Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2024

ENSAYO PROYECTO 1 IPC2

202307534 – Daniel Alejandro Portillo García

Resumen

El manejo dinámico de datos es un tema en la programación de vital importancia para el desarrollo de un ingeniero en sistemas, nos permite comprender todo el campo de comprensión del cómo funcionan las estructuras de base de datos o simplemente manejo de datos, pero para comprender este tema se necesita de suficiente practica y comprensión de sistemas de listas para comprender como almacenar objetos y atributos y poder manejarlos cada estructura de datos en un entorno de programación, ya sea Python, C#, C++, Java, entre otros entornos de programación. La importancia de manejo de listas es importante de aprender, puesto que muchas de las herramientas de entornos de programación están en su mayoría hechas con esta lógica llamadas listas enlazadas, que a no ser de estas herramientas no nos seria posible poder dar solución a muchas problemáticas informáticas hoy en día, por ello se demuestra en problemas simples de matrices su manejo para mejorar la comprensión de estas estructuras.

Palabras clave

Datos, Listas, procesar, programación, enlazada.

Abstract

Dynamic data management is a topic in programming of vital importance for the development of a systems engineer, it allows us to understand the entire field of understanding how database structures work or simply data management, but to understand this topic you need enough practice and understanding of list systems to understand how to store objects and attributes and be able to handle each data structure in a programming environment, be it Python, C #, C ++, Java, among other programming environments. The importance of list management is important to learn, since many of the tools of programming environments are mostly made with this logic called linked lists, which without these tools it would not be possible for us to be able to solve many computer problems today, therefore its handling is demonstrated in simple matrix problems to improve the understanding of these structures.

Keywords

Data, Lists, Processing, Programming, Linked

Introducción

Determinar el alojamiento de datos de forma accesible es un problema muy común en el manejo de información dinámica, y a menudo se utiliza programación orientada a objetos POO para dar soluciones a problemas simples, como leer y procesar cadenas de datos para interpretarlos, ya sean tablas de nombre, números, matrices y operaciones aritméticas. Y para muchos problemas reales se usan la combinación NP-Hard, que se especializan en el manejo de memoria por medio de métodos de agrupamiento, podemos aplicar esta lógica en ambientes de programación como Python que nos permiten manejar datos agrupados de forma siempre, pero, que pasaría si las herramientas que tenemos en nuestros entornos de programación no existieran, tendríamos que aplicar lógica de listas para poder simular el manejo de datos en memoria, Por medio del manejo de listas enlazadas y nodos que ayudaran al programador en comprender esta lógica en un problema cotidiano en la vida de un ingeniero en sistemas.

Desarrollo del tema

Para demostrar el manejo de estructuras de datos por medio de listas enlazadas, lectura de archivos e interpretación grafica por medio de herramientas de Python necesitamos saber la problemática que nos solicitan resolver. Utilizaremos herramientas externas para la lectura y creación gráfica, xml.dom.minidom como herramienta de lectura y creación y lenguaje dot interpretado con Graphiz.

El problema principal radica en la lectura de datos iniciales por medio de un archivo XML el cual nos mostrara la estructura de matrices las cuales tendremos que leer e interpretar por medio de la lectura de sus etiquetas, su estructura es:

Matrices: definirá la creación raíz del archivo determinando la creación del conjunto de matrices

Matriz: Contendrá el indicador diferenciador de cuáles son las matrices a leer, en este caso leer una cantidad x de matrices, con un nombre n cantidad de filas y m cantidad de columnas.

Valor: llevara la coordenada en la posición de la matriz x, y junto al valor de cada celda a almacenar

Al momento de procesar los datos se almacenarán en listas enlazadas y nodos que contendrán los atributos de cada matriz para luego procesarla. Para el poder procesar las matrices se pide el uso de matrices código en la cuales cada numero en la matriz representará un numero 1 y cada valor vacío o cero será representado con un 0. Esta matriz se formará recorriendo cada nodo por medio de listas circulares donde se formara una copia representando la matriz código obtenida y compararla, de pues se recorrerán columnas y filas para la matriz código en la cuales se encontraran coincidencias entre filas. Por cada coincidencia encontrada la matriz se reducirá en el numero de filas y sumará las celdas de las fulas encontradas, para así mostrar la matriz resultante y almacenarla manteniendo también las filas sin coincidencias dentro de nuestra matriz resultante.

