TECNOLÓGICA DEL ORIENTE
INSTITUCIÓN DE EDUCACIÓN SUPERIOR

FUNDAMENTOS DE PROGRAMACIÓN

CICLO FOR

CICLO FOR:
ESTRUCTURA CLAVE PARA ITERACIONES CONTROLADAS

El ciclo for es una herramienta fundamental en programación que permite ejecutar un bloque de instrucciones un número determinado de veces. Su funcionamiento se basa en una iteración controlada, es decir, un proceso donde se utiliza una variable que se inicializa, se evalúa en cada vuelta y se actualiza automáticamente. Esto lo convierte en la opción ideal para tareas en las que ya se conoce cuántas veces debe repetirse una acción, como recorrer arreglos, generar secuencias numéricas o ejecutar cálculos repetitivos.

Ciclo for (Iteración Controlada)

El ciclo for es una estructura iterativa ampliamente utilizada en programación para repetir un bloque de código un número determinado de veces. Se le conoce como iteración controlada porque su ejecución depende de un contador o variable de control que se inicializa, evalúa y actualiza dentro de la misma estructura (Cormen et al., 2022).

Este tipo de ciclo es particularmente útil cuando se conoce de antemano la cantidad de repeticiones necesarias, como en el recorrido de arreglos, generación de secuencias numéricas y cálculos repetitivos.

Sintaxis del Ciclo for

La estructura básica del ciclo for en la mayoría de los lenguajes de programación sigue el siguiente formato:

Figura 1. Sintaxis ciclo for

```
for (inicialización; condición; actualización) {
    // Bloque de código a ejecutar
}
```

Componentes del for

1. **Inicialización:** Se declara e inicializa una variable de control (contador).
2. **Condición:** Se evalúa antes de cada iteración; si es true, el ciclo continúa, si es false, el ciclo termina.
3. **Actualización:** Se modifica la variable de control después de cada iteración.

Figura 2. Ejemplo Básico del Ciclo for:

```
int main() {
    for (int i = 1; i <= 5; i++) {
        cout << "Iteración: " << i << endl;
    }
    return 0;
}
```

Explicación:

1. Se inicializa i = 1.
2. La condición i <= 5 se evalúa antes de cada iteración.
3. Si la condición es true, se ejecuta el bloque cout.
4. Después de ejecutar el bloque, i se incrementa (i++).
5. El ciclo se repite hasta que i sea mayor que 5.

Variantes del Ciclo for

for con Paso Personalizado

La actualización del contador no siempre tiene que ser un simple incremento de 1; se puede modificar según sea necesario.

Figura 3. Ejemplo en C++ (Ciclo de 2 en 2)

```
for (int i = 0; i <= 10; i += 2) {
    cout << "Valor de i: " << i << endl;
}
```

Figura 4. Salida por consola

```
Valor de i: 0
Valor de i: 2
Valor de i: 4
Valor de i: 6
Valor de i: 8
Valor de i: 10
```

for en Orden Descendente

Es posible hacer que el ciclo for decrementa en lugar de incrementar.

Figura 5. Ejemplo en JavaScript (Ciclo Descendente)

```
for (let i = 5; i >= 1; i--) {
    console.log("Cuenta regresiva:", i);
}
```

Uso de Ciclos Anidados

Los ciclos anidados son una estructura fundamental en la programación que permite iterar sobre conjuntos de datos multidimensionales o realizar tareas repetitivas dentro de otras tareas repetitivas. Se emplean en diversas situaciones, como el procesamiento de matrices, la generación de patrones y la optimización de algoritmos de búsqueda y ordenamiento (Marzal Varó et al., 2016).

Un ciclo anidado se define cuando un bucle se encuentra dentro de otro. El ciclo externo controla la cantidad de veces que el ciclo interno debe ejecutarse completamente antes de continuar con la siguiente iteración. Esta relación permite la ejecución de operaciones más complejas con un alto grado de control sobre el flujo de la repetición (Cormen et al., 2022).

for Anidado

Los ciclos for pueden anidarse para trabajar con estructuras bidimensionales como matrices.

Figura 7. Ejemplo en Python (Imprimir Tabla de Multiplicar)

```python
for i in range(1, 4):
    for j in range(1, 4):
        print(i, "*", j, "=", i * j)
    print()
```

Figura 8. Salida del For anidado

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
```

Bibliografía

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4ª ed.). MIT Press. https://mitpress.mit.edu/books/introduction-algorithms-fourth-edition
- Marzal Varó, A., García Sevilla, P., & Gracia Luengo, I. (2016). Introducción a la programación con Python 3. Universitat Jaume I. Servei de Comunicació i Publicacions. https://elibro.net/es/lc/tecnologicadeloriente/titulos/51760