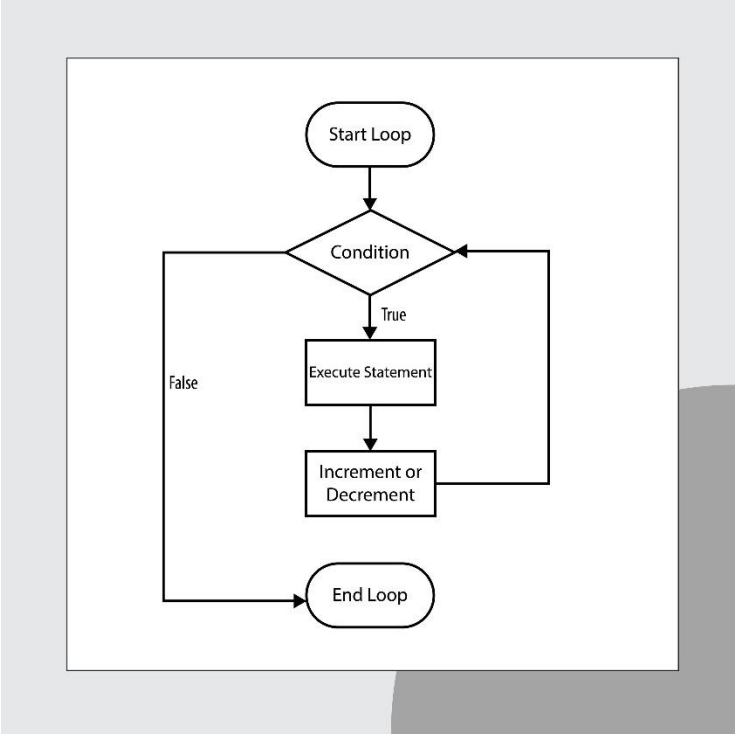
Tipos de Ciclos en C++

Ciclo while

Se utiliza cuando no sabemos exactamente cuántas veces se repetirá el bloque, ya que depende de una condición lógica. Si la condición es falsa desde el inicio, el código nunca se ejecuta.



C++

```
while (condicion) {
   // Código a repetir
}
```

Ciclo do-while

Es el hermano rebelde del while. La diferencia clave es que se ejecuta al menos una vez, porque la condición se evalúa al final del bloque.

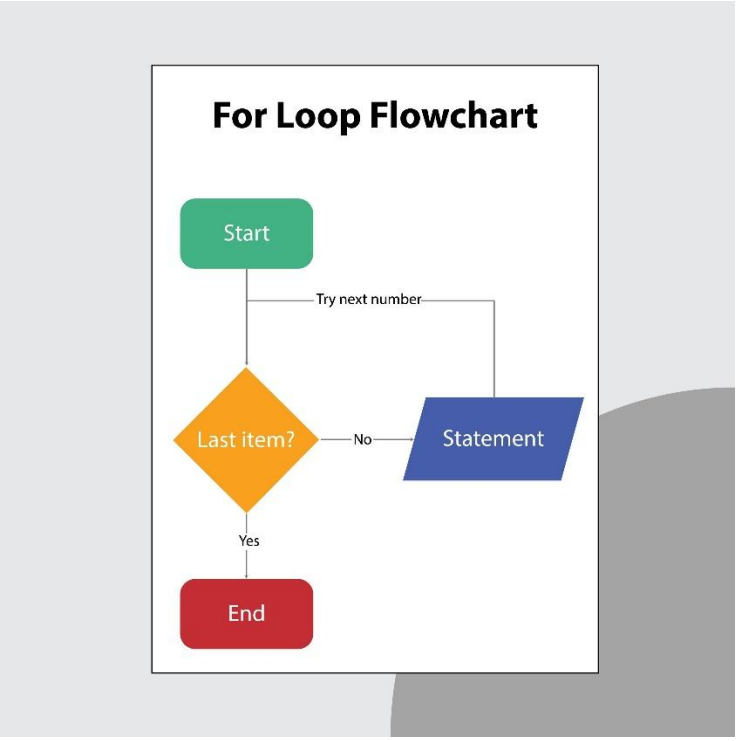
C++

```
do {
   // Código a repetir
} while (condicion);
```

Ciclo for

Es el estándar cuando sabemos el número exacto de iteraciones (por ejemplo, del 1 al 100). Agrupa la inicialización, la condición y el incremento en una sola línea.

