|  |
| --- |
| OPTIMIZACION DE BASES DE DATOS UTILIZANDO LA METODOLOGIA NP-HARD |
| **201908327-Diego André Gómez Esturbán** |

**Resumen**

Se realizo un programa basado en la resolución de los problemas NP-Hard, siendo estos los conjuntos de problemas de decisiones que contienen los problemas H tales que todo problema L en NP puede ser transformado polinomialmente en H, también se realizó una reducción de alojamiento de objetos para determinar de que manera se obtendría un mejor costo/beneficio al mantener activos los servidores.

Se crearon nuevos esquemas basados en los datos proporcionados originalmente por el usuario, siendo este nuevo esquema el que daría mayor beneficio a toda la empresa para la cual se calculan los datos.

Se utilizaron archivos de tipo XML como formato de entrada y salida de datos para poder manejar un estándar único de etiquetas y facilitar asi la lectura y la escritura tanto como para los usuarios al crear los archivos que ingresaran al programa, como también para el programa mismo al momento de leer el archivo proporcionado por los usuarios y también para cuando se requiera que el propio programa cree su propio archivo de tipo XML.

**Palabras clave**

* Tuplas
* Binario
* Matriz
* Frecuencia
* XML

***Abstract***

*A program based on the resolution of the NP-Hard problems was carried out, these being the sets of decision problems that contain the problems H such that every problem L in NP can be polynomially transformed into H, a reduction of accommodation of objects to determine how a better cost / benefit would be obtained by keeping the servers active.*

*New schemes were created based on the data originally provided by the user, this new scheme being the one that would give the greatest benefit to the entire company for which the data is calculated.*

*XML-type files were used as data input and output format to be able to handle a single standard of labels and thus facilitate reading and writing both for users when creating the files that will enter the program, as well as for the program itself. when reading the file provided by users and also when the program itself is required to create its own XML file.*

***Keywords***

* *Tuples*
* *Binary*
* *Matrix*
* *Frecuency*
* *XML*

**Introducción**

El proyecto se basó en la utilización del modelo NP-Hard para la resolución de modelos combinatorios, para esto se determinó el alojamiento de datos de forma que los costos de acceso y comunicaciones fueran minimizados.

Esto se logró al analizar la matriz que el usuario ingresa en un archivo XML para poder verificaren cuales puntos se registran acceso, gracias a realizar esto se crea una matriz binaria en la que se representan los accesos con un 1 y en las que no se tiene acceso con un 0, con la misma información se crea una matriz de frecuencias utilizando las tuplas que tienen el mismo patrón, al realizar esto es posible determinar cuáles son los puntos de acceso más utilizados.

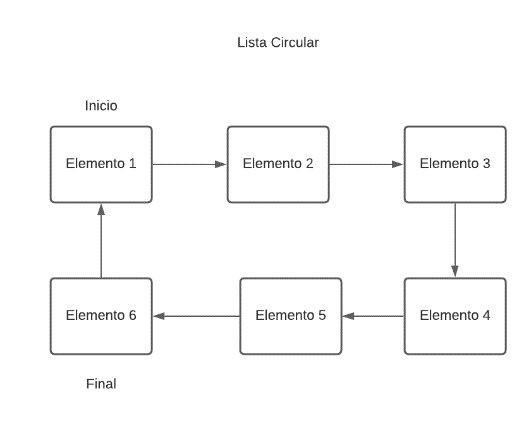
Se utilizo la herramienta grafica Graphviz para poder crear un grafo que represente la estructura del archivo previamente procesado, gracias a esto se facilita la lectura para que el usuario comprenda todo de una manera cómoda y sencilla

**Desarrollo del tema**

El programa permite al usuario poder cargar un archivo de extensión XML, este archivo debe cumplir con cierta estructura para que el programa sea capaz de procesar el archivo, al incluir las etiquetas indicadas el programa es capaz de realizar una matriz binaria y una matriz de frecuencias basada en la matriz ingresada previamente por el usuario, para esto el usuario debe ingresar la dirección física del archivo que desea procesar, esta opción se presenta en el menú de la consola como la primera opción.

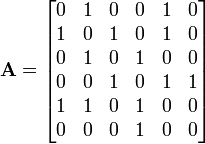
Para poder cargar el archivo el usuario debe ingresar la dirección física del archivo que desea procesar, gracias a esto el programa guarda el archivo en una variable que puede ser utilizada luego para realizar el análisis y procesamiento del archivo.

Para procesar el archivo previamente ingresado el usuario deberá utilizar la segunda opción mostrada en el menú de consola, esta función utilizará el método ProcesarArchivo el cual creará una lista circular, también recogerá los datos de cuantas Tuplas y columnas se encuentran en el archivo marcado por las etiquetas, se recorrerán los hijos de las etiquetas y se guardarán los datos para posteriormente almacenarlos en un objeto que sea capaz de contener todos los datos recopilados del archivo. Gracias a lo previamente hecho se crea una matriz binaria revisando que en cada espacio de la matriz donde dato guardado sea diferente de cero sea reemplazado con un numero “1”, de esta manera obtendríamos una segunda matriz.



