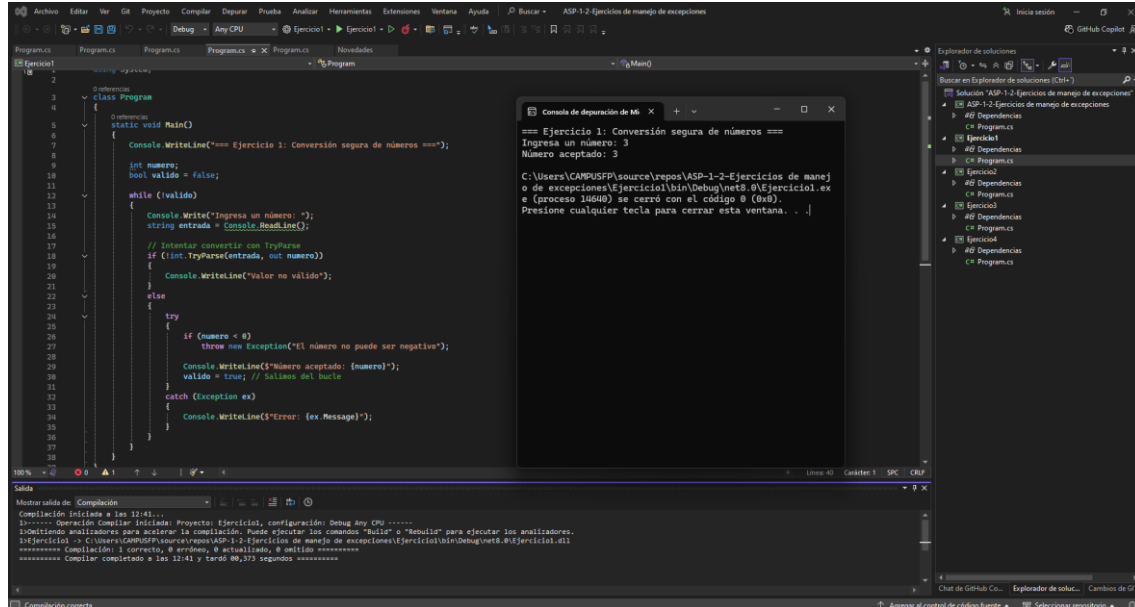


ASP-A1.2: Ejercicios de manejo de excepciones

Ejercicios sobre manejo de excepciones en C#

EJERCICIO 1



```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("=== Ejercicio 1: Conversión segura de números
===");

        int numero;
        bool valido = false;

        while (!valido)
        {
            Console.Write("Ingresa un número: ");
            string entrada = Console.ReadLine();

            // Intentar convertir con TryParse
            if (!int.TryParse(entrada, out numero))
            {
                Console.WriteLine("Valor no válido");
            }
            else
            {
                try
                {
                    if (numero < 0)
                        throw new Exception("El número no puede ser
negativo");

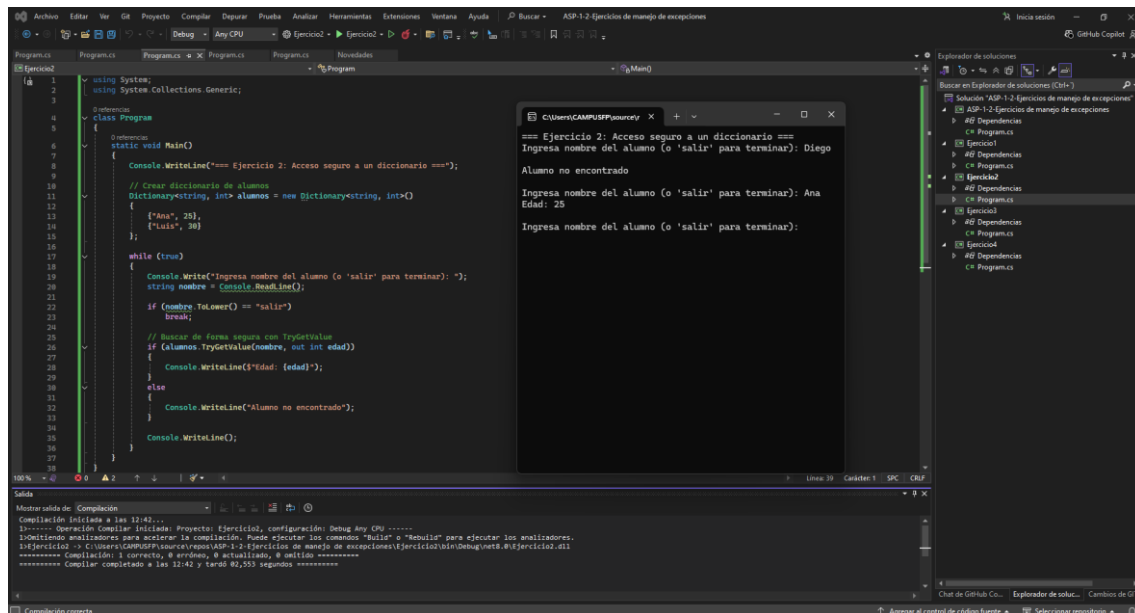
                    Console.WriteLine($"Número aceptado: {numero}");
                    valido = true; // Salimos del bucle
                }
                catch (Exception ex)
                {
                    Console.WriteLine($"Error: {ex.Message}");
                }
            }
        }
    }
}
```

```

        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}
}

