

Actividad 1 – Menú Tipos de datos en Consola

Andrés Felipe Gonzalez Noreña

Servicio Nacional de Aprendizaje – SENA

Programa: Análisis y Desarrollo de Software

Ficha: 3144585

Instructor: Luis Fernando

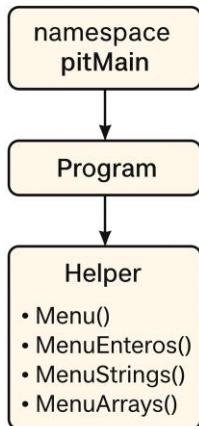
2025

Introducción

La presente actividad consiste en el desarrollo de una aplicación en C# basada en menús y submenús, con el propósito de comprender el funcionamiento de distintos tipos de datos como enteros, cadenas de texto (strings) y arreglos (arrays). Mediante esta estructura interactiva se busca fortalecer la comprensión de operaciones básicas, manipulación de datos y lógica fundamental del lenguaje.

Estructura del Proyecto

El proyecto se organiza dentro del namespace pitMain, donde se integran las clases esenciales. La clase Program contiene el punto de entrada del programa, instanciando la clase Helper y ejecutando el menú principal. La clase Helper centraliza la lógica mediante los métodos Menu(), MenuEnteros(), MenuStrings() y MenuArrays(). Esta estructura favorece un diseño modular y de fácil mantenimiento.



Menú Principal

El menú principal emplea un ciclo while que se ejecuta hasta que el usuario elige la opción Salir. A través de un switch, se dirige al submenú correspondiente según la selección del usuario, actuando como controlador principal del flujo del programa.

```
C# helper.cs 9+ X
pitMain > C# helper.cs > Helper > MenuEnteros
1  using System;
2
3  namespace pitMain
4  {
5      public class Helper
6      {
7          public void Menu()
8          {
9              int opcion = 0;
10             while (opcion != 4)
11             {
12                 Console.WriteLine("Menu principal");
13                 Console.WriteLine("1. Enteros");
14                 Console.WriteLine("2. Strings");
15                 Console.WriteLine("3. Arrays");
16                 Console.WriteLine("4. Salir");
17                 opcion = int.Parse(Console.ReadLine());
18
19                 switch (opcion)
20                 {
21                     case 1:
22                         MenuEnteros();
23                         break;
24                     case 2:
25                         MenuStrings();
26                         break;
27                     case 3:
28                         MenuArrays();
29                         break;
30                 }
31             }
32         }
33
34         public void MenuEnteros()
35         {
36             int opcion = 0;
37             while (opcion != 6)
38             {
39                 Console.WriteLine("Menu Enteros");
40                 Console.WriteLine("1. Sumar");
41                 Console.WriteLine("2. Restar");
42                 Console.WriteLine("3. Dividir");
43                 Console.WriteLine("4. multiplicacion y modulo");
44                 Console.WriteLine("5. Otra");
45                 Console.WriteLine("6. Salir");
46                 opcion = int.Parse(Console.ReadLine());
47
48                 switch (opcion)
49                 {
50                     case 1:
51                         Console.Write("primero numero que se sumara: "); a = int.Parse(Console.ReadLine());
52                         Console.Write("segundo numero que se sumara: "); b = int.Parse(Console.ReadLine());
53                         Console.WriteLine("El resultado de la suma es: ", a + b);
54                         break;
55                     case 2:
56                         Console.Write("primero numero que se restara: "); a = int.Parse(Console.ReadLine());
57                         Console.Write("segundo numero que se restara: "); b = int.Parse(Console.ReadLine());
58                         Console.WriteLine("El resultado de la resta es: ", a - b);
59                 }
60             }
61         }
62     }
63 }
```

