

Principios de Programación

Bucle for

Repetición controlada con contador, recorridos definidos, acumulación, conteo y anidamiento.

Base previa

- Ya se trabajó con `if`, `if-else` y `switch`
- Ya se escribieron condiciones con operadores de comparación y lógicos
- Ya aparecieron programas con caminos alternativos

SIN ESTRUCTURA DE ITERACIÓN

```
printf("1\n");  
printf("2\n");  
printf("3\n");  
...  
printf("98\n");  
printf("99\n");  
printf("100\n");
```

Problema

- El patrón es siempre el mismo
- Cambia solo el dato que se procesa
- Si la cantidad crece, el código deja de ser manejable

Observación

Repetir instrucciones manualmente no escala y aumenta la probabilidad de error.

Definición

Un bucle permite repetir un bloque de instrucciones siguiendo una regla de control que decide cuándo continuar y cuándo terminar.

Funcionamiento

- Se establece un estado inicial
- Se ejecuta un mismo bloque varias veces
- En cada vuelta cambia alguna variable de control
- El proceso termina cuando ya no corresponde repetir

PSEWEB

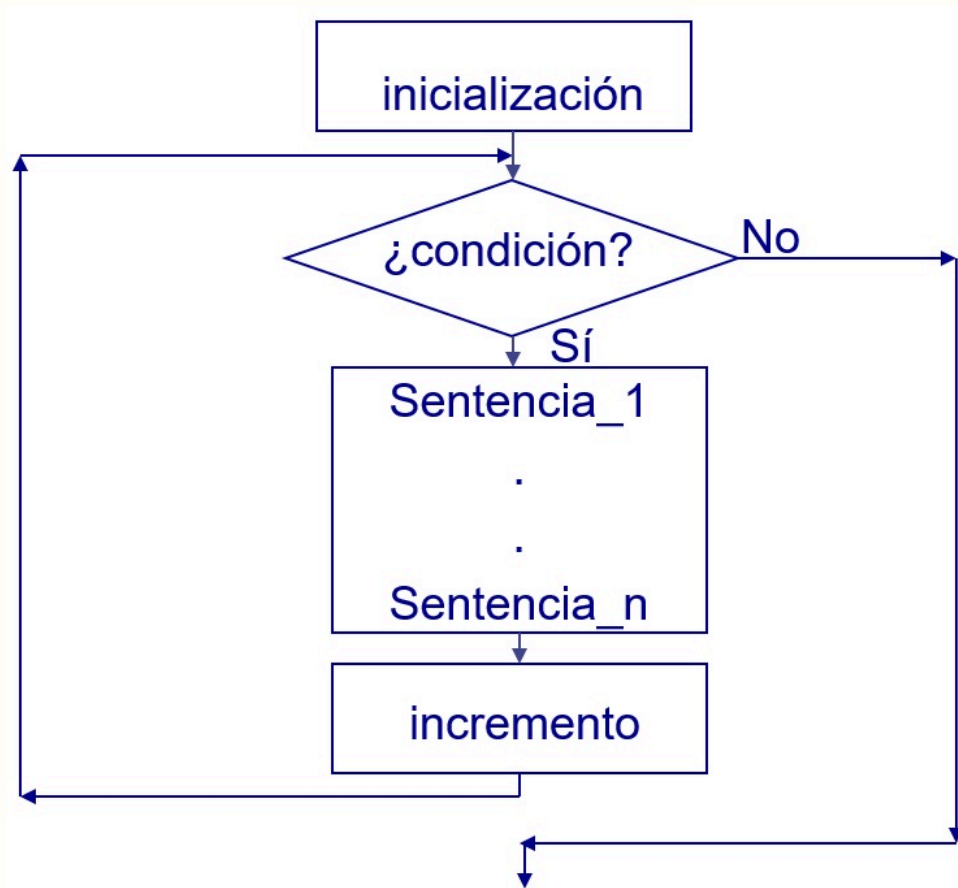
```
Algoritmo RecorrerDel1a1N
  Definir contador , N Numero
  Leer N
  Para contador = 1 Hasta N
    Escribir contador
  FinPara
FinAlgoritmo
```