```

EJERCICIO 2



```

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.WriteLine("=== Ejercicio 2: Acceso seguro a un diccionario ===");

        // Crear diccionario de alumnos
        Dictionary<string, int> alumnos = new Dictionary<string, int>()
        {
            {"Ana", 25},
            {"Luis", 30}
        };

        while (true)
        {
            Console.Write("Ingresa nombre del alumno (o 'salir' para
terminar): ");
            string nombre = Console.ReadLine();

            if (nombre.ToLower() == "salir")
                break;

            // Buscar de forma segura con TryGetValue
            if (alumnos.TryGetValue(nombre, out int edad))
            {
                Console.WriteLine($"Edad: {edad}");
            }
            else

```

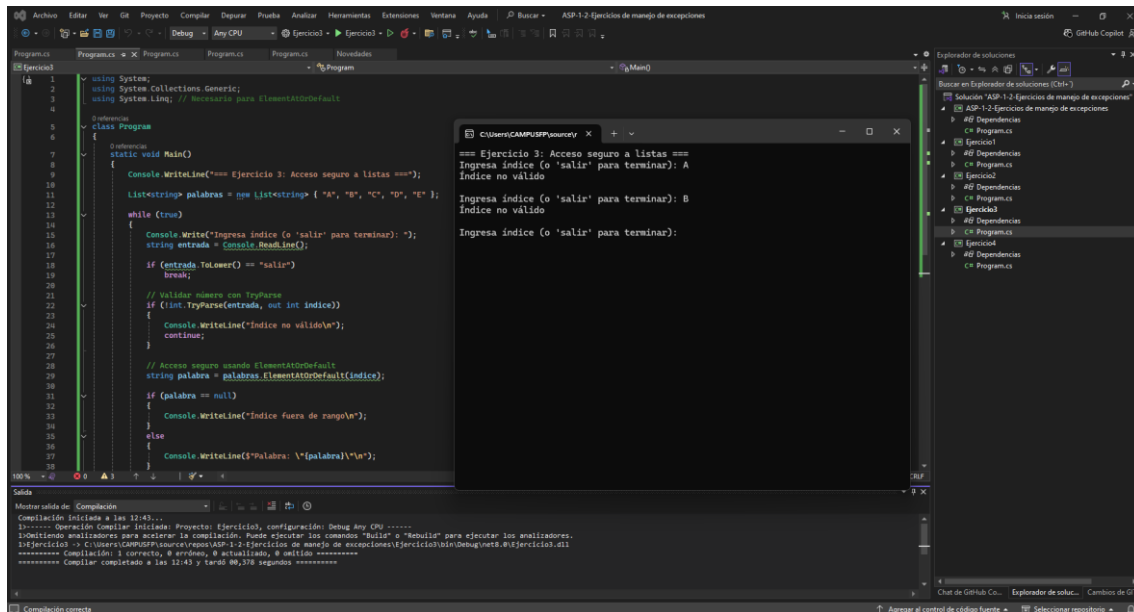
```

    {
        Console.WriteLine("Alumno no encontrado");
    }

    Console.WriteLine();
}
}
}

