



**UNIVERSIDAD AMERICANA**

**Facultad de Ingeniería y arquitectura**

**Asignatura: Metodología y programación estructurada**

**Grupo 2**

**Integrantes:**

**-Andrés Humberto Porras Romero**

**-Cristopher Amaru Rodríguez Arauz**

**Docente:**

**MSc. Silvia Gigdalia Ticay López**

**Fecha: 21 de octubre de 2024**

## Ejercicio 3

```

Program3.cs x Funciones.cs
Ejercicio3 Ejercicio3.Program Main(string[] args)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Ejercicio3
8 {
9     0 referencias
10     internal class Program
11     {
12         0 referencias
13         static void Main(string[] args)
14         {
15             Console.WriteLine("Bienvenidos. Este programa se centra en las temperaturas y sus distintas conversiones(Celsius, Kelvin y Fahrenheit). A continuaci
16             Console.ReadLine();
17             Console.Clear();
18
19             //Listas para almacenar las temperaturas en Celsius, Fahrenheit y Kelvin
20             List<double> temperaturasCelsius = new List<double>();
21             List<double> temperaturasFahrenheit = new List<double>();
22             List<double> temperaturasKelvin = new List<double>();
23
24             bool continuar = true;
25
26             //Bucle principal para ingresar las temperaturas y convertirlas
27             while (continuar)
28             {
29                 Console.WriteLine("Ingrese una temperatura en grados Celsius:");
30                 double tempCelsius = double.Parse(Console.ReadLine());
31                 temperaturasCelsius.Add(tempCelsius);
32
33                 //Convierte la temperatura a Fahrenheit y Kelvin
34                 double tempFahrenheit = Funciones.ConvertirAFahrenheit(tempCelsius);
35                 double tempKelvin = Funciones.ConvertirAKelvin(tempCelsius);
36
37                 //Almacena las temperaturas convertidas en las listas anteriormente mencionadas
38                 temperaturasFahrenheit.Add(tempFahrenheit);
39                 temperaturasKelvin.Add(tempKelvin);
40
41                 //Muestra las temperaturas convertidas
42                 Console.WriteLine($"Temperatura en Fahrenheit: {tempFahrenheit}");
43                 Console.WriteLine($"Temperatura en Kelvin: {tempKelvin}\n");
44
45                 //Pregunta al usuario si desea ingresar otra temperatura
46                 Console.WriteLine("¿Desea ingresar otra temperatura? (s/n):");
47                 //Pregunta al usuario si desea ingresar otra temperatura
48                 Console.WriteLine("¿Desea ingresar otra temperatura? (s/n)");
49                 continuar = Console.ReadLine().ToLower() == "s";
50                 Console.Clear();
51             }
52
53             //Pregunta al usuario si desea eliminar alguna temperatura anteriormente convertida
54             Console.WriteLine("\n¿Desea eliminar alguna temperatura convertida? (s/n)");
55             if (Console.ReadLine().ToLower() == "s")
56             {
57                 Funciones.EliminarTemperatura(temperaturasFahrenheit, temperaturasKelvin);
58             }
59
60             //Muestra las listas de las temperaturas ya convertidas
61             Console.WriteLine("\nTemperaturas en Fahrenheit:");
62             Funciones.MostrarLista(temperaturasFahrenheit);
63
64             Console.WriteLine("\nTemperaturas en Kelvin:");
65             Funciones.MostrarLista(temperaturasKelvin);
66         }
67     }
68 }

