
PROYECTO 2

201903873 – Joaquin Emmanuel Aldair Coromac Huevo

Resumen

El enunciado propone el uso de matriz ortogonales para el procesamiento de estas sea minimizado, debido a la fuerte demanda de recursos de memoria y tiempo, para esto se propone una metodología de agrupamiento.

La aplicación consiste en una forma de representar imágenes utilizando listas ortogonales y permitir realizar operaciones sobre estas imágenes.

Se requiere poder gestionar “N” imágenes en una lista simple ordenada. Estas imágenes tendrán un nombre y una dimensión. La dimensión de la imagen se determina por la cantidad de filas “f” y la cantidad de columnas “c” de esta imagen.

Las operaciones pueden variar desde el resultado de combinar 2 imágenes o modificar una sola imagen.

Palabras clave

> Matriz > Objeto
>Reporte > Lista > Grafo

Abstract

The statement proposes the use of orthogonal matrix to minimize the processing of these images, due to the high demand of memory resources and time, for this a grouping methodology is proposed.

The application consists of a way to represent images using orthogonal lists and allow to perform operations on these images.

It is required to be able to manage "N" images in a simple ordered list. These images will have a name and a dimension. The dimension of the image is determined by the number of rows "f" and the number of columns "c" of this image.

The operations can vary from the result of combining 2 images or modifying a single image.

Keywords

> Matrix > Object
>Report > List > Network.

Introducción

Teniendo como solución la metodología de agrupamiento, y optando por el paradigma de programación la orientada a objetos (POO), se desarrolló una Lista Simple General donde en cada objeto se tiene los datos de las matrices que nos encontramos en el archivo XML de entrada.

Para así procesar cada matriz y poder realizar las distintas operaciones de una o dos imágenes y graficar los resultados obtenidos mediante graphviz.

Para cada evento realizado se genera un objeto Reporte que nos indica la hora y fecha, tipo, matriz y sus datos para llevar un informe de lo realizado por el programa en su funcionamiento.

Desarrollo del tema

Lógica Programa

➤ Archivo(proyecto.py)

- Clases
 - class Reporte:
 - class Ventana:

➤ Archivo(matriz.py)

- Clases
 - class Nodo:
 - class Matriz:

➤ Archivo(lista.py)

- Clases
 - class Lista:
 - class nodo:

➤ Archivo(grafo.dot)

- Archivo generado para cada matriz de entrada que desee graficar

Paradigma de programación en uso

Programación orientada a objetos(POO)

Es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

Este muestra un realce mayor al momento de guardar en espacio de memoria cada matriz en un objeto de la lista simple.

Lógica de Matriz Dispersa y lista simple

Cada matriz tiene los encabezados de todas sus filas y columnas.

En cada variable data de los nodos de la lista simple se guarda el objeto matriz o reporte para poder manejar de forma sencilla en memoria dinámica esos datos.

