
PROYECTO 1 METODOLOGÍA DE AGRUPAMIENTO

201900955 – Diego Fernando Cortez López

Resumen

Obtener un esquema de alojamiento de objetos de base de datos donde el costo de acceso y comunicación sean minimizados.

El proyecto se aplicará una metodología de agrupamiento, que consiste en transformar una matriz de frecuencias de acceso a una matriz de patrones de acceso.

El método consiste en tener la matriz de frecuencia de acceso y transformarla en una matriz de patrones binarios para agrupar las tuplas con el mismo patrón.

Palabras clave

Tupla, patrón, lista, nodo y valor.

Abstract

Obtaining a host scheme is database objects where the cost of access and communication are minimized.

The project will apply a grouping methodology, which involves transforming an array of access frequencies to an array of access patterns.

The method is to have the access frequency matrix and transform it into an array of binary patterns to group the tuples with the same pattern.

Keywords

Tuple, pattern, list, node, and value

Introducción

Se elaboró un método que recibe una matriz de patrones de n-filas y n-columnas, y transfórmala en una matriz binaria para operar las tuplas idénticas y poder reducir la matriz de patrones. El objetivo del método es reducir el recurso de memoria necesario para almacenar datos en una base de datos en sitios distribuidos.

Los datos de la matriz son almacenados en listas circulares, que es una estructura de datos abstractos que almacena los datos de una forma ordenada y de forma dinámica. Se logra almacenar una colección de elementos en los nodos, que almacenan los datos y ligan a otros nodos, para poder localizarse en cualquier parte de la memoria, utilizando una referencia que relaciona el nodo.

Para la elaboración de los métodos de la aplicación se realizaron mediante programación orientada a objetos, donde se contienen los métodos para la reducción de la matriz y el almacenamiento de los datos de los patrones.

Desarrollo del tema

Documentos XML, es un documento con lenguaje de enmarcado que permite definir etiquetas personalizadas para descripción y organización de datos. XML significa lenguaje extensible de marcas, y es utilizada para representar información estructurada en la web, de modo que esa información pueda ser almacenada, transmitida, procesada y visualizada por diversas aplicaciones y dispositivos.

a. Carga de Archivo

La aplicación solicita la ruta en la que se encuentra ubicado el archivo XML, desde cualquier lugar en la computadora, ingresando la ruta de acceso. El archivo de entrada XML está limitado por las etiquetas:

- Matrices: es la etiqueta raíz o padre, en la cual se derivarán las demás etiquetas del archivo.
- Matriz: esta etiqueta es la que se encarga de indicar la matriz que será utilizada para el análisis. Los atributos que posee son nombre, el número de filas y el número de columnas.
- Dato: esta etiqueta contiene los valores que contiene la matriz en cada celda, Los atributos que contiene es la posición en la fila y la posición de la columna.

Se almacenan los datos del archivo en listas circulares para mantenerlos en la memoria y poder manipular los datos en el programa.

Una lista enlazada es una colección de elementos denominados nodos, y cada nodo almacena un dato y un valor de siguiente nodo. La lista circular es un tipo de lista enlazada que contiene un nodo head como el primer nodo de la lista y que esta enlazada como el nodo siguiente del último elemento de la lista.

b. Procesar el Archivo

Con los datos almacenados en listas, se obtienen los grupos que poseen patrones binario similares.

El programa maneja la matriz como una lista anidada, donde una lista contiene sublistas que serían las filas de la matriz original. Se crea la matriz de patrones con la función “crearMatrizPatrones()”, que mediante la matriz guardada crea una nueva matriz binaria, posicionando un 1 en los valores mayores a 0. Luego con la matriz binaria se recorre cada fila y se buscan las filas similares, con la función “grupos_Matriz()”, que retorna una lista con las posiciones de los grupos que son idénticos en la matriz original.

Recorre la matriz original para evaluar las listas idénticas, los resultados se almacenan en una lista para luego almacenar la lista a otra y obtener la matriz reducida.

