

Intro. a la Programación con C# .NET

Contenido del módulo

- Instrucciones de control de flujo
- Instrucción if...else
- Instrucción switch
- Instrucción while
- Ejercitación

Intro. a la Programación con C# .NET

Instrucciones de control de flujo

Hemos visto hasta ahora que los programas van ejecutando las líneas de código con un **orden secuencial**. Sin embargo, hay situaciones en las que necesito alterar ese orden:

- **bifurcando** o
- **repitiendo** líneas de código.

Por ejemplo, si el programa pide una clave de acceso, deberá continuar con la ejecución normal en caso que la clave sea correcta, y deberá salir del mismo en caso contrario.

Para estas alteraciones en la ejecución tenemos las Instrucciones de control de flujo.

Intro. a la Programación con C# .NET

Instrucciones de control de flujo

En C# tenemos varios tipos de estas instrucciones:

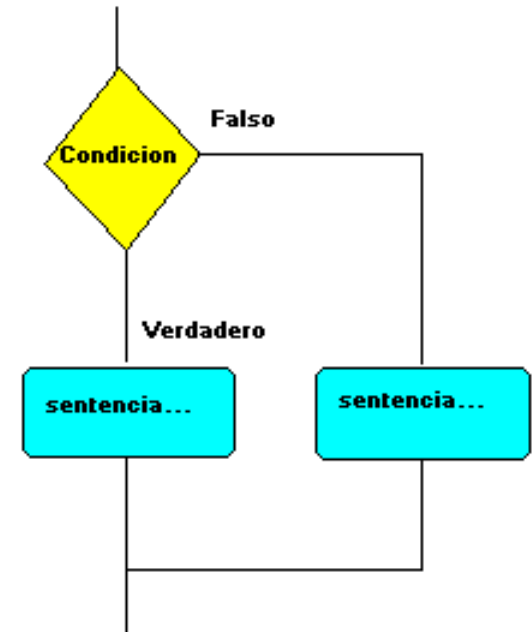
- **Condicionales**: **if...else**, **switch** permiten ejecutar bloques de instrucciones sólo si se da una determinada condición (True/False).
- **Iterativas**: **while**, **do**, **for**, **foreach** permiten ejecutar repetidas veces una instrucción o un bloque de instrucciones mientras se cumpla una condición. (Bucles o ciclos).
- **Instrucciones de salto**: **break**, **continue**, **return**, **goto** permiten variar el orden normal en que se ejecutan las instrucciones (**no recomendadas salvo el **return** en creación de funciones o métodos y **break** en estructura switch**).

Intro. a la Programación con C# .NET

Instrucción if...else

La instrucción if permite ejecutar ciertas instrucciones sólo si se da una determinada condición. Su sintaxis de uso es la sintaxis:

```
if (<condición>) {  
    <instruccionesIf>  
}  
Else {  
    <instruccionesElse>  
}
```



Intro. a la Programación con C# .NET

Instrucción if...else

```
if (<condición>) {  
    <instruccionesIf>  
}  
Else {  
    <instruccionesElse>  
}
```

Se evalúa la expresión **<condición>**, que devuelve un valor lógico **True/False**. Si devuelve **True** se ejecutan las **<instruccionesIf>**, y si devuelve **False** se ejecutan las **<instruccionesElse>**.

La rama else es opcional.

Tanto **<instruccionesIf>** como **<instruccionesElse>** pueden ser una única instrucción o un bloque de instrucciones.

