

Ejercicios

Ejercicio 1

Para repasar los conceptos de teoría vamos a realizar varios ejercicios donde probaremos de distintas formas cada una de las estructuras de control.

Vamos a crear un proyecto nuevo en Visual Studio llamado **Ejercicio1_Tema5**, en el que realizaremos diversos ejercicios que devolverán por consola los resultados.

A) Vamos a hacer diversas pruebas con la estructura **if-else**.

Si recordamos la teoría, if es una estructura en la que se decide si ejecutar o no un bloque de código según una condición.

```
if (condicion)
{
    // Acción a realizar si la condición se cumple
}
else
{
    // Acción a realizar si la condición NO se cumple
}
```

Dentro de **static void Main(string[] args)** añade:

```
Console.WriteLine("EJERCICIO 1");
Console.WriteLine("¿Está lloviendo?");
Console.WriteLine("Sí / No");
String llueve = Console.ReadLine();
//Comprobar si llueve. En el caso de que llueva, añadir la siguiente línea
Console.WriteLine("¿Tienes paraguas?");
Console.WriteLine("Sí/No");
String paraguas = Console.ReadLine();
//Comprobar si hay paraguas
```

Recordad que tendréis que crear las variables llueve y paraguas.

Ahora, mediante **if-else**, generad una respuesta al usuario en la que se le aconseje coger el paraguas o irse a la playa según llueva o no. Y si está lloviendo y no tiene paraguas que vaya a comprárselo.

Podemos hacerlo con tres **if** con las condiciones adecuadas:

1. Si llueve y sí tenemos paraguas - Respuesta: “**Coge el paraguas**”
2. Si llueve y no tenemos paraguas - Respuesta: “**Compra un paraguas**”
3. Si no llueve - Respuesta: “**Vamos a la playa**”

- B)** Vamos a crear un pequeño programa que devuelva cuantos días tiene un mes, en el cual el usuario elegirá el mes a mostrar. Utilizaremos la estructura **switch-case** para generar una respuesta.

Del mismo modo que en el ejercicio anterior, el usuario tendrá que introducir un valor en la consola.

```
Console.WriteLine("EJERCICIO 2");
Console.WriteLine("Selecciona un mes");
Console.WriteLine("1 - Enero / 2 - Febrero / 3 - Marzo / 4 - Abril / "
+ "5 - Mayo / 6 - Junio / 7 - Julio / 8 - Agosto / 9 - Septiembre / "
+ " 10 - Octubre / 11 - Noviembre / 12 - Diciembre");

// Permite capturar lo que se escribe por teclado
String mes = Console.ReadLine();
```

Con la estructura **switch** podemos devolver la respuesta adecuada al usuario. Ha de ser algo similar al ejemplo que tenemos a continuación:

```
switch (mes)
{
    case "1":
        Console.WriteLine("31 días");
        break;
}
```

Recuerda que:

Enero - 31 días
Febrero - 28 días
Marzo - 31 días
Abril - 30 días
Mayo - 31 días
Junio - 30 días
Julio - 31 días
Agosto - 31 días
Septiembre - 30 días
Octubre - 31 días
Noviembre - 30 días
Diciembre - 31 días

Ampliación apartado B

Modificando el código podemos simplificarlo mucho escribiendo menos respuestas, quitando unos pocos **break** y añadiendo el **default**. Aquí tenemos un ejemplo:

```
switch (mes)
{
    case "2":
        Console.WriteLine("28 días");
        break;
    case "4":
    case "6":
    case "9":
    case "11":
        Console.WriteLine("30 días");
        break;
    default:
        Console.WriteLine("31 días");
        break;
}
```

Comenta el código anterior y utiliza el del ejemplo. Comprobaremos que devuelve los mismos valores. **Para comentar grandes cantidades de código podemos seleccionarlo y luego pulsar CTRL + E a la vez y luego C. De esta forma veremos cómo se comenta cada línea seleccionada. Si volvemos a pulsar CTRL + E a la vez y luego U se descomentará el código.**

- C) Mediante un bucle, crea una operación de multiplicar donde solo podremos usar el operador suma.

Cuando multiplicamos dos números realmente estamos haciendo la misma suma del primer operando tantas veces como indica el segundo operando.

Ejemplo: Multiplicar 5 por 3 equivale a sumar 3 veces 5.

$$5 \times 3 = 15$$

$$5 + 5 + 5 = 15$$

```
Console.WriteLine("EJERCICIO 3");
int num1 = 5;
int num2 = 3;
int resultado = 0;
// Aquí el bucle
Console.WriteLine("El resultado " + resultado);
```

De esta forma, usando for o while, debes implementar la operación de multiplicar utilizando solo sumas.

Cambia los valores a números mucho más altos. Con los procesadores actuales este tipo de operaciones son instantáneas.