Lcdo. Gustavo Jiménez.



C++

```
for (inicializacion; condicion; incremento) {
   // Código a repetir
}
```

Ciclo for basado en rangos (C++11 en adelante)

Ideal para recorrer contenedores (como vectores o arreglos) de forma elegante y segura.

C++

```
for (tipo elemento : contenedor) {
   // Uso de elemento
}
```

2. Ejemplos Prácticos

Ejemplo 1: Validación de entrada (do-while)

Imagina que pides una nota al usuario y debe estar entre 0 y 10. Si introduce algo mal, el programa insiste.

```
#include <iostream>
int main() {
   float nota;
   do {
      std::cout << "Introduce una nota (0-10): ";
      std::cin >> nota;
   } while (nota < 0 || nota > 10);

   std::cout << "Nota valida: " << nota << std::endl;
   return 0;
}
```

Ejemplo 2: Tabla de multiplicar (for)

Clásico ejemplo para mostrar la estructura de control de un contador.

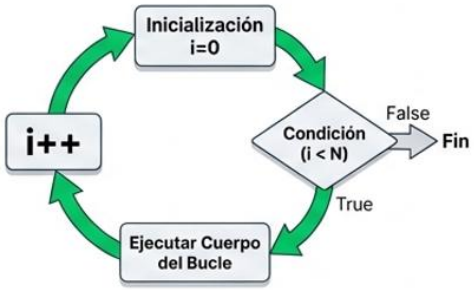
```
#include <iostream>
int main() {
   int num = 7;
   for (int i = 1; i <= 10; ++i) {
      std::cout << num << " x " << i << " = " << num * i << std::endl;
   }
   return 0;
}
```

3. Comparativa Rápida

Ciclo	¿Cuándo usarlo?	Evaluación de condición
while	Cuando la repetición depende de un evento externo.	Al inicio (Pre-test).
do-while	Menús de usuario o validaciones obligatorias.	Al final (Post-test).
for	Recorrer rangos numéricos conocidos.	Al inicio.
range-for	Procesar todos los elementos de una lista/vector.	Automática.

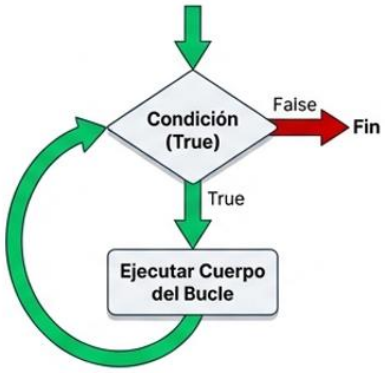
Nivel 1: El Flujo de Ejecución (Bucles)

For



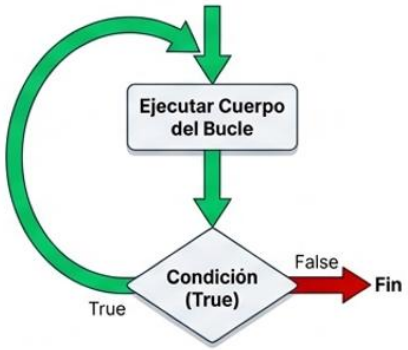
Ideal para iteraciones conocidas. conocidas.
Ejemplo: Recorrer un arreglo de tamaño N.

While



Se ejecuta mientras la condición sea verdadera.
Ejemplo: Leer input hasta el final del archivo.

Do-While



Cheat Sheet

Sintaxis

```
for(int i=0; i<N; i++) { ... }
```

⚠ **Tip:** Cuidado con el Ciclo Infinito (TLE).

NotebookLM

Bibliografía Recomendada

Para estudiar C++ en serio, estas son las fuentes de autoridad:

1. Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley. (Escrito por el creador del lenguaje).
2. Deitel, P. J., & Deitel, H. M. (2017). *C++ How to Program*. Pearson. (Excelente para pedagogía y ejemplos claros).
3. Standard C++ Foundation. isocpp.org. La referencia oficial para los estándares modernos.
4. cppreference.com. Control structures - Loops. La "biblia" técnica online para consultar sintaxis exacta.