

Ejercicios programación en lenguaje I.Net

Johanna Milena Garcia Bautista

2184588

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN



Ejercicio # 1

1. Leer un número por teclado y definir si es par o impar.

ANALISIS DEL PROBLEMA

➤ **Entradas:**

Inicio

"Digite un numero entero"

Leer numero

➤ **Proceso:**

➤ Int numero=0

Si

numero es % 2 y su resultado es = 0

➤ **Salidas:**

{

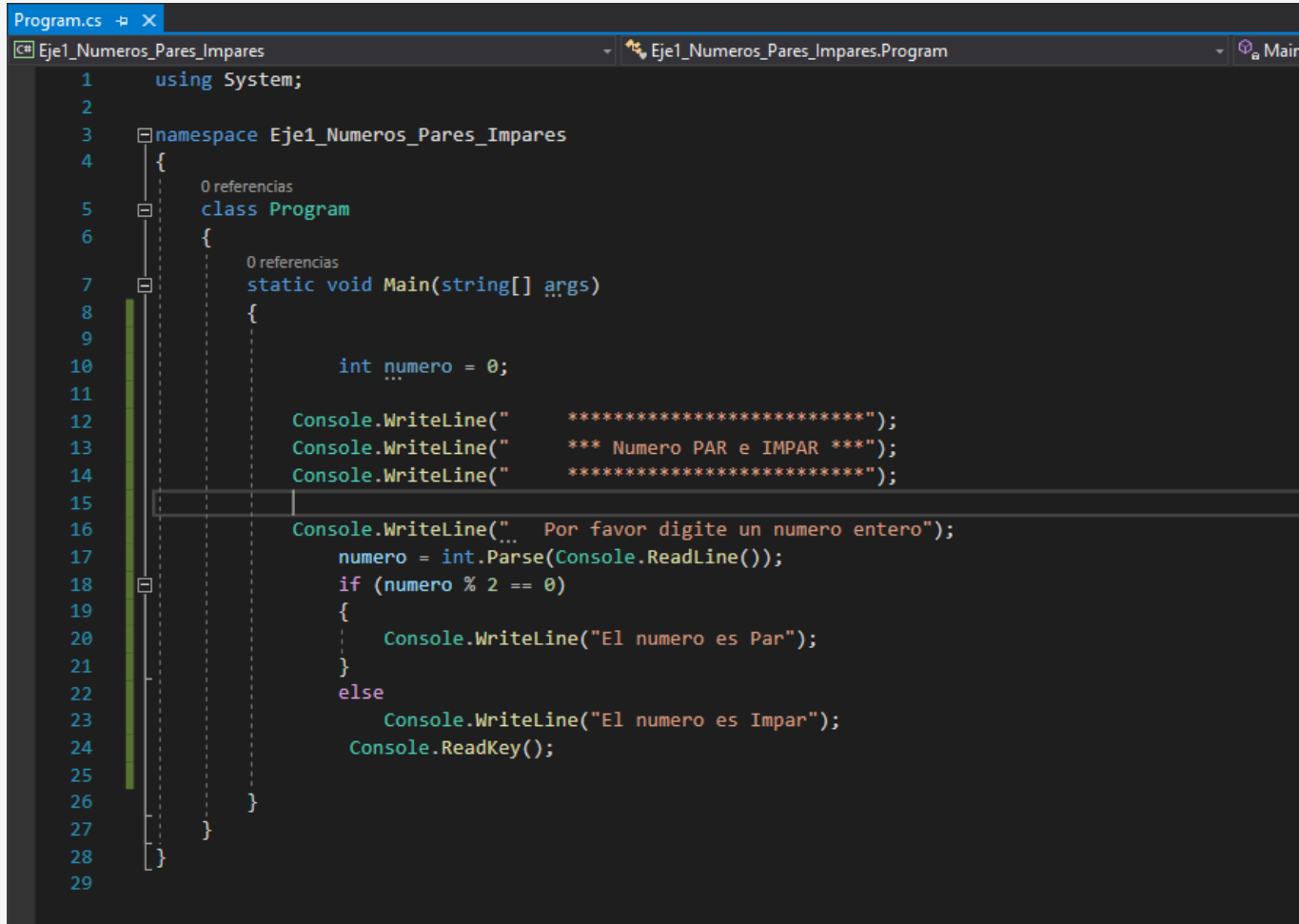
"El numero es par"

}

Sino "el numero es impar"

Fin

CODIFICACIÓN (C #)



