nueva licencia se denominará Python Software Foundation Licence.



*Figura 1. Logo de Python*

Fuente: Python Software Foundation. (2020)

### Listas

Una lista es una estructura dinámica de datos que contiene una colección de elementos homogéneos (del mismo tipo) de manera que se establece entre ellos un orden. Es decir, cada elemento, a excepción del primero, tiene un predecesor, y cada elemento, menos el último, tiene un sucesor.

Las listas usualmente son implementadas tanto como listas enlazadas (simple o doblemente enlazada), o como una matriz (array), normalmente de longitud variable (dinámico). Pueden ser implementada como arboles binarios de búsqueda balanceados manteniendo pares de índice valor, proporcionando acceso en el mismo tiempo a cualquier elemento.

### Listas simplemente enlazadas

Una lista simplemente enlazada es una estructura de datos en la que cada elemento apunta al siguiente. De este modo, teniendo la referencia del principio de la lista podemos acceder a todos los elementos de la misma.

Este TDA se caracteriza por ser de estructura dinámica, por lo que no es necesario conocer los elementos que puede contener.

Los operadores básicos de una lista simplemente enlazada son:

- Insertar
- Eliminar
- Buscar
- Localizar
- Vaciar

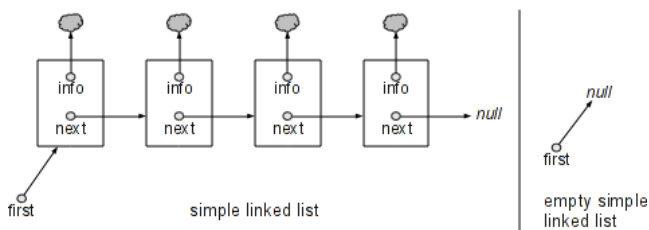


Figura II. Ejemplo de lista simplemente enlazada

Fuente: Universidad Carlos III de Madrid, (2011), Página web propia

### Listas doblemente enlazadas

Una lista doblemente enlazada es una estructura de datos en la que cada elemento apunta a dos direcciones, hacia adelante y atrás específicamente.

Los operadores básicos de una lista doblemente enlazada son:

- Eliminar
- Eliminar ultimo
- Agregar inicio
- Agregar siguiente



Figura III. Ejemplo de lista doblemente enlazada

Fuente: J. Valdivia, (2005), Página web propia

### Punteros

Un puntero es una variable que referencia una región (dirección) de memoria de la siguiente manera: Existe la variable “p” de tipo puntero que contiene una dirección de memoria en la que se encuentra almacenado el valor “v”, diciendo que “p” apunta a “v”.

### Nodos

Un nodo es uno de los elementos de una lista simple o doblemente enlazada. Cada nodo será una estructura o registro que dispondrá de varios campos donde al menos uno de esos campos es un puntero o referencia a otro nodo, de forma que, conocido un nodo será posible tener acceso a otros nodos de la estructura. Los nodos son herramientas esenciales.

### Graphviz

Graphviz es un software de creación y visualización grafica código abierto con la capacidad de crear una extensa cantidad de tipos de diagramas de gráficos de cualquier tipo y uso.

La visualización de gráficos es una forma de representar información estructural como diagramas de gráficos y redes abstractas. Tiene importantes aplicaciones en redes, bioinformática, ingeniería de software, diseño de bases de datos y web, aprendizaje automático en interfaces visuales para otros dominios técnico.

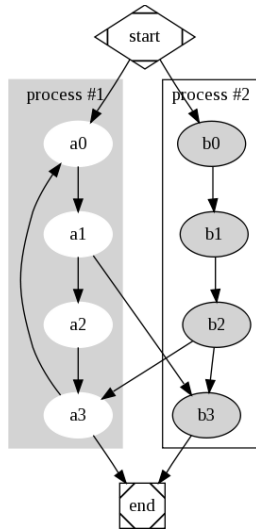


Figura IV. Ejemplo grafico generado con Graphviz

Fuente: Graphviz, (2021), Página web propia

Graphviz toma descripciones de los gráficos en un lenguaje de texto simple y realiza los diagramas en los formatos necesarios como imágenes y archivos de extensión .svg para ser visualizado en páginas web.

