

Programa 03: Grafos

30 de octubre de 2022

Carlos Reyes Rico. 217458353.

Estructura De Datos 2.



## OBJETIVO

Codificar un grafo con las siguientes características:

- 1) Debe de hacer uso de memoria dinámica (apuntadores)
- 2) Crear una clase interfaz o en el main para el uso completo de todas sus operaciones
- 3) Crear los métodos de guardar/cargar para toda la lista simple, el archivo debe llamarse file01.txt y la información se almacenará con delimitadores de campo "|" y delimitador de registro "\*\*"
- 4) Crear una clase que será su dato en cada nodo. (3 atributos privados, sus métodos get y set, así como los métodos que crean que sean de utilidad)
  - a) A la plataforma debe de subir su programa y una reporte con impresiones de pantalla del funcionamiento de su programa.
    - a.1) Debe de tener una explicación breve que hace su programa
    - a.2) Debe de explicar como organizo sus datos en la TDA
    - a.3) Debe de explicar la estrategia que uso para almacenar y recuperar sus dato.
    - a.4) El reporte debe de tener el siguiente nombre reporte01-PrimerApellido SegundoApellido Nombre.pdf
  - b) El nombre del programa debe ser prog01-PrimerApellido SegundoApellido Nombre.cpp o comprimir todo el proyecto en un solo archivo esté nombre

## QUE HACE EL PROGRAMA

El programa realizado es un grafo que crea vértices de ciudades y enlaza esos mismos vértices con ellas mismas denominándolas aristas con un peso que se refiere a kilómetros de distancia entre su recorrido entre las ciudades. Al enlazar los vértices denominamos cuales están conectados y cuales no, esto nos serviría para realizar búsquedas o recorridos dentro del grafo.

## EXPLICACIÓN DE LA TDA

Para la organización de datos en mi TDA cree 3 clases las cuales denomine, Vértices, Aristas y Grafo, vértices se encarga del proceso de el nombre de ciudades, mientras que aristas denomina el peso entre las conexiones creadas y la clase grafos es la que se encarga de todas las operaciones de nuestro grafo.

### Operaciones

- Insertar Vértice.

Esta operación se encarga de crear los vértices de nuestro grafo y agregarlos a lista de adyacencia.

```
C:\Users\chris\OneDrive\Escritorio\Grafo\bin\Debug\Grafo.exe

.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir

R= 1
Ingresa nombre del vertice: Guadalajara
Presione una tecla para continuar . . .
```

```
Guadalajara->
Puebla->
DF->
Zacatecas->
Michoacan->
Los Cabos->
Presione una tecla para continuar . . .
```

- Insertar Arista.

Esta operación se encarga de crear las aristas de nuestro grafo, solicitando el vértice origen del cual se va a enlazar la arista y un vértice destino que va a ser la arista enlazada con el vértice, además aquí se declara el peso entre ciudades.

Así mismo esta arista es agregada a la lista de adyacencia.

```
C:\Users\chris\OneDrive\Escritorio\Grafo\bin\Debug\Grafo.exe

.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir

R= 2
Ingresa nombre del vertice origen: Guadalajara
Ingresa nombre del vertice destino: DF
Ingresa el peso: 200
```

```
Guadalajara->DF->Zacatecas->Los Cabos->
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
Michoacan->Guadalajara->DF->
Los Cabos->Michoacan->
Presione una tecla para continuar . . .
```

- Eliminar Vértice.

Esta operación se encarga de eliminar un vértice creado además de también eliminar las conexiones que tenga este vértice con otros vértices en sus aristas.

```
C:\Users\chris\OneDrive\Escritorio\Grafo\bin\Debug\Grafo.exe

.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir
R= 5
Ingrese el vertice a eliminar: Guadalajara
Presione una tecla para continuar . . .
```

```
Guadalajara->DF->Zacatecas->Los Cabos->
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
Michoacan->Guadalajara->DF->
Los Cabos->Michoacan->
Presione una tecla para continuar . . .
```

```
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
Michoacan->DF->
Los Cabos->Michoacan->
Presione una tecla para continuar . . .
```

- Eliminar Arista.

Esta operación se encarga de eliminar una arista de nuestro grafo, quitando la conexión de esa arista con su respectivo vértice sin afectar a los demás vértices.

```
.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir
R= 6
Ingrese vertice origen: Guadalajara
Ingrese vertice destino: DF
Presione una tecla para continuar . . .
```

```
Guadalajara->DF->Zacatecas->Los Cabos->
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
Michoacan->Guadalajara->DF->
Los Cabos->Michoacan->
Presione una tecla para continuar . . .
```

```
Guadalajara->Zacatecas->Los Cabos->
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
Michoacan->Guadalajara->DF->
Los Cabos->Michoacan->
Presione una tecla para continuar . . .
```

- Anular.

