
PROYECTO 2

201906588 – Byron Estuardo Solís González

Resumen

Se desarrolló una aplicación en el lenguaje de programación Python cuyo principal objetivo era la visualización de matrices de datos como imágenes y contrastarlas con las imágenes resultantes de las modificaciones hechas por parte del usuario a la imagen original.

La aplicación cuenta con una interfaz gráfica simple con paneles intercambiables, listas de opciones, entradas de texto y botones oyentes de eventos con los que el usuario puede interactuar para la manipulación de las imágenes. Los datos le son proveídos a través de archivos de tipo xml que el usuario puede elegir a través del selector de archivos incluido en esta.

Para mejorar el manejo de memoria se optó por implementar matrices ortogonales en lugar de las matrices convencionales para guardar la información de cada imagen, esto se logró desarrollando desde cero un tipo de dato abstracto de matriz ortogonal con la capacidad de insertar y mostrar datos en estos.

Palabras clave

Matriz Ortogonal DTA Manejo Memoria

Abstract

An application was developed in the Python programming language whose main objective was the visualization of data matrices as images and contrast them with the images resulting from the modifications made by the user to the original image.

The application has a simple graphical interface with interchangeable panels, option lists, text inputs and event listener buttons with which the user can interact to manipulate the images. The data is provided through xml files that the user can choose through the file selector included in it.

To improve memory management, it was decided to implement orthogonal matrices instead of conventional matrices to save the information of each image, this was achieved by developing from scratch a type of abstract orthogonal matrix die with the ability to insert and display data in these .

Keywords

Orthogonal Matrix DTA Memory Management

Introducción

Debido a que la memoria a nivel local, área a la que se orienta la aplicación, es un recurso limitado se debe buscar su optimización.

En el proyecto se maneja la información de imágenes a por su posición en horizontal y vertical es adecuado el uso de matrices, pero al ser los datos de las imágenes únicamente de 2 tipos (forma y fondo) se puede hacer un mejor uso de la memoria el guardar solamente la información de una de estas e interpretar la otra al momento de mostrar los datos. Esto se logra a través de utilizar matrices de tipo ortogonal, cuya característica es el únicamente guardar en memoria las posiciones cuyos nodos tengan contenido, lo cual es ideal si se trabaja con una gran cantidad.

Esto sumado con una interfaz gráfica simple y la capacidad de la creación de imágenes constituyen esta aplicación.

Desarrollo del tema

Para el desarrollo de la aplicación se abordaron funciones de menor tamaño, las cuales se describen a continuación.

1. Interfaz grafica:

Esta fue desarrollada utilizando la librería de Python Tkinter, gracias a su capacidad para crear ventanas emergentes, ventanas de contenido, paneles, entradas de texto, botones, selector de archivos y su capacidad de mostrar imágenes fue posible la creación de una interfaz con la que el usuario puede interactuar sin acceder al procesamiento interno de la aplicación.

2. Procesamiento xml:

La aplicación toma como materia prima información estructurada donde por cada imagen se especifica el nombre, la cantidad de filas, la cantidad de columnas e imagen, siendo esta última un conjunto de filas de texto donde se especifica común símbolo “*” las posiciones llenas y con un “-” las posiciones vacías. Por tener una estructura rígida el uso del formato xml es ideal.

Por esto el usuario para ingresar información a la aplicación debe hacerlo seleccionando un archivo de dicho tipo desde el selector de archivos, la estructura de este debe ser rígida igualmente, colocando todas las matrices dentro de la etiqueta “matrices” y delimitando cada matriz individual con la etiqueta “matriz”, dentro de esa última deben estar los datos antes mencionados correspondientes a cada una de las matrices (nombre, cantidad de filas, cantidad de columnas e imagen).

Dentro de la aplicación se verifica que el archivo de entrada no tenga errores, y si se encuentra alguno es reportado e ingresado al reporte de operaciones de la aplicación en la categoría de errores.