Ya procesando nuestros datos requerimos formar un archivo XML con las matrices procesadas, este proceso se implementa leyendo las matrices nuevas almacenada e imprimiendo el archivo con sus respectivas etiquetas, matrices, matriz, nombre, n, m, dato, x, y y su valor almacenado, todo en formato XML.

Ahora necesitamos crear el reporte de graficas procesadas, esto se hace con lenguaje tipo dot por medio de la herramienta de graphiz en donde en nuestras listas le diremos que recorra los nodos según

los atributos de cada matriz y genera la estructura de grafico con los datos leídos de una matriz en específico según solicite el usuario. Al final creara un archivo tipo dot que se usara para la creación del archivo grafico en tipo pdf.

Terminado el proceso se solicita mostrar datos de creador del sistema y salir del sistema

Flujo del programa:

A continuación, se ejemplificara el proceso de muestra de nuestro programa con entrada de datos, procesamiento, salda de archivos y reporte de salida

Se Muestra el menú con las 6 opciones disponibles dentro del progrma

```
=========Menu Principal=========
1. Cargar Archivo
2. Procesar Archivo
3. Escribir Archivo de Salida
4. Mostrar datos del Estudiante
5. Generar Grafica
6. Salida
================================
Ingresar opcion:
```

Al Seleccionar la opción 1 nos solicitara el archivo que queremos cargar, en este caso es el archivo ejemplo.xml. El archivo al ser cargado imprimirá las matrices que se encuentran dentro del archivo xml con el formato de matriz, su nombre, y dimensiones. En este caso se imprimen 2 matrices.

```
----------------------------------------------------------------
Se eligio la opcion 1
Ingrese la ruta del archivo:
```

```
Ingrese la ruta del archivo: ejemplo.xml
matrices
================================================================================
Nombre de la Matriz: Matriz 1 | Con Filas N: 3 y Columnas M: 3
|  0 ||  1 ||  2 |
|  3 ||  4 ||  5 |
|  6 ||  7 ||  8 |
================================================================================
================================================================================
Nombre de la Matriz: Matriz 2 | Con Filas N: 2 y Columnas M: 2
|  0 ||  1 |
|  3 ||  4 |
================================================================================
Datos leidos con éxito con ElementTree
```

Ahora al seleccionar la opción 3 se procesaran las matrices, mostrando primero las matrices originales, para después mostrar las matrices código y formar las matrices resultantes

```
================================================================================
Nombre de la Matriz: Matriz 1 | Con Filas N: 3 y Columnas M: 3
|  0 ||  1 ||  2 |
|  3 ||  4 ||  5 |
|  6 ||  7 ||  8 |
================================================================================
================================================================================
Nombre de la Matriz: Matriz 2 | Con Filas N: 2 y Columnas M: 2
|  0 ||  1 |
|  3 ||  4 |
================================================================================
Matrices Nuevas//////////////////////////////
Matriz con filas idénticas sumadas 'Matriz 1':
================================================================================
Nombre de la Matriz: Matriz 1 | Con Filas N: 3 y Columnas M: 3
|  0 ||  1 ||  1 |
|  1 ||  1 ||  1 |
|  1 ||  1 ||  1 |
================================================================================
================================================================================
!!!!!!Nombre de la Matriz Reducida!!!!!: Matriz 1 | Con Filas N: 3 y Columnas M: 3
|  0 ||  1 ||  2 |
|  9 ||  11 ||  13 |
|  6 ||  7 ||  8 |
================================================================================
```

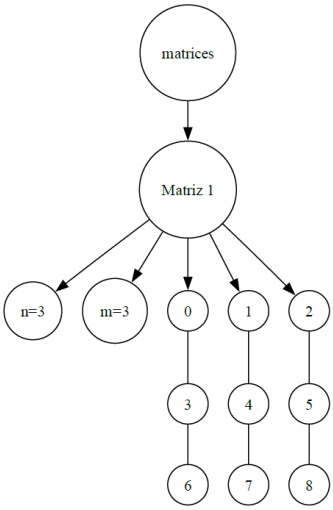
```
================================================================================
Nombre de la Matriz: Matriz 2 | Con Filas N: 2 y Columnas M: 2
|  0 ||  1 |
|  1 ||  1 |
================================================================================
================================================================================
!!!!!!Nombre de la Matriz Reducida!!!!!: Matriz 2 | Con Filas N: 2 y Columnas M: 2
|  0 ||  1 |
|  3 ||  4 |
================================================================================
```

