|  |
| --- |
| **PROYECTO NO. 2** |
| **201901429 – Diego Abraham Robles meza** |

**Resumen**

El problema planteado en este programa fue el de como gestionar los datos de un archivo con extensión xml, el cual contiene una cantidad ‘n’ de matrices ortogonales, a las cuales se les tiene que poder realizar ciertas operaciones tales como: rotar verticalmente, obtener su transpuesta, limpiar una zona, etc. Para luego en una interfaz gráfica poder visualizar los cambios hechos.

Se empleo la programación orientada a objetos para crear la estructura de todas las matrices ortogonales leídas desde el archivo de entrada y se almacenaron todas las que tuvieran correctamente el formato establecido para las matrices ortogonales.

El programa cuenta con 12 operaciones en total, las cuales se muestran en una interfaz realizada con Python, y estas operaciones se pueden realizar tanto a una matriz (en el caso de las primeras 8 operaciones), o a dos matrices si así se requiere.

**Palabras clave**

* Matriz
* Ortogonal
* Imagen
* TDA

***Abstract***

*The problem posed in this program was how to manage the data of a file with xml extension, which contains a number 'n' of orthogonal matrices, which must be able to perform certain operations such as: rotate vertically, get its transpose, clean a zone, etc.. Then, a graphical interface is used to visualize the changes made.*

*Object oriented programming was used to create the structure of all the orthogonal matrices read from the input file and all those that had the correct format established for orthogonal matrices were stored.*

*The program has 12 operations in total, which are displayed in a Python interface, and these operations can be performed either on one matrix (in the case of the first 8 operations), or on two matrices if required.*

***Keywords***

* Matrix
* Orthogonal
* Image
* TDA

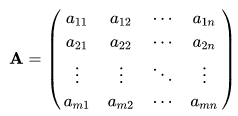
**Introducción**

Las matrices ortogonales y la estructura de una matriz en general son muy utilizadas en muchos campos de la ciencia, una matriz ortogonal es una matriz cuadrada cuya matriz inversa es igual a su matriz transpuesta. Se decidió emplear esta estructura para el manejo de imágenes, porque al ser tan utilizada en muchas áreas, es mucho más fácil trabajar con este tipo de estructuras, ya que teniendo el conocimiento de cómo funcionan es un tema del que ya se posee bastante documentación en internet y en las universidades en general.

Una de las mayores dificultades presentadas en este proyecto fue el de como emplear estas estructuras en el lenguaje de programación Python, ya que es un tema un poco abstracto en cuanto a código se refiere. Pero como anteriormente se menciona, hay bastante documentación para poder tener una idea de cómo estructurarla de la manera correcta.

**Matrices Ortogonales:**

Es lo que a matemáticas se refiere, una matriz es un arreglo bidimensional que cuenta con filas y columnas, las cuales están estructuradas de la siguiente manera:



Donde el primer subíndice de cada elemento representa el número de fila al que pertenece y el segundo subíndice representa el número de columna al que pertenece. Existen diferentes tipos de matrices, las cuales se clasifican de la siguiente forma:

a. Matriz fila

b. Matriz columna

c. Matriz identidad

d. Matriz rectangular

e. Matriz cuadrada

f. Matriz nula

g. Matriz ortogonal

Matriz fila:

Como su nombre lo indica, esta matriz consta de una sola fila, y se ve de la siguiente manera:



*Figura 1.* Matriz fila.

Fuente: Elaboración propia, 2021.

Matriz Columna:

Como su nombre lo indica, esta matriz consta de una

sola columna, y se ve de la siguiente manera:

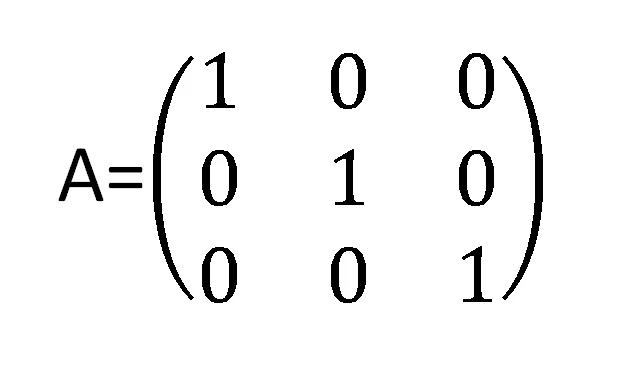


*Figura I1.* Matriz columna.