Intro. a la Programación con C# .NET

Ejemplo con IF sin ELSE

```
// Ejemplo IF sin ELSE
int condicion = 3;
Console.WriteLine("Condición = " + condicion);

// Escribir IF y pulsar TAB. El IDE arma la estructura
if (condicion == 3)
{
    Console.WriteLine("La condición se cumplió");
    Console.ReadKey();
}
```

Intro. a la Programación con C# .NET

Ejemplo con IF sin ELSE abreviado

```
// Ejemplo IF sin ELSE modo abreviado
int condicion = 4;
Console.WriteLine("Condición = " + condicion);

// solo cuando hay una sola sentencia en el IF
if (condicion == 4)
    Console.WriteLine("La condición se cumplió");

Console.ReadKey();
```

Intro. a la Programación con C# .NET

Ejemplo con IF y ELSE

```
// Ejemplo IF con ELSE
```

```
int valor = 0;
```

```
Console.Write("Ingrese una valor entero:");
```

```
valor=Convert.ToInt32(Console.ReadLine());
```

```
if (valor == 10)
```

```
{
```

```
    Console.WriteLine("Valor ingresado {0} es igual 10",valor);
```

```
}
```

```
else
```

```
{
```

```
    Console.WriteLine("Valor ingresado {0} es distinto de 10: ",valor);
```

```
}
```

```
Console.ReadKey();
```

Probar ingresando cualquier valor entero y luego 10

Intro. a la Programación con C# .NET

Ejemplo con IF y ELSE abreviado

```
// Ejemplo IF con ELSE abreviado
// una sola sentencia para el IF y para el ELSE
int valor = 0;

Console.Write("Ingrese una valor entero:");
valor = Convert.ToInt32(Console.ReadLine());

if (valor == 10)
    Console.WriteLine("Valor ingresado {0} es igual 10", valor);
else
    Console.WriteLine("Valor ingresado {0} es distinto de 10 ", valor);

Console.ReadKey();
```

Probar ingresando cualquier valor entero y luego 10

Intro. a la Programación con C# .NET

Ejemplo con IF y ELSE anidado

Dados 3 notas de un alumno, promediarlas.

- Según el promedio obtenido mostrar:
- si es 4 "**Regular**"
- si es menor o igual a 3 "**No aprobado**"
- si es mayor o igual a 7 "**Promocionado**"

Probar modificando los valores de las variables para obtener los 3 resultados

Intro. a la Programación con C# .NET

```
int nota1=7, nota2=8, nota3=9;
int prom=(nota1 + nota2 + nota3) / 3;
if (prom>=7)
{
    Console.Write("Promocionado con {0}",prom);
}
else
{
    if (prom>=4)
    {
        Console.Write("Regular con {0}", prom);
    }
    else
    {
        Console.Write("Reprobado con {0}", prom);
    }
}
Console.ReadKey();
```

**Ejemplo
con IF y
ELSE
anidado**

Intro. a la Programación con C# .NET

Lab con IF y ELSE anidado

Tomar el ejercicio del promedio de notas y agregarle:

- Ingreso por pantalla de las 3 notas
- Uso de constantes
- Control para nota mayor a 10

Intro. a la Programación con C# .NET

Lab con IF y ELSE anidado

Solución

```
int nota1, nota2, nota3;
const int Regular = 4;
const int AprobadoDesde = 7;
const int AprobadoHasta = 10;
Console.Write("Ingrese primer nota:");
nota1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Ingrese segunda nota:");
nota2 = Convert.ToInt32(Console.ReadLine());
Console.Write("Ingrese tercer nota:");
nota3 = Convert.ToInt32(Console.ReadLine());

int prom = (nota1 + nota2 + nota3) / 3;
if (prom >= AprobadoDesde && prom <= AprobadoHasta)
{
    Console.Write("Promocionado con {0}", prom);
}
else
{
    if (prom >= Regular)
    {
        Console.Write("Regular con {0}", prom);
    }
    else
    {
        Console.Write("Reprobado con {0}", prom);
    }
}
Console.ReadKey();
```

Intro. a la Programación con C# .NET

Instrucción Switch

La instrucción switch permite ejecutar unos u otros bloques de instrucciones según el valor de una cierta expresión. Su estructura es:

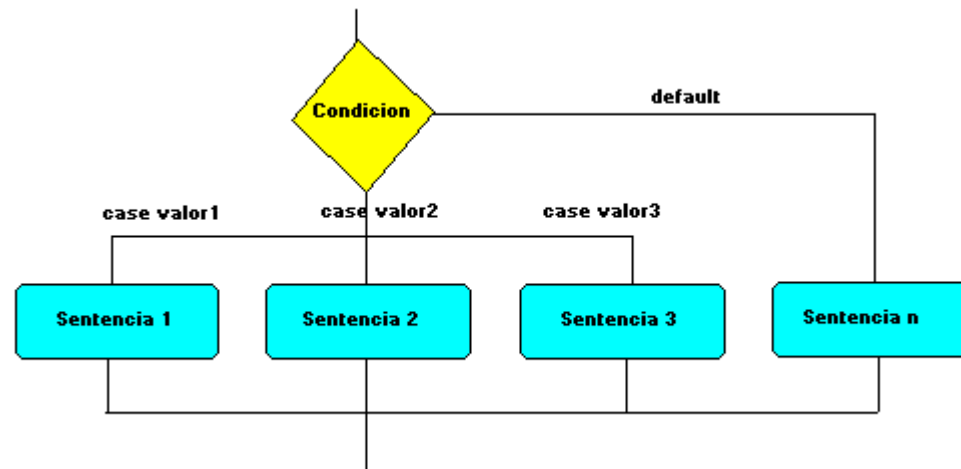
```
switch (<expresión>
{
    case valor1: Sentencias1; <siguiente accion>;
    case valor2: Sentencias2; <siguiente accion>;
    case valor3: Sentencias3; <siguiente accion>;
    default: Sentencias4;
}
```

Intro. a la Programación con C# .NET

Instrucción Switch

Se evalúa **<expresión>**. Si su valor es **<valor1>** se ejecuta el **<bloque1>**, si es **<valor2>** se ejecuta **<bloque2>**, y así sucesivamente. Si no es igual a ninguno y se incluye la rama **default**, se ejecuta el **<bloqueDefault>**, pero si no se incluye se pasa a ejecutar la instrucción siguiente al switch.

No puede haber más de una rama con el mismo valor.



Intro. a la Programación con C# .NET

Ejemplo con Switch

- Probar con 11
- Luego cambiar por otro valor
- Probar quitar el break

```
//Ejemplo con switch
int cond = 11;

switch (cond)
{
    case 10:
        Console.WriteLine("El Valor 10");
        break;
    case 11:
        Console.WriteLine("El Valor 11");
        break;
    case 12:
        Console.WriteLine("El Valor 12");
        break;
    case 13:
        Console.WriteLine("El Valor 13");
        break;
    default:
        Console.WriteLine("El Valor otro");
        break;
}

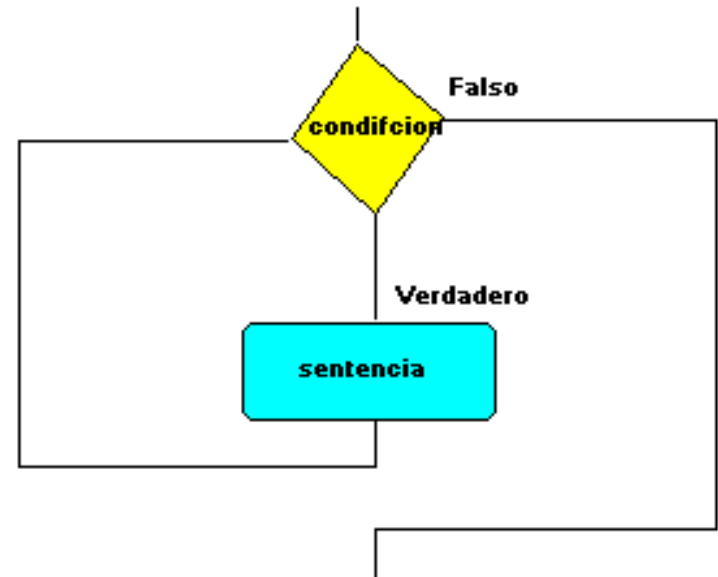
Console.ReadKey();
```


