

Ejercicios de repaso

Iris Pérez Aparicio
2º DAM
Campus FP Emprende Humanes

Nivel 1 – Básico: Operaciones simples y lectura de datos

Ejercicio_1

- Suma de dos números
 - Crea un programa que pida dos números por consola y muestre su suma.
 - Bonus: muestra también la resta, multiplicación y división

```
using System;

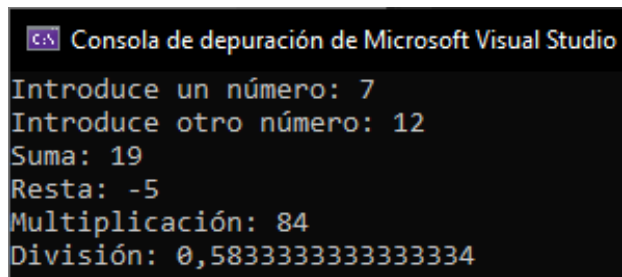
class SumaDosNumeros
{
    static void Main(string[] args)
    {
        Console.Write("Introduce un número: ");
        double num1 = Convert.ToDouble(Console.ReadLine());

        Console.Write("Introduce otro número: ");
        double num2 = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Suma: " + (num1 + num2));
        Console.WriteLine("Resta: " + (num1 - num2));
        Console.WriteLine("Multiplicación: " + (num1 * num2));

        if (num2 != 0)
            Console.WriteLine("División: " + (num1 / num2));
        else
            Console.WriteLine("División: No se puede dividir por cero");
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Microsoft Visual Studio
Introduce un número: 7
Introduce otro número: 12
Suma: 19
Resta: -5
Multiplicación: 84
División: 0,5833333333333334
```

Ejercicio_2

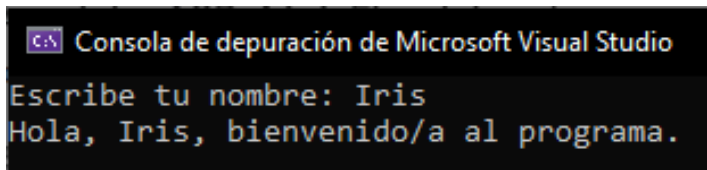
- Saludo personalizado
 - Pide al usuario su nombre y muestra un mensaje: “Hola, [nombre], bienvenido al programa”.

```
using System;

class SaludoPersonalizado
{
    static void Main(string[] args)
    {
        Console.Write("Escribe tu nombre: ");
        string nombre = Console.ReadLine();

        Console.WriteLine($"Hola, {nombre}, bienvenido al programa.");
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Microsoft Visual Studio
Escribe tu nombre: Iris
Hola, Iris, bienvenido/a al programa.
```

Ejercicio_3

- Edad en meses
 - Pide la edad del usuario y calcula cuántos meses ha vivido.
 - Muestra el resultado en consola.

```
using System;

class EdadEnMeses
{
    static void Main(string[] args)
    {
        Console.Write("Introduce tu edad: ");
        int edad = Convert.ToInt32(Console.ReadLine());
    }
}
```

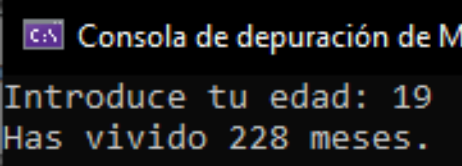
```

        int edadMeses = edad * 12;

        Console.WriteLine($"Has vivido {edadMeses} meses.");
    }
}

```

EJECUCIÓN:



```

C:\> Consola de depuración de M
Introduce tu edad: 19
Has vivido 228 meses.

