

---

## APLICACIÓN DE CONSOLA PARA ARCHIVOS XML

---

201906053 – Jose Rodrigo Rodas Palencia

### Resumen

El desarrollo de esta aplicación implica el uso de matrices las cuales contienen direcciones de memoria las cuales son guardados y reducidos para su posterior acceso por parte de un usuario.

Estas matrices de acceso se definen como objetos los cuales para acceder a los datos almacenados en esta misma se utilizan estructuras de datos para mantener un control sobre los datos.

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 dos librerías se construyen y se acceden a estos archivos para su acceso posterior.

### Palabras clave

Estructuras de datos

Matriz

XML

Memoria

### Abstract

*The develop of this application implicates use of matrixes that contains memory direccions that are saved an reduxed for their future acces from user.*

*This matrixs of acces are defind as objects that for get acces to the saved data on them it's use data structures to have a control on the data*

*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*

*Memory*

## 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 representan accesos a una base de datos por varios equipos realizando los cuales realizan queries, el objetivo de este proyecto es reducir estas matrices de acceso para una buena administración de recursos ya que los queries se pueden unificar para varios dispositivos ahorrando memoria al momento de querer acceder a un dato dentro de la base de datos, las reducciones de estas matrices implican utilizar una matriz binaria la cual nos define si se está o no accediendo a memoria para luego encontrar las coincidencias y unificarlas dando como resultado una matriz con tuplas nuevas.

## 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 de acceso las cuales se definen de la siguiente forma:

```
<matrices>
  <matriz nombre="" n="" m="">
    <dato x="" y="">1</dato>
    <dato x="" y="">1</dato>...
  </matriz>...
</matrices>
```

Figura 1. Modelo xml para las matrices de acceso

Fuente: Elaboración Propia

Con este modelo se busca obtener los datos y almacenarlos para su posterior acceso y con ello realizar la reducción para luego escribir un nuevo archivo xml el cual contendrá las matrices reducidas y las frecuencias las cuales son las veces que se encontraron coincidencias, se define de la siguiente manera:

```
<matrices>
  <matriz nombre="" n="" m="" f="">
    <dato x="" y="">1</dato>
    <dato x="" y="">1</dato>...
    <frecuencia g=""></frecuencia>
  </matriz>...
</matrices>
```

Figura 2. Modelo xml para las matrices reducidas

Fuente: Elaboración Propia

**Matriz de accesos:** la matriz de accesos se define como una matriz la cual contiene datos los cuales representan accesos a una bd o en este caso queries los cuales solicitan los datos a la bd, la estructura de estos es el uso de nodos y tda los cuales almacenan estos datos como se presenta:

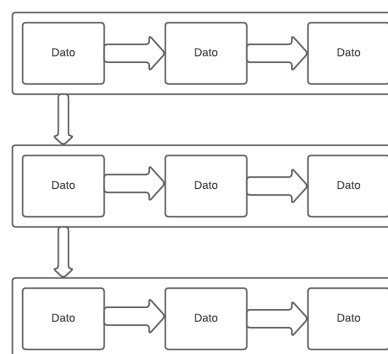


Figura 3. Modelo para las matrices de acceso

Fuente: Elaboración Propia

Los métodos los cuales posee una matriz son los siguientes:

- a. Binario: el método binario lo que realiza es tomar cada fila de nuestro objeto matriz y obtener los datos y dividirlos entre estos mismos devolviendo así un objeto matriz la cual posee datos en lenguaje binario y se le devuelven al método tuplas.
- b. Tuplas: el método tuplas toma la matriz binaria y toma cada fila y la compara con las filas las cuales les siguen sin volver a la fila anterior la cual se está comparando, al momento de encontrar una coincidencia entre filas marca la posición y agrega un valor más a una lista la cual almacena la frecuencia con la que se encontraron coincidencias y también se verifica que no se estén almacenando datos duplicados ya que al momento de comparar la primera fila y encontrar más coincidencias se verifican que estos valores ya han sido almacenados y no duplicarlos ya que las frecuencias funcionan como un rango de búsqueda para las posiciones donde hay coincidencias.
- c. Sumas: con las listas de frecuencias y coincidencias se realizan los recorridos siendo el valor de frecuencia nuestro rango de búsqueda para las primeras coincidencias de filas, con ello se toman los valores de filas que poseen coincidencias y se definen dos objetos los cuales almacenan diferentes listas de datos, uno almacena la fila anterior, si no es la primera vez que se entra al ciclo, y uno que almacena la lista de datos actual y con ello se realiza la suma de dos listas para luego