Al elegir la opción de generar el archivo de salida creara el archivo de salida con las matrices almacenadas, el nombre del archivo será salida.

```
----------------------------------------------------------------
Se eligio la opcion 3
Archivo XML generado con éxito.
----------------------------------------------------------------
```

En la opción 5 se creara el reporte grafico de las matrices, en el cual deberemos de seleccionar la matriz almacenada que se desea graficar, creara un archivo .dot y .pdf para mostrar la matriz, el archivo pdf se creara con el nombre de la matriz

```
Se eligio la opcion 5
Ingrese el nombre de la matriz que desea graficar: Matriz 1
```

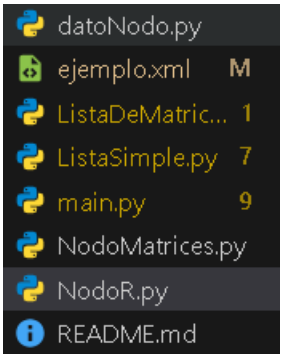


Por último tenemos la opción de mostrar datos del creador del programa y el cierre del programa.

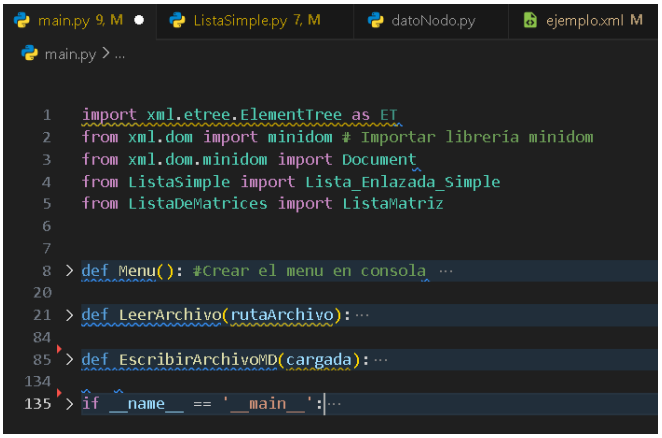
```
---------------------------------
Ingresar opcion: 4
-----------------------------------------------------------------
Nombre: Daniel Alejandro Portillo Garcia
Carnet: 202307534
IPC2 Proyecto 1 2024
-----------------------------------------------------------------
```

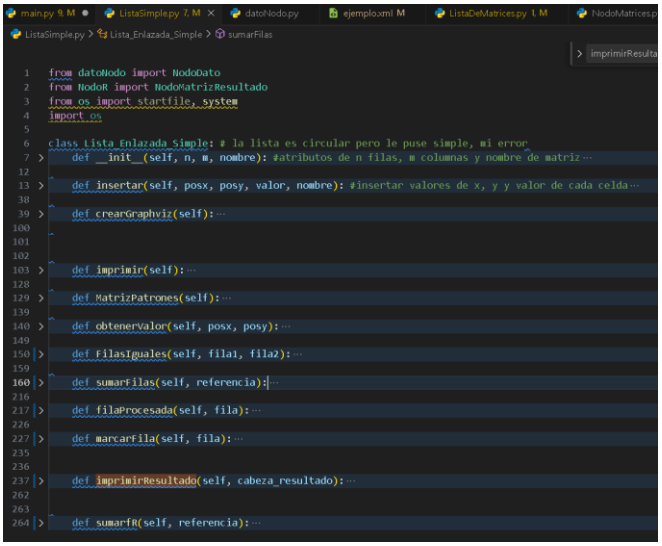
```
=================================
Ingresar opcion: 6
-----------------------------------------------------------------
Adios :)
```