```

Nivel 2 – Condicionales y operadores

Ejercicio_4

- Mayor de dos números
 - Pide dos números y muestra cuál es mayor.
 - Si son iguales, muestra un mensaje “Los números son iguales”.

```

using System;

class MayorDeDosNumeros
{
    static void Main(string[] args)
    {
        Console.Write("Introduce un número: ");
        double num1 = Convert.ToDouble(Console.ReadLine());

        Console.Write("Introduce otro número: ");
        double num2 = Convert.ToDouble(Console.ReadLine());

        if (num1 > num2)
        {
            Console.WriteLine($"{num1} es mayor que {num2}");
        }
        else if (num2 > num1)
        {
            Console.WriteLine($"{num2} es mayor que {num1}");
        }
    }
}

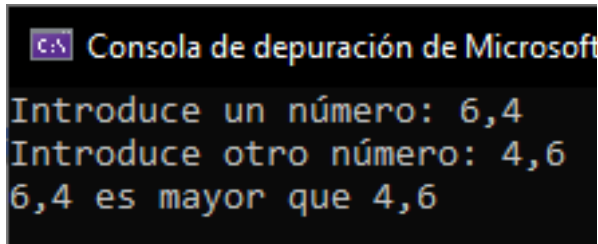
```

```

        else
        {
            Console.WriteLine("Los números son iguales");
        }
    }
}

```

EJECUCIÓN:



```

C:\> Consola de depuración de Microsoft
Introduce un número: 6,4
Introduce otro número: 4,6
6,4 es mayor que 4,6

```

Ejercicio_5

- Aprobado o suspendido
 - Pide una nota y muestra “Aprobado” si es ≥ 5 o “Suspendido” si es < 5 .

```

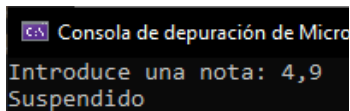
using System;

class AprobadoOSuspendido
{
    static void Main(string[] args)
    {
        Console.Write("Introduce una nota: ");
        double nota = Convert.ToDouble(Console.ReadLine());

        if (nota >= 5)
        {
            Console.WriteLine("Aprobado");
        }
        else
        {
            Console.WriteLine("Suspendido");
        }
    }
}

```

EJECUCIÓN:



```

C:\> Consola de depuración de Micro
Introduce una nota: 4,9
Suspendido

```

Ejercicio_6

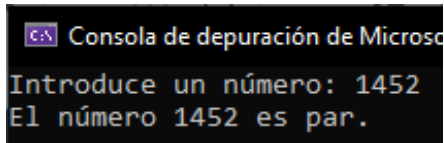
- Número par o impar
 - Pide un número y muestra si es par o impar usando el operador módulo %.

```
using System;

class NumeroParOImpar
{
    static void Main(string[] args)
    {
        Console.Write("Introduce un número: ");
        int num = Convert.ToInt32(Console.ReadLine());

        if (num % 2 == 0)
        {
            Console.WriteLine($"El número {num} es par.");
        }
        else
        {
            Console.WriteLine($"El número {num} es impar.");
        }
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Microso
Introduce un número: 1452
El número 1452 es par.
```

Nivel 3 - Métodos y funciones

Ejercicio_7

- Pide al usuario el **nombre de un corredor**.
- Pide al usuario los **tiempos de tres carreras** en segundos.
- Crea un **método llamado CalcularPromedio** que reciba esos tres tiempos y devuelva el **tiempo medio**.

- Muestra un mensaje en pantalla con el siguiente formato:
Hola, [nombre], tu tiempo medio es: [promedio] segundos

```
using System;

class CalcularPromedioTiempos
{
    static void Main(string[] args)
    {
        Console.Write("Introduce el nombre de un corredor: ");
        string nombre = Console.ReadLine();

        Console.Write("Introduce el tiempo de la primera carrera (segundos): ");
        double tiempo1 = Convert.ToDouble(Console.ReadLine());

        Console.Write("Introduce el tiempo de la segunda carrera (segundos): ");
        double tiempo2 = Convert.ToDouble(Console.ReadLine());

        Console.Write("Introduce el tiempo de la tercera carrera (segundos): ");
        double tiempo3 = Convert.ToDouble(Console.ReadLine());

        double promedio = CalcularPromedio(tiempo1, tiempo2, tiempo3);

        Console.WriteLine($"Hola, {nombre}, tu tiempo medio es: {promedio} segundos");
    }

    static double CalcularPromedio(double t1, double t2, double t3)
    {
        return (t1 + t2 + t3) / 3;
    }
}
```