Fuente: Elaboración propia, 2021.

Matriz identidad:

La particularidad de este tipo de matrices es que en su diagonal principal los elementos de esta son ‘1’:



*Figura I1I.* Matriz identidad.

Fuente: Elaboración propia, 2021.

Matriz rectangular:

La matriz rectangular tiene distinto número de filas que, de columnas, siendo su dimensión m x n.

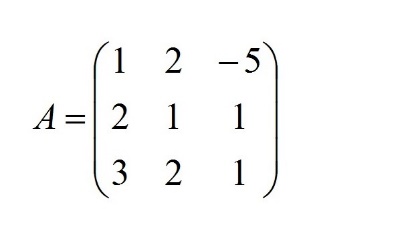


*Figura IV.* Matriz rectangular.

Fuente: Elaboración propia, 2021.

Matriz cuadrada:

Este tipo de matriz se caracteriza por tener el mismo numero de filas y columnas.



*Figura V.* Matriz cuadrada.

Fuente: Elaboración propia, 2021.

Matriz nula:

En esta matriz todos sus elementos son ‘0’:



*Figura VI.* Matriz nula.

Fuente: Elaboración propia, 2021.

Matriz ortogonal:

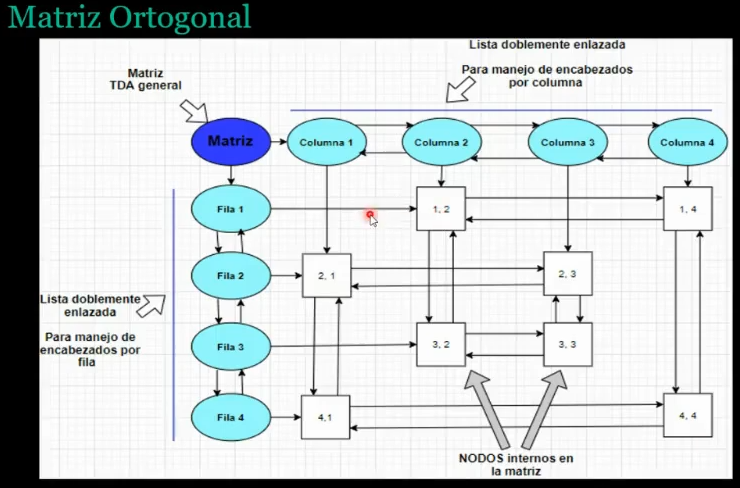
Una matriz ortogonal es una matriz cuadrada cuya matriz inversa coincide con su matriz traspuesta.



*Figura VII.* Matriz ortogonal.

Fuente: Elaboración propia, 2021.

Esta fue la estructura utilizada para el programa, en la siguiente imagen se puede ver cómo es la lógica para implementar las matrices:

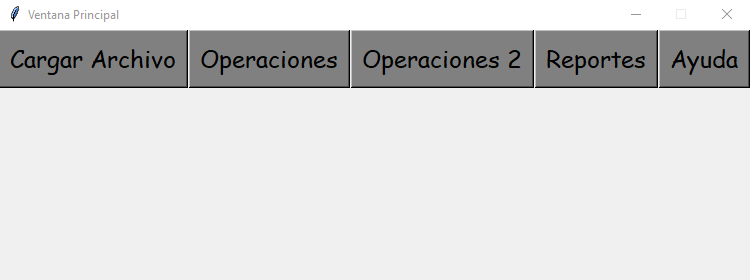


¿Cómo funciona?

Esta matriz hecha en Python está basada en dos listas de encabezados, uno de filas y uno de columnas, las listas son doblemente enlazadas, como puede verse en la imagen y cada uno de los nodos de la lista de encabezados apunta hacia un nodo interno, esto se hace para que las listas estén unidas y no se pierda la referencia a cada nodo.