SINTAXIS

```
for (inicializacion; condicion; actualizacion) {  
    // Instrucciones a repetir  
}
```

Correspondencias

- **Inicialización:** se ejecuta una sola vez al comienzo
- **Condición:** se evalúa antes de cada iteración
- **Actualización:** modifica la variable de control al final de cada vuelta



Lectura del ciclo

- La inicialización ocurre una sola vez
- La condición se consulta antes de cada iteración
- Si la condición es true, se ejecuta el bloque
- La actualización se aplica antes de volver a evaluar

CÓDIGO EN C

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }

    return 0;
}
```

SALIDA

```
1
2
3
...
10
```

Lectura

- *i* comienza en 1
- El bucle sigue mientras $i \leq 10$
- En cada iteración se aplica $i++$

PASO DE 2

```
for (int i = 0; i <= 10; i += 2) {  
    printf("%d\n", i);  
}
```

RECORRIDO INVERSO

```
for (int i = 10; i >= 0; i--) {  
    printf("%d\n", i);  
}
```

Paso distinto de 1

El cambio puede ser += 2, += 5 u otra actualización fija.

Decremento

El recorrido también puede ir hacia atrás si condición y actualización son coherentes.

CÓDIGO EN C

```
#include <stdio.h>

int main() {
    int N;
    int suma = 0;

    printf("Ingrese N: ");
    scanf("%d", &N);

    for (int i = 1; i <= N; i++) {
        suma = suma + i;
    }

    printf("La suma de 1 a %d es: %d\n", N, suma);
    return 0;
}
```

Variables

- N fija el final del recorrido
- suma es el acumulador y empieza en 0
- i toma cada valor desde 1 hasta N

CÓDIGO EN C

```
#include <stdio.h>

int main() {
    int nota;
    int aprobados = 0;

    for (int i = 1; i <= 5; i++) {
        printf("Ingrese nota %d: ", i);
        scanf("%d", &nota);

        if (nota >= 60) {
            aprobados++;
        }
    }

    printf("Cantidad de aprobados: %d\n", aprobados);
    return 0;
}
```

El recorrido asegura exactamente cinco lecturas.

aprobados solo se incrementa cuando la nota cumple la condición de aprobación.

ESQUEMA

```
for (int i = 1; i <= N; i++) {  
    for (int j = 1; j <= M; j++) {  
        // Instrucciones que usan i y j  
    }  
}
```

Definición

Un bucle anidado es un bucle dentro de otro.

Uso típico

Permite recorrer combinaciones de valores, filas y columnas o pares ordenados.

Observación

Cada iteración del bucle externo ejecuta completo el bucle interno.

CÓDIGO EN C

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 3; i++) {
        for (int j = 1; j <= 3; j++) {
            printf("%d x %d = %d\n", i, j, i * j);
        }
        printf("-----\n");
    }

    return 0;
}
```

Bucle externo

i selecciona qué tabla se está recorriendo.

Bucle interno

j recorre cada multiplicación dentro de la tabla actual.

FRAGMENTO

```
for (int i = 1; i <= 10; i = i + 2) {  
    printf("%d ", i);  
}
```

Valores de i

Iteración	i	Salida
1	1	1
2	3	3
3	5	5
4	7	7
5	9	9

Resultado

El printf se ejecuta cinco veces y la salida contiene los impares entre 1 y 9.

FRAGMENTO

```
int suma = 0;

for (int i = 1; i <= 5; i++) {
    suma = suma + i;
}

printf("Suma: %d\n", suma);
```

Traza resumida

Iteración	i	suma
Inicial	-	0
1	1	1
2	2	3
3	3	6
4	4	10
5	5	15

Resultado

El cuerpo del bucle se ejecuta cinco veces y al final se imprime Suma: 15.

Ejercicio

Escribir un programa que muestre las tablas de multiplicar desde 1 hasta N usando bucles anidados.

Ejercicio

Leer enteros hasta que el usuario ingrese 0 y calcular el promedio sin contar el valor de corte.