Intro. a la Programación con C# .NET

Instrucción While

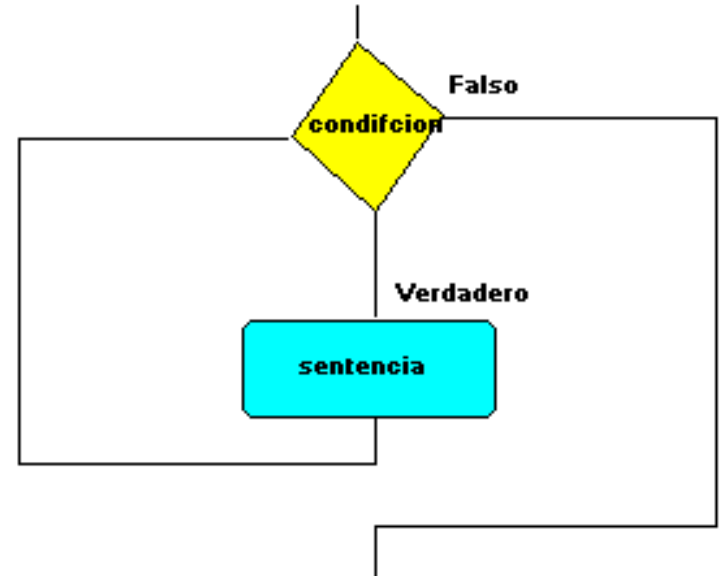
La instrucción while permite ejecutar un bloque de instrucciones mientras se de una cierta instrucción. Su sintaxis de uso es:

```
while (<condición>) {  
    <instrucciones>  
}
```



Intro. a la Programación con C# .NET

Instrucción While



Su significado es el siguiente:

Se evalúa la **<condición>** indicada. Si es cierta (**True**) se ejecutan las **<instrucciones>** (sentencia) y se repite el proceso de evaluación de **<condición>** y ejecución de **<instrucciones>** hasta que deje de serlo. Cuando sea falsa (**False**) se ejecuta la instrucción siguiente al while.

Intro. a la Programación con C# .NET

Ejemplo con While

- Luego de probarlo, decommentar el a++ y probar
- Lab:** Mostrar los números en orden descendente

```
//WHILE
int a = 1;
while (a <= 5)
{
    Console.WriteLine(a);
    a = a + 1;
    //a++;
}
Console.WriteLine("-- fin del while --");
Console.WriteLine("Llegó a {0} ", a);

Console.ReadKey();
```

Intro. a la Programación con C# .NET

Lab con While - Solución

```
//Lab WHILE orden descendente
int a = 5;
while (a >= 1)
{
    Console.WriteLine(a);
    a--;
}
Console.WriteLine("-- fin del while --");
Console.WriteLine("Llegó a {0} ", a);

Console.ReadKey();
```

Intro. a la Programación con C# .NET

Resolver los ejercicios del Lab 2

Ejercicio 1: Dado el siguiente código:

```
int nro1 = 100, nro2 = 500, nro3 = 250;
```

Informar cuál de los tres números es mayor.

Ejercicio 4: Ingresar dos números y ofrecer al usuario un menú con las siguientes opciones: “1-suma 2-resta 3-producto 4-división”. Luego, mostrar el resultado de la operación aritmética elegida.

Tratar de hacerlos sólo sin mirar la solución!!!

Intro. a la Programación con C # .NET

Más ejercicios: While

Lab 3, Ejercicio 2: Imprimir los números del 1 al 10 saltando de a 2 uno abajo del otro.

Lab3, Ejercicio 4: Imprimir los números del 1 al 10 sin imprimir números 2,5 y 9 uno abajo del otro.

Lab3, Ejercicio 5: Imprimir los números del 1 al 30 sin imprimir números entre el 10 y el 20 uno abajo del otro.

Lab3, Ejercicio 6: Imprimir la suma de los números del 1 al 10.

Lab3, Ejercicio 7: Imprimir la suma de los números pares del 1 al 25.