The image shows a screenshot of a C# code editor with a dark theme. The code is for a program named 'Program.cs' within a namespace 'Eje1_Numeros_Pares_Impares'. The program defines a class 'Program' with a static method 'Main' that takes an array of strings 'args'. Inside 'Main', it declares an integer 'numero' and initializes it to 0. It then prints three lines of text: a separator of asterisks, the title '*** Numero PAR e IMPAR ***', and another separator of asterisks. It prompts the user to enter an integer and reads the input using 'Console.ReadLine()'. The input is parsed to an integer using 'int.Parse()'. An if-else statement checks if the number is even ('numero % 2 == 0'). If true, it prints 'El numero es Par'; otherwise, it prints 'El numero es Impar'. Finally, it calls 'Console.ReadKey()' to wait for a key press before exiting.

```
1  using System;
2
3  namespace Eje1_Numeros_Pares_Impares
4  {
5      0 referencias
6      class Program
7      {
8          0 referencias
9          static void Main(string[] args)
10         {
11
12             int numero = 0;
13
14             Console.WriteLine("*****");
15             Console.WriteLine("*** Numero PAR e IMPAR ***");
16             Console.WriteLine("*****");
17
18             Console.WriteLine(".. Por favor digite un numero entero");
19             numero = int.Parse(Console.ReadLine());
20             if (numero % 2 == 0)
21             {
22                 Console.WriteLine("El numero es Par");
23             }
24             else
25                 Console.WriteLine("El numero es Impar");
26             Console.ReadKey();
27         }
28     }
29 }
```

Ejercicio # 2

2. Leer un número por teclado y generar con ese número la tabla de multiplicar del número 1 al 10.

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

"Digite el numero de la tabla a multiplicar"

Leer numero

➤ Proceso:

Int contador=1

Int numero =0

Int resultado =0

Mientras contador <= 10

{

Resultado = contador * numero

➤ Salidas:

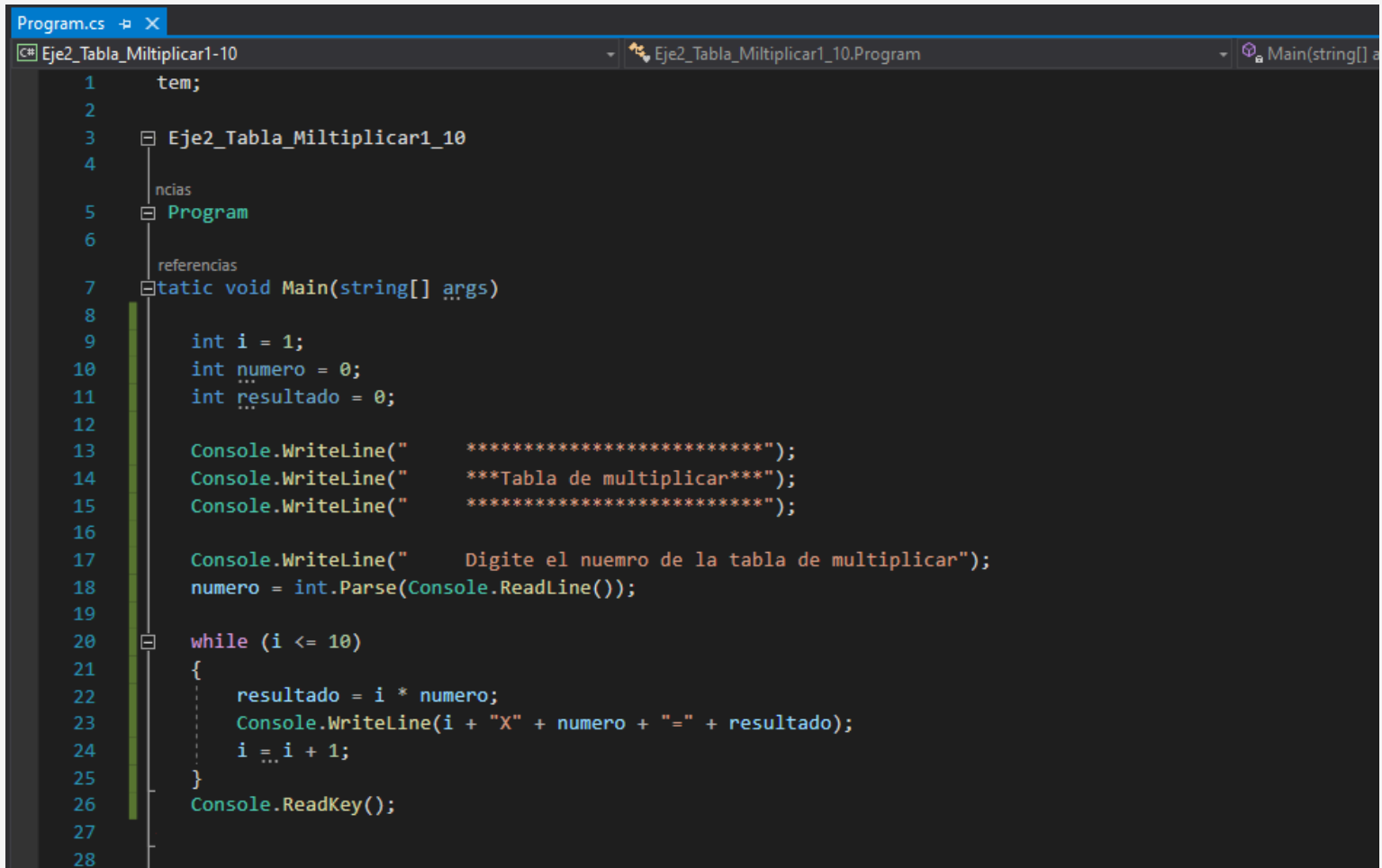
Escribir (contador + "X" + numero + "=" + resultado)