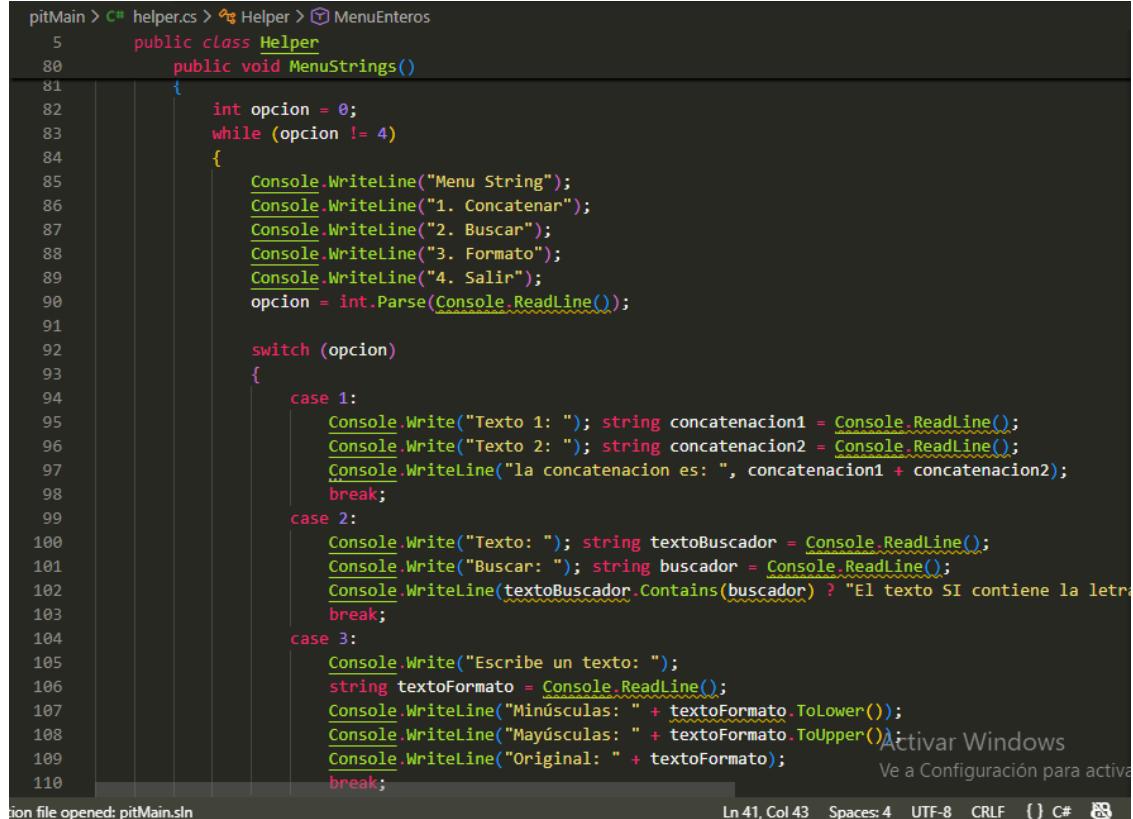
Submenú Enteros

Este submenú permite realizar operaciones básicas con números enteros, como suma, resta, división, multiplicación, módulo y raíz cuadrada mediante Math.Sqrt(). El programa solicita valores por consola y devuelve resultados, ejemplificando el manejo de variables int en C#.

```
C# helper.cs 9+ X
pitMain > C# helper.cs > Helper > MenuEnteros
5  public class Helper
6  {
7      public void Menu()
8      {
9          }
10
11         public void MenuEnteros()
12         {
13             int opcion = 0;
14             while (opcion != 6)
15             {
16                 Console.WriteLine("Menu Enteros");
17                 Console.WriteLine("1. Sumar");
18                 Console.WriteLine("2. Restar");
19                 Console.WriteLine("3. Dividir");
20                 Console.WriteLine("4. multiplicacion y modulo");
21                 Console.WriteLine("5. Otra");
22                 Console.WriteLine("6. Salir");
23                 opcion = int.Parse(Console.ReadLine());
24
25                 switch (opcion)
26                 {
27                     case 1:
28                         Console.Write("primero numero que se sumara: "); a = int.Parse(Console.ReadLine());
29                         Console.Write("segundo numero que se sumara: "); b = int.Parse(Console.ReadLine());
30                         Console.WriteLine("El resultado de la suma es: ", a + b);
31                         break;
32                     case 2:
33                         Console.Write("primero numero que se restara: "); a = int.Parse(Console.ReadLine());
34                         Console.Write("segundo numero que se restara: "); b = int.Parse(Console.ReadLine());
35                         Console.WriteLine("El resultado de la resta es: ", a - b);
36                 }
37             }
38         }
39     }
40 }
```

