

```
using System.Collections.ObjectModel;
using System.ComponentModel.Design;
using System.Data;
using System.Numerics;
using Microsoft.VisualBasic;

class Program
{
    static void Main(string[] args)
    {
        OperacionesMatrices operacionesMatrices = new
OperacionesMatrices();
        char opcion = 'a';

        operacionesMatrices.CrearMatriz();
        operacionesMatrices.IngresarDatosMatriz();

        while (opcion != 'd')
        {
            Console.WriteLine("Menú opciones");
            Console.WriteLine(" a) Multiplicación de matriz por
escalar");
            Console.WriteLine(" b) Devolver todas las posiciones por
fila de un elemento menor");
            Console.WriteLine(" c) Vector de pares");
            Console.WriteLine(" d) Salir");
            Console.WriteLine("Ingrese opción: ");
            opcion = Console.ReadLine()[0];

            switch (opcion)
            {
                case 'a':
                    Console.WriteLine("Ingrese un escalar");
                    int escalar = Convert.ToInt32(Console.ReadLine());
operacionesMatrices.MultiplicaciónMatrizEscalar(escalar);
                    operacionesMatrices.ImprimirMatriz(escalar);
                    break;
                case 'b':
                    Console.WriteLine("Ingrese valor de busqueda");
```

```
                int nBusqueda =
Convert.ToInt32(Console.ReadLine());
                operacionesMatrices.Devolverposiciones(nBusqueda);
                operacionesMatrices.ImprimirMatriz2();
                break;

            case 'c':
                int Npares = Convert.ToInt32(Console.ReadLine());
                operacionesMatrices.VectorEnteros(Npares);
                operacionesMatrices.ImprimirMatriz3(Npares);
                break;
        }
    }
}

using System.Data;
using System.Runtime.CompilerServices;
using Microsoft.VisualBasic;

class OperacionesMatrices
{
    // Ejercicio a Multiplicación de matriz por escalar
    public int[,] matriz = new int[0,0];

    public OperacionesMatrices()
    {

    }

    public void CrearMatriz()
    {
        int cantidadFilas = 0;
        int cantidadCols = 0;

        Console.WriteLine("Ingrese la cantidad de filas de la matriz");
        cantidadFilas = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Ingrese la cantidad de columnas de la
matriz");
        cantidadCols = Convert.ToInt32(Console.ReadLine());

        matriz = new int[cantidadFilas,cantidadCols];
    }
}
```

```
public void IngresarDatosMatriz()
{
    for (int fila = 0; fila < matriz.GetLength(0); fila++)
    {
        for (int columna = 0; columna < matriz.GetLength(1);
columna++)
        {
            Console.WriteLine($"Ingrese valor para la posicion
[{{fila}}][{{columna}}]");
            matriz[fila,columna] =
Convert.ToInt32(Console.ReadLine());
        }
    }

    public int[,] MultiplicaciónMatrizEscalar(int escalar)
    {
        int[,] matrizMultiplicada = new
int[matriz.GetLength(0),matriz.GetLength(1)];

        for (int fila = 0; fila < matriz.GetLength(0); fila++)
        {
            for (int columna = 0; columna < matriz.GetLength(1);
columna++)
            {
                matrizMultiplicada[fila,columna] = matriz[fila,columna]
* escalar;
            }
        }

        return matrizMultiplicada;
    }

    public void ImprimirMatriz(int escalar)
    {
        for (int fila = 0; fila < matriz.GetLength(0); fila++)
        {
            for (int columna = 0; columna < matriz.GetLength(1); columna++)
            {
                Console.WriteLine("" + matriz[fila,columna]*escalar);
            }
        }
    }
}
```

```
        Console.WriteLine(" ");
    }
}

// Ejercicio b Devolver todas las posiciones por fila de un
// elemento menor

public int [,] Devolverposiciones (int nBusqueda, int [,] matriz)
{
    int[,] resultados = new int[matriz.GetLength(0), 1];

    for (int fila = 0; fila < matriz.GetLength(0); fila++)
    {
        for (int columna = 0; columna < matriz.GetLength(1);
columna++)
        {
            Console.WriteLine($"Cantidad de veces que se encontro
un valor menor al ingresado por fila: [{fila}][{columna}]");
            matriz[fila, columna] =
Convert.ToInt32(Console.ReadLine());

            if (matriz[fila, columna] < nBusqueda)
            {
                resultados[fila, 0]++;
            }
        }
    }

    return resultados;
}

public void ImprimirMatriz2()
{
    int[,] resultados = new int[matriz.GetLength(0), 1];
    for (int fila = 0; fila < matriz.GetLength(0); fila++)
    {
        for (int columna = 0; columna < matriz.GetLength(1); columna++)
        {
            Console.WriteLine($"Fila {fila + 1}: {resultados[fila,
0]}");
        }
    }
    Console.WriteLine(" ");
}
```

```
    }  
}  
  
// Ejercicio c Vector de pares  
  
public int [] VectorEnteros (int[,] nPares)  
{  
    int[] Npares = new int[0];  
    int contador = 0;  
    int fila = 0;  
    int columna = 0;  
  
    for (int i = 0; i < fila; i++ )  
    {  
        for (int j = 0; j < columna; i++)  
        {  
            if (matriz[i, j] % 2 == 0) //agregado  
            {  
                Npares[contador] = matriz[i, j];  
                contador++;  
            }  
        }  
    }  
    return Npares;  
}  
  
public void ImprimirMatriz3(int Npares)  
{  
    for (int i = 0; i < matriz.GetLength(0); i++)  
    {  
        for (int j = 0; j < matriz.GetLength(1); j++)  
        {  
            Console.WriteLine("Los numeros pares de la matriz son: " +  
Npares);  
        }  
    }  
}
```