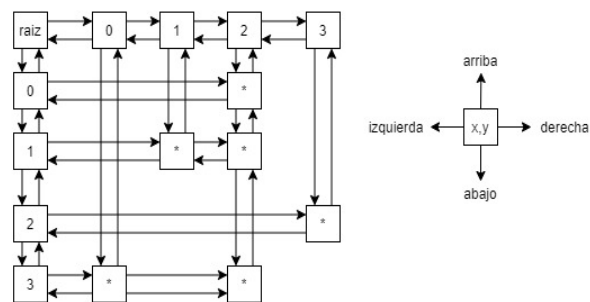


Diagrama Flujo

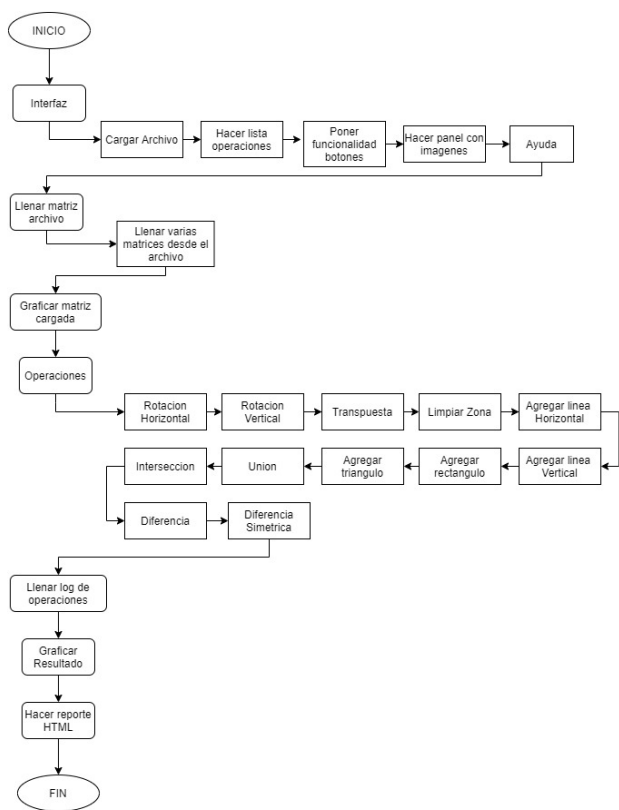
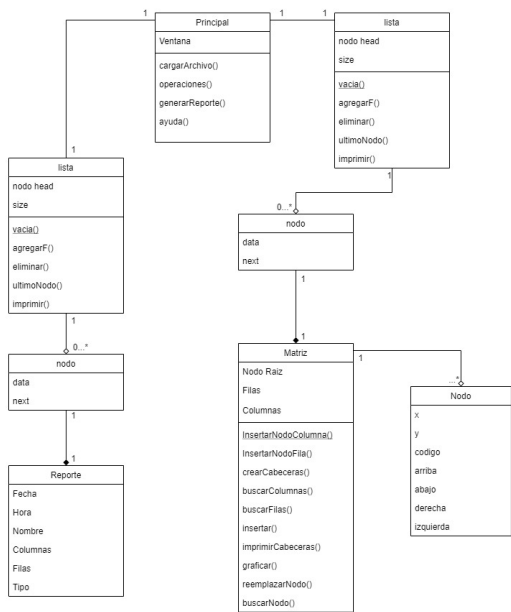


Diagrama de clases



Especificaciones

Especificaciones de Editor de Texto, versión Python y Graphviz

Visual Studio Code

python.3.8.1

Graphviz 2.46.0

Especificaciones de computadora

Procesador: Intel(R) Core™ i7-9750H
CPU@

2.60 GHz 2.59GHz

RAM instalada: 16.0GB (15.9 GB usable)

Tipo de sistema: Sistema operativo de 64 bits,
Procesador basado en x64

Windows 10.

Control de versiones

Para el proyecto se trabajó con GitHub para controlar las versiones durante su realización.

Enlace Repositorio.

https://github.com/jeach27/IPC2_Proyecto2_201903873.git

Forma de Uso Programa

Menu principal

Escoja una opción entre 1-6 para realizar la función requerida.