3. Matriz ortogonal:

Para aumentar la eficiencia del uso de memoria por parte de la aplicación se optó por desarrollar e implementar desde cero un dato abstracto de tipo matriz ortogonal, aprovechando las propiedades de la programación orientada a objetos, en este se guardarían solamente las posiciones llenas de la matriz de datos de la imagen para a la hora de crear la imagen interpretar las posiciones no existentes como posiciones vacías. El desarrollo de esta se describe a continuación:

- **Nodos:** Se crearon dos tipos de nodos, de tipo interno y de tipo encabezado.

- **Nodo Interno:** Estos contienen los datos que se quieren guardar en la matriz, en este caso número 1, los cuales son asignados a las posiciones llenas de la matriz de imagen proporciona por el usuario. Cada uno de estos tiene como atributos la fila y columna que deben ocupar, su valor (un número uno en este caso) y los apuntadores que le ayudarán a ubicarse dentro de la matriz; Apuntador hacia la derecha, hacia la izquierda, hacia arriba y hacia abajo. Los apuntadores le son asignados en el transcurso de inserción de nuevos nodos, ya que irán cambiando, el resto de atributos se le asignan al momento de su creación.

- **Nodo Encabezado:** Estos están enlazados a la el primer nodo de cada fila o columna, son los que nos dan acceso a los nodos internos. Manejan un identificador que contiene el número de fila o columna del cual son encabezado.

Sus atributos son el identificador anteriormente descrito, un acceso, que apunta al primer nodo de fila o columna de la que son encabezado además de dos apuntadores uno hacia la derecha y a otro hacia la

izquierda, esto debido a que pertenecen a una lista doblemente enlazada de nodos encabezado que representa ya sea las filas o columnas de la matriz, la cual en descrita más adelante.

El único atributo que se le asigna al momento de su creación es el identificador, ya que el resto son asignados en el transcurso de la incisión de nuevos nodos, tanto de tipo internos como de tipo encabezado.

- **Encabezado:** Esta estructura es una lista doblemente enlazada ordenada que contiene a todos los nodos encabezados ya sea del conjunto de filas o columnas de la matriz. Por cada matriz existen dos instancias de este tipo de objeto, una para el conjunto de filas y otra para el conjunto de columnas. Estos al momento de su creación reciben un nodo encabezado como parámetro, ya que solo se crea al haber un nodo interno queriendo ingresar a la matriz, el nodo encabezado que apunta al este es el que pasamos como parámetro.

Dentro de este se realiza el proceso de ubicación de cada nuevo nodo encabezado que se agrega a la matriz, proceso que se agiliza al ser una lista doblemente enlazada ordenada.

- **Matriz:** Este es nuestro DTA de matriz ortogonal, en este contendremos a nuestras

instancias de tipo encabezado, que a su vez contendrán nodos encabezado, los cuales apuntarán a los nodos internos.

Sus atributos son dos instancias del objeto Encabezado, uno para filas y otro para columnas. Además este recibe los nodos internos que se quieren ingresar a la matriz, y los ubica en la posición adecuada, además de colocar los apuntadores de cada una de las instancias de las estructuras que contiene.

○ Proceso de inserción

1. Recorrerlos encabezados, estos deben estar ordenados, esto se hace para verificar si existe en encabezado adecuado para el nuevo nodo o si se debe crear uno nuevo.
2. Si el encabezado existe quiere decir que ya hay elementos en esa dimensión, por lo que de debe insertar el nuevo nodo basándose en el orden de los elementos ya ingresados. Buscamos la ubicación adecuada en la lista del encabezado correspondiente.
3. Recorremos los encabezados de la siguiente dimensión y procedemos como en el primer paso, si antes recorrimos filas, ahora recorreremos columnas o viceversa; Buscando el encabezado necesario, si este no existe lo creamos e insertamos el nuevo nodo.
4. Si el encabezado de la siguiente dimensión ya existe quiere decir que hay elementos allí, al igual que en segundo paso buscamos la ubicación e insertamos en nuevo no.