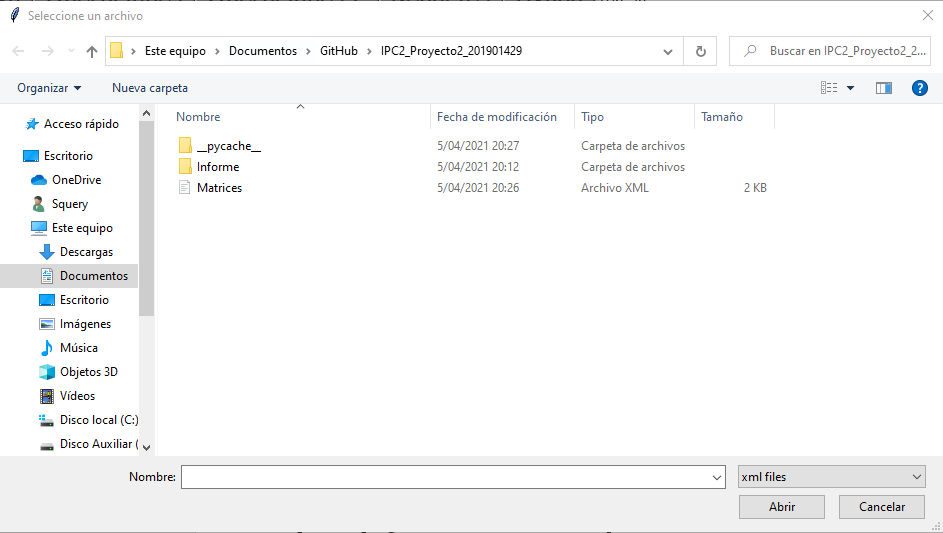
**Funcionalidad del programa:**

En la ventana principal se muestran las distintas funcionalidades que este programa puede realizar.



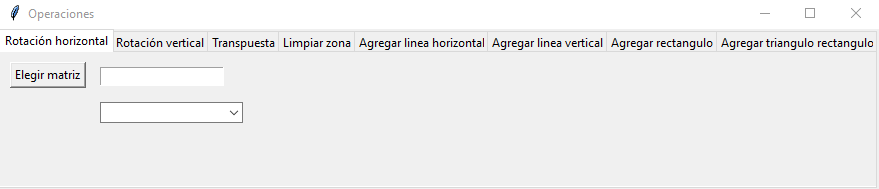
**Cargar Archivo**:

Al presionar esta opción se despliega una ventana para elegir algún archivo de nuestra computadora para luego ser cargado en el programa.



**Operaciones:**

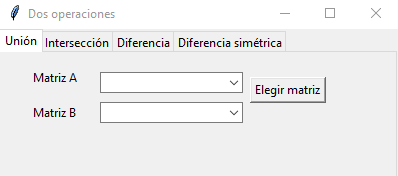
Al presionar esta opción se desplegarán las distintas operaciones que pueden realizar el programa sobre una imagen(matriz).



En algunas operaciones solamente es necesario ingresar o seleccionar el nombre de la matriz a la cual deseemos aplicar dicha operación, pero en otras nos pedirá una coordenada inicial, coordenada final o el numero de elementos.

**Operaciones 2:**

Al presionar esta opción se desplegarán todas las operaciones que podemos realizar con dos imágenes(matrices).



En estas operaciones únicamente es necesario seleccionar las matrices que deseamos operar y acto seguido presionar el botón de elegir matriz para realizar las operaciones.

**Conclusiones**

1. Los tipos de datos abstractos dan mucho pie a la imaginación, con este tipo de estructuras se pueden programas muchísimas cosas que a veces ni imaginamos que se pueden hacer.
2. El manejo optimo de los nodos representa una funcionalidad clave a la hora de trabajar con datos en un programa, ya que esto influye en la fluidez de este tipo de funciones.
3. Todo lo relacionado con las matrices, ya sea en matemáticas o en programación o en muchísimas otras ramas de la ciencia es muy útil, y este programa es una muestra de ello, ya que vimos que es posible implementar y gestionar de una manera correcta todas las operaciones planteadas con matrices.

**Referencias bibliográficas**

GraphViz (8 de marzo de 2021) En Wikipedia.

<https://es.wikipedia.org/w/index.php?title=Graphviz&oldid=119439121>

Matriz ortogonal (06 de noviembre de 2020) En Wikipedia.

https://es.wikipedia.org/w/index.php?title=Matriz\_ortogonal&oldid=130689405