

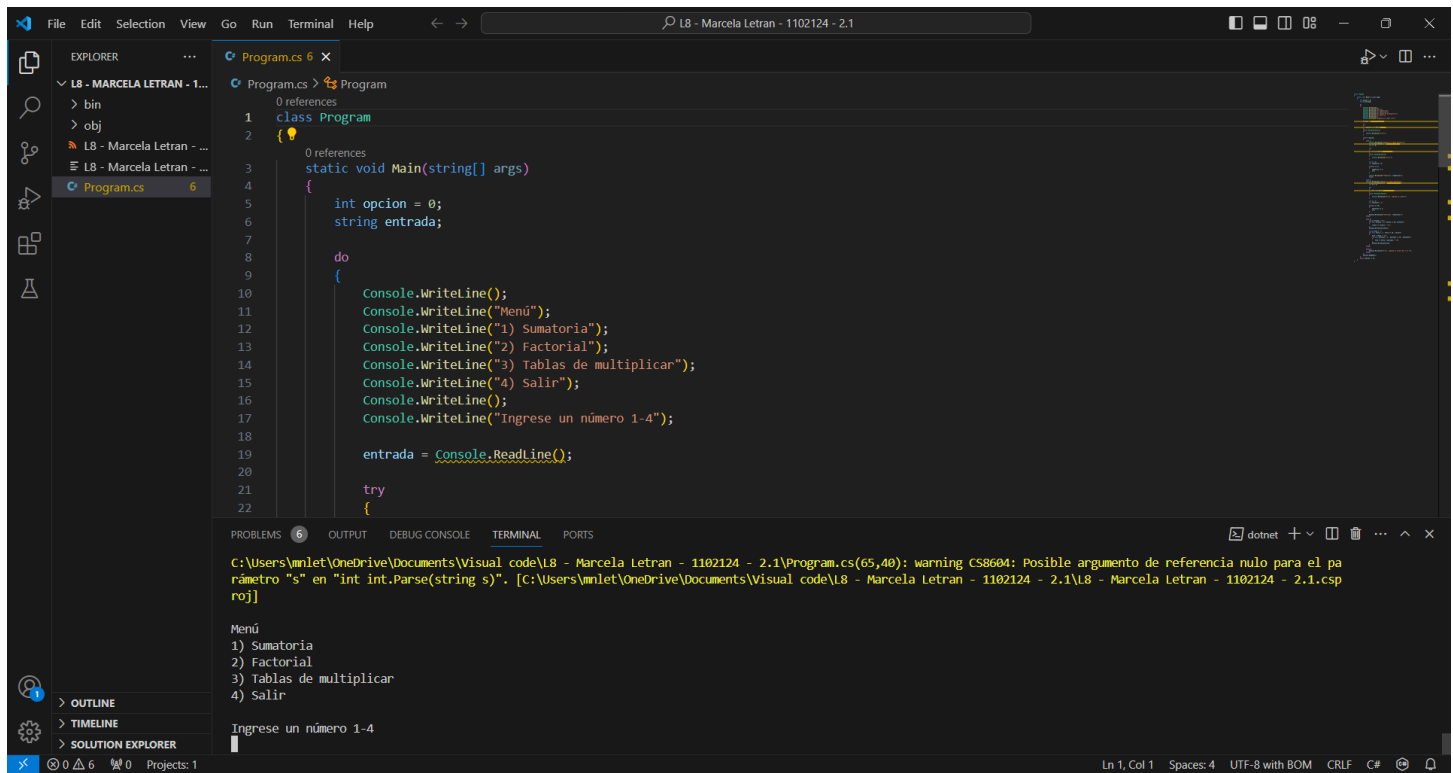
Marcela Nicole Letran Lee

Carné: 1102124

Carrera: Licenciatura en ingeniería en informática y sistemas

Curso: Pensamiento computacional (práctica)

Sección: 15



```
class Program
{
    static void Main(string[] args)
    {
        int opcion = 0;
        string entrada;

        do
        {
            Console.WriteLine();
            Console.WriteLine("Menú");
            Console.WriteLine("1) Sumatoria");
            Console.WriteLine("2) Factorial");
            Console.WriteLine("3) Tablas de multiplicar");
            Console.WriteLine("4) Salir");
            Console.WriteLine();
            Console.WriteLine("Ingrese un número 1-4");

            entrada = Console.ReadLine();
```