4. Generación de imágenes:

Para la creación de imágenes se utilizó la librería PIL específicamente su módulo Image, esta permite la creación de imágenes a partir de matrices cuyos elementos sean listas con los valores en RGB de color que se quiere aplicar al pixel en la misma posición del elemento; Por lo que para crear una imagen del tamaño adecuada para visualizarla de forma cómoda se agregaron varias veces cada posición a la matriz.

5. Generación de Reportes

Al procesar un archivo de entrada de matrices se crea un archivo HTML en el cual se especifican el nombre de la matriz, número de espacios vacíos y llenos junto con la hora y fecha en que se procesa cada matriz registrada, además de en caso de que una operación genere un error, éste deberá quedar descrito en este reporte indicando fecha y hora, y una descripción del error.

Conclusiones

El uso de la programación orientada a objetos y la creación de propios DTA permiten darle una solución más personalizada y/o eficiente a la problemática de un proyecto de programación.

La memoria es un recurso máspreciado y una forma de manejarlo eficientemente es guardar en ella solamente lo estrictamente necesario, una herramienta para lograr esto es la implementación de matrices ortogonales en lugar de las matrices convencionales, con esto podemos conservar todas las propiedades de una matriz convencional sin perder memoria si se tuviera alguna posición vacía.

Se debe incluir en la estimación del tiempo de programación una buena parte de tiempo para corregir errores y solucionar imprevistos, ya que de no ser así puede que el proyecto no se termine a tiempo.

Referencias bibliográficas

Academia Coder. (2020, 17 mayo). *INTERFACES GRÁFICAS con PYTHON 3 - TKinter 01 - Curso de Python DESDE CERO - #67* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=Nn0AVbGVMhk>

Codemy.com. (2020, 18 marzo). *Delete Frame Children Widgets - Python Tkinter GUI Tutorial #50* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=A6m7TmjNzww>

Convertir una matriz NumPy a una imagen PIL. (2020, 7 agosto). javaer101.
<https://www.javaer101.com/es/article/1810513.html>

DelftStack. (2020, 25 junio). *Cómo crear un menú desplegable en Tkinter.*
<https://www.delftstack.com/es/howto/python-tkinter/how-to-create-dropdown-menu-in-tkinter/>

Academia Coder. (2020, 17 mayo). *INTERFACES GRÁFICAS con PYTHON 3 - TKinter 01 - Curso de Python DESDE CERO - #67* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=Nn0AVbGVMhk>

Python Software Foundation. (2021, 1 abril). *tkinter — Interface de Python para Tcl/Tk.* docs.python.
<https://docs.python.org/es/3/library/tkinter.html#tkinter-modules>

Recursos Python. (2018, 14 agosto). *Cuadros de diálogo (messagebox) en Tcl/Tk (tkinter).*
<https://recursospython.com/guias-y-manuales/cuadros-de-dialogo-messagebox-en-tkinter/>

Rodríguez, D. (2019, 4 septiembre). *Numpy básico: Creación de un Array de Numpy a partir de una listas o tuplas.* Analytics Lane.
<https://www.analyticslane.com/2019/09/04/numpy-basico-creacion-de-un-array-de-numpy-a-partir-de-una-listas-o-tuplas/>

SemicolonWorld. (2017, 5 junio). *How to update the image of a Tkinter Label widget? - SemicolonWorld.* SemicolonWorld - Best Place for Developers.
<https://www.semicolonworld.com/question/56158/how-to-update-the-image-of-a-tkinter-label-widget>

