

CODIGO DEL PROGRAMA

DAYSJ JARA ARISACA

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
using namespace std;

void generarGrafo(const vector<vector<int>>& listaAdyacencia, int n) {
    ofstream archivoDot("grafo.dot");

    if (archivoDot.is_open()) {
        archivoDot << "digraph G {" << "\n";

        for (int i = 0; i < n; i++) {
            archivoDot << "    \"" << i + 1 << "\";" << "\n";
        }

        for (int i = 0; i < n; i++) {
            for (int j : listaAdyacencia[i]) {
                archivoDot << "    \"" << i + 1 << "\" -> \"" << j + 1 <<
                "\";" << "\n";
            }
        }

        archivoDot << "}" << "\n";
        archivoDot.close();

        system("dot -Tpdf grafo.dot -o grafo.pdf");
        system("dot -Tpng grafo.dot -o grafo.png");
        system("start grafo.pdf");
        system("start grafo.png");

        cout << "\nEl grafo se ha generado en formato PDF y PNG " ;
    } else {
        cout << "\nError con el archivo grafo.dot." ;
    }
}

int main() {
    char opcion;
    bool salir = false;
    int numero[100][100];
    int n = 0;
    bool regresarMenu = false;
    vector<vector<int>> listaAdyacencia;
```

```

while (!salir) {
    cout << "\n      MENU";
    cout << "\nEscribe una opcion";
    cout << "\n";
    cout << "\n- Crear grafo(a)";
    cout << "\n- Representacion del grafo(b)";
    cout << "\n- Valencias del grafo(d)";
    cout << "\n- Generar grafo en formato PDF Y PNG(e)";
    cout << "\n- Salir(f)";
    cout << "\n";
    cout << "\nOpcion=";
    cin >> opcion;
    cout << "\n";

    //para observar mejor la opcion ejecutada, se debe subir un poco
    con el mouse
    // porque instantaneamente se regresa al menu para seleccionar
    una opcion nuevamente
    //por lo que puede no observarse bien la opcion ejecutada

    switch (opcion) {
        case 'a':
            cout << "Ingrese el número de vértices: ";
            cin >> n;

            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    cout << "Ingresa un valor en [" << i << "][" << j
<< "]: ";

                    cin >> numero[i][j];
                }
                listaAdyacencia.push_back(vector<int>());
            }

            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    if (numero[i][j] != 0) {
                        listaAdyacencia[i].push_back(j);
                    }
                }
            }
            break;

        case 'b':
            char opcionb;
            regresarMenu = false;

            while (!regresarMenu) {
                cout << "\nEscoge una opcion";

```

```

cout << "\n - Matriz de Adyacencia(a)";
cout << "\n - Lista de Adyacencia(b)";
cout << "\n - Matriz de Incidencia(c)";
cout << "\n Volver al menu (f)";
cout << "\n";
cout << "\n opcion=";
cin >> opcionb;
cout << "\n";

switch (opcionb) {
    case 'a':
        cout << "Matriz de Adyacencia";
        cout << "\n";
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                cout << numero[i][j] << " ";
            }
            cout << "\n";
        }
        break;

    case 'b':
        cout << "Lista de Adyacencia";
        cout << "\n";
        for (int i = 0; i < n; i++) {
            cout << i + 1 << " -> ";
            for (int j : listaAdyacencia[i]) {
                cout << j + 1 << " ";
            }
            cout << "\n";
        }
        break;

    case 'c':
        cout << "Matriz de Incidencia";
        cout << " \n ";
        for (int j = 0; j < n; j++) {
            cout << " e" << j + 1 << " ";
        }
        cout << "\n";

        for (int i = 0; i < n; i++) {
            cout << "v" << i + 1 << " ";
            for (int j = 0; j < n; j++) {
                if (numero[i][j] != 0) {
                    cout << "1 ";
                } else {
                    cout << "0 ";
                }
            }
        }
    }
}

```

```

        }
        cout << "\n";
    }
    break;

    case 'f':
        regresarMenu = true;
        break;

    default:
        cout << "Opción no válida";
    }
}
break;

case 'd':
    cout << "Valencias del Grafo";
    cout << "\n";
    for (int i = 0; i < n; i++) {
        int valencia = listaAdyacencia[i].size();
        cout << i + 1 << " --> valencia= " << valencia <<
"\n";
    }
    break;

case 'e':
    generarGrafo(listaAdyacencia, n);
    break;

case 'f':
    salir = true;
    break;

default:
    cout << "Opción no válida";
}
}

return 0;
}

```

EJECUCION DEL PROGRAMA

```
MENU
Escribe una opcion

- Crear grafo(a)
- Representacion del grafo(b)
- Valencias del grafo(d)
- Generar grafo en formato PDF Y PNG(e)
- Salir(f)

Opcion=
```

OPCION A:

```
MENU
Escribe una opcion

- Crear grafo(a)
- Representacion del grafo(b)
- Valencias del grafo(d)
- Generar grafo en formato PDF Y PNG(e)
- Salir(f)

Opcion=a

Ingresa el número de vértices: 3
Ingresa un valor en [0][0]: 1
Ingresa un valor en [0][1]: 0
Ingresa un valor en [0][2]: 0
Ingresa un valor en [1][0]: 1
Ingresa un valor en [1][1]: 0
Ingresa un valor en [1][2]: 1
Ingresa un valor en [2][0]: 0
Ingresa un valor en [2][1]: 1
Ingresa un valor en [2][2]: 1
```

OPCION B:

```

      MENU
Escribe una opcion

- Crear grafo(a)
- Representacion del grafo(b)
- Valencias del grafo(d)
- Generar grafo en formato PDF Y PNG(e)
- Salir(f)

Opcion=b

Escoge una opcion
- Matriz de Adyacencia(a)
- Lista de Adyacencia(b)
- Matriz de Incidencia(c)
  Volver al menu (f)
```

OPCION A, B, C – MATRIZ DE ADYACENCIA, LISTA DE ADYACENCIA, MATRIZ DE INCIDENCIA:

```
Escoge una opcion
- Matriz de Adyacencia(a)
- Lista de Adyacencia(b)
- Matriz de Incidencia(c)
Volver al menu (f)
```

```
opcion=a
```

```
Matriz de Adyacencia
```

```
1 0 0
1 0 1
0 1 1
```

```
Escoge una opcion
- Matriz de Adyacencia(a)
- Lista de Adyacencia(b)
- Matriz de Incidencia(c)
Volver al menu (f)
```

```
opcion=b
```

```
Lista de Adyacencia
```

```
1 -> 1
2 -> 1 3
3 -> 2 3
```

```
Escoge una opcion
- Matriz de Adyacencia(a)
- Lista de Adyacencia(b)
- Matriz de Incidencia(c)
Volver al menu (f)
```

```
opcion=c
```

```
Matriz de Incidencia
```

```
  e1  e2  e3
v1 1 0 0
v2 1 0 1
v3 0 1 1
```

REGRESO AL MENU PRINCIPAL Y OPCION D – VALENCIA DEL GRAFO

Escoge una opcion

- Matriz de Adyacencia(a)
- Lista de Adyacencia(b)
- Matriz de Incidencia(c)
- Volver al menu (f)

opcion=f

MENU

Escribe una opcion

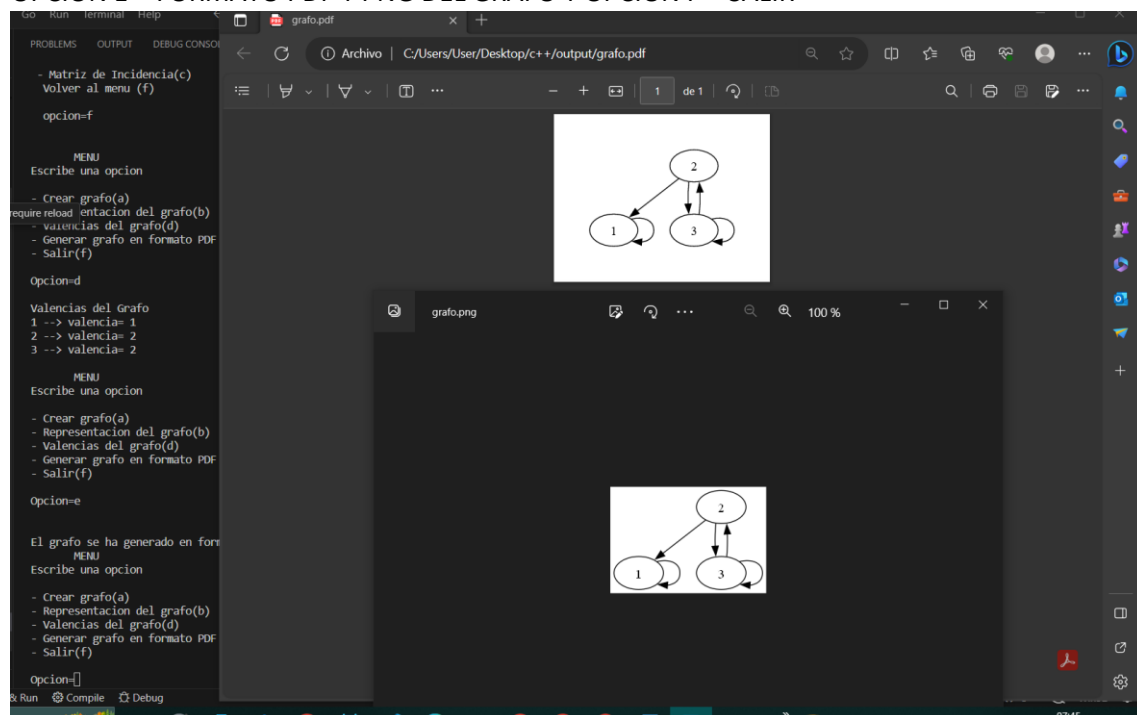
- Crear grafo(a)
- Representacion del grafo(b)
- Valencias del grafo(d)
- Generar grafo en formato PDF Y PNG(e)
- Salir(f)

Opcion=d

Valencias del Grafo

- 1 --> valencia= 1
- 2 --> valencia= 2
- 3 --> valencia= 2

OPCION E – FORMATO PDF Y PNG DEL GRAFO Y OPCION F – SALIR



MENU

Escribe una opcion

- Crear grafo(a)
- Representacion del grafo(b)
- Valencias del grafo(d)
- Generar grafo en formato PDF Y PNG(e)
- Salir(f)

Opcion=e

El grafo se ha generado en formato PDF y PNG



MENU

Escribe una opcion

- Crear grafo(a)
- Representacion del grafo(b)
- Valencias del grafo(d)
- Generar grafo en formato PDF Y PNG(e)
- Salir(f)

Opcion=f

PS C:\Users\User\Desktop\c++\output> █

Run  Compile  Debug