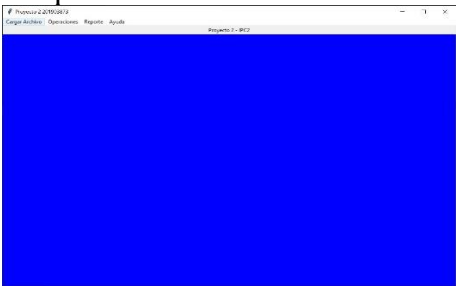


Imagen 1. Menu Principal

Fuente: Programa, elaboración propia

Opcion1 Cargar Archivo

Escoja archivo XML mediante filedialog

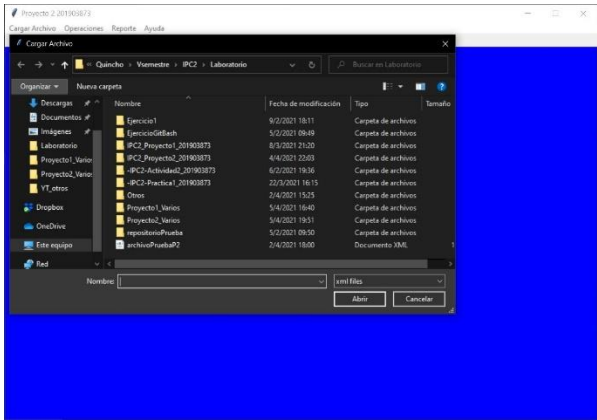


Imagen 2. Cargar Archivo

Fuente: Programa, elaboración propia

Opcion2 Procesar Archivo

Procesa cada matriz y va identificando cada acción realizada y muestra combobox para elegir que imágenes desea usar para realizar las operaciones



Imagen 3. Procesar Archivo

Fuente: Programa, elaboración propia

Opcion4 Datos Estudiantiles

Muestra los datos del estudiante desarrollador.

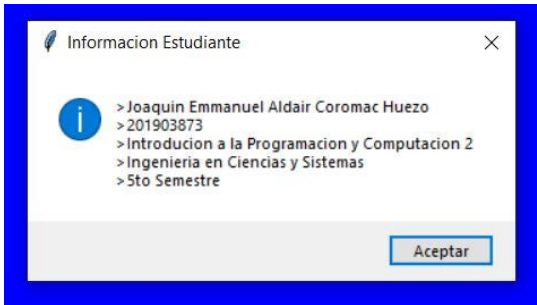


Imagen 4. Datos Estudiantiles

Fuente: Programa, elaboración propia

Opcion5 Generar Grafica

Muestra tabla de reporte de

No	Fecha	Hora	Nombre	Filas	Columnas	Tipo
1	5/4/2021	19:16				IngresosArchivos
2	5/4/2021	19:16				IngresosArchivos
3	5/4/2021	19:16	Matriz_1	10	10	ReporteConfidencial

Imagen 5. Generar Grafica

Fuente: Programa, elaboración propia

Modelo Grafica

Matriz_2	0	1	2	3	4	5	6	7
0								
1		*		*		*		*
2			*				*	
3		*		*			*	
4								
5		*	*	*			*	
6			*			*		*
7		*	*	*			*	

Imagen 6. Modelo Grafica

Fuente: Programa, elaboración propia

Conclusiones

El uso de Paradigma de programación orientado a objetos nos permite trabajar de forma más sencilla cada objeto de la Lista de matrices.

Los archivos XML es una forma sencilla y segura de guardar datos de este tipo.

La herramienta Graphviz para realizar los grafos es bastante fácil al momento de generar sus archivos .dot.

Referencias bibliográficas

9.3. Programación orientada a objetos —

Materiales del entrenamiento de programación en Python - Nivel básico. (s. f.). Materiales del entrenamiento de programación en Python - Nivel básico. Recuperado 24 de febrero de 2021, de <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion9/poo.html>

GeeksforGeeks. (s. f.). *Parse XML using Minidom in Python.* Recuperado 1 de marzo de 2021, de <https://www.geeksforgeeks.org/parse-xml-using-minidom-in-python/>

graphviz. (2020, 24 diciembre). PyPI. <https://pypi.org/project/graphviz/>

Salcedo, L. (s. f.). *Linked List: Listas enlazadas - Implementación en Python.* Mi Diario Python. Recuperado 29 de febrero de 2021, de <https://pythondiario.com/2018/07/linked-list-listas-enlazadas.html>

uniwebsidad. (s. f.). *Capítulo 16. Listas enlazadas*

(Algoritmos de programación con Python).

Recuperado 27 de febrero de 2021, de <https://uniwebsidad.com/libros/algoritmos-python/capitulo-16>