Graphviz tiene muchas funciones útiles para diagramas concretos, como opciones para colores, fuentes, diseños de nodos tabulares, estilos de líneas, hipervínculos, formas personalizadas entre otros.

```
digraph M{{
    node[shape=circle fillcolor="#FFFFFF" style=filled]

    subgraph cluster{{
        label= "{nombreterrenoformateado}"
        raiz[label = "0,0"]
        edge[dir = "none"]
        /* creacion cabeceras filas*/
        F1[label="1", group=1];
        F2[label="2", group=1];
        F3[label="3", group=1];
        F4[label="4", group=1];
        F5[label="5", group=1];
        /* enlaces nodos de las filas */
        F1->F2;
        F2->F3;
        F3->F4;
        F4->F5;
        /*enlazar nodos de columnas*/
        C1[label = "1", group=2];
        C2[label = "2", group=3];
        C3[label = "3", group=4];
        C4[label = "4", group=5];
        C5[label = "5", group=6];
    }}
```

Figura V. Ejemplo código de Graphviz para crear un diagrama gráfico.

Fuente: Elaboración propia.

## Archivos XML

XML, o eXtensible Markup Language en inglés, es un metalenguaje que fue diseñado para estructurar, almacenar y para intercambiar datos entre distintas aplicaciones, convirtiéndose en un estándar ya que es extensible y puede ser utilizado por cualquier aplicación independientemente de la plataforma. Tiene una serie de características que lo hacen un estándar internacional, las cuales son:

- Eficiente
- Libre
- Manejable
- Simplifica el intercambio de datos

```
<terreno nombre="terreno1">
|   <posicioninicio>
|       <x>7</x>
|       <y>9</y>
|   </posicioninicio>
|   <posicionfin>
|       <x>7</x>
|       <y>15</y>
|   </posicionfin>
|   <combustible>7</combustible>
|   <posicion x="1" y="1">1</posicion>
|   <posicion x="1" y="2">1</posicion>
|   <posicion x="3" y="2">1</posicion>
|   <posicion x="5" y="2">1</posicion>
|   <posicion x="5" y="3">1</posicion>
|   <posicion x="5" y="4">1</posicion>
|   <posicion x="5" y="5">1</posicion>
| </terreno>
```

Figura VI. Ejemplo de un archivo XML

Fuente: Elaboración propia.

Un XML está conformado por las siguientes partes:

- Prologo: Se especifica que el documento en cuestión es XML, se define su versión y la codificación de caracteres que se utilizó en el archivo.
- Elementos: se tiene una etiqueta o tag inicial y un tag final

- Epilogo: Datos adicionales como encriptación. Generalmente los tokens son agregados en este apartado

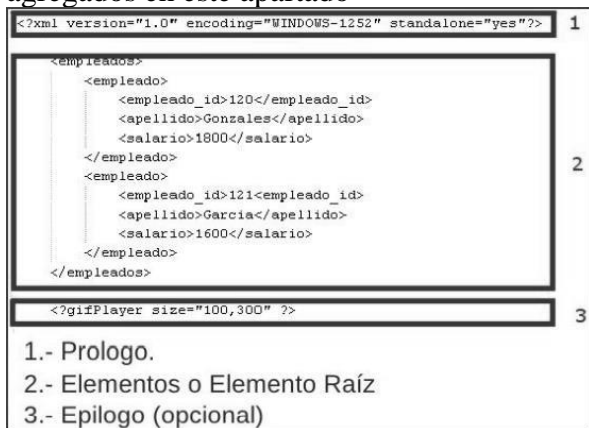


Figura VII. Partes de un archivo XML

Fuente: M. Muñoz, (2021), Laboratorio de “Introducción a la Programación y computación” sección “E”.

## Conclusiones

Los distintos TDA presentados aportan un manejo de datos eficientes en la memoria del computador, permitiendo un mejor acceso a los datos de mejor manera que otros TDA que están quedando obsoletos o se vuelven ineficientes con la evolución de los datos a lo largo de los años

Graphviz es una herramienta de diseño de estructuras graficas dinámico que permite crear un sinfín de estructuras graficas tanto solo una vez, como en ejecución con distintos datos que pueden ser proporcionados.

