<!--Bucle-->

Ciclo For en C#

{

<Por="Dimas Samuel"/>

}

Agenda

01 Que es?

02 Como funciona?

03 Representacion grafica

05 Sintaxis en C#

06 Ejemplos

07 Ejercicios

Que es? {

Los ciclos for son lo que se conoce como estructuras de control de flujo cíclicas o simplemente estructuras cíclicas, estos ciclos, como su nombre lo sugiere, nos permiten ejecutar una o varias líneas de código de forma iterativa, conociendo un valor especifico inicial y otro valor final, además nos permiten determinar el tamaño del paso entre cada "giro" o iteración del ciclo.

- Se utiliza cuando se necesita ejecutar repetidamente una sentencia o bloque de sentencias, un número de veces conocido.
- Se le conoce o se le llama bucle desde -hasta

}

Como funciona? {

La instrucción for permite repetir una instrucción o una instrucción compuesta un número especificado de veces. El cuerpo de una instrucción for se ejecuta cero o más veces hasta que una condición opcional sea falsa.

}

Como funciona? {

Inicializacion

Actualizacion

```
for ( i=0; i<=10; i++)
```

Condicion

}

Como funciona? {

Comienzo

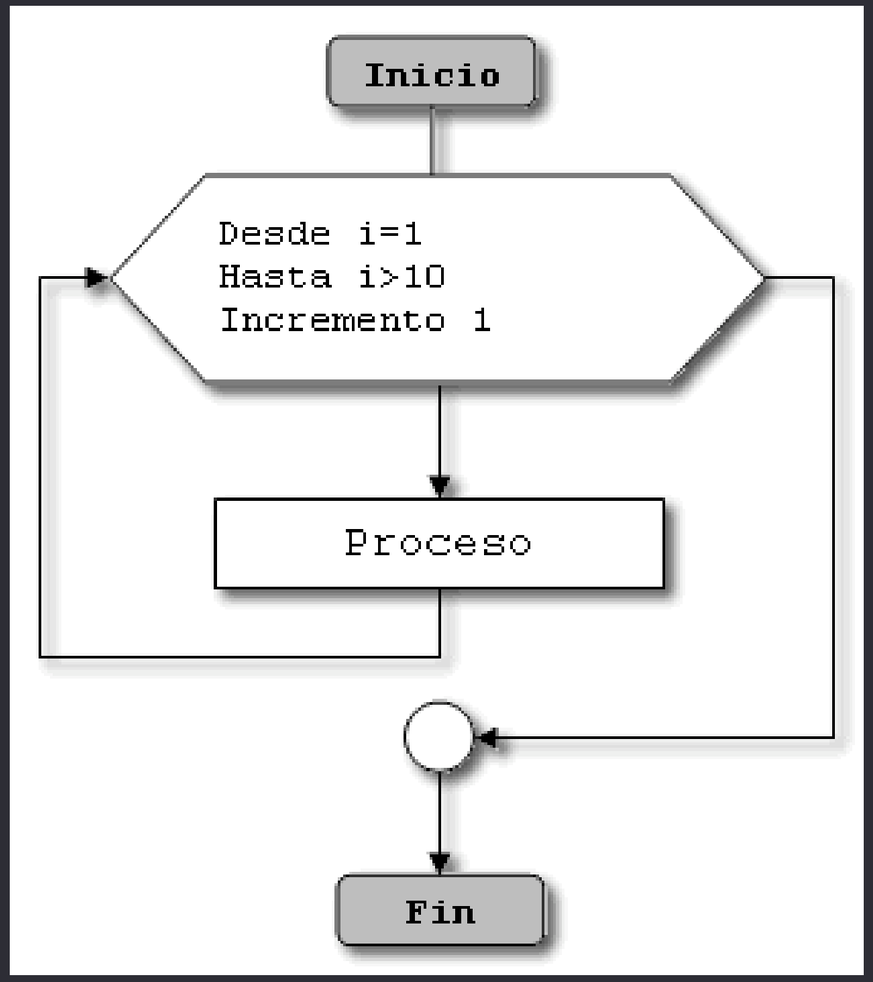
Avance

```
for ( i=0; i<=10; i++)
```

Final

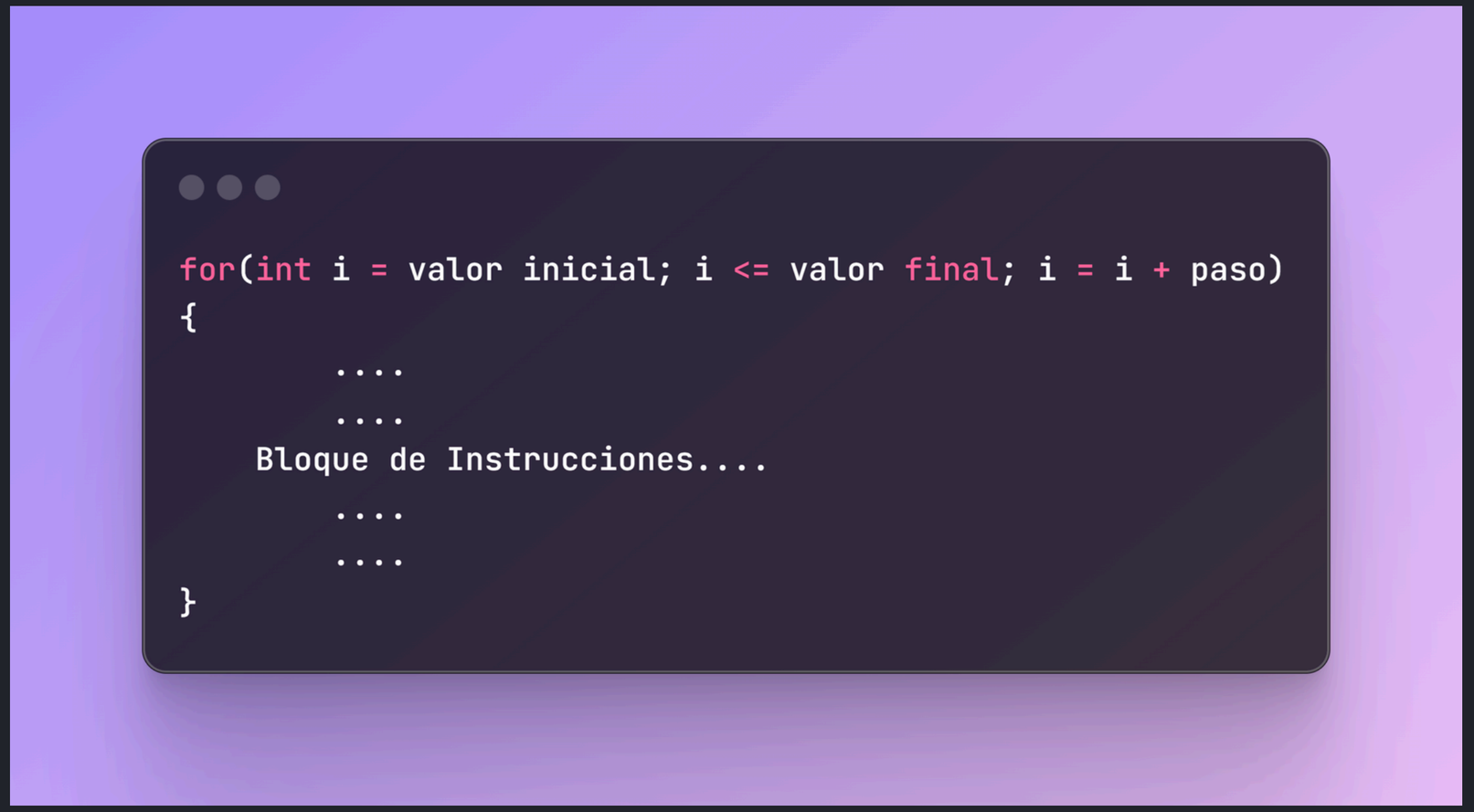
}

# Representacion grafica
{



}

Sintaxis en C# {



```
for(int i = valor inicial; i <= valor final; i = i + paso)
{
        ....
        ....
    Bloque de Instrucciones....
        ....
        ....
}
```

}

Ejemplos {

```
using System;

namespace Ejemplo_1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int i=0;
            // Imprime los números del 1 al 5
            for (i = 1; i <= 5; i++)
            {
                Console.WriteLine(i);
            }
            Console.ReadKey();
        }
    }
}
```

- Inicialización: Se crea la variable de control i y se le asigna 1.
- Condición: El bucle sigue ejecutándose mientras i sea menor o igual que 5.
- Incremento: Al final de cada iteración, i aumenta en 1.