retornar la fila sumada si no hay más coincidencias y si hay más se sigue sumando hasta terminar el ciclo, esta fila nueva identificada como fila sumada se devuelve y se almacena a un nuevo objeto el cual se define como matriz redux la cual es un objeto Matriz el cual almacena las listas ya reducidas y sumadas para luego buscar las filas las cuales no tienen coincidencias y se agregan a esta matriz redux y se devuelve la matriz redux junto con las frecuencias para su posterior generación en un archivo XML.

- d. Gráfico: la matriz posee el método grafico el cual con el uso de la herramienta graphiz se realiza una representación gráfica de la matriz agregando objetos llamados nodos por cada dato que la matriz posee se crea un nuevo nodo recorriendo fila por fila y cada nodo apuntando al nodo que le sigue en su misma posición “y” pero diferente posición en “x”, con estos nodos se construye un archivo de formato dot, el cual es la extensión la cual utiliza graphiz para trabajar, dentro de este se realiza un tipo de apuntador ya que para hacer una relación entre nodos se deben apuntar entre ellos para crear las relaciones. Echadas todas las relaciones y recorrida toda la matriz se procede a llamar a la consola de la computadora y se utiliza el siguiente comando:

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

El cual convoca al archivo dot creado y lo genera como una imagen en formato “png” ya que los archivos dot son muy parecidos a los

HTML ya que al momento de abrirlos solo se mostrarán líneas con código de etiquetas sin generar una forma gráfica hasta mostrarlo desde un navegador.

**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 3 tipos de estructuras de datos los cuales son los siguientes:

- a. **Lista Simple Enlazada:** una lista simple es una lista la cual posee apuntadores únicamente para el siguiente valor de la lista y el ultimo apuntando a nulo, teniendo su nodo un atributo next, esta lista se define de 3 maneras para almacenar 3 tipos de datos, pero al momento de definir una lista enlazada se define si este almacena strings, datos u una lista enlazada de datos o strings. Las listas enlazadas son definidas para construir una matriz sin el uso de listas simples, el objetivo del mismo es tener un conocimiento de cómo funcionan las listas internamente. La estructura de una lista simple es la siguiente:



Figura 4. Modelo lista simple enlazada

Fuente: Elaboración Propia

- b. **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:

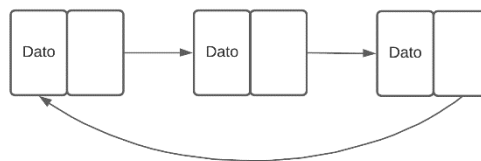
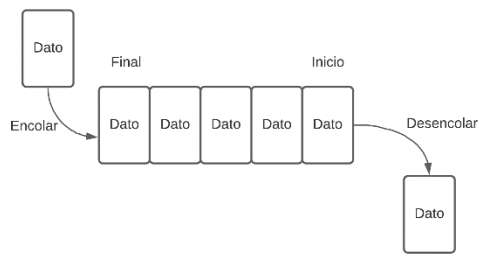


Figura 5. Modelo lista circular

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.

- c. **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 6. 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.

### Conclusiones

La idea principal del proyecto es resuelta con el uso de los tda con ello se cumple el objetivo del proyecto, pero estos mismo implican un problema al momento de realizar comparaciones u obtener datos de una manera simple y rápida.

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 matrices las cuales el usuario desea graficar.

Se mantiene enterado al usuario de los procesos los cuales el programa está realizando.