EJECUCIÓN:

```
Consola de depuración de Microsoft Visual Studio
Introduce el nombre de un corredor: Rafa
Introduce el tiempo de la primera carrera (segundos): 246
Introduce el tiempo de la segunda carrera (segundos): 312
Introduce el tiempo de la tercera carrera (segundos): 168
Hola, Rafa, tu tiempo medio es: 242 segundos
```

Nivel 4 – Introducción a POO (clases y objetos)

Ejercicio_8

- Clase Producto básica
 - Crea una clase Producto con atributos: Nombre, descripción, precio.
 - Constructor que inicialice estos valores.
 - Método Datos() que muestre la información del producto.
 - Crea 2-3 objetos en Main() y llama al método Datos() de cada uno.

```
using System;

class Producto
{
    public string _nombre;
    public string _descripcion;
    public double _precio;

    public Producto(string nombre, string descripcion, double precio)
    {
        _nombre = nombre;
        _descripcion = descripcion;
        _precio = precio;
    }

    public void Datos()
    {
        Console.WriteLine($"Nombre: {_nombre}");
        Console.WriteLine($"Descripción: {_descripcion}");
        Console.WriteLine($"Precio: {_precio} euros\n");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Producto producto1 = new Producto("Portátil", "Ordenador portátil de 15 pulgadas", 799.99);
        Producto producto2 = new Producto("Ratón", "Ratón inalámbrico ergonómico", 24.50);
        Producto producto3 = new Producto("Monitor", "Monitor Full HD de 24 pulgadas", 159.90);
    }
}
```

```

        Console.WriteLine("Datos del primer producto: ");
        producto1.Datos();
        Console.WriteLine("Datos del segundo producto: ");
        producto2.Datos();
        Console.WriteLine("Datos del tercer producto: ");
        producto3.Datos();
    }
}

```

EJECUCIÓN:

```

C# Consola de depuración de Microsoft Visual Studio
Datos del primer producto:
Nombre: Portátil
Descripción: Ordenador portátil de 15 pulgadas
Precio: 799,99 euros

Datos del segundo producto:
Nombre: Ratón
Descripción: Ratón inalámbrico ergonómico
Precio: 24,5 euros

Datos del tercer producto:
Nombre: Monitor
Descripción: Monitor Full HD de 24 pulgadas
Precio: 159,9 euros

```

Ejercicio_9

- Compra
 - En Main(), pide al usuario 3 productos con sus precios.
 - Calcula el total de la compra.
 - Bonus: hazlo sin duplicar la clase Producto, es decir, reutilizando el código de esta clase que has desarrollado para el ejercicio 7.

```

using System;
class Producto
{
    public string _nombre;
    public string _descripcion;
    public double _precio;
}

```



```

    public Producto(string nombre, string descripcion, double precio)
    {
        _nombre = nombre;
        _descripcion = descripcion;
        _precio = precio;
    }

    public Producto(string nombre, double precio)
    {
        _nombre = nombre;
        _precio = precio;
    }

    public void Datos()
    {
        Console.WriteLine($"Producto: {_nombre}");
        Console.WriteLine($"Descripción: {_descripcion}");
        Console.WriteLine($"Precio: {_precio} euros\n");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Producto[] productos = new Producto[3];
        double total = 0;

        for (int i = 0; i < productos.Length; i++)
        {
            Console.Write($"Introduce el nombre del producto {i + 1}:
");
            string nombre = Console.ReadLine();

            Console.Write("Introduce el precio: ");
            double precio = Convert.ToDouble(Console.ReadLine());

            productos[i] = new Producto(nombre, precio);
            total += precio;
            Console.WriteLine();
        }

        Console.WriteLine("Resumen de la compra:");
        foreach (var producto in productos)
        {
            producto.Datos();
        }
        Console.WriteLine($"Total de la compra: {total} euros");
    }
}