Contador = contador +1

}

Fin

CODIFICACIÓN (C #)



The image shows a screenshot of a C# code editor with a dark theme. The editor window is titled "Program.cs" and shows the code for a program named "Eje2_Tabla_Multiplicar1_10". The code is as follows:

```
1  tem;
2
3  Eje2_Tabla_Multiplicar1_10
4
5  ncias
6
7  Program
8
9  referencias
10 static void Main(string[] args)
11
12     int i = 1;
13     int numero = 0;
14     int resultado = 0;
15
16     Console.WriteLine("*****");
17     Console.WriteLine("***Tabla de multiplicar***");
18     Console.WriteLine("*****");
19
20     Console.WriteLine("  Digite el nuemro de la tabla de multiplicar");
21     numero = int.Parse(Console.ReadLine());
22
23     while (i <= 10)
24     {
25         resultado = i * numero;
26         Console.WriteLine(i + "X" + numero + "=" + resultado);
27         i = i + 1;
28     }
29     Console.ReadKey();
```

Ejercicio # 3

2. Generar las tablas de multiplicar desde el numero 2 al 9, cada una desde el numero 1 al 10

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

➤ Proceso:

Int numTabla

Int numero

Int resultado

```
para (numTabla =2; numTabla <= 9; numTabla ++)
```

```
{
```

```
  Decir ("la tabla del" + numTabla)
```

```
  para (numero = 1; numero <= 10; numero++)
```

```
{
```

```
  Resultado = numero * numTabla
```

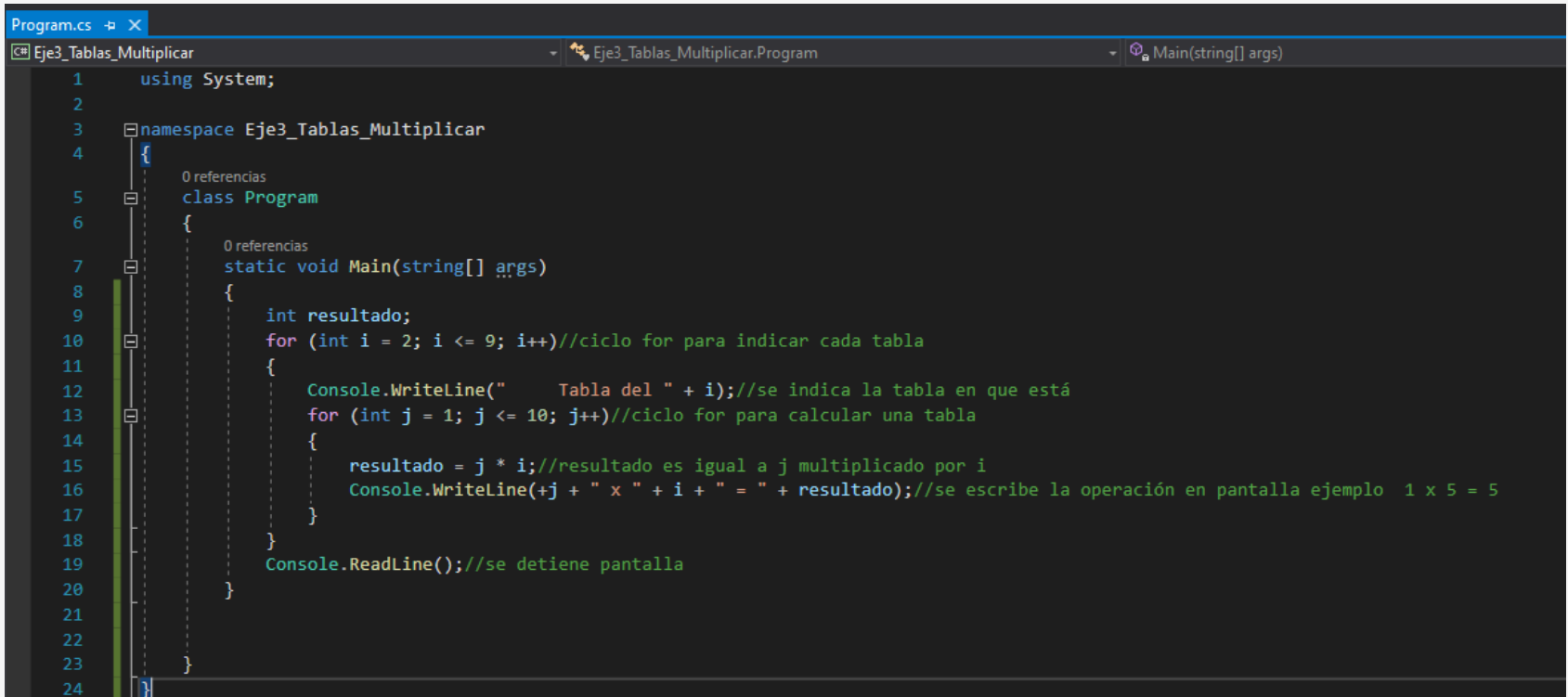
➤ Salidas:

```
Escribir (numero + "X" + numTabla + "=" + resultado)
```

```
}
```