Submenú Strings

Este submenú incluye funciones como concatenación de textos, búsqueda mediante Contains() y cambios de formato utilizando ToUpper() y ToLower(). Representa una introducción práctica al manejo del tipo string en el lenguaje.



```
pitMain > C# helper.cs > Helper > MenuEnteros
 5     public class Helper
 80    public void MenuStrings()
 81    {
 82        int opcion = 0;
 83        while (opcion != 4)
 84        {
 85            Console.WriteLine("Menu String");
 86            Console.WriteLine("1. Concatenar");
 87            Console.WriteLine("2. Buscar");
 88            Console.WriteLine("3. Formato");
 89            Console.WriteLine("4. Salir");
 90            opcion = int.Parse(Console.ReadLine());
 91
 92            switch (opcion)
 93            {
 94                case 1:
 95                    Console.Write("Texto 1: "); string concatenacion1 = Console.ReadLine();
 96                    Console.Write("Texto 2: "); string concatenacion2 = Console.ReadLine();
 97                    Console.WriteLine("la concatenacion es: ", concatenacion1 + concatenacion2);
 98                    break;
 99                case 2:
100                    Console.Write("Texto: "); string textoBuscador = Console.ReadLine();
101                    Console.Write("Buscar: "); string buscador = Console.ReadLine();
102                    Console.WriteLine(textoBuscador.Contains(buscador) ? "El texto SI contiene la letra" : "No contiene la letra");
103                    break;
104                case 3:
105                    Console.Write("Escribe un texto: ");
106                    string textoFormato = Console.ReadLine();
107                    Console.WriteLine("Minúsculas: " + textoFormato.ToLower());
108                    Console.WriteLine("Mayúsculas: " + textoFormato.ToUpper());
109                    Console.WriteLine("Original: " + textoFormato);
110                    break;
111            }
112        }
113    }
114
115    ion file opened: pitMain.sln
116
117    Line 41, Col 43  Spaces: 4  UTF-8  CRLF  { } C#  🇺🇸
```

Submenú Arrays

En este submenú se implementa la creación de arreglos, ordenamiento con Array.Sort(), búsqueda de elementos con Array.IndexOf() y ampliación de arreglos mediante copia y reasignación. Estas operaciones permiten trabajar con estructuras de datos más complejas.

```
C# helper.cs 9+ X
pitMain > C# helper.cs > Helper > MenuArrays
157 }
158 |
159     1 reference
160     public void MenuArrays()
161     {
162         int opcion = 0; int[] arr = new int[0];
163         while (opcion != 5)
164         {
165             Console.WriteLine("Menu Arrays");
166             Console.WriteLine("1. Crear");
167             Console.WriteLine("2. Ordenar");
168             Console.WriteLine("3. Buscar");
169             Console.WriteLine("4. Agregar");
170             Console.WriteLine("5. Salir");
171             opcion = int.Parse(Console.ReadLine());
172
173             switch (opcion)
174             {
175                 case 1:
176                     Console.Write("Tamaño: ");
177                     int n = int.Parse(Console.ReadLine());
178                     arr = new int[n];
179                     for (int i = 0; i < n; i++) arr[i] = int.Parse(Console.ReadLine());
180                     break;
181                 case 2:
182                     Array.Sort(arr);
183                     Console.WriteLine(string.Join(", ", arr));
184                     break;
185                 case 3:
186                     Console.Write("Número: ");
187                     int x = int.Parse(Console.ReadLine());
188                     int p = Array.IndexOf(arr, x);
189                     Console.WriteLine(p);
190             }
191         }
192     }
193 }
```