```

```

Program3.cs  Funciones.cs  Ejercicio3
Ejercicio3  Funciones.cs  ConvertirAFahrenheit(double celsius)

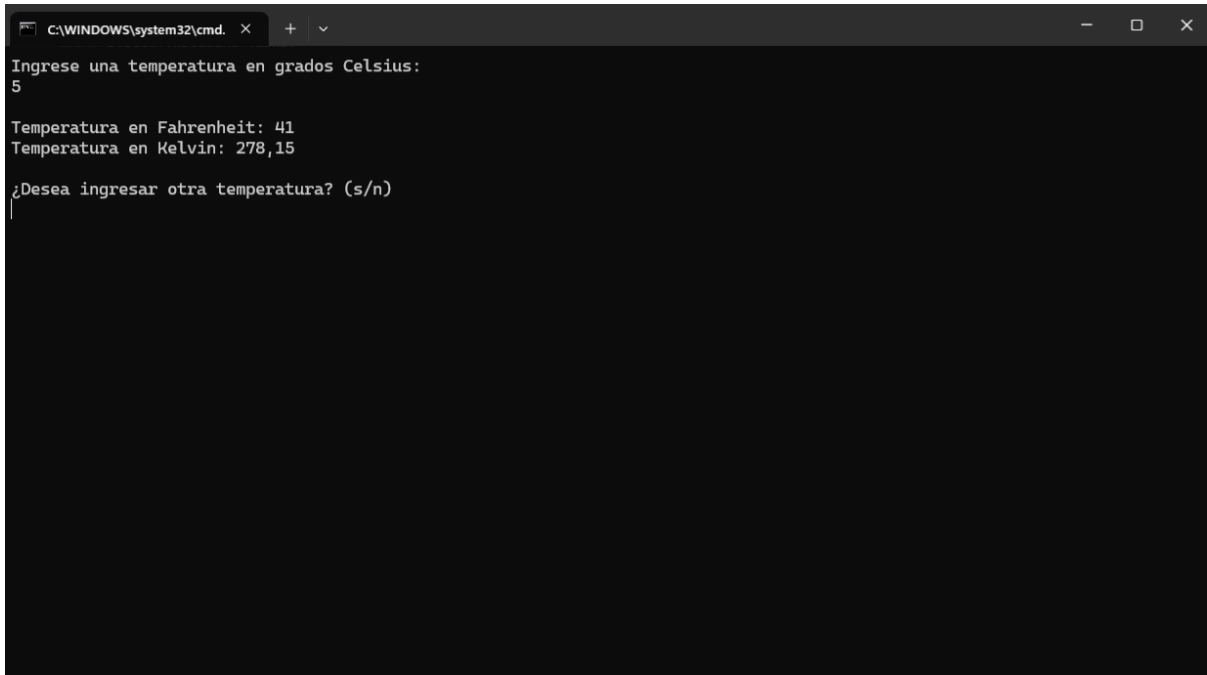
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Ejercicio3
8  {
9      5 referencias
10     internal class Funciones
11     {
12         //Convierte de Celsius a Fahrenheit
13         1 referencia
14         public static double ConvertirAFahrenheit(double celsius)
15         {
16             return (celsius * 9 / 5) + 32;
17         }
18         //Convierte de Celsius a Kelvin
19         1 referencia
20         public static double ConvertirAKelvin(double celsius)
21         {
22             return celsius + 273.15;
23         }
24         //Muestra los elementos de una lista
25         2 referencias
26         public static void MostrarLista(List<double> lista)
27         {
28             foreach (var item in lista)
29             {
30                 Console.WriteLine(item);
31             }
32         }
33         //Elimina una temperatura de una lista seleccionada
34         1 referencia
35         public static void EliminarTemperatura(List<double> listaFahrenheit, List<double> listaKelvin)
36         {
37             Console.WriteLine("Seleccione la lista de la que desea eliminar una temperatura:");
38             Console.WriteLine("1. Fahrenheit");
39             Console.WriteLine("2. Kelvin");
40             int opcion = int.Parse(Console.ReadLine());
41             Console.Clear();
42
43             //Selecciona la lista correspondiente según la opción del usuario

```

```

41             //Selecciona la lista correspondiente según la opción del usuario
42             List<double> listaSeleccionada = opcion == 1 ? listaFahrenheit : listaKelvin;
43
44             //Pide al usuario el índice de la temperatura a eliminar
45             Console.WriteLine("Ingrese el índice de la temperatura que desea eliminar:");
46             int indice = int.Parse(Console.ReadLine());
47             Console.Clear();
48
49             //Verifica si el índice es válido y elimina dicha temperatura
50             if (indice >= 0 && indice < listaSeleccionada.Count)
51             {
52                 listaSeleccionada.RemoveAt(indice);
53                 Console.WriteLine("Temperatura eliminada.");
54             }
55             else
56             {
57                 Console.WriteLine("Índice no válido.");
58             }
59         }
60     }
61 }
62

```



```
C:\WINDOWS\system32\cmd. x + v
Ingrese una temperatura en grados Celsius:
5
Temperatura en Fahrenheit: 41
Temperatura en Kelvin: 278,15
¿Desea ingresar otra temperatura? (s/n)
```

## Estructura del Programa

El programa se divide en dos archivos: Program.cs y Funciones.cs.