Fin

CODIFICACIÓN (C #)



The screenshot shows a C# code editor with a dark theme. The file name is 'Program.cs'. The code is for a namespace 'Eje3_Tablas_Multiplicar' and a class 'Program'. It contains a static method 'Main' that uses nested loops to calculate and display multiplication tables. The code is as follows:

```
1  using System;
2
3  namespace Eje3_Tablas_Multiplicar
4  {
5      0 referencias
6      class Program
7      {
8          0 referencias
9          static void Main(string[] args)
10         {
11             int resultado;
12             for (int i = 2; i <= 9; i++)//ciclo for para indicar cada tabla
13             {
14                 Console.WriteLine("    Tabla del " + i);//se indica la tabla en que está
15                 for (int j = 1; j <= 10; j++)//ciclo for para calcular una tabla
16                 {
17                     resultado = j * i;//resultado es igual a j multiplicado por i
18                     Console.WriteLine(+j + " x " + i + " = " + resultado);//se escribe la operación en pantalla ejemplo  1 x 5 = 5
19                 }
20             }
21             Console.ReadLine();//se detiene pantalla
22         }
23     }
24 }
```

Ejercicio # 4

2. Leer un número por teclado y definir si es primo o no.

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

"Por favor digite un numero entero"

Lea numero

➤ Proceso:

divisor<-0

divisores<-1

Si (Numero mod divisores=0) entonces

Divisor<- divisor+1

Sino

Si Divisor =2

➤ Salidas:

Numero, "Es numero primo"

Sino

Numero "No es numero primo"

Fin

CODIFICACIÓN (C #)

```
Program.cs X
Eje4_Numero_Primo Eje4_Numero_Primo.Program Main(string[])
1 using System;
2
3 namespace Eje4_Numero_Primo
4 {
5     0 referencias
6     class Program
7     {
8         0 referencias
9         static void Main(string[] args)
10        {
11            int numero;
12            int divisor = 0;
13            int divisores = 1;
14            Console.WriteLine("*****");
15            Console.WriteLine("*** NUMERO PRIMO ***");
16            Console.WriteLine("*****");
17
18            Console.WriteLine("    digite un numero");
19            numero = Convert.ToInt32(Console.ReadLine());
20
21            for (divisores = 2; divisores < numero; divisores++)
22            {
23                if ((numero % divisores) == 0)
24                {
25                    divisor = divisor + 1;
26                    Console.WriteLine("El numero No es Primo");
27                }
28                else
29                {
30                    divisor = 2;
31                    Console.WriteLine("El numero es Primo");
32                }
33            }
34        }
35    }
```

Ejercicio # 5

5. Dado el vector edad = [12,50,23,11,18,35,41,85,16,45], ordenarlos en forma ascendente.

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

Vector = (números a ordenar)

Int aux

➤ Proceso:

//Método Burbuja

//for inicial es el que selecciona el primer digito

For (int i = 0; i < vector; i++)

{

//ahora se compara el primer digito con los demás para así ordenarlos

For (int j= 1; j < vector; j++)

{

//Se crea auxiliar, la cual va a reservar el valor de la posición a cambiar

//Ej: 10 15 5

//10>15 no cambia

//10 > 5 se intercambian los datos

//Quedando 5 en la primera posición.

Si vector i > vector j

{

//Se hace el intercambio comparando los números en las posiciones

Aux = vector[i]

vector[i] = vector[j]

vector[j] = aux

}

}

}

➤ Salidas:

//Mostrar los datos ordenados

For (int i = 0; i < vector; i++)

