

---

## GUI PARA LA MANIPULACION DE IMAGENES

---

201906053 – Jose Rodrigo Rodas Palencia

### Resumen

El desarrollo de esta aplicación implica el uso de matrices las cuales contienen imágenes en forma de series de caracteres las cuales se busca mostrar de manera gráfica con la herramienta graphviz y a su vez hacer operaciones dentro de la misma.

Estas matrices se definen como objetos los cuales para acceder a los datos almacenados de una manera rápida y simple ya que no trabajan con nodos vacíos y utilizan cuatro apuntadores por cada nodo and una serie de encabezados que apuntan a los nodos.

Como parte de la solución se aplica el manejo de archivos con extensión xml los cuales en su contenido posee las etiquetas que identifican a las matrices y sus datos, con el uso de librerías se acceden a estos archivos para su acceso posterior.

### Palabras clave

Estructuras de datos

Matriz

XML

Lista Ortogonal

### Abstract

*The develop of this aplicacion implicates the use of matrixes that contains images in form of characters that it search to show them in a graphic way using graphviz language and do operation with this matrixes.*

*This matrixs are definid as objects that for get acces to the saved data on them it's in a fast and simple way because it don't work with empty nodes and they use four pointers for each node and headers that point to acces nodes.*

*Part of the solution implicates the manage of files with extension xml that in their content has the tags that identificate matrixes an their data, using two librarys it's build and access to those files for their future access.*

### Keywords

*Data structures*

*Matrix*

*XML*

*Ortogonal List*

## Introducción

El proyecto se fundamenta en el análisis y uso de archivos con estructuras de etiquetas conocido como XML el cual es un estándar usado para la comunicación entre servidores, equipos, paginas, etc. Los archivos a tratar con esta estructura contienen n matrices en ellas estas matrices contienen un numero de filas, un numero de columnas y una imagen la cual se representa por medio de caracteres, lo que se busca es que por medio de estructuras de datos se logre hacer una representación grafica por medio de la herramienta graphviz y a la vez estas imágenes se deben almacenar en una matriz ortogonal para hacer un rápido acceso y evitar el desperdicio de memoria buscando la mejor opción para el desarrollo de este proyecto. También se busca implementar diferentes operaciones entre matrices y operaciones de adición.

## Desarrollo del tema

**XML:** se conoce como xml un modelo de datos semiestructurados y lenguajes de consultas los cuales aprovechan de forma eficaz los datos almacenados en estos tipos de archivos<sup>1</sup>. El estándar xml es utilizado para guardar varias matrices las cuales se definen de la siguiente forma:

```
<matrices>
  <matriz>
    <nombre>Matriz</nombre>
    <filas>1</filas>
    <columnas>1</columnas>
    <imagen>
      *___*_*_*_
      ...
    </imagen>
  </matriz>...
</matrices>
```

Figura 1. Modelo xml para las matrices

Fuente: Elaboración Propia

Con este modelo se busca obtener la imagen que contiene la matriz y guardarla en una lista ortogonal para su posterior acceso y manipulación.

**Imagen:** una imagen es una etiqueta perteneciente a una matriz la cual posee n caracteres los cuales pueden ser “-” los cuales representan un pixel vacío y “\*” que representa un pixel lleno y estos pixeles poseen una coordenada x y una coordenada y la cual define en que parte de la matriz se va a almacenar.

**Lista Ortogonal:** una lista ortogonal se define como una lista que posee encabezados las cuales son listas doblemente enlazadas que apuntan a un nodo el cual se conoce como nodo de acceso, a la vez estos nodos poseen cuatro apuntadores y estos apuntadores a su vez son listas simplemente enlazadas que apuntan a otro nodo. Las listas ortogonales se definen como un TDA que no posee nodos vacíos, en el caso de este proyecto los guiones representan espacios en blanco en una imagen y estos se ignoran y se ignora un espacio por lo que la lista va creando los encabezados junto con los nodos en desorden lo cual representaría un problema, una representación gráfica seria la siguiente:

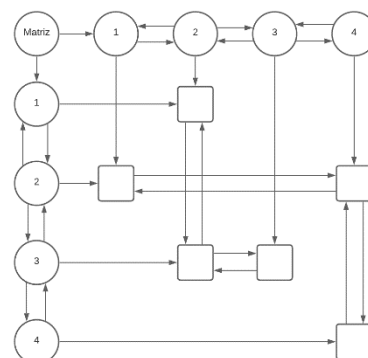


Figura 2. Gráfico lista ortogonal

Fuente: Elaboración Propia

Los métodos los cuales posee una matriz son los siguientes:

- a. **Insertar:** el método insertar agrega un nodo nuevo con un valor "\*" el cual posee 4 apuntadores los cuales pueden estar apuntando o no a 4 diferentes direcciones de memoria, a este se le envían 2 parámetros los cuales son el numero de fila y el numero de columna y con esto se valida la existencia de un encabezado y si no existe se crea dependiendo si hay un encabezado de mayor o menor valor y se coloca en orden y con ello también se valida el nodo de acceso de este encabezado y dependiendo del índice del nodo y del encabezado se determina el acceso de nodo y a su vez se determina a donde apuntaran los 4 apuntadores del nodo a insertar. Los objetos encabezados se definen como listas doblemente enlazadas que almacenan un tipo de nodo encabezado el cual posee tres apuntadores como lo son siguiente, anterior y el acceso a nodo el cual apunta a otro tipo de nodo el cual es el nodo el simboliza un pixel de una imagen lleno.
- b. **Rotación Horizontal y Vertical:** estos métodos lo que realiza es llamar al método devolverFilas y devolverColumnas cada uno respectivamente con su método para luego borrar todos los nodos de la lista ortogonal y luego con los métodos llamados con anterioridad rearmar la matriz utilizando el método insertar restando las filas o columnas con los valores actuales de los nodos a insertar.

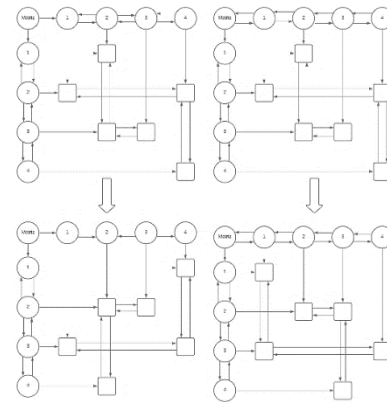


Figura 3. Rotaciones de imágenes

Fuente: Elaboración Propia

- c. **Transpuesta:** este método lo único que realiza es llamar rearmar la matriz con el método devolverColumnas y con los índices rearmar la matriz haciendo las columnas filas y luego llamar al método de rotación horizontal.
- d. **Agregar línea Horizontal y Vertical:** estos métodos lo único que realizan es llamar al método insertar en un recorrido horizontal o vertical con un ciclo for el cual se construye a través de los parámetros enviados los cuales son el número de fila y columna donde debe empezar la adición de nodos y se adicionan n nodos que es un número que es enviado como un parámetro llamado elementos.
- e. **Agregar Triángulo y Rectángulo:** estos métodos reciben 3 parámetros los cuales son el número de fila y columna donde se empezara a construir el triángulo o el rectángulo y la dimensión que en el caso de ser el triángulo solo se envía un valor numérico el cual determina las dimensiones del mismo haciendo referencia a la altura,

base y diagonal de este por lo que se llama al método de insertar una línea horizontal para construir la base y se llama al método de insertar una línea vertical para construir la altura y lo mismo se hace para el rectángulo ya que solo se llama 2 veces al mismo método, para diferenciar las posiciones donde se empezaran a construir se llevan diferentes contadores.

f. **Borrar:** el método borrar recibe dos parámetros los cuales definen la ubicación del nodo a eliminar por lo que se recorre la lista ortogonal, únicamente los nodos existentes, hasta encontrar la coincidencia en el numero de fila y en el numero de columna con ello se hacen las validaciones al nodo preguntando por cada apuntador que posee se valida que este se nulo o no y dependiendo de la respuesta se vuelve a realizar otra validación para saber si se va a volver a apuntar a una dirección de memoria u a los encabezados o bien si el encabezado se elimina ya que no existirá ninguna dirección de memoria a la cual apuntar.