Program.cs

El archivo Program.cs contiene el punto de entrada principal. En él se crea una instancia de Helper y se llama al método Menu(), lo que da inicio a toda la aplicación. Este archivo permite comprender el flujo inicial del programa.

```

pitMain > C# program.cs > program > Main
1
2 //el namespace el cual contiene toda mi app
3 namespace pitMain
4 {
5     //esta es la clase la cual da acceso a las funciones de la aplicacion
6     public class program()
7     {
8         //agreggo un objeto de tipo AppWeb, con el cual voy a instanciar o llamar al otro metodo construct
9
10        static void Main()
11        {
12            Helper H = new Helper();
13            H.Menu();
14        }
15
16    }
17
18 //puedo crear otro archivo en el cual puedo poner otra clase con sus atributos y metodos, tiene que es
19

```

Tablas

Tabla Clase Helper:

Método	Descripción
Menu()	Controla el menú principal y permite navegar entre las opciones generales.
MenuEnteros()	Gestiona las operaciones matemáticas básicas con números enteros.
MenuStrings()	Permite realizar operaciones con cadenas como concatenación, búsqueda y formato.
MenuArrays()	Maneja arreglo de enteros: creación, ordenamiento, búsqueda y ampliación.

Tablas submenús y funciones:

Submenú	Opciones que ofrece
Enteros	Suma, resta, división, multiplicación, módulo, raíz cuadrada.
Strings	Concatenar textos, buscar palabra, convertir mayúsculas/minúsculas.
Arrays	Crear arreglo, ordenar, buscar elemento, agregar nuevos datos.

Tabla de problemas durante la realización

Problema identificado	Descripción
Falta de experiencia en el lenguaje	Este es un lenguaje relativamente nuevo para mi y durante la realización de este código había ocasiones que no entendí muchas cosas especialmente en el método Array
Errores con el método Array	Este al tener un código un poco mas complejo tuve demasiados errores y tuve que recurrir bastante a la IA ya que me ocasionaba muchos dolores de cabeza
Errores con los "int.Parse()	Yo había hecho de forma normal esta parte, pero a la hora de hacer las operaciones básicas recurrió a ayudas como chatGPT las cuales me recomendaron usar int.Parse junto a los readlines para que así sea más completo el Submenú de enteros en vez de la operación ahí quemada

Referencias

Chatgpt: Esta IA me ayudo bastante durante la realización de este proyecto, fue útil para ayudarme a darle forma con las normas APA a este documento, también me dio ideas

Para apoyar mi documento y fue de gran utilidad para darme una idea de como debí hacer este documento informativo.

Conclusiones

Conclusión 1

El desarrollo de la aplicación permitió comprender de manera práctica el funcionamiento de los tipos de datos fundamentales en C#, así como su manipulación mediante menús interactivos en consola. El uso de enteros, cadenas y arreglos dentro de un entorno estructurado fomentó el fortalecimiento de habilidades en lógica de programación, control de flujo y diseño modular, aspectos esenciales en la formación de un programador.

Conclusión 2

La implementación de los submenús y sus respectivas operaciones evidenció la importancia de organizar el código en métodos y clases, lo que mejora la claridad, reutilización y mantenibilidad del programa. A su vez, la elaboración de diagramas UML y tablas descriptivas contribuyó a comprender de forma visual la arquitectura del proyecto, reforzando la capacidad de documentar software siguiendo estándares técnicos y académicos.