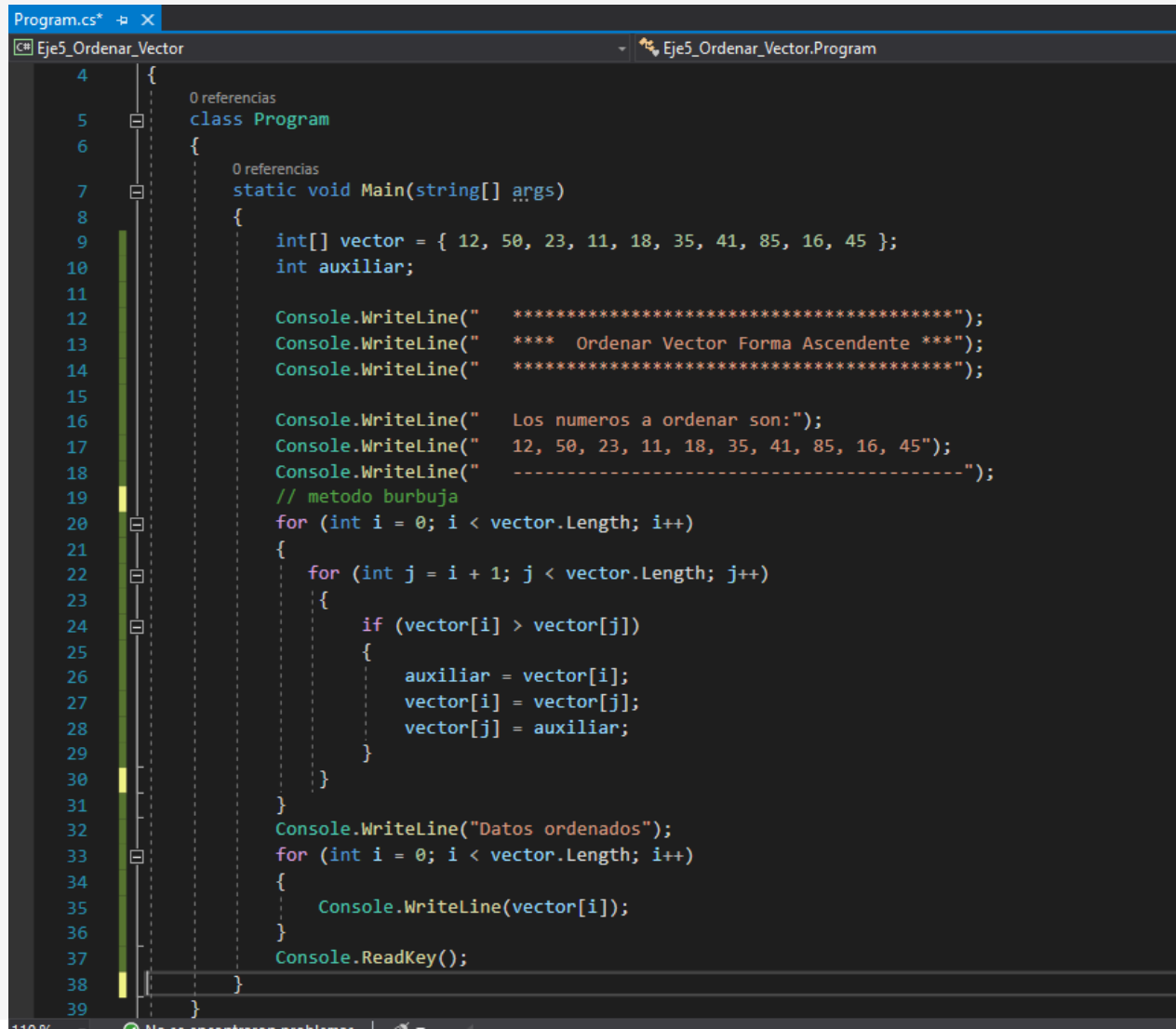
{

Escriba vector[i]

}

Fin

CODIFICACIÓN (C #)



```
Program.cs* X
Eje5_Ordenar_Vector Eje5_Ordenar_Vector.Program
4 {
5     0 referencias
6     class Program
7     {
8         0 referencias
9         static void Main(string[] args)
10        {
11            int[] vector = { 12, 50, 23, 11, 18, 35, 41, 85, 16, 45 };
12            int auxiliar;
13
14            Console.WriteLine(" *****");
15            Console.WriteLine(" **** Ordenar Vector Forma Ascendente ****");
16            Console.WriteLine(" *****");
17
18            Console.WriteLine(" Los numeros a ordenar son:");
19            Console.WriteLine(" 12, 50, 23, 11, 18, 35, 41, 85, 16, 45");
20            Console.WriteLine(" -----");
21            // metodo burbuja
22            for (int i = 0; i < vector.Length; i++)
23            {
24                for (int j = i + 1; j < vector.Length; j++)
25                {
26                    if (vector[i] > vector[j])
27                    {
28                        auxiliar = vector[i];
29                        vector[i] = vector[j];
30                        vector[j] = auxiliar;
31                    }
32                }
33            }
34            Console.WriteLine("Datos ordenados");
35            for (int i = 0; i < vector.Length; i++)
36            {
37                Console.WriteLine(vector[i]);
38            }
39            Console.ReadKey();
40        }
41    }
42 }
```

110% No se encontraron problemas

Ejercicio # 6

6. Dados los vectores `edad = [12,50,23,11,18,35,41,85,16,45]`
`nombre = ["juan", "maria", "tereza", "pedro", "javier", "ana",
"diana", "jorge", "dayana", "lady"]` Leer un nombre por
teclado y definir si existe, y en caso de existir mostrar su
edad de lo contrario un mensaje que el nombre no existe

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

Vector edades = (números)

Vector nombres = (nombres)

➤ Proceso:

//Método imprimir vector

Se hace el arreglo y recorre el vector buscando el elemento que se relaciona, en este caso el nombre con la edad.

```
foreach (string elemento in vector)
{
    Console.WriteLine(" [ " + elemento + " ] ");
}
```

// método bool existe nombre

Se recorre el vector comprobando que el nombre que se digite exista dentro de el.

```
foreach (string elemento in vector)
{
    if (elemento == persona)
        existe = true;
}
return existe;
```

// método Posicion persona

Se recorre el vector dándole el mismo valor en posición del nombre para la edad que le corresponde

```
foreach (string elemento in vector)
{
    if (elemento == persona)
    {
        posicion = indice;
    }
    indice++;
}
return posicion;
```

➤ Salidas:

//Mostrar la edad correspondiente al nombre

Si Existe Nombre

```
{
    (" El Usuario " + nomBus + " si existe y tiene " +
    edad[PosocionPersona(nombre, nomBus)] + " Anos de edad ");
}
sino (" El Usuario " + nomBus + " No existe ")
Fin
```

CODIFICACIÓN (C #)

```
Program.cs
Eje6_Nombre_Edad - Eje6_Nombre_Edad.Program - Main(string[] args)
1 using System;
2
3 namespace Eje6_Nombre_Edad
4 {
5     0 referencias
6     class Program
7     {
8         0 referencias
9         static void Main(string[] args)
10         {
11             string[] edad = { "12", "50", "23", "11", "18", "35", "41", "85", "16", "45" };
12             string[] nombre = { "juan", "maria", "tereza", "pedro", "javier", "ana", "diana", "jorge", "dayana", "lady" };
13             string nomBus;
14
15             Console.Clear();
16             ImprimirVector(edad, " Edades ");
17             ImprimirVector(nombre, " Nombres ");
18             Console.WriteLine("\n Digite el nombre a buscar ");
19             nomBus = Console.ReadLine();
20
21             if (ExisteNombre(nombre, nomBus))
22             {
23                 Console.WriteLine(" El Usuario " + nomBus + " si existe y tiene " + edad[PosicionPersona(nombre, nomBus)] + " Anos de edad ");
24             }
25             else
26             {
27                 Console.WriteLine(" El Usuario " + nomBus + " No existe ");
28             }
29
30             2 referencias
31             static void ImprimirVector(Array vector, string titulo)
32             {
33                 Console.WriteLine("\n Arreglo " + titulo);
34
35                 foreach (string elemento in vector)
36                 {
37                     Console.Write(" [ " + elemento + " ] ");
38                 }
39             }
40         }
41     }
42 }
```

```
33 static bool ExisteNombre(Array vector, string persona)
34 {
35     bool existe = false;
36     foreach (string elemento in vector)
37     {
38         if (elemento == persona)
39             existe = true;
40     }
41     return existe;
42 }
43
44 1 referencia
45 static int PosicionPersona(Array vector, string persona)
46 {
47     int indice = 0;
48     int posicion = 0;
49     foreach (string elemento in vector)
50     {
51         if (elemento == persona)
52         {
53             posicion = indice;
54         }
55         indice++;
56     }
57     return posicion;
58 }
59
60 }
```

Ejercicio # 7

7. Partiendo de los vectores anteriores buscar el menor y el mayor y presentar sus respectivos nombres con su edad.

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

Vector edades = (números)

Vector nombres = (nombres)

➤ Proceso:

//Se recorre el arreglo con un for, el cual indicara la posición y buscara el mayor y el menor.

Se hace el arreglo y recorre

For{

Si (edad[indice] > edadMayor)

{ entonces ubicara el numero mayor para indicar la posición y así dar el nombre correspondiente}

Si (edad[indice] < edadMenor)

{ entonces ubicara el numero menor para indicar la posición y así dar el nombre correspondiente}

➤ Salidas:

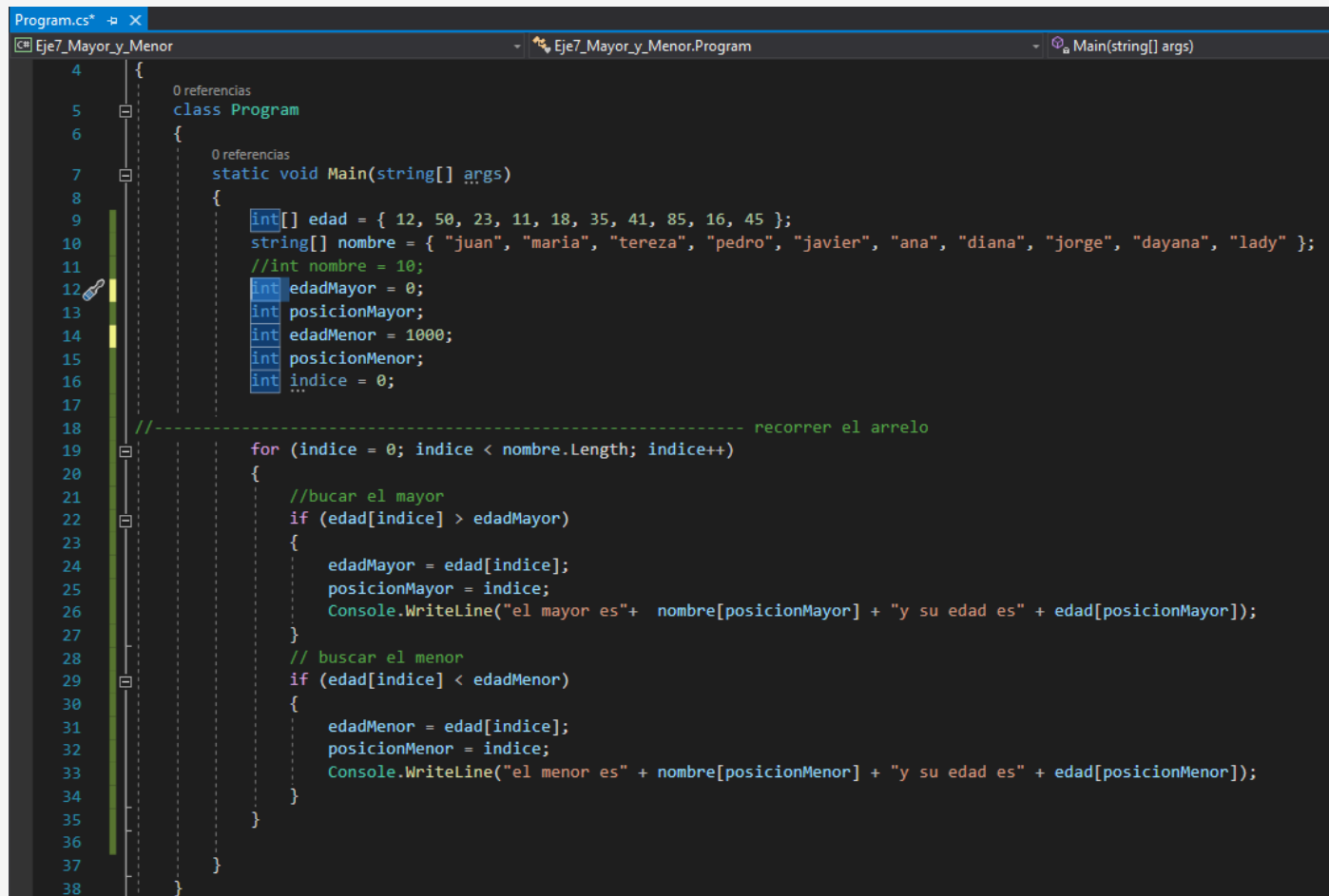
//Mostrar la edad correspondiente al nombre

("el mayor es" + nombre[posicionMayor] + "y su edad es" + edad[posicionMayor])

Console.WriteLine("el menor es" + nombre[posicionMenor] + "y su edad es" + edad[posicionMenor]);

Fin

CODIFICACIÓN (C #)



```
Program.cs* x
Eje7_Mayor_y_Menor Eje7_Mayor_y_Menor.Program Main(string[] args)
4 {
5     0 referencias
6     class Program
7     {
8         0 referencias
9         static void Main(string[] args)
10        {
11            int[] edad = { 12, 50, 23, 11, 18, 35, 41, 85, 16, 45 };
12            string[] nombre = { "juan", "maria", "tereza", "pedro", "javier", "ana", "diana", "jorge", "dayana", "lady" };
13            //int nombre = 10;
14            int edadMayor = 0;
15            int posicionMayor;
16            int edadMenor = 1000;
17            int posicionMenor;
18            int indice = 0;
19            //----- recorrer el arreglo
20            for (indice = 0; indice < nombre.Length; indice++)
21            {
22                //bucar el mayor
23                if (edad[indice] > edadMayor)
24                {
25                    edadMayor = edad[indice];
26                    posicionMayor = indice;
27                    Console.WriteLine("el mayor es" + nombre[posicionMayor] + "y su edad es" + edad[posicionMayor]);
28                }
29                // buscar el menor
30                if (edad[indice] < edadMenor)
31                {
32                    edadMenor = edad[indice];
33                    posicionMenor = indice;
34                    Console.WriteLine("el menor es" + nombre[posicionMenor] + "y su edad es" + edad[posicionMenor]);
35                }
36            }
37        }
38    }
```

Ejercicio # 8

8. Dado una palabra definir si es palíndromo o no.

ANALISIS DEL PROBLEMA

➤ Entradas:

Inicio

Digite la palabra que desea verificar si es palíndromo.

Lea la palabra

➤ Proceso:

//Se crea un método bool para false o true

Se llama la cadena

Si las letras de la palabras son <2

return es true;

Si tiene mas de dos letras se empieza a comparar la primera y la ultima, si coinciden se llama a la función de nuevo con la cadena cortada, hasta que se acaben las coincidencias.