Dentro de **Program.cs**:

### Importaciones y Espacio de Nombres:

Se importa la biblioteca necesaria para poder hacer uso de las listas genericas (System.Collections.Generic).

Define el espacio de nombres Ejercicio3.

### Clase Principal y Main:

Se crean las respectivas listas para almacenar temperaturas en Celsius, Fahrenheit y Kelvin.

Se utiliza un bucle while para permitir al usuario ingresar múltiples temperaturas en Celsius, convertirlas y almacenarlas en las listas correspondientes.

Muestra las temperaturas convertidas

Pregunta al usuario si este desea eliminar alguna temperatura convertida y llama a la función correspondiente

Muestra las listas de temperaturas convertidas

## **Funciones.cs**

Define una clase estática Funciones que contiene métodos auxiliares.

### **Métodos de Conversión:**

ConvertirAFahrenheit: Convierte una temperatura de Celsius a Fahrenheit.

ConvertirAKelvin: Convierte una temperatura de Celsius a Kelvin.

### **Método para Mostrar Listas:**

MostrarLista: Muestra todos los elementos de una lista

### **Método para Eliminar Temperaturas:**

EliminarTemperatura: Permite al usuario seleccionar una lista (Fahrenheit o Kelvin) y eliminar una temperatura especificando su índice (Iniciando desde el 0).

### **Flujo del Programa:**

**Ingreso de Temperaturas:** El usuario ingresa temperaturas en Celsius, que se convierten a Fahrenheit y Kelvin y se almacenan en las listas.

**Mostrar Temperaturas:** Las temperaturas convertidas se muestran

**Eliminar Temperaturas:** El usuario puede eliminar una temperatura de las listas convertidas especificando el índice(Que va desde el 0).

**Listas Finales:** Al final, se muestran las listas de temperaturas convertidas si es que no se desea eliminar ni agregar una nueva conversión.

## Ejercicio 5

```

Program5.cs x Funciones.cs
Ejercicio5 Ejercicio5.Program Main(string[] args)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Ejercicio5
8 {
9     0 referencias
10    internal class Program
11    {
12        0 referencias
13        static void Main(string[] args)
14        {
15            Console.WriteLine("Bienvenido Usuario. Este programa se encarga de imprimir en orden inverso de una serie de numeros que elijas, ademas de mostrar");
16            Console.ReadLine();
17            Console.Clear();
18
19            //Se Solicita al usuario que ingrese el tamaño del arreglo
20            Console.WriteLine("Ingrese el tamaño del arreglo (Cantidad de numeros):");
21
22            if (int.TryParse(Console.ReadLine(), out int tamaño) && tamaño > 0)
23            {
24                //Inicializa el arreglo con el tamaño especificado
25                int[] arreglo = new int[tamaño];
26
27                //Solicita al usuario que ingrese los elementos del arreglo (los numeros)
28                for (int i = 0; i < tamaño; i++)
29                {
30                    Console.WriteLine($"Ingrese el numero {i + 1}:");
31                    while (!int.TryParse(Console.ReadLine(), out arreglo[i]))
32                    {
33                        Console.WriteLine("Entrada no válida. Por favor, ingrese un número entero:");
34                    }
35                }
36
37                Console.Clear();
38
39                //Muestra el arreglo original
40                Funciones.MostrarArreglo(arreglo, "Arreglo original:");
41
42                //Invierte el arreglo y lo almacena dentro de una nueva variable
43                int[] arregloInvertido = Funciones.InvertirArreglo(arreglo);
44
45                //Muestra el arreglo invertido
46                Funciones.MostrarArreglo(arregloInvertido, "Arreglo invertido:");
47
48                //Cuenta la cantidad de valores impares dentro del arreglo invertido
49                int cantidadImpares = Funciones.ContarImpares(arregloInvertido);
50
51                //Muestra la cantidad de valores impares dentro del arreglo
52                Console.WriteLine($"Cantidad de valores impares en el arreglo invertido: {cantidadImpares}");
53            }
54            else
55            {
56                Console.WriteLine("Tamaño del arreglo no válido. Debe ser un número entero y positivo.");
57            }
58        }
59    }
60