Se almacena los datos de la nueva matriz reducida en una nueva clase. Se almacena el nombre, el número de filas, el número de columnas, los grupos que se crearon y la frecuencia de cada grupo.

c. Escribir Archivo de salida

Al seleccionar esta opción se crea un archivo XML con las matrices reducidas que resultaron en la opción “Procesar el Archivo”. El archivo de salida contendrá las mismas etiquetas y la misma estructura que el archivo de entrada, pero con algunos cambios que son:

- La etiqueta **matriz** contendrá un atributo que representa el número de grupos que contiene la matriz reducida.
- Aparte de la etiqueta **dato**, dentro de la etiqueta **matriz**, se agregará la etiqueta **frecuencia**. Esta etiqueta contiene de atributo los grupos formados y la frecuencia en la que aparecían en la matriz de patrones original.

La ubicación del archivo será determinada por el usuario, mediante una ruta absoluta o una ruta relativa.

d. Mostrar datos del estudiante

Se muestra en consola el nombre, carne y la información del curso del estudiante que realizo la aplicación.

Graphviz es un paquete que facilita la creación y representación de descripciones de gráficos en el lenguaje DOT, que es un lenguaje de descripción de gráficos. Graphviz trabaja con nodos, que son donde se almacenará el dato y contendrá un valor para identificar cada nodo; y bordes, que son los que conectan un nodo con otro.

d. Generar gráfica

Se implementó la librería de Graphviz para generar el reporte. Primero se muestra en consola las matrices que están almacenadas en memoria para seleccionar la que se desea generar una gráfica. Al escribir el nombre de la matriz, se busca en los nombres almacenados, retornando un valor booleano para validar la búsqueda.

Se recorre en memoria para seleccionar la matriz que pertenezca el nombre ingresado. Se crea un nodo inicial llamado “Matrices” que se conectara a un nodo con el nombre de la matriz. Se crean un nodo para el numero de filas y otro para el numero de columnas, se agregan los atributos de color y “shape”, para crear un doble circulo, que se conectaran con el nodo del nombre.

Se crean un nodo para cada valor de la matriz, haciendo que la primera fila conecte con el nodo de nombre y las demás filas se conecten sucesivamente una detrás de la otra.

Se implementó la programación orientada a objetos para la realización de las funciones de operar la matriz.

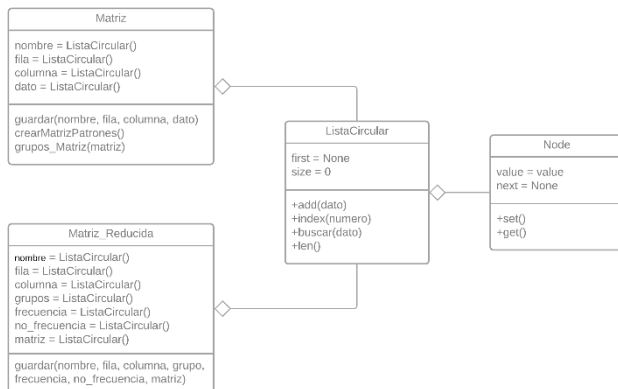


Figura 1. Diagrama de clases de la aplicación.

Fuente: elaboración propia.

Conclusiones

- La programación orientada a objetos es una forma de facilitar la programación y reducir líneas de código. Permite que el programa se pueda modificar de forma sencilla, sin que afecte otra parte del programa. Permite que el programa sea fácil de leer y comprender

permitiendo que sean visibles los detalles relevantes del programa.

- Los tipos de datos abstractos son conjuntos de valores y operaciones que se definen mediante cualquier representación. Las listas son una forma de manipular TDA, y permitir guardarlos en memoria. La lista permite que la memoria se adapte dinámicamente dependiendo del número de datos almacenados en cada momento, lo que permite que se pueda agregar más elementos conforme transcurre el programa.
- Graphviz nos permite visualizar de mejor manera los datos mediante grafos, lo que permite que tenga un orden de forma clara y sencilla.

Referencias bibliográficas

Calzadilla, J. C. F., Herrera, A. N., & Delfino, E. La enseñanza de los arrays estáticos, dinámicos y listas enlazadas ¿cuál usar? Análisis de códigos The teaching of static arrays, dynamics and linked lists. What to use? Code analysis.

Díaz, G. *Listas Enlazadas* (Doctoral dissertation, Universidad de Los Andes, Mérida 5101 Venezuela).

Gómez Dueñas, L. F. (2011). XML, la base de la interoperabilidad en los sistemas de información documental. *Códices*, 3(2).

graphviz. (s. f.). Recuperado 1 de marzo de 2021, de <https://pypi.org/project/graphviz/>

López, L. (2006). Programación orientada a objetos.

Diagrama de Clases