Marcela Nicole Letran Lee

Carné: 1102124

Carrera: Licenciatura en ingeniería en informática y sistemas

Curso: Pensamiento computacional (práctica)

Sección: 15

```
try
{
    opcion = int.Parse(entrada);
}
catch (FormatException)
{
    Console.WriteLine("Error");
}

switch (opcion)
{
    case 1:
        Console.WriteLine("Ingrese un número positivo");
        string entradaNumero = Console.ReadLine();
        int n = 0;

        try
        {
            n = int.Parse(entradaNumero);
        }
        catch (FormatException)
        {
            Console.WriteLine("Error");
        }

        int i = 1;
        int sumatoria = 0;

        while(i <= n)
        {
            sumatoria += 1;
            i++;
        }

        Console.WriteLine($"Sumatoria: {sumatoria}");
        break;

    case 2:
        Console.WriteLine("Ingrese un número positivo");
        string entradaNumeroF = Console.ReadLine();
        int nF = 0;

        try
        {
```

Marcela Nicole Letran Lee

Carné: 1102124

Carrera: Licenciatura en ingeniería en informática y sistemas

Curso: Pensamiento computacional (práctica)

Sección: 15

```
        nF = int.Parse(entradaNumeroF);
    }
    catch (FormatException)
    {
        Console.WriteLine("Error, ingrese un número");
    }

    int j = 1;
    int factorial = 1;

    while(j <= nF)
    {
        factorial *= j;
        j++;
    }

    Console.WriteLine($"Factorial: {factorial}");
    break;

case 3:
    string titulo = "\t";
    for (int iTitulo = 1; iTitulo <= 10; iTitulo++)
    {
        titulo += iTitulo + "\t";
    }
    Console.WriteLine(titulo);

    string fila = "";
    for (int iFila = 1 ; iFila <= 10 ; iFila++)
    {
        fila = iFila + "\t";
        for (int multiplo = 1 ; multiplo <= 10 ; multiplo++)
        {
            fila += iFila * multiplo + "\t";
        }
        Console.WriteLine(fila);
    }
    break;

default:
    Console.WriteLine("Error, ingrese un número del 1 al 4");
    break;
}
Console.ReadKey();
```

Marcela Nicole Letran Lee
 Carné: 1102124
 Carrera: Licenciatura en ingeniería en informática y sistemas
 Curso: Pensamiento computacional (práctica)
 Sección: 15

```

    }
    while (opcion != 4);
  }
}

```

Diagramas UML:

Ejercicio 1 : Entidades URL

Estudiante	Cursos	Profesor	BECA
+ 2nombres: string + 2apellidos: string + carné: int + correoURL: string + usuarioEST: string + facultad: string + carrera: string + portalURL (usuarioEST, contraseña1): PORTestu + cursos (PORTestu): asigCURSOS	+ nombre: string + profesor: string + sección: int + salón: char + jornada: string + material (PORTAlestu): matCURSO + semestreSIG (cursos, PORTAlestu): cursosSIG	+ 2nombres: string + 2apellidos: string + carnéADMIN: int + correoURLadmin: string + usuarioADMIN: string + cursoIMP: string + portalURL(usuarioADMIN, contraseña2): PORTadmin + nomina (2nombres, 2apellidos, NIT, cuentaBAC, DPI): pagoMENS	+ 2personas: string + 2apellidos: string + carné: int + exmADaprob: char + requisitosPROM: string + requisitosPERFíl: string + requisitosPART: string + beneficio (requisitos, carné, carrera): benMONE + reconocimiento (BECA, requisitos): reconTÍTULO

Ejercicio 2: Cuenta Bancaria

Cuenta Bancaria
+ NUMcuenta: int + dineroCRED: int + dineroEFEC: int + DPI: int + NIT/TRUsistema) : int + debe: char + haber: char + acreditar (haber, NUMcuenta, dineroEFEC): dineACRED + debitar (dineroEFEC, partida, debe): dineDEBI + mostrarSALDO (NIT/RTU, NUMcuenta, DPI, : SALDO