```

```
Program5.cs  Funciones.cs
Ejercicio5  Ejercicio5.Funciones  InvertirArreglo(int[] arreglo)

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Ejercicio5
8  {
9      4 referencias
10     internal class Funciones
11     {
12         //Muestra los elementos del arreglo en la consola
13         2 referencias
14         public static void MostrarArreglo(int[] arreglo, string mensaje)
15         {
16             Console.WriteLine(mensaje);
17             foreach (int valor in arreglo)
18             {
19                 Console.Write(valor + " ");
20             }
21             Console.WriteLine();
22         }
23         //Invierte el arreglo
24         1 referencia
25         public static int[] InvertirArreglo(int[] arreglo)
26         {
27             int[] invertido = new int[arreglo.Length];
28             for (int i = 0; i < arreglo.Length; i++)
29             {
30                 //Se asignan los elementos del arreglo en orden inverso
31                 invertido[i] = arreglo[arreglo.Length - 1 - i];
32             }
33             return invertido;
34         }
35         //Cuenta los valores impares en el arreglo
36         1 referencia
37         public static int ContarImpares(int[] arreglo)
38         {
39             int contador = 0;
40             foreach (int valor in arreglo)
41             {
42                 //Incrementa el contador si el valor es impar
43                 if (valor % 2 != 0)
44                 {
45                     contador++;
46                 }
47             }
48             return contador;
49         }
50     }
```

```
42     {
43         contador++;
44     }
45 }
46 return contador;
47 }
48 }
49 }
50 }
```

```
C:\WINDOWS\system32\cmd.  +  v
Arreglo original:
4 0 5 9 3
Arreglo invertido:
3 9 5 0 4
Cantidad de valores impares en el arreglo invertido: 3
Presione una tecla para continuar . . .
```

## **Ejercicio5.cs (Archivo principal)**

### **Interacción con el Usuario:**

Solicita al usuario que ingrese el tamaño del arreglo y valida la entrada.

Inicializa un arreglo con el tamaño especificado.

Pide al usuario que ingrese los elementos del arreglo, validando cada entrada.

### **Llamadas a Funciones:**

Llama a Funciones.MostrarArreglo para mostrar el arreglo original.

Llama a Funciones.InvertirArreglo para invertir el arreglo.

Llama a Funciones.MostrarArreglo para mostrar el arreglo invertido.

Llama a Funciones.ContarImpares para contar los valores impares en el arreglo invertido y muestra el resultado.

## **Funciones.cs (Archivo de funciones)**

### **Clase Estática:**

public static class Funciones: Define una clase estática que contiene todas las funciones auxiliares.

### **Mostrar Arreglo:**

public static void MostrarArreglo(int[] arreglo, string mensaje): Imprime los elementos del arreglo en la consola junto con un mensaje.

### **Invertir Arreglo:**

public static int[] InvertirArreglo(int[] arreglo): Invierte los elementos del arreglo y devuelve un nuevo arreglo invertido.

### **Contar Impares:**

public static int ContarImpares(int[] arreglo): Cuenta y devuelve la cantidad de valores impares en el arreglo.



## Ejercicio 1

```

Program.cs  Class1.cs
ConsoleApp31 ConsoleApp31.Program Main(string[] args)
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp31
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("El siguiente programa Solicita la base su altura de un triangulo y C");
14
15             Console.WriteLine("Ingrese la altura del triangulo");
16             double Altura = double.Parse(Console.ReadLine());
17
18             Console.WriteLine("Ingrese la base del triangulo");
19             double Base = double.Parse(Console.ReadLine());
20
21             double Area = Calcular_Area_Triangulo.calcular_area_triangulo(Altura, Base);
22             Calcular_Area_Triangulo.ImprimirResultado(Area);
23
24             Console.ReadKey();
25         }
26     }
27 }
28

```

```

Program.cs  Class1.cs
ConsoleApp31 ConsoleApp31.Calcular_Area_Triangulo calcular_area_triangulo(double Base, double Altura)
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp31
8  {
9      internal class Calcular_Area_Triangulo
10     {
11         //metodo para calcular el area
12
13         public static double calcular_area_triangulo(double Base, double Altura)
14         {
15             double Area;
16             Area = Base * Altura / 2;
17             return Area;
18         }
19
20         //Metodo para imprimir el resultado
21         public static void ImprimirResultado(double Area)
22         {
23             Console.WriteLine($"El Area del triangulo es: {Area:F2}");
24         }
25     }
26 }
27
28