```

EJECUCIÓN:

```
Consola de depuración de Microsoft Visual Studio
Introduce el nombre del producto 1: Teclado
Introduce el precio: 79,89

Introduce el nombre del producto 2: Altavoces
Introduce el precio: 120,65

Introduce el nombre del producto 3: Ratón
Introduce el precio: 30,29

Resumen de la compra:
Producto: Teclado
Descripción:
Precio: 79,89 euros

Producto: Altavoces
Descripción:
Precio: 120,65 euros

Producto: Ratón
Descripción:
Precio: 30,29 euros

Total de la compra: 230,83 euros
```

Ejercicio_10

- Lista de productos
 - Guarda varios objetos Productos en un array o en una lista.
 - Usa un bucle foreach para mostrar la presentación de todos los productos.
 - Muestra el precio total de los productos.
 - Añade el poder meter un descuento del 15%.
 - Bonus: hazlo sin duplicar la clase Producto, es decir, reutilizando el código de esta clase que has desarrollado para el ejercicio 7.

```
using System;
using System.Collections.Generic;

class Producto
{
```

```

private string _nombre;
private string _descripcion;
private double _precio;

public string Nombre
{
    get { return _nombre; }
    set { _nombre = value; }
}

public string Descripcion
{
    get { return _descripcion; }
    set { _descripcion = value; }
}

public virtual double Precio
{
    get { return _precio; }
    set { _precio = value; }
}

public Producto(string nombre, string descripcion, double precio)
{
    _nombre = nombre;
    _descripcion = descripcion;
    _precio = precio;
}

public virtual void Datos()
{
    Console.WriteLine($"Producto: {_nombre}");
    Console.WriteLine($"Descripción: {_descripcion}");
    Console.WriteLine($"Precio: {_precio} euros\n");
}
}

class ProductoDetallado : Producto
{
    private Dictionary<string, string> _caracteristicas = new
Dictionary<string, string>();

    public ProductoDetallado(string nombre, string descripcion, double
precio)
        : base(nombre, descripcion, precio)
    {
        _caracteristicas = new Dictionary<string, string>();
    }

    public ProductoDetallado(string nombre, string descripcion, double
precio, Dictionary<string, string> caracteristicas)
        : base(nombre, descripcion, precio)
    {

