

Ejercicios

Ejercicio 1

Crea una aplicación de consola que pida un valor y me diga si es positivo, negativo o cero



Ejercicio 2

El objetivo de los ejercicios es asimilar todas las operaciones y tipos de variables primitivas con las que podemos trabajar en C#.

Crearemos un nuevo proyecto en Visual Studio llamado Ejercicio2_Tema4 y crearemos además una clase llamada del mismo modo.

A continuación se adjunta un pequeño programa que permite al usuario seleccionar una operación de las 3 disponibles.

1. Mostrará por consola las soluciones de todas las operaciones posibles.
2. El usuario verá algunas respuestas al jugar con valores lógicos.
3. Utilizaremos los strings (cadenas de texto).

Una vez resuelta la operación el programa preguntará otra vez el tipo de operación que quiere hacer el usuario. Si elige “0” como operación la aplicación se cerrará.

El programa es muy sencillo pero mucha parte del código no la hemos visto aún en lenguaje C#, como por ejemplo la estructura de control switch que se verá en el próximo tema.

Antes de pasar a los ejercicios vamos a ejecutar el programa para comprobar que lo hemos copiado correctamente y podemos usar la consola.

```
namespace Ejercicio1_Tema4
{
    O referencias
    class Program
    {
        O referencias
        static void Main(string[] args)
        {
            int operacion;
            bool salir = false;
            while (!salir)
            {
                Console.WriteLine("Elige el tipo de operación que quieras hacer: ");
                Console.WriteLine(" 1 - Numerica / 2 - Logicas / 3 - Cadenas");
                // Permite capturar lo que se escribe por teclado y lo deja en la variable operación.
                // En este caso también utilizamos una conversión personalizada de tipo String a Int
                operacion = Convert.ToInt32(Console.ReadLine());

                switch (operacion)
                {
                    case 1:
                        Console.WriteLine("1 - Numérica");
                        // EJERCICIO 1
                        // RELLENAR
                        //--
                        // No tocar las estructura, no borrar los break ni los cases
                        break;
                    case 2:
                        Console.WriteLine("2 - Lógicas");
                        // EJERCICIO 2
                        // RELLENAR
                        //--
                        break;
                    case 3:
                        Console.WriteLine("3 - Cadenas");
                        // EJERCICIO 3
                        // RELLENAR
                        //--
                        break;
                    case 0:
                        Console.WriteLine("Bye Bye");
                        salir = true;
                        break;
                    default:
                        Console.WriteLine("No se seleccionó ninguna operación");
                        break;
                }
            }
            Console.ReadKey();
        }
    }
}
```

Ejecuta el proyecto y veremos cómo nos muestra por consola:

"Elige el tipo de operación que quieras hacer: 1 - Numérica / 2 - Lógicas / 3 - Cadenas"

Escribimos 1, 2, 3 o 0 y pulsamos Intro. En este momento veremos cómo responde. El programa no está preparado para recibir otra cosa que no sean enteros. Si intentamos introducir un carácter o un número real dará un error durante la ejecución y se cerrará la aplicación.

```
Elige el tipo de operación que quieras hacer:  
1 - Numerica / 2 - Logicas / 3 - Cadenas
```

De esta forma podemos interactuar con la consola. Ahora que ya hemos comprobado que funciona correctamente pasaremos a hacer unos ejercicios.

En los ejercicios solo debemos completar el código después de los comentarios **// RELLENAR**. iremos repasando todas las operaciones y viendo posibles respuestas el programa.

Vamos a escribir un código en el que mostraremos por pantalla el resultado con todas las operaciones que conocemos (suma, resta, división, multiplicación, resto, incremento) sobre dos variables “a” y “b” y devolveremos el resultado en la variable “resultado”.

La primera operación debemos hacerla de este modo:

Primero hemos inicializado las variables, en segundo lugar sumamos los operandos y por último mostramos el resultado por la consola. Ahora hacer las demás operaciones mostrando el resultado en cada una de ellas, la operación incremento la dejaremos para el final.

Una vez tengamos todas las operaciones cambiar el valor de “a” y “b”.

Caso 1: a = 6 -- b = 2 -- Suma

Caso 2: a = 6 -- b = 4 -- Resta

Caso 3: a = 8 -- b = 4 -- División

Caso 4: a = 5 -- b = 1 -- Multiplicación

Caso 5: a = 4 -- b = 3 -- Resto

Vemos en el caso de la división que si la solución no es un entero devuelve el número truncado sin decimales. Y si intentamos dividir entre 0 vemos en consola un error y la aplicación se cierra.

Debemos tener mucho cuidado con las operaciones, ya que el código era correcto sintácticamente pero ha ocurrido un error durante la ejecución. Más adelante, en el tema de excepciones, veremos que podemos controlar estos errores y evitar que la aplicación se cierre completamente.

```
int a = 0;  
int b = 0;  
int resultado = a + b;  
Console.WriteLine("Suma: " + resultado);
```

La operación incremento tiene una característica curiosa según dónde la situemos respecto a la variable. Podemos ver con este código los dos casos (volver a dejar los valores de “a” y “b” con valor 0):

¿Qué está sucediendo? Escribir en un comentario la respuesta, justo debajo de los incrementos.

Ampliación. Cambiar el tipo de variable de integer a float y volver a comprobar los resultados. Float también acepta números reales. Pero para ello debemos indicar al valor que se guardará como float ya que C# entiende que cualquier número real es doble por defecto y debemos forzarlo.

```
resultado = a++;
Console.WriteLine("Incrementar a: " + a);
Console.WriteLine("Incrementar a: " + resultado);
resultado = ++b;
Console.WriteLine("Incrementar b: " + b);
Console.WriteLine("Incrementar b: " + resultado);
```

Esta forma de asignar el valor una variable se llama coerción o casting: estamos diciéndole al compilador que intente introducir un valor de un tipo distinto en una variable. Debemos tener cuidado, ya que no funciona en todos los casos.

Case 2: Ahora vamos a crear un código similar pero usando todas las operaciones lógicas.

Creamos dos variables numéricas: c y d. Las inicializaremos a los valores 2 y 5.

Aquí tenemos el ejemplo de “c es menor que d”. Ahora debemos hacer:

- c es mayor que d
- c es menor igual que d
- c es mayor igual que d
- c es igual a d
- c es distinto de d
- c es mayor que d o c es menor que d
- 2*c es mayor que d y c es menor que d

Podemos fijarnos que “c es mayor que d o c es menor que d” da el mismo resultado que “c es distinto de d”. Tenemos muchas formas de usar operaciones lógicas. Utilizando menos operaciones conseguiremos un código más eficiente.

```
int c = 0;
int d = 0;
bool resultado2 = false;

resultado2 = c < d;
Console.WriteLine("c es menor que d: " + resultado2);
```

En el case 3 usaremos las operaciones de cadenas de texto. Realizar todas las operaciones al igual que en los dos ejercicios anteriores.

Vamos a crear dos strings, **n** y **m**, las inicializamos a “Hola” y “Buenos días”. A continuación aplicaremos las siguientes funciones:

- Aplicamos en **n** ToUpper() y en **m** ToLower().
- Mostramos la longitud de las cadenas.
- Comprobamos si las cadenas contienen las letras a y b, respectivamente.
- Utilizamos el operador “+” para unir dos cadenas.