Anexos

Figura 1. Diagrama de clases de la aplicación

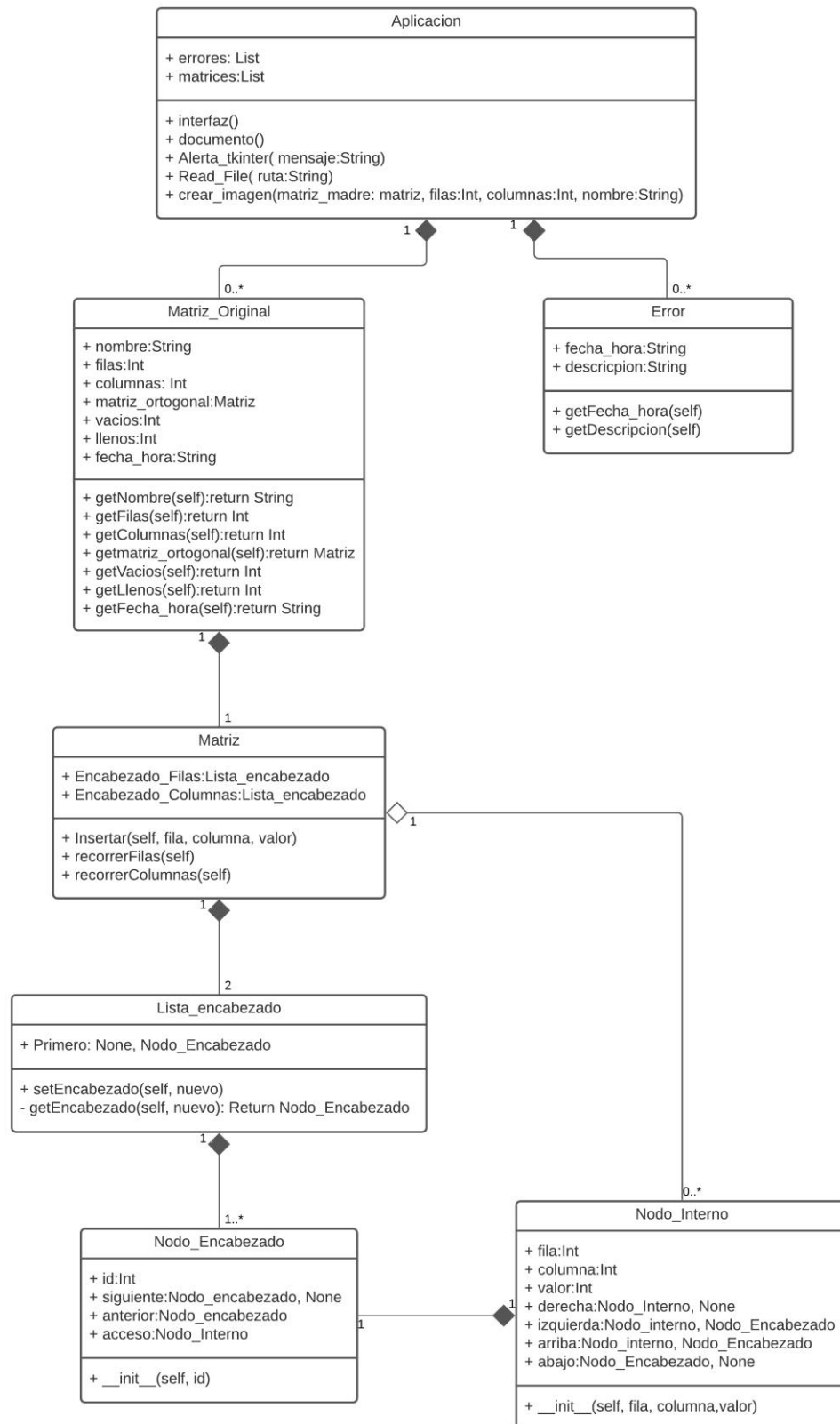


Figura 1. Diagrama de clases de la aplicación
 Fuente: elaboración propia