```

EJERCICIO 3



```

using System;
using System.Collections.Generic;
using System.Linq; // Necesario para ElementAtOrDefault

class Program
{
    static void Main()
    {
        Console.WriteLine("=== Ejercicio 3: Acceso seguro a listas ===");

        List<string> palabras = new List<string> { "A", "B", "C", "D", "E" };

        while (true)
        {
            Console.Write("Ingresa índice (o 'salir' para terminar): ");
            string entrada = Console.ReadLine();

            if (entrada.ToLower() == "salir")
                break;

            // Validar número con TryParse
            if (!int.TryParse(entrada, out int indice))
            {
                Console.WriteLine("Índice no válido\n");
                continue;
            }

            // Acceso seguro usando ElementAtOrDefault
            string palabra = palabras.ElementAtOrDefault(indice);

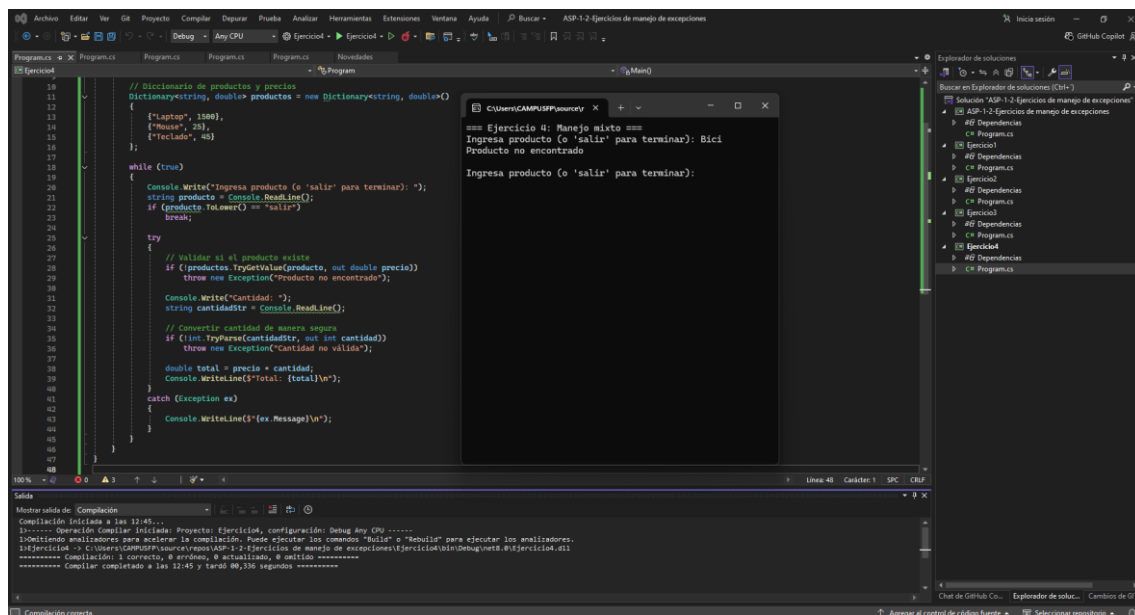
```

```

        if (palabra == null)
        {
            Console.WriteLine("Índice fuera de rango\n");
        }
        else
        {
            Console.WriteLine($"Palabra: \"{palabra}\"\\n");
        }
    }
}
}

```

EJERCICIO 4



```

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.WriteLine("=== Ejercicio 4: Manejo mixto ===");

        // Diccionario de productos y precios
        Dictionary<string, double> productos = new Dictionary<string,
double>()
        {
            {"Laptop", 1500},
            {"Mouse", 25},
            {"Teclado", 45}
        };

        while (true)
        {
            Console.Write("Ingresa producto (o 'salir' para terminar): ");
            string producto = Console.ReadLine();
            if (producto.ToLower() == "salir")
                break;

            try

```

```

{
    // Validar si el producto existe
    if (!productos.TryGetValue(producto, out double precio))
        throw new Exception("Producto no encontrado");

    Console.Write("Cantidad: ");
    string cantidadStr = Console.ReadLine();

    // Convertir cantidad de manera segura
    if (!int.TryParse(cantidadStr, out int cantidad))
        throw new Exception("Cantidad no válida");

    double total = precio * cantidad;
    Console.WriteLine($"Total: {total}\n");
}
catch (Exception ex)
{
    Console.WriteLine($"{ex.Message}\n");
}
}
}
}

```