

# **UNIVERSIDAD NACIONAL DE LOJA**



## **TEORÍA DE LA PROGRAMACIÓN**

**UNIDAD: 2**

**TEMA:** Cuadro comparativo entre las estructuras comparativas.

**DOCENTE:** Ing. Lissette López

**ESTUDIANTE:** José Daniel  
Maldonado Rodriguez

## Cuadro comparativo entre las estructuras repetitivas

En programación, los bucles o ciclos desempeñan un papel fundamental al permitir la repetición de instrucciones o fragmentos de código hasta que una condición deje de cumplirse, momento en el cual el programa continúa con el siguiente bloque. Su funcionamiento está controlado por variables o condiciones que determinan la cantidad de repeticiones, lo que permite gestionar correctamente el flujo del programa y evitar la aparición de bucles infinitos [1].

**Tabla Comparativa de Estructuras Repetitivas**

Tipo de Bucle	Estructura	Uso
FOR	<p>Según [1] Tiene 4 parámetros:</p> <ul style="list-style-type: none"><li>-Variable de inicio: donde se comienza a ejecutar el ciclo.</li><li>-Condición del ciclo: solo se repite si la condición es verdadera.</li><li>-Modificador: modificará la variable en cada iteración.</li><li>-Cuerpo: donde se añaden las instrucciones a repetir.</li></ul> <p>En un lenguaje de programación se visualiza así:</p> <pre>for (int i = 0; i &lt; n; i++) {     // Instrucciones a ejecutar }</pre>	<p>Se usa cuando se conoce cuántas veces se repetirá el ciclo. Ideal para conteos y recorridos, como:</p> <ul style="list-style-type: none"><li>-Tablas de multiplicar.</li><li>-Leer varios datos del usuario.</li><li>-Calcular el promedio de varios estudiantes.</li><li>-Recorrer una lista de elementos.</li></ul>
WHILE	<p>Evalúa una condición lógica antes de ejecutar el bloque. Si la condición es falsa, no se ejecuta ninguna vez [2]. Tiene la siguiente estructura:</p> <pre>while (condicion) {     // Instrucciones a ejecutar }</pre>	<p>Se usa cuando no se sabe cuántas iteraciones serán necesarias; depende de una condición. Ejemplos:</p> <ul style="list-style-type: none"><li>-Para validar información hasta que sea correcta.</li><li>-Procesos que dependan del consumo de un recurso.</li></ul>
DO...WHILE	<p>Ejecuta el bloque primero y luego evalúa la condición. Garantiza al menos una ejecución [2]. Es decir, un código se repetirá, después de ser ejecutado, cuando se cumpla la condición [1].</p>	<p>Se usa cuando es necesario que el bloque se ejecute mínimo una vez. Ejemplos:</p> <ul style="list-style-type: none"><li>-Menús.</li><li>-Solicitudes de datos.</li></ul>

	Tiene la siguiente estructura:  <pre>do {     // Instrucciones a ejecutar } while (condicion);</pre>	-Para procesar elementos en orden.
--	--	------------------------------------

### Ejercicio propuesto en la plataforma megaUp

Escribe un programa que dada las calificaciones de un grupo con n alumnos calcule el promedio del grupo.

Entrada:

Un número n ( $1 \leq n \leq 1000$ ), representando el número de alumnos en la clase. Le siguen n líneas indicando las calificaciones de cada uno de los alumnos del grupo. Las calificaciones son números reales entre 0 y 10 con un sólo decimal.

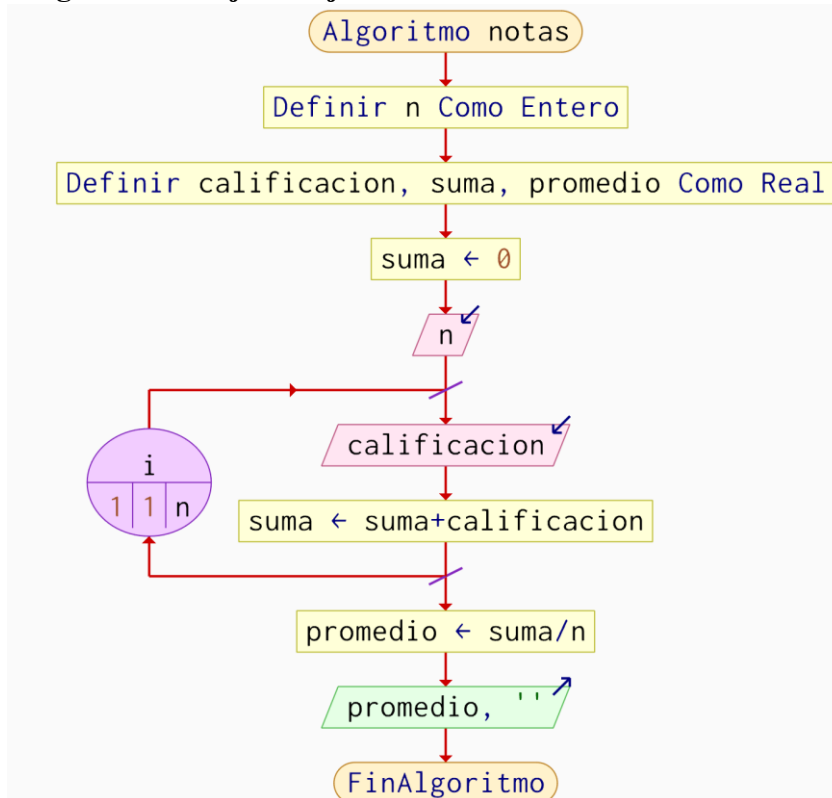
Salida:

El promedio del grupo con una precisión de 2 decimales.

### Desarrollo:

- Se elaboró un diagrama de flujo para representar visualmente la secuencia lógica del programa.
- En la herramienta Visual Studio Code, se declararon las variables necesarias: n (contador de estudiantes), nota (nota ingresada para cada estudiante), suma (acumulador de todas las notas) y total (cantidad total de estudiantes).
- El programa solicita al usuario ingresar el número de estudiantes. Este valor se almacena en la variable n, y posteriormente se copia en la variable total, ya que n será modificada dentro del ciclo.
- Mediante un bucle while, se solicita al usuario la nota correspondiente a cada estudiante. El ciclo se repite hasta que el contador llegue a cero, almacenando cada valor en la variable nota y sumándolo al acumulador suma.
- Finalmente, se calcula el promedio dividiendo la suma total de las notas entre el número de estudiantes y se muestra el resultado en pantalla.

### Diagrama de flujo del ejercicio:



### Implementación en el lenguaje C:

```
C promedio.c > main()
1  #include <stdio.h>
2
3  int main() {
4      int n;
5      float nota, suma = 0;
6
7      printf("Ingrese el numero de estudiantes: ");
8      scanf("%d", &n);
9
10     int total = n;
11
12     while (n-- > 0) {
13         printf("Ingrese la nota del estudiante: \n");
14         scanf("%f", &nota);
15         suma += nota;
16     }
17
18     printf("El promedio de las notas de todos los estudiantes es de: %.2f\n", suma / total);
19     return 0;
20 }
21
```

### Prueba en la terminal:

```
PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C> gcc promedio.c -o promedio
PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C> ./promedio.exe
Ingrese el numero de estudiantes: 6
Ingrese la nota del estudiante:
5
Ingrese la nota del estudiante:
9.2
Ingrese la nota del estudiante:
7.3
Ingrese la nota del estudiante:
8.5
Ingrese la nota del estudiante:
10
Ingrese la nota del estudiante:
4.9
El promedio de las notas de todos los estudiantes es de: 7.48
```

Las estructuras repetitivas constituyen un elemento esencial en la programación, ya que permiten automatizar tareas que requieren la ejecución reiterada de instrucciones, optimizando así el tiempo, los recursos y la eficiencia de un programa. Su correcto uso facilita la resolución de problemas que involucran procesos cíclicos, como la lectura continua de datos, el procesamiento de listas extensas o la realización de cálculos acumulativos. A través de su aplicación en ejercicios prácticos —como el cálculo del promedio de calificaciones de un grupo de alumnos— se evidencia cómo los ciclos `for`, `while` y `do...while` ofrecen soluciones según la naturaleza de cada problema. Comprender su funcionamiento y aplicarlos adecuadamente no solo mejora la calidad del código, sino que también fortalece la capacidad de diseñar soluciones computacionales mas complejas.

### Referencias:

[1] R. Maldonado, «¿Qué son los ciclos de programación o bucles y qué tipos hay? [2025]». Accedido: 4 de diciembre de 2025. [En línea]. Disponible en: <https://keepcoding.io/blog/bucles-o-ciclos-de-programacion/>

[2] «Qué es un bucle en programación y sus tipos principales». Accedido: 4 de diciembre de 2025. [En línea]. Disponible en: <https://tecnologia.euroinnova.com/bucle>

### Declaración del uso de IA:

Para la elaboración del presente informe se utilizó el apoyo de herramientas de inteligencia artificial, específicamente ChatGPT, con el fin de mejorar la redacción, obtener explicaciones técnicas y optimizar la presentación del contenido.