g. **Gráfico:** la lista ortogonal posee el método grafico el cual con el uso de la herramienta graphiz se realiza una representación gráfica de la matriz creando un mapa de los datos de filas\*columnas y se llama al método devolverFilas y con ello se agregan los nodos existentes en la lista, con estos nodos se construye un archivo de formato dot, el cual es la extensión la cual utiliza graphiz para trabajar. Construida la tabla la cual representa la lista ortogonal se procede a llamar a la

consola del sistema y se utiliza el siguiente comando:

```
dot -T png grafo.dot -o imagen.png
```

**Estructuras de Datos:** se conoce como estructuras de datos a diferentes objetos los cuales almacenan una serie de datos de forma dinámica ya que son definidos objetos nodos los cuales son como apuntadores los cuales se definen como su mismo nombre lo dice ya que apuntan a direcciones de memoria. Para la solución del proyecto se utilizaron los tipos de estructuras de datos los cuales son los siguientes:

a. **Lista Circular:** una lista circular es un tipo de tda el cual es muy parecido a una lista simple, pero con la diferencia de que los nodos apuntan al siguiente y el final apunta a la cabeza dando la posibilidad que al momento de recorrerla ningún nodo apunta al valor null, su estructura se define a continuación:

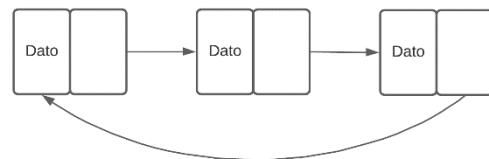


Figura 4. Modelo lista circular

Fuente: Elaboración Propia

El uso de la lista circular implica la validación de que una matriz la cual se quiere utilizar su método graficar se encuentre en la lista ya que el usuario ingresa el nombre de la matriz la cual se quiere crear un grafo con ello se crea

un método buscar el cual recibe como parámetro el nombre de la matriz y recorre la matriz hasta validar que se encuentre.

- b. **Cola:** una cola es un tipo de tda la cual usa el estándar “FIFO” el cual significa First In First Out, el cual su nombre en español nos dice que el primer dato que almacena va a ser el primero en salir de la cola a diferencia de los demás una cola no utiliza nodos y al momento de sacar un valor de la misma este es eliminado por lo tanto esta puede almacenar cualquier tipo de datos y no puede devolver un valor en una posición específica, posee los métodos encolar y desencolar. Su estructura es la que se muestra a continuación:

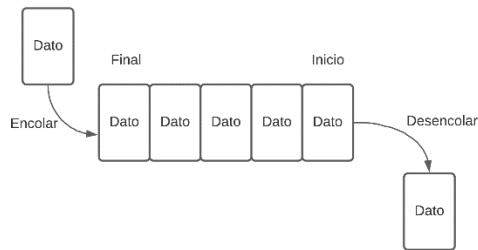


Figura 5. Modelo cola

Fuente: Elaboración Propia

La función del tda cola es almacenar los valores tipo Dato que poseen la matriz para almacenarlo de forma ordenada en filas y columnas.

**Interfaz:** la interfaz se define como la GUI con la que el usuario se comunica con el programa, como primer lugar se debe cargar un archivo de entrada el cual contiene dentro las matrices por lo que con la librería dom se extraen todos los datos que están en las etiquetas del xml y estas se guardan como objetos de tipo ortogonal los cuales poseen un nombre un

número de filas y columnas y m nodos que se almacenan en una lista ortogonal, estos objetos ortogonal se guardan en una lista circular la cual posee la capacidad de retornar un objeto con saber su etiqueta nombre. La interfaz en su barra menú posee diferentes opciones las más básicas son desplegar la documentación y los datos del estudiante y un reporte html que lleva un informe sobre las acciones que realiza el programa, las acciones avanzadas son las operaciones las cuales se pueden realizar entre matrices o bien una matriz, esta posee diferentes cajas de texto y 3 place holders donde se despliegan las imágenes, algunas operaciones son el uso de métodos que se definen al momento de crear un objeto ortogonal pero además de eso están las operaciones entre matrices las cuales son las siguientes:

- a. **Unión:** la operación unión llama a la lista circular y está devuelve dos objetos matrices basándose en los parámetros que le envían las cajas de texto y con ello las dos matrices llamas a su método recorrerFilas y con los datos que devuelven únicamente se crear un nuevo objeto ortogonal el cual tendrá el nombre “Matriz\_U” y tendrá las dimensiones de la primera matriz y con ello se inserta un nodo por cada dato de la primera y segunda matriz. Con ello echo se muestran en pantalla la grafica de la primera matriz y la segunda matriz y se muestra la matriz resultante.
- b. **Intersección:** la operación intersección llama a la lista circular y está devuelve dos objetos matrices basándose en los parámetros que le envían las cajas de texto y con ello las dos matrices llamas a su método recorrerFilas y con los datos que devuelven únicamente se crear un nuevo objeto ortogonal el cual tendrá el nombre “Matriz\_I” y que tendrá las

dimensiones de la primera matriz y con ello se inserta un nodo por cada dato que tengan en común la primera y segunda matriz. Con ello echo se muestran en pantalla la gráfica de la primera matriz y la segunda matriz y se muestra la matriz resultante.

- c. **Diferencia:** la operación diferencia llama a la lista circular y está devuelve dos objetos matrices basándose en los parámetros que le envían las cajas de texto y con ello las dos matrices llamas a su método recorrerFilas y con los datos que devuelven únicamente se crear un nuevo objeto ortogonal el cual tendrá el nombre "Matriz\_D" y que tendrá las dimensiones de la primera matriz y con ello se inserta un nodo en el cual este en la primera matriz pero que no tenga coincidencia en la segunda matriz. Con ello echo se muestran en pantalla la gráfica de la primera matriz y la segunda matriz y se muestra la matriz resultante.
- d. **Diferencia Simétrica:** la operación diferencia simétrica llama a la lista circular y está devuelve dos objetos matrices basándose en los parámetros que le envían las cajas de texto y con ello las dos matrices llamas a su método recorrerFilas y con los datos que devuelven únicamente se crear un nuevo objeto ortogonal el cual tendrá el nombre "Matriz\_DS" y tendrá las dimensiones de la primera matriz y con ello se llama a el método de diferencia dos veces una donde se realiza la operación  $A \cdot B$  y luego  $B \cdot A$  para luego unirlos, con ello se muestra en pantalla las dos matrices y la matriz resultante.

## Conclusiones

Se logra implementar una lista ortogonal para el almacenamiento de datos.

Se logra el cometido del uso de archivos con el estándar XML dando el conocimiento de una estructura de un árbol el cual posee una raíz y n números de hijos.

Con el uso de la herramienta graphiz se logra graficar las diferentes imágenes que poseen las diferentes matrices y a su vez las operaciones entre estas.

Se mantiene enterado al usuario de los errores que ocurren al momento de fallar alguna operación seleccionada por el mismo.