Figura 2. Diagrama de la estructura de la matriz ortogonal

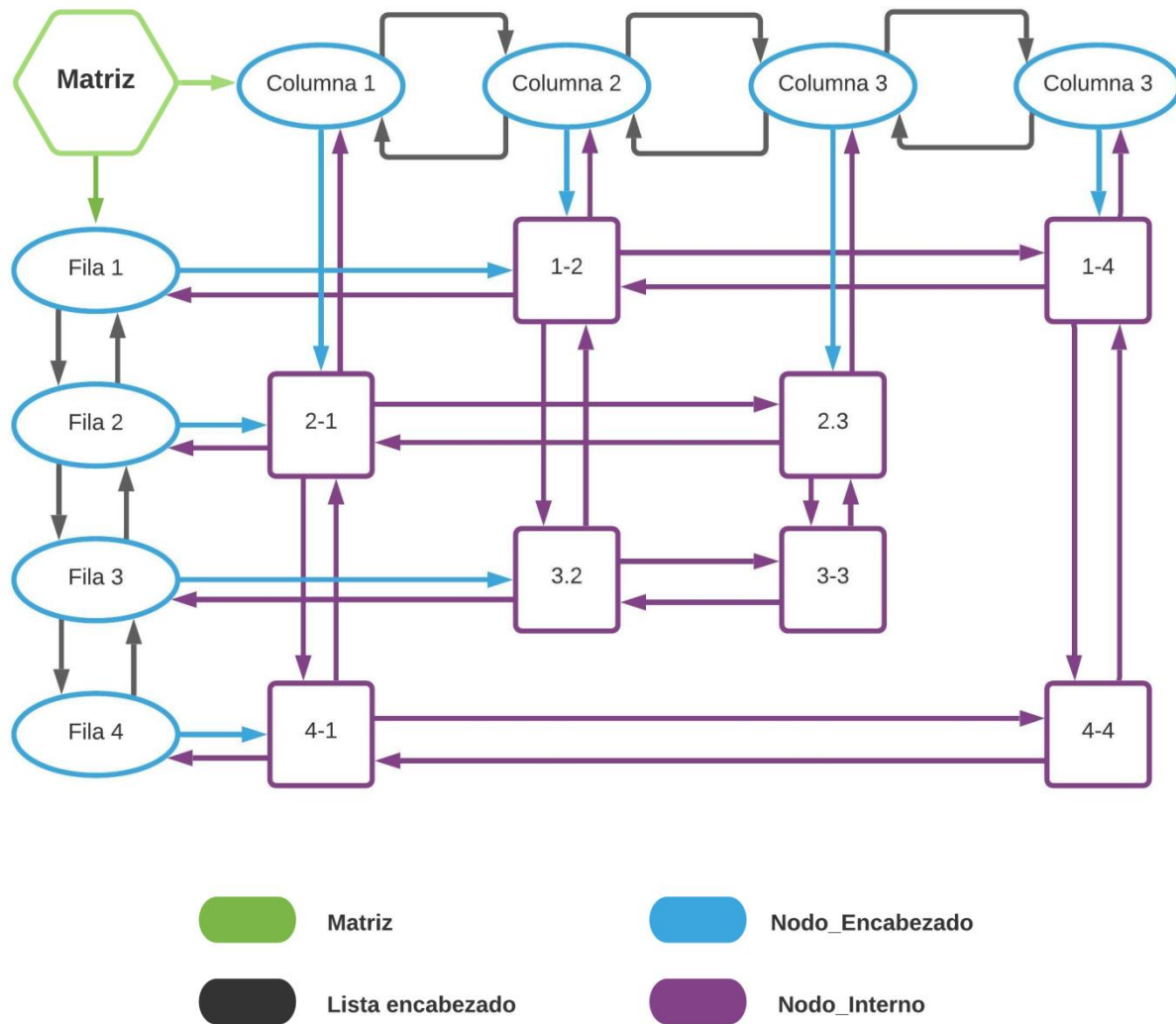


Figura 2 . Diagrama de la estructura de la matriz ortogonal
Fuente: elaboración propia