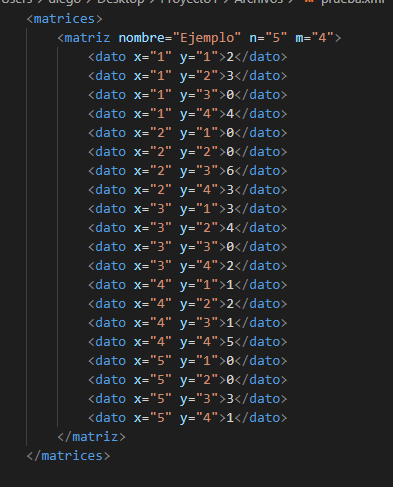
Representacion grafica de Lista Circular

Para llegar a obtener una matriz reducida de frecuencia es necesario compara las tuplas creadas en la matriz binaria, si las tuplas coinciden entre si se procede a realizar la suma de los datos de cada tupla con las coincidentes, las tuplas que no tengan ningún coincidente pasaran a ser un grupo solitario, al realizar este procedimiento se creara una nueva matriz con los datos obtenidos.



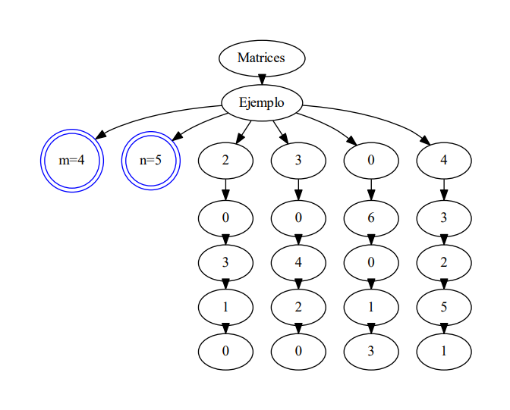
Representacion de una matriz Binaria

Para realizar la escritura de un archivo el programa utiliza las matrices de frecuencias para obtener los datos, de estas se escribe un archivo de extensión XML, con las mismas etiquetas del archivo de entrada original, pero añadiendo las frecuencias que presentan los grupos creados mediante el método del procesado del archivo.



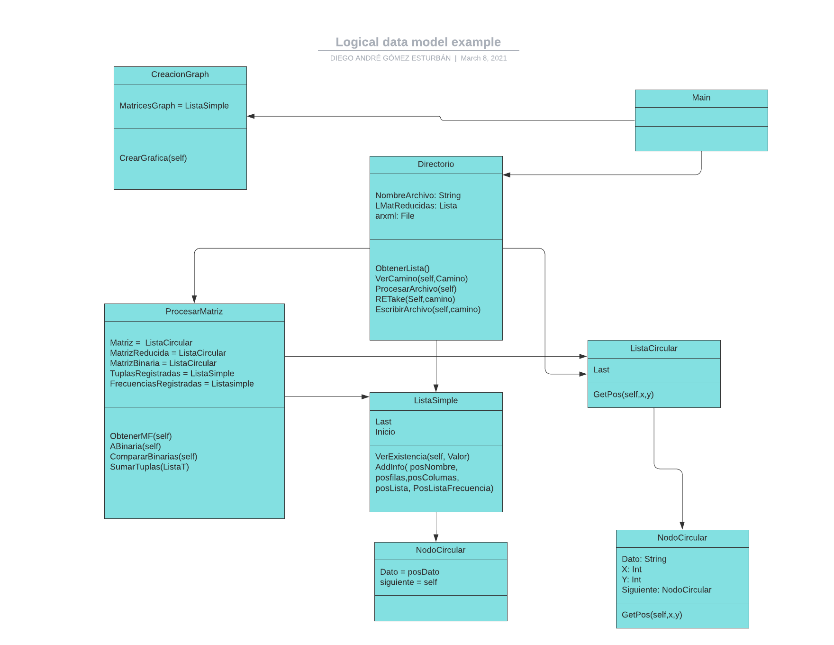
Ejemplo del archivo de salida generado por el programa

La creación de una grafica a partir del archivo ingresado por el usuario está diseñado utilizando Graphviz, esto permite mantener una estética ordenada sin que el usuario tenga que preocuparse al momento de cargar sus archivos de entrada. El método crea un archivo “.dot” en el que se colocaran los datos recopilados por el segundo método, en este método se recorren las filas y las columnas y al hacer esto es capaz de imprimir los datos que se encuentran en cada posición siendo luego enlazados entre sí, al terminar esto se crea un archivo de extensión pdf que se abre al terminar de ser escrito.



Ejemplo de una grafica creada por el programa

DIAGRAMA DE CLASES



**Conclusiones**

El programa fue desarrollado utilizando programación orientada a objetos, gracias a esto se pueden implementar nuevas funcionalidades sin necesidad de reescribir el código ya existente.

El estándar XML es increíblemente útil para proporcionar datos de manera legible y que tanto el programador como el usuario tengan entendimiento de lo que están leyendo.

Al analizar los problemas con el modelo NP-Hard se puede llegar a obtener modelos mucho mas eficientes que los que se tenían establecidos originalmente, ahorrando muchos costos a las empresas.

**Referencias bibliográficas**

A. (s. f.). Python - Leer archivo XML. decodigo.com. Recuperado 2 de marzo de 2021, de <https://decodigo.com/python-leer-archivo-xml>

Listas en Python. (2017, 19 septiembre). DevCode Tutoriales. <https://devcode.la/tutoriales/listas-python/>

uniwebsidad. (s. f.). 17.2. Colas (Algoritmos de Programación con Python). Recuperado 4 de marzo de 2021, de <https://uniwebsidad.com/libros/algoritmos-python/capitulo-17/colas>