```

```

        _caracteristicas = caracteristicas;
    }

    public string this[string key]
    {
        get
        {
            if (_caracteristicas.ContainsKey(key))
                return _caracteristicas[key];
            else
                return null;
        }
        set
        {
            _caracteristicas[key] = value;
        }
    }

    public override void Datos()
    {
        Console.WriteLine($"Producto: {Nombre}");
        Console.WriteLine($"Descripción: {Descripcion}");
        Console.WriteLine($"Precio: {Precio} euros");
        Console.WriteLine("Características:");
        foreach (var kvp in _caracteristicas)
        {
            Console.WriteLine($"- {kvp.Key}: {kvp.Value}");
        }
        Console.WriteLine();
    }
}

class Program
{
    static void Main(string[] args)
    {
        ProductoDetallado producto1 = new ProductoDetallado("Portátil",
"Ordenador portátil", 800.0);
        producto1["peso"] = "2 Kg";
        producto1["color"] = "Gris";
        producto1["batería"] = "6 horas";

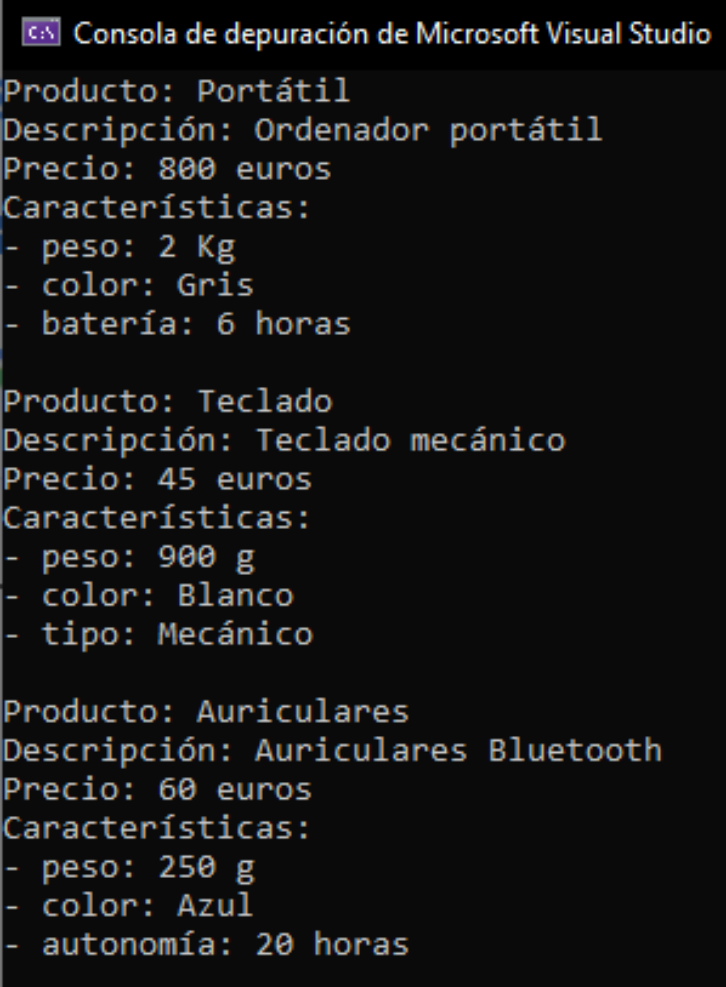
        ProductoDetallado producto2 = new ProductoDetallado("Teclado",
"Teclado mecánico", 45.0);
        producto2["peso"] = "900 g";
        producto2["color"] = "Blanco";
        producto2["tipo"] = "Mecánico";

        ProductoDetallado producto3 = new
ProductoDetallado("Auriculares", "Auriculares Bluetooth", 60.0);
        producto3["peso"] = "250 g";
        producto3["color"] = "Azul";
        producto3["autonomía"] = "20 horas";
    }
}

```

```
        producto1.Datos();  
        producto2.Datos();  
        producto3.Datos();  
    }  
}
```

EJECUCIÓN:



```
C:\> Consola de depuración de Microsoft Visual Studio  
  
Producto: Portátil  
Descripción: Ordenador portátil  
Precio: 800 euros  
Características:  
- peso: 2 Kg  
- color: Gris  
- batería: 6 horas  
  
Producto: Teclado  
Descripción: Teclado mecánico  
Precio: 45 euros  
Características:  
- peso: 900 g  
- color: Blanco  
- tipo: Mecánico  
  
Producto: Auriculares  
Descripción: Auriculares Bluetooth  
Precio: 60 euros  
Características:  
- peso: 250 g  
- color: Azul  
- autonomía: 20 horas
```

Ejercicio_11

- Clase ProductoDetallado
 - Esta clase hereda de la clase Producto.
 - Tiene una propiedad indexada con las características del producto. Por ejemplo: *myProducto["peso"] = "10 Kg"*;
 - Se accede a sus atributos mediante propiedades Get y Set.
 - El valor de la propiedad precio no puede ser negativo.

- Método Datos() que muestre la información detallada del producto.
- Crea 2-3 objetos en Main() y llama al método Datos() de cada uno.
- Si tienes que modificar la clase Producto, hazlo de forma que no afecte a los proyectos anteriores que la utilizan.