```

110 % No se encontraron problemas. Línea: 10 Carácter: 9 SPC CRLF

## Ejercicio 1 (Distinto Documento)

```

Program.cs
ConsoleApp34
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 class Program
8 {
9     7 referencias
10     struct Persona
11     {
12         public string Nombre;
13         public string Direccion;
14         public string Telefono;
15         public int Edad;
16     }
17
18     //List<persona> significa que creamos la lista llamada persona pero la pasamos a funcion con el personas
19     1 referencia
20     static void MostrarNombres(List<Persona> personas)
21     {
22         //el Foreach es un bucle especializado en recorrer colecciones, como listas o arrays.
23         //el tipo de dato var le dice al compilador que deduzca el tipo de la variable basada en el valor que se le asigna
24         foreach (var persona in personas)
25         {
26             Console.WriteLine(persona.Nombre);
27         }
28
29         //sobre la misma /List<persona> agregamos de tipo entero edad
30         1 referencia

```

```

Program.cs
ConsoleApp34
28 static void MostrarPorEdad(List<Persona> personas, int edad)
29 {
30     foreach (var persona in personas)
31     {
32         if (persona.Edad == edad)
33         {
34             Console.WriteLine($"{persona.Nombre} {persona.Direccion} {persona.Telefono} {persona.Edad}");
35         }
36     }
37 }
38
39 //sobre la misma List<persona> agregamos de tipo string
40 1 referencia
41 static void MostrarPorNombre(List<Persona> personas, string nombre)
42 {
43     foreach (var persona in personas)
44     {
45         if (persona.Nombre.Equals(nombre, StringComparison.OrdinalIgnoreCase))
46         {
47             Console.WriteLine($"{persona.Nombre} {persona.Direccion} {persona.Telefono} {persona.Edad}");
48         }
49     }
50 }
51
52 0 referencias
53 static void Main()
54 {
55     //creacion del /List<persona> personas
56     List<Persona> personas = new List<Persona>();
57     char opcion;

```

```

Program.cs
ConsoleApp34
56
57 Console.WriteLine("El siguiente programa primero solicitara el nombre la direccion telefono y edad");
58 Console.WriteLine("Presione cualquier tecla para continuar\n");
59 Console.ReadKey();
60
61 //persona
62 do
63 {
64     Persona p = new Persona();
65     Console.Write("Ingrese el nombre: ");
66     p.Nombre = Console.ReadLine();
67     Console.Write("Ingrese la direccion: ");
68     p.Direccion = Console.ReadLine();
69     Console.Write("Ingrese el telefono: ");
70     p.Telefono = Console.ReadLine();
71     Console.Write("Ingrese la edad: ");
72     p.Edad = int.Parse(Console.ReadLine());
73     personas.Add(p);
74
75     // Limpiar la pantalla después de ingresar los datos
76     Console.Clear();
77
78     Console.Write("Desea ingresar otra persona? (s/n): ");
79     opcion = Console.ReadLine().ToLower()[0];
80 } while (opcion == 's');
81
82 do
83 {
84     Console.WriteLine("\nMenu:");
85     Console.WriteLine("1. Mostrar todos los nombres");
86     Console.WriteLine("2. Mostrar personas por edad");

```

```

Program.cs
ConsoleApp34
86 Console.WriteLine("2. Mostrar personas por edad ");
87 Console.WriteLine("3. Mostrar personas por nombre");
88 Console.WriteLine("4. Salir");
89 Console.Write("Seleccione una opcion: ");
90 opcion = Console.ReadLine().ToLower()[0];
91
92 // Limpiar la pantalla después de mostrar el menú
93 Console.Clear();
94
95 switch (opcion)
96 {
97     case '1':
98         MostrarNombres(personas);
99         break;
100     case '2':
101         Console.Write("Ingrese la edad: ");
102         int edad = int.Parse(Console.ReadLine());
103         MostrarPorEdad(personas, edad);
104         break;
105     case '3':
106         Console.Write("Ingrese el nombre: ");
107         string nombre = Console.ReadLine();
108         MostrarPorNombre(personas, nombre);
109         break;
110     case '4':
111         Console.WriteLine("Saliendo del programa.");
112         break;
113     default:
114         Console.WriteLine("Opcion invalida.");
115         break;
116 }

```

```

116 }
117
118 // Limpiar la pantalla después de ejecutar cada opción
119 Console.Clear();
120
121 } while (opcion != '4');
122 }
123 }

```