Esta operación vacía nuestro grafo eliminando todos los datos del mismo

```

.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir

R= 7

Anulando datos...
Datos Anulados
Presione una tecla para continuar . . .

```

```

C:\Users\chris\OneDrive\Escritorio\Grafo\bin\Debug\Grafo

.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir

R= 3

El grafo esta vacio
Presione una tecla para continuar . . .

```

- Lista De Adyacencia.

En esta operación mostramos todos los vértices creados junto con sus respectivas conexiones de aristas.

```

.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir

R= 3

Guadalajara->DF->Zacatecas->Los Cabos->
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
Michoacan->Guadalajara->DF->
Los Cabos->Michoacan->
Presione una tecla para continuar . . .

```

- Guardar

El método guardar se encarga de realizar un respaldo de datos de nuestro grafo para posteriormente recuperar esos datos al correr el programa.

```
C:\Users\chris\OneDrive\Escritorio\Grafo\bin\Debug\Grafo.exe
.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir
R= 8
Guardando datos...
Datos Guardados
Presione una tecla para continuar . . .
```

```
file01: Bloc de notas
Archivo Edición Formato Ver Ayuda
Guadalajara*Puebla*DF*Zacatecas*Michoacan*Los Cabos*$
```

```
file02: Bloc de notas
Archivo Edición Formato Ver Ayuda
Guadalajara|DF|300*Guadalajara|Zacatecas|200*Guadalajara|Los Cabos|400*Puebla|Michoacan|100*Puebla|DF|300*DF|Los Cabos|
```

- Cargar

El método cargar se encarga de recuperar los datos guardados por el método guardar, y así mismo cada que se abra el programa podemos recuperar los datos anteriores.

```
C:\Users\chris\OneDrive\Escritorio\Grafo\bin\Debug\Grafo.exe
.:MENU:.
1. Ingresar Vertice
2. Ingresar Arista
3. Lista de Adyacencia
4. Tamano
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir
R= 3
El grafo esta vacio
Presione una tecla para continuar . . .
```

```
5. Eliminar Vertice
6. Eliminar Arista
7. Anular
8. Guardar
9. Cargar
0. Salir
R= 9
Informacion cargada
Cargando datos...
Datos Cargados
Presione una tecla para continuar . . .
```

```
8. Guardar
9. Cargar
0. Salir
R= 3
Guadalajara->DF->Zacatecas->Los Cabos->
Puebla->Michoacan->DF->
DF->Los Cabos->
Zacatecas->Puebla->Los Cabos->
```

## MÉTODO GUARDAR Y CARGAR

- Guardar

Para el funcionamiento del método guardar utilizamos la librería `fstream` con el método `ofstream`, dentro de nuestra función declaramos un auxiliar el cual posicionamos en la primer posición ahora verificamos que no genere algún error crear el archivo, si no hay error pasamos a un ciclo `while` el cual recorrerá nuestro grafo hasta que el dato sea nulo, dentro del `while` cargaremos en nuestra variable de `ofstream` el nombre del vertice, cuando termine el ciclo pasamos a guardar las aristas en otra función.

El método de guardar aristas funciona de la misma forma que el de guardar vértices, solamente que ahora guardaremos los enlaces del grafo, agregando a nuestro archivo el origen, destino y peso de cada arista.

- Cargar

Para el funcionamiento del método cargar utilizamos la librería `fstream` con el método `ifstream`, dentro de nuestra función declaramos los datos a recuperar, ahora verificamos que no genere algún error crear el archivo, si no hay error pasamos a leer el archivo con un `getline`, ahora creamos un ciclo `while` el cual recorrerá nuestro grafo con el mismo `getline` creado hasta que el texto sea igual a '\$', dentro del `while` y con un `try catch` leemos el texto con un `getline` hasta que el texto sea '|', ese texto leído lo declaramos como nombre y llamamos el método insertar vértice para agregarlo a nuestro grafo, cuando se rompa el ciclo comenzamos a cargar las aristas.

El método de cargar aristas funciona de la misma forma que el de cargar vértices, solamente que ahora cargaremos los enlaces del grafo, para ello leeremos el siguiente texto creado y con `getline` declararemos el origen, destino y peso de la arista, para después mandar a llamar la función insertar arista pasando como parámetro los datos leídos.