Los archivos XML se han convertidos en estándares en el mundo de la programación ya que con su versatilidad y fácil accesibilidad a los datos ya sean de entrada o de salida se ha implementado en muchas aplicaciones y permite mejor manejo de datos en la aplicación y fuera de ella.

## Referencias bibliográficas

- A. Donwney, (2002). *Aprenda a pensar como un Programador con Python*. Green Tea Press.
- C. Severance, (2009). *Python para todos. Explorando la información con Python 3*. Green Tea Press
- J. Bullinaria, (2019). *Data Structures and Algorithms*. Green T. School of Computer Science, University of Birmingham, UK.
- M. Acera, (2011). *XML Edición 2012*. Anaya Multimedia
- V. Das, (2006) *Principles of Data Structures using C and C++*. New Age International (P) Ltd., Publishers.

Apéndice A

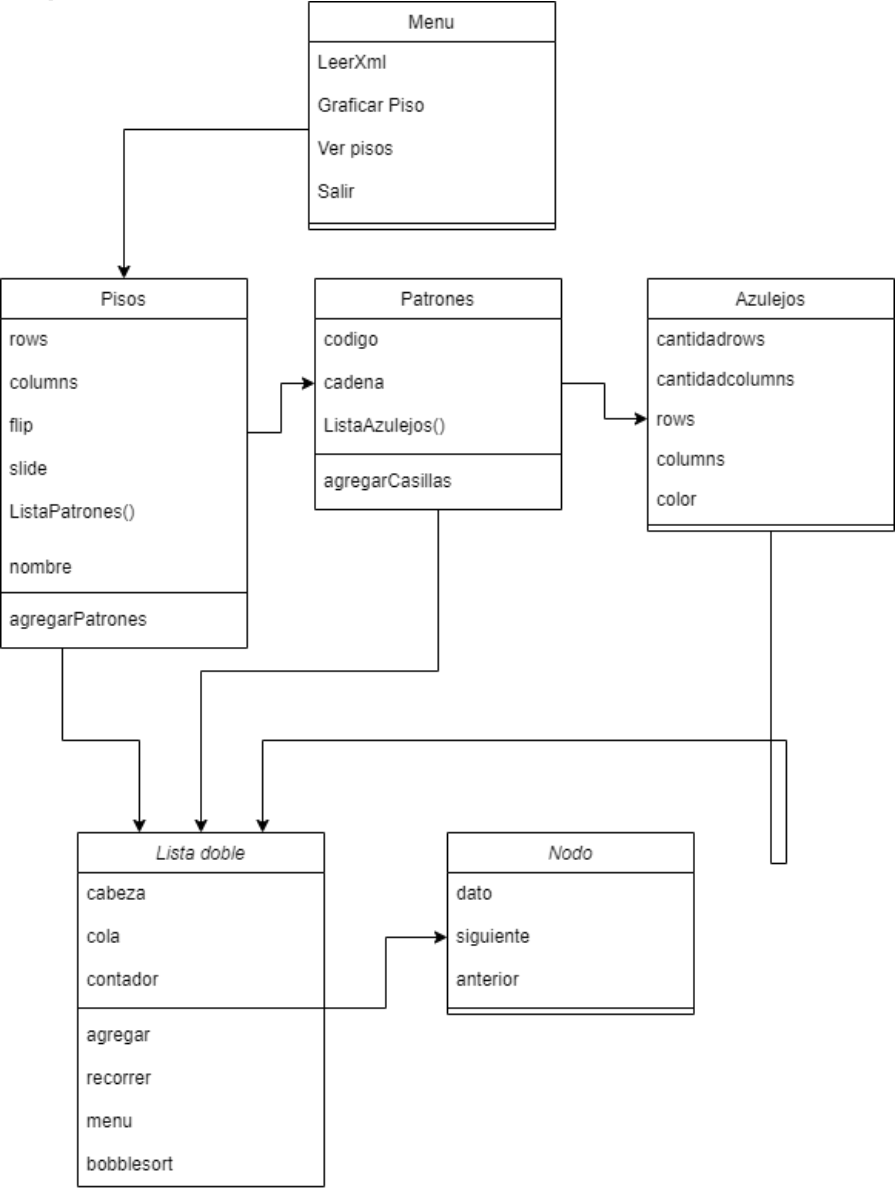


Figura VIII. Diagrama de clases proyecto 1

Fuente: Elaboración propia