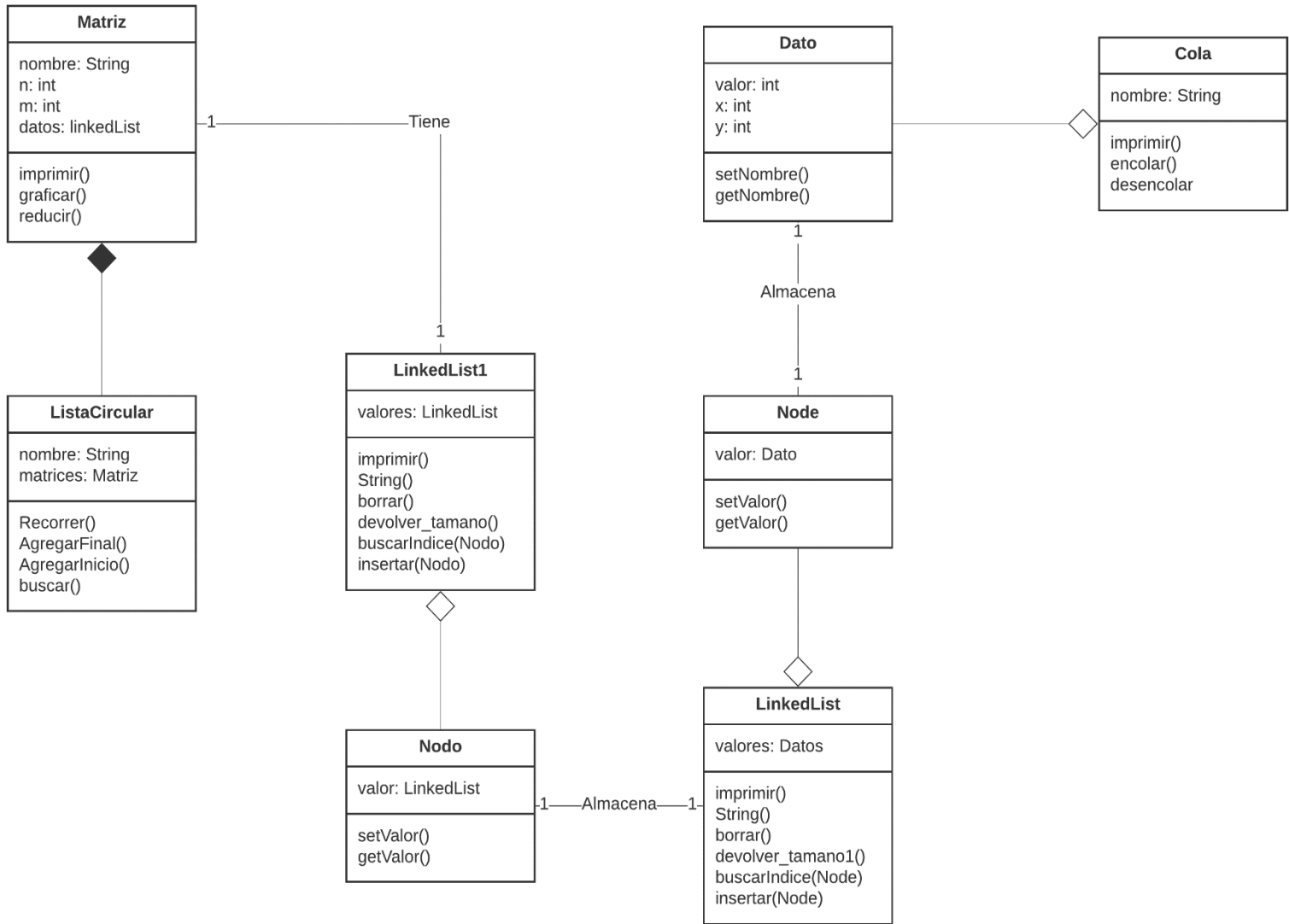
### Referencias bibliográficas

DE, M. Tabla 8 Matriz de acceso y control de recursos y beneficios. Cooperación descentralizada pública, 126.

He, J. S. K. T. G., & Naughton, C. Z. D. D. J. (2008). Relational databases for querying XML documents: Limitations and opportunities. In Proceedings of VLDB (pp. 302-314).

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: Logica del programa

### Extraccion de la matriz