➤ Salidas:

Si coinciden todas las letras {

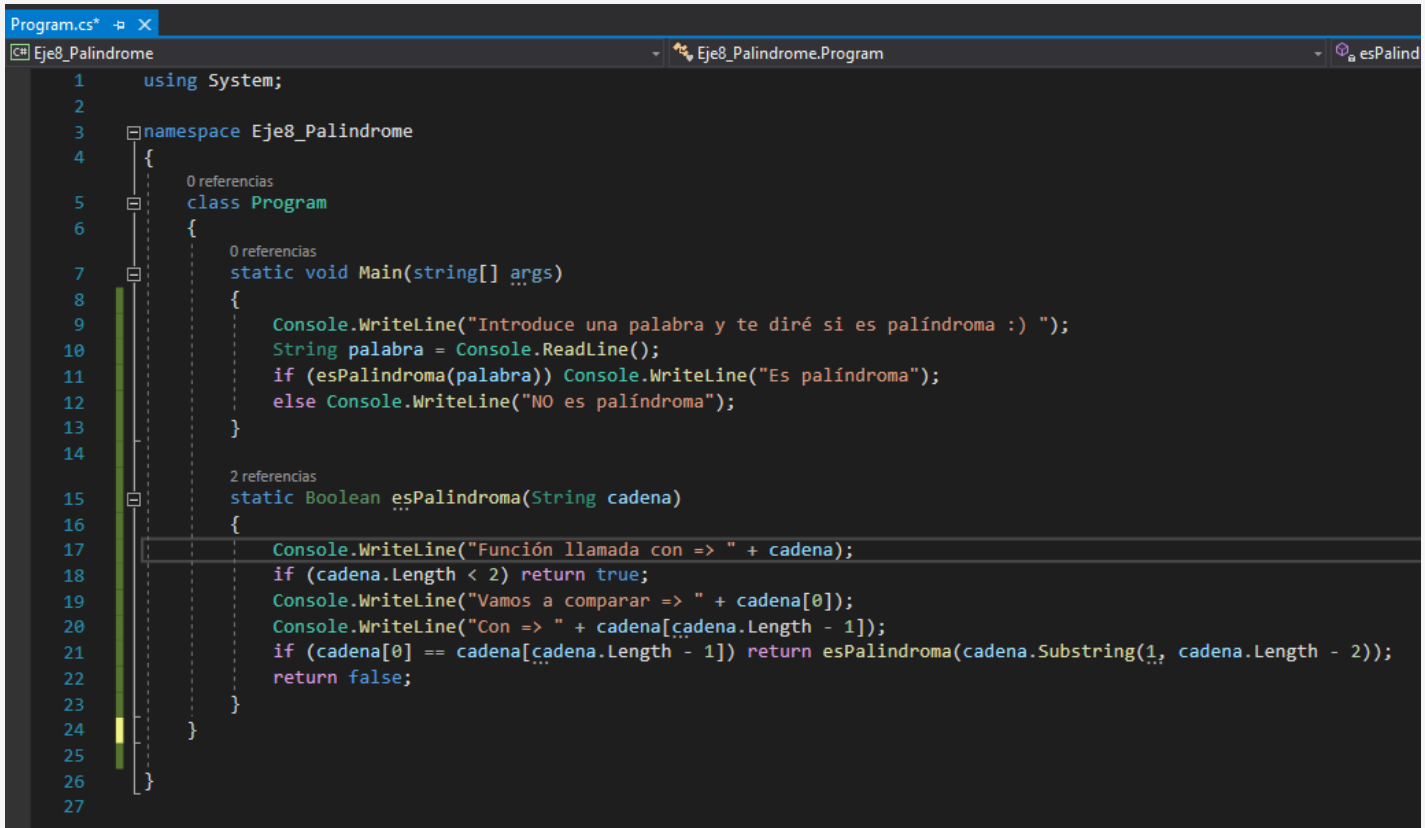
“La palabra es palíndroma”

} else

“La palabra no es palíndroma”

Fin

CODIFICACIÓN (C #)



```
Program.cs* X
Eje8_Palindrome Eje8_Palindrome.Program esPalind

1  using System;
2
3  namespace Eje8_Palindrome
4  {
5      0 referencias
6      class Program
7      {
8          0 referencias
9          static void Main(string[] args)
10         {
11             Console.WriteLine("Introduce una palabra y te diré si es palíndroma :) ");
12             String palabra = Console.ReadLine();
13             if (esPalindroma(palabra)) Console.WriteLine("Es palíndroma");
14             else Console.WriteLine("NO es palíndroma");
15         }
16
17         2 referencias
18         static Boolean esPalindroma(String cadena)
19         {
20             Console.WriteLine("Función llamada con => " + cadena);
21             if (cadena.Length < 2) return true;
22             Console.WriteLine("Vamos a comparar => " + cadena[0]);
23             Console.WriteLine("Con => " + cadena[cadena.Length - 1]);
24             if (cadena[0] == cadena[cadena.Length - 1]) return esPalindroma(cadena.Substring(1, cadena.Length - 2));
25             return false;
26         }
27     }
```