}

Ejemplos {

```
using System;

namespace Ejemplo_2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string frase = "Horas Sociales";
            int i = 0;
            int contarVocales = 0;

            // Recorremos cada carácter de la frase
            for (i = 0; i < frase.Length; i++)
            {
                char c = char.ToLower(frase[i]);   // lo pasamos a minúscula

                // ¿Es una vocal?
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
                {
                    contarVocales++;
                }
            }

            Console.WriteLine("La frase tiene {0} vocales.", contarVocales);
            Console.ReadKey();
        }
    }
}
```

- frase.Length define el número de iteraciones.
- Dentro del for, tomamos cada carácter, lo convertimos a minúscula con char.ToLower para no distinguir entre mayúsculas y minúsculas.
- Si el carácter es una de las cinco vocales, incrementamos contarVocales.

}

Ejercicios {

- Ejercicio 1: Imprime en pantalla los números del 1 al 20.
- Ejercicio 2: Muestra los números pares del 2 al 50 en una sola línea, separados por espacios.
- Ejercicio 3: Imprime una cuenta atrás desde 10 hasta 1.
- Ejercicio 4: Pide un número al usuario y muestra su tabla de multiplicar del 1 al 10.
- Ejercicio 5: Solicita un entero positivo y calcula la suma de todos los números desde 1 hasta el numero ingresado.

}

Ejercicios {

- Ejercicio 6: Pide un número entero no negativo e imprime su factorial.
- Ejercicio 7: Pide al usuario la cantidad de calificaciones que ingresará, luego solicita cada nota y muestra el promedio final.
- Ejercicio 8: Lee una palabra y construye otra cadena insertando un guion entre cada letra (ej. “hola” → “h-o-l-a”).
- Ejercicio 9: Solicita una frase y cuenta cuántas vocales contiene.
- Ejercicio 10: Solicita un entero positivo y encuentra el primer número primo mayor o igual a al numero entero.

}

<!--Bucle-->

Gracias {

<Por="Dimas Samuel"/>

}