Producto.cs

```
using System;

class Producto
{
    //private string _nombre = "";
    private double _precio = 0;

    private string _descripcion = "";

    private Dictionary<string, string> _características = new
Dictionary<string, string>();

    public string Nombre { set; get; }

    public virtual double Precio
    {
        set { if (value < 0) _precio = 0;
        else _precio = value;}
        get { return _precio; }
    }

    public string Descripcion { set; get; }

    public Producto(string nombre, string descripcion, double precio)
    {
        Nombre = nombre;
        Precio = precio;
        Descripcion = descripcion;
    }

    public virtual void Datos()
    {
        Console.WriteLine($"Producto: {Nombre}");
        Console.WriteLine($"Precio: {Precio} euros\n");
        Console.WriteLine($"Descripción: {Descripcion}");
    }
}
```

ProductoDetallado.cs

```
using System;
class ProductoDetallado : Producto
{
    private Dictionary<string, string> _caracteristicas = new
Dictionary<string, string>();

    public ProductoDetallado(string nombre, string descripcion, double
precio)
        : base(nombre, descripcion, precio)
    {
        _caracteristicas = new Dictionary<string, string>();
    }

    public ProductoDetallado(string nombre, string descripcion, double
precio, Dictionary<string, string> caracteristicas)
        : base(nombre, descripcion, precio)
    {
        _caracteristicas = caracteristicas;
    }

    // propiedad indexada
    public string this[string key]
    {
        get { return _caracteristicas[key]; }
        set { _caracteristicas[key] = value; }
    }

    /* sobrescribo el método Datos para que muestre también
    * las características específicas de los productos */
    public override void Datos()
    {
        Console.WriteLine($"Producto: {Nombre}");
        Console.WriteLine($"Descripción: {Descripcion}");
        Console.WriteLine($"Precio: {Precio} euros");
        Console.WriteLine("Características:");
        foreach (var caracteristica in _caracteristicas)
        {
            Console.WriteLine($"- {caracteristica.Key}:
{caracteristica.Value}");
        }
        Console.WriteLine();
    }
}
```

Program.cs

```
using System;

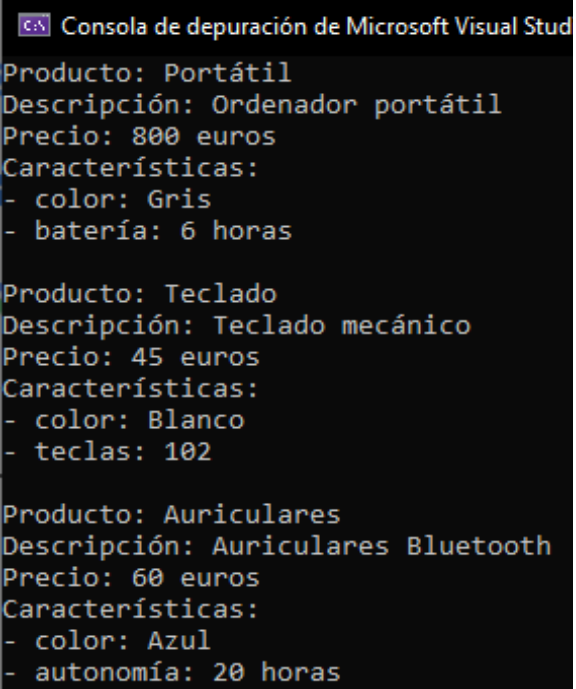
class Program
{
    static void Main(string[] args)
    {
        ProductoDetallado producto1 = new ProductoDetallado("Portátil",
"Ordenador portátil", 800.0);
        producto1["color"] = "Gris";
        producto1["batería"] = "6 horas";

        ProductoDetallado producto2 = new ProductoDetallado("Teclado",
"Teclado mecánico", 45.0);
        producto2["color"] = "Blanco";
        producto2["teclas"] = "102";

        ProductoDetallado producto3 = new
ProductoDetallado("Auriculares", "Auriculares Bluetooth", 60.0);
        producto3["color"] = "Azul";
        producto3["autonomía"] = "20 horas";

        producto1.Datos();
        producto2.Datos();
        producto3.Datos();
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Microsoft Visual Stud

Producto: Portátil
Descripción: Ordenador portátil
Precio: 800 euros
Características:
- color: Gris
- batería: 6 horas

Producto: Teclado
Descripción: Teclado mecánico
Precio: 45 euros
Características:
- color: Blanco
- teclas: 102

Producto: Auriculares
Descripción: Auriculares Bluetooth
Precio: 60 euros
Características:
- color: Azul
- autonomía: 20 horas
```