- D) Con un bucle do...while vamos a preguntar al usuario si quiere salir del mismo, y hasta que no pulse 1 no saldremos del bucle. Recordar el utilizar sentencias que ya hemos visto anteriormente para capturar la respuesta del teclado:

```
Console.WriteLine("¿Quieres salir del bucle?");
String salir = Console.ReadLine();
```

- E) Vamos a crear un pequeño juego en el que el programa se “inventará” un número aleatorio del 0 al 100 y el usuario tendrá que adivinarlo según las respuestas.

Podemos necesitar el método **Parse** de **Int32**, para leer el entero de consola.

Secuencia:

1. El programa generará un número aleatorio.
2. Pedirá al usuario un número.
3. Lo comparará y devolverá si es mayor, menor o el número exacto.
 - Si es mayor o menor volverá al paso 2.
 - Si es el número exacto irá al paso 4.
4. Mostrará en consola: “¡¡Correcto!! Acertaste”
5. Fin del programa.

Crear un nuevo proyecto llamado **NumeroAleatorio** con una clase llamada **Main**. Utilizando las estructuras que hemos visto hasta ahora, seguir los pasos antes mencionados. Para realizar este código deberéis controlar los ejercicios anteriores y usar estructuras anidadas. Podéis apoyaros en este código:

```
static void Main(string[] args)
{
    Random random = new Random();
    int numero;
    int respuesta;
    Console.WriteLine("Generando número");

    // Genera un número aleatorio
    numero = random.Next(100);

    Console.WriteLine("¿Serás capaz de averiguarlo?");

    // LÓGICA DEL JUEGO
    // RELLENAR

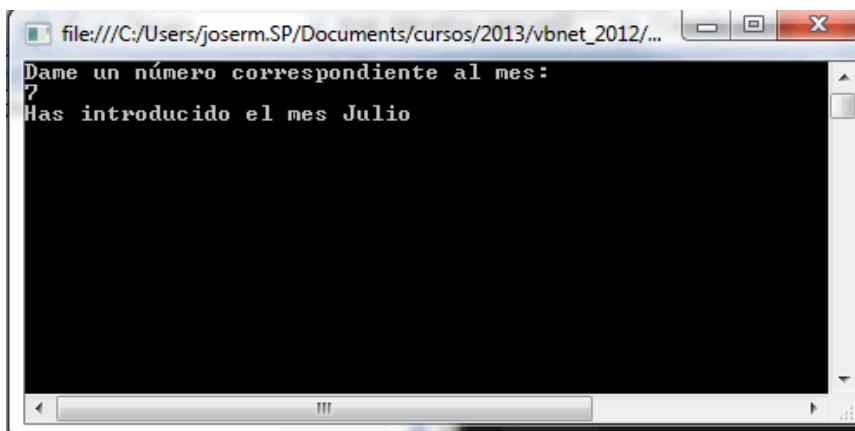
    Console.WriteLine("¡¡Correcto!! Acertaste");
    Console.ReadKey();
}
```

Ampliación apartado E

Tenemos varias opciones para realizar el ejercicio anterior. En este apartado realizaremos la implementación utilizando la palabra clave **goto**, si hemos realizado antes esta implementación lo haremos ahora mediante un bucle **while**.

Ejercicio 2

Haz una aplicación de consola que pida un número de mes y nos diga el literal del mes y en caso de ser mayor que 12 nos lo indique con un mensaje. Utiliza el **switch - case**



Ejercicio 3

Cálculo de sueldos. Aplicación de consola que permita determinar el sueldo básico, descuento y neto que debe percibir un empleado de acuerdo a las horas de trabajo y a un coste por hora.

Hay que tener en cuenta los siguientes aspectos:

- Debe introducir el nombre del empleado, horas trabajadas y coste por hora.
- El sueldo básico se obtiene del producto entre las horas trabajadas y el coste por hora.
- El descuento se obtiene aplicando el 12% al sueldo básico.
- El neto se obtiene de la diferencia entre el sueldo básico y el descuento.
- Mostrar los resultados con el formato de moneda en euros.

```
--PAGO DE EMPLEADOS--  
Nombre del empleado: Raúl  
Horas trabajadas: 40  
Coste por hora: 9  
-----  
EMPLEADO: Raúl  
HORAS TRABAJADAS: 40  
COSTE HORA: 9 Euros  
-----  
BÁSICO: 360 Euros  
DESCUENTO: 43,2 Euros  
NETO: 316,8 Euros
```

Ejercicio 4

Promedio de notas. Aplicación de consola que permita determinar el promedio de cuatro notas de un alumno teniendo cada una de ellas un peso diferente.

