



INSTITUTO
KHIPU

Semestre I

Sesión 29

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**FUNDAMENTOS DE
PROGRAMACIÓN**

Tema

**ESTRUCTURA DE CONTROL REPETITIVA
“PARA”**

Conceptos Basicos

La estructura repetitiva **for** es una construcción de control de flujo en muchos lenguajes de programación, incluido C#. Esta estructura se utiliza para ejecutar un bloque de código un número específico de veces. Aquí tienes una explicación detallada sobre la estructura for:

Sintaxis de la estructura for:

La sintaxis básica de la estructura for es la siguiente:

```
for (inicialización; condición; actualización)
{
    // Bloque de código que se ejecuta repetidamente
}
```

- **Inicialización:** Es una expresión o declaración que se ejecuta una vez antes de que comience el bucle. Se utiliza generalmente para inicializar la variable de control del bucle.
- **Condición:** Es una expresión booleana que se evalúa antes de cada iteración del bucle. Si la condición es verdadera, el bucle continúa; de lo contrario, el bucle termina.
- **Actualización:** Es una expresión que se ejecuta después de cada iteración del bucle. Se utiliza generalmente para actualizar la variable de control del bucle.
- **Bloque de código:** Es el conjunto de instrucciones que se ejecutan repetidamente mientras la condición se evalúe como verdadera.

Funcionamiento:

1. La inicialización se ejecuta una vez al inicio del bucle y generalmente se utiliza para inicializar la variable de control del bucle.

2. Luego, se evalúa la condición. Si la condición es verdadera, el bloque de código dentro del bucle se ejecuta; de lo contrario, el bucle se detiene y el control se transfiere fuera del bucle.
3. Después de cada iteración del bucle, se ejecuta la actualización, que generalmente se utiliza para modificar la variable de control del bucle.
4. Este proceso de evaluación de la condición, ejecución del bloque de código y actualización de la variable de control se repite hasta que la condición se evalúe como falsa.

Características clave:

- La estructura for es útil cuando se sabe de antemano cuántas veces se debe ejecutar el bloque de código.
- La variable de control del bucle se declara localmente dentro de la estructura for y solo es visible dentro del bucle.
- Se pueden utilizar múltiples inicializaciones y actualizaciones separadas por comas en la sección correspondiente de la estructura for.
- La estructura for se puede utilizar para iterar sobre arreglos, colecciones y otros tipos de datos indexados.
- Es importante asegurarse de que la condición en la estructura for eventualmente se vuelva falsa para evitar bucles infinitos.

Ejemplos

Suma de los 10 primeros números.

```
// demo de un bucle for
#include <iostream>
```

```
using namespace std;

int main() {

    int suma = 0;
    for (int n = 1; n <= 10; n++)
        suma = suma + n;
    cout << "La suma de los números 1 a 10 es "
        << suma << endl;

    return 0; }
```

Precauciones en el Uso de FOR

Un bucle for se debe construir con gran precaución, asegurándose de que la expresión de inicialización, la condición del bucle y la expresión de incremento harán que la condición del bucle se convierta en falsa en algún momento. En particular: «si el cuerpo de un bucle de conteo modifica los valores de cualquier variable implicada en la condición del bucle, entonces el número de repeticiones se puede modificar».

Esta regla anterior es importante, ya que su aplicación se considera una mala práctica de programación. Es decir, no es recomendable modificar el valor de cualquier variable de la condición del bucle dentro del cuerpo de un bucle for, ya que se pueden producir resultados imprevistos.

Por ejemplo, la ejecución de

```
int limite = 1;
for (int i = 0; i <= limite; i++)
{
    cout << i << endl;
    limite++; }
```

produce una secuencia infinita de enteros (puede terminar si el compilador tiene constantes MAXINT, con máximos valores enteros, entonces la ejecución terminará cuando i sea MAXINT y limite sea $\text{MAXINT}+1 = \text{MININT}$).

0 1 2 3 ...

ya que a cada iteración, la expresión `limite++` incrementa `limite` en 1, antes de que `i++` incremente `i`. A consecuencia de ello, la condición del bucle `i <= limite` siempre es cierta.

Otro ejemplo es el bucle

```
int limite = 1;
for (int i = 0; i <= limite; i++)
{
    cout << i << endl;
    i; }
```

que producirá infinitos ceros

0 0 0 ..

ya que en este caso la expresión `i` del cuerpo del bucle decrementa `i` en 1 antes de que se incremente la expresión `i++` de la cabecera del bucle en 1. Como resultado `i` es siempre 0 cuando el bucle se comprueba.

Ejemplo

- Imprimir los primeros 10 numeros pares

Para `i` desde 2 hasta 20 con paso 2

Escribir `i`

Fin Para

- Calcular la suma de los primeros 100 primeros enteros

suma = 0

Para i desde 1 hasta 100

 suma = suma + i

Fin Para

Escribir "La suma de los primeros 100 números enteros es: ", suma





INSTITUTO
KHIPU