## Referencias bibliográficas

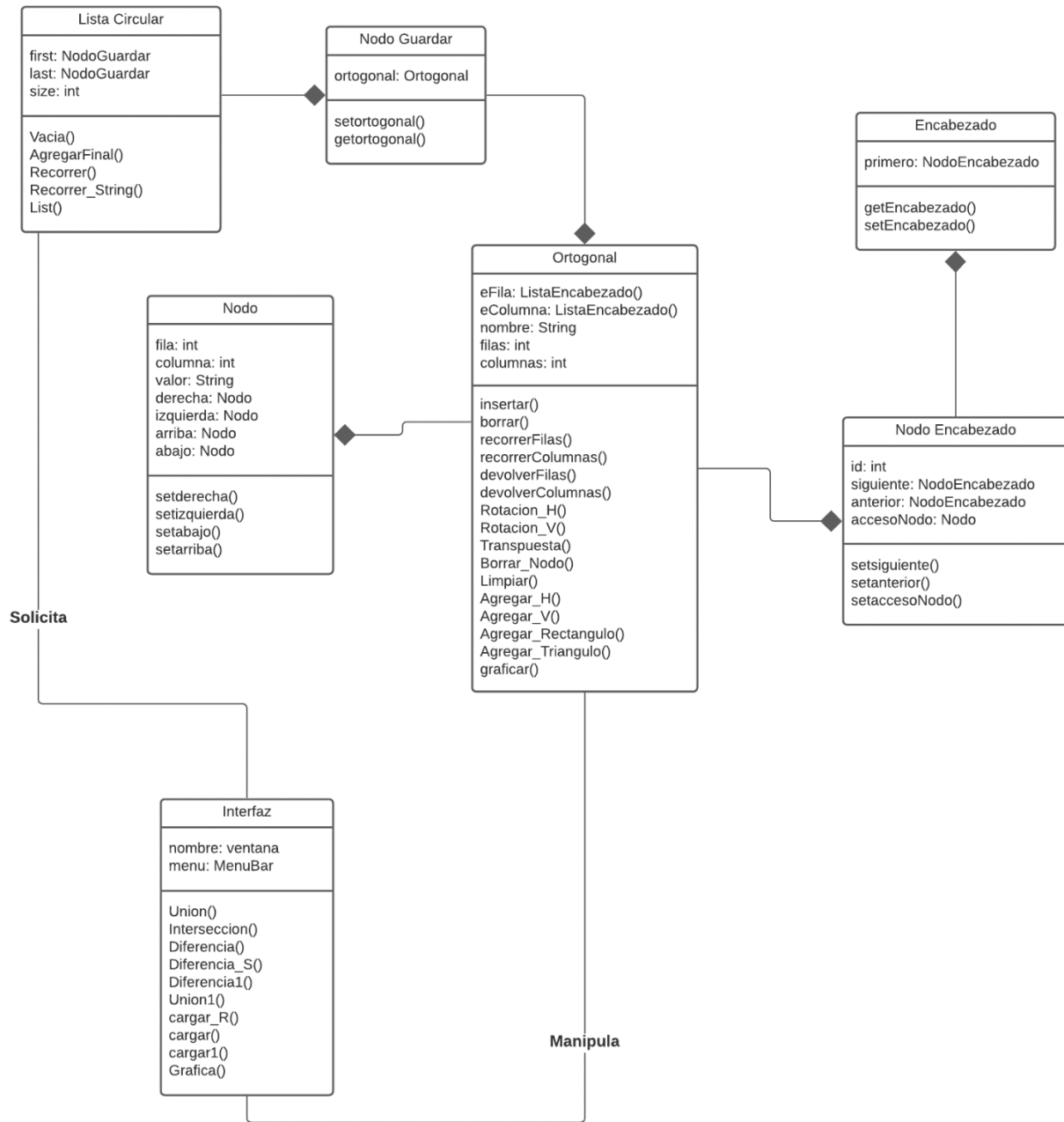
Jones, C. A., & Drake Jr, F. L. (2001). Python & XML: XML Processing with Python. " O'Reilly Media, Inc.".Rodríguez, K. O., & Oviedo, B. J.

(2021). Ejemplos básicos de álgebra lineal con python: Basic examples of linear algebra with python. Revista Digital: Matemática, Educación e Internet, 21(1).

Shipman, J. W. (2013). Tkinter 8.4 reference: a GUI for Python. New Mexico Tech Computer Center, 54.

Ray, J., & Trovati, M. (2017, August). A survey of topological data analysis (TDA) methods implemented in Python. In International conference on intelligent networking and collaborative systems (pp. 594-600). Springer, Cham.

## Apéndice A: Diagrama de clases



## Apéndice B: Diagrama de flujo del método insertar