Hay que tener en cuenta los siguientes aspectos:

- Debe introducir el nombre del alumno y sus cuatro notas.
- Los pesos de las notas son 10%, 20%, 30% y 40%, según el orden de introducción.
- El promedio de notas se obtiene del valor porcentual de cada nota según el peso.
- Mostrar los resultados en un formato numérico de dos decimales.

```
----- PROMEDIO DE NOTAS -----  
Introduzca el nombre del alumno: Óscar  
Introduzca nota 1: 4  
Introduzca nota 2: 7  
Introduzca nota 3: 6  
Introduzca nota 4: 8  
-----  
ALUMNO: Óscar  
NOTA 1|NOTA 2|NOTA 3|NOTA 4  
4,00    7,00    6,00    8,00  
-----  
PROMEDIO: 6,80
```

Ejercicio 5

Conversión de grados. Aplicación que permite determinar la temperatura en grados Kelvin (K) y Rankine (R) a partir de un valor en grados centígrados.

Hay que tener en cuenta los siguientes aspectos:

- Debe introducir el valor en grados centígrados.
- Usa las siguientes fórmulas:
 - $rankine = \frac{9 \times Centígrados}{5} + 491,67$
 - $kelvin = Centígrados + 273,15$
- Mostrar los valores resultantes a dos decimales usando el método ToString ("0.00").

```
----- CONVERSIÓN DE GRADOS -----
Introduzca un valor en grados centigrados: 14
-----
CENTIGRADOS      : 14,00
-----
RANKINE          : 516,87
KELVIN           : 287,15
```

Ejercicio 6

Venta de coches. Una empresa de venta de automóviles nuevos necesita una aplicación que permita determinar el precio de venta de un coche determinado, sabiendo que la empresa tiene como política ganar el 25% sobre el valor del coche.

Hay que tener en cuenta los siguientes aspectos:

- Debe introducir el coste real del coche.
- Determinar el precio de venta para obtener una ganancia del 25% sobre el coste real del coche.
- Mostrar los valores resultantes en formato moneda euro.

```
----- CONTROL DE PRECIOS DE VENTA - AUTOMOVIL -----
Introduzca el coste real del coche: 17500
-----
COSTE REAL        : 17.500 Euros
-----
PORCENTAJE DE GANANCIA : 4.375 Euros
PRECIO DE VENTA    : 21.875 Euros
```

Ejercicio 7

Realiza un bucle en un programa de consola que escriba los valores pares del 0 al 100. Escríbelos seguidos separados por un guión.

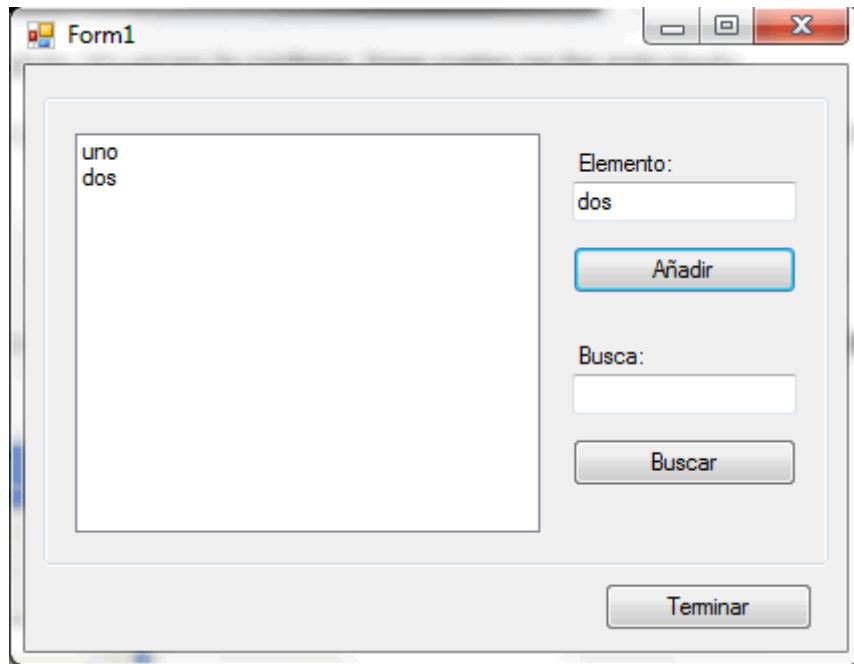
Escribe otro bucle en el mismo programa para que escriba n veces en líneas distintas una cadena que pidas por teclado.

En la pantalla se han escrito los valores y se ha repetido 30 veces la cadena Jose como se ha solicitado.

Nota: Para ver cuántas veces debe realizarse el bucle puede que necesites convertir la cadena leída de string a entero. Utiliza la clase de conversión Convert, por ejemplo.

Ejercicio 8

Lee una serie de elementos y los introduce en un cuadro de lista. A continuación se introduce un elemento en un cuadro de texto y el programa debe decir si está o no en la lista. Comprueba que no se inserten cadenas vacías e implementa el algoritmo del centinela.



Nota: Para saber cuántos elementos hay en una lista utiliza la propiedad Count:
lista.Items.Count