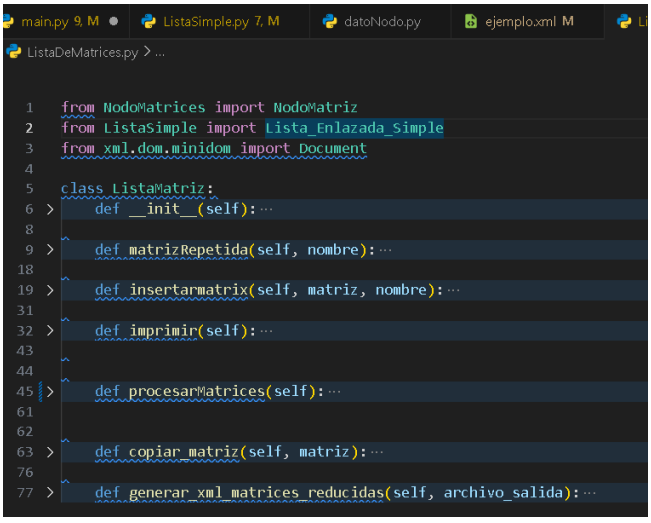
Archivos de clases utilizadas dentro del programa



Clases, funciones y importaciones de cada archivo





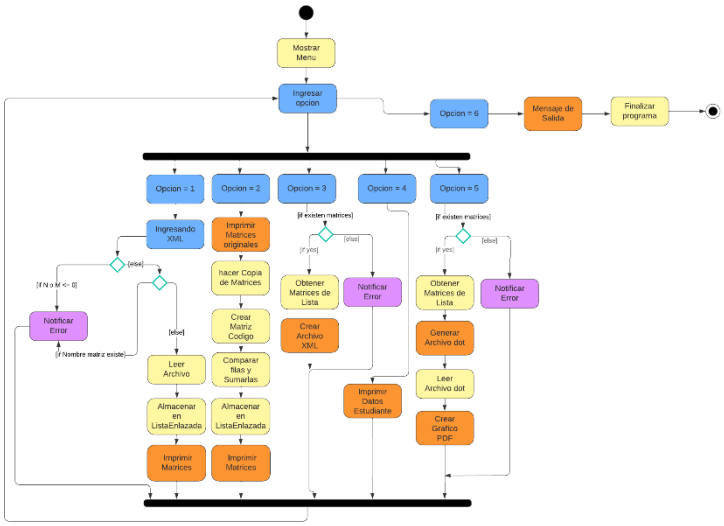


```python
class NodoMatriz:
    def __init__(self, matriz, nombre):
```

```python
class NodoMatrizResultado:
    def __init__(self, fila, columna, valor):
```

Conclusiones

- Podemos observar que el uso de manejo de datos por medio de listas nos permite comprender el proceso dentro de la lectura y manejo de datos por medio de POO desde su forma más compleja pero básica dentro de la programación.
- El uso de listas enlazadas nos permite llevar una secuencia de manejo de datos dinámica y mas ordenada.
- La lectura, inserción, eliminación y el mostrar datos por medio de almacenamiento de nodos y listas es un proceso mas complejo que el usar las herramientas nativas del sistema de programación, en este caso Python



Referencias bibliográficas

Graphviz. (s.f.). *Graphviz*. PyPI. Recuperado de https://pypi.org/project/graphviz/#files

Anexos

Diagrama de Clases

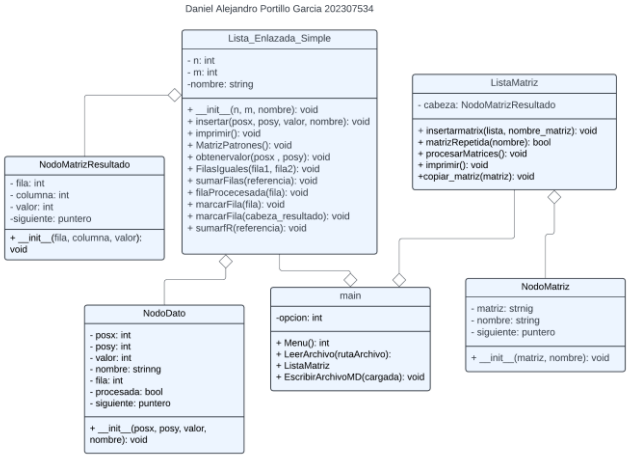


Diagrama de Actividades