

```
#include <iostream>
#include <conio.h>
#include <stdio.h>
#include <iomanip>

#define MOT 3
#define DIS 5

using namespace std;

int main() {
    // Variables
    int iDis, iMot, auxMot, hayDatos, dni, nroDis, transp, estado,
    cantDistriSinPermi, contPermiXDistri, disMin, motMin;

    // Vectores
    int vecCodMot[MOT];

    // Matrices
    int matCantPermisos[MOT] [DIS];
    int matCantPermi0torgados[MOT] [DIS];

    // Inicializo las matrices en 0
    for (iMot = 0; iMot < MOT; iMot++) {
        for (iDis = 0; iDis < DIS; iDis++) {
            matCantPermisos[iMot][iDis] = 0;
            matCantPermi0torgados[iMot][iDis] = 0;
        }
    }

    // Ingreso de datos

    // MOTIVO
    cout << "----Ingreso de Motivos----" << endl;
    for (iMot = 0; iMot < MOT; iMot++) {
        cout << "Ingrese el codigo del motivo (entre 2000 y 10000): ";
        cin >> auxMot;

        // Validación
        while (auxMot < 2000 || auxMot > 10000) {
            cout << "ERROR! Ingrese un codigo valido (entre 2000 y 10000): ";
            cin >> auxMot;
        }
        vecCodMot[iMot] = auxMot;
    }

    cout << "-----" << endl;
    cout << "----Ingreso de Permisos----" << endl;

    cout << "Hay datos para cargar (1: SI / 2:N0): ";
    cin >> hayDatos;
```

```
// Validación
while (hayDatos != 1 && hayDatos != 2) {
    cout << "ERROR! Opcion invalida" << endl;
    cout << "Hay datos para cargar (1: SI / 2:NO): ";
    cin >> hayDatos;
}

// Ingreso al bucle
while (hayDatos == 1) {
    cout << "Ingrese DNI: ";
    cin >> dni;

    do {
        cout << "Ingrese el codigo del motivo (entre 2000 y 10000): ";
        cin >> auxMot;

        // Validación
        while (auxMot < 2000 || auxMot > 10000) {
            cout << "ERROR! Ingrese un codigo valido (entre 2000 y
10000): ";
            cin >> auxMot;
        }
        iMot = 0;
        while (iMot < MOT && vecCodMot[iMot] != auxMot) {
            iMot++;
        }
        if (iMot == MOT) {
            cout << "El codigo de distrito no existe." << endl;
        }
    } while (iMot == MOT){};

    cout << "Ingrese el nro de distrito (entre 1 - 22): ";
    cin >> nroDis;

    // Validación
    while (nroDis < 1 || nroDis > 22) {
        cout << "ERROR!. Distrito invalido." << endl;
        cout << "Ingrese el nro de distrito (entre 1 - 22): ";
        cin >> nroDis;
    }

    cout << "Ingrese transporte (10: Transporte Publico - 20: Vehiculo
- 30: Bicicleta): ";
    cin >> transp;

    // Validación
    while (transp != 10 && transp != 20 && transp != 30) {
        cout << "ERROR! Transporte invalido." << endl;
        cout << "Ingrese transporte (10: Transporte Publico - 20:
Vehiculo - 30: Bicicleta): ";
        cin >> transp;
    }
```

```
cout << "Ingrese el estado (1: Pendiente - 2: Otorgado): ";
cin >> estado;

// Validación
while (estado != 1 && estado != 2) {
    cout << "ERROR! Opcion invalida." << endl;
    cout << "Ingrese el estado (1: Pendiente - 2: Otorgado): ";
    cin >> estado;
}

matCantPermisos[iMot][nroDis - 1]++;

if (estado == 2) {
    matCantPermiOtorgados[iMot][nroDis - 1]++;
}

cout << "-----"
---" << endl;

// Pregunta para salir del bucle
cout << "Hay datos para cargar (1: SI / 2:N0): ";
cin >> hayDatos;

// Validación
while (hayDatos != 1 && hayDatos != 2) {
    cout << "ERROR! Opcion invalida" << endl;
    cout << "Hay datos para cargar (1: SI / 2:N0): ";
    cin >> hayDatos;
}
}

cout << "-----SALIDA DE DATOS-----"
<< endl;
cout << "-----"
<< endl;

// 1. Por cada DIS y MOT, la cantidad de permisos solicitados
for (iDis = 0; iDis < DIS; iDis++) {
    for (iMot = 0; iMot < MOT; iMot++) {
        cout << "Distrito: " << iDis + 1 << " Motivo: " <<
vecCodMot[iMot] << " Cantidad Permisos: " << matCantPermisos[iMot][iDis]
<< endl;
    }
}

// 2. Por cada MOT y DIS, la cantidad de permisos solicitados en
// estado OTORGADO
cout << "-----"
<< endl;
for (iMot = 0; iMot < MOT; iMot++) {
    for (iDis = 0; iDis < DIS; iDis++) {
        cout << "Motivo: " << vecCodMot[iMot] << " Distrito: " << iDis
+ 1 << " Cantidad de permisos otorgados: " << matCantPermiOtorgados[iMot]
[iDis] << endl;
```

```
        }
    }

    // 3. Por cada MOT, la cantidad de DIS de los cuales no se han
    solicitados permisos
    cout << "-----"
<< endl;
    cout << "Cantidad de Distritos de los cuales no se han solicitado
permisos por Motivos " << endl;
    for (iMot = 0; iMot < MOT; iMot++) {
        cantDistriSinPermi = 0;
        for (iDis = 0; iDis < DIS; iDis++) {
            if (matCantPermisos[iMot][iDis] == 0) {
                cantDistriSinPermi++;
            }
        }
        cout << "Motivo: " << vecCodMot[iMot] << " Cantidad Distritos: "
<< cantDistriSinPermi << endl;
    }

    // 4. La cantidad de permisos solicitados por cada DIS
    cout << "-----"
<< endl;
    cout << "Cantidad de permisos solicitados por cada distrito" << endl;
    for (iDis = 0; iDis < DIS; iDis++) {
        contPermiXDistri = 0;
        for (iMot = 0; iMot < MOT; iMot++) {
            contPermiXDistri += matCantPermisos[iMot][iDis];
        }
        cout << "Distritos: " << iDis + 1 << " Permisos solicitados: " <<
contPermiXDistri << endl;
    }

    // 5. Cual fue el MOT y en que DIS, y cual fue en donde se solicito
    MENOR cantidad de permisos
    cout << "-----"
<< endl;
    for (iMot = 0; iMot < MOT; iMot++) {
        for (iDis = 0; iDis < DIS; iDis++) {
            if (iMot == 0 && iDis == 0) {
                disMin = 0;
                motMin = 0;
            } else {
                if (matCantPermisos[iMot][iDis] < matCantPermisos[motMin]
[disMin]) {
                    disMin = iDis;
                    motMin = iMot;
                }
            }
        }
    }
    cout << "El motivo: " << vecCodMot[motMin] << " Distrito " << disMin +
1 << " solicito la menor cantidad de permisos: " <<
matCantPermisos[motMin][disMin] << endl;
```

```
getch();  
return 0;  
}
```