



**Asignatura:** Teoría de la Programación - Unidad 2

**Estudiante:** Emerson Chamba Galarza

**Docente:** Ing. Lissette Geoconda López Faicán

**Unidad:** 2

**Tema:** Cuadro comparativo entre las estructuras repetitivas

### Tabla Comparativa:

Tipo de estructura	Estructura (Sintaxis genera)	Uso / Cuándo se utiliza
For	for (inicialización; condición; actualización) { instrucciones; }	Se usa cuando <b>se conoce el número exacto de iteraciones</b> o cuando hay un contador que cambia en forma controlada. Ideal para recorrer rangos, arreglos o ejecutar procesos repetidos determinadas veces.
While	while (condición) { instrucciones; }	Se usa cuando <b>no se sabe cuántas veces se repetirá el ciclo</b> y depende estrictamente de una condición inicial. Primero evalúa, luego ejecuta. Puede ejecutarse cero veces si la condición no se cumple.
Do... while	do { instrucciones; } while (condición);	Se usa cuando <b>necesariamente debe ejecutarse al menos una vez</b> antes de evaluar la condición. Ideal para menús, lectura de datos o procesos obligatorios iniciales.

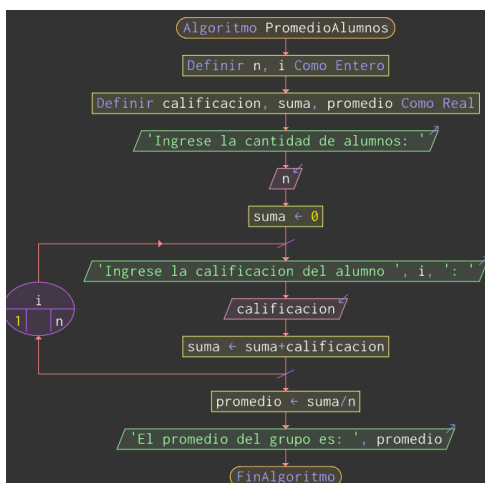
### Ejercicio en Omega Up

El ejercicio consiste en desarrollar un programa que reciba el número N de alumnos en un grupo y posteriormente las N calificaciones de cada uno de ellos. El objetivo es calcular el promedio general del grupo utilizando las notas ingresadas.

Las calificaciones son números reales entre 0 y 10, y pueden incluir un decimal. La salida debe mostrar el promedio con exactamente 2 decimales, lo cual implica formatear la impresión final.

Este ejercicio integra estructuras repetitivas (para leer N calificaciones) y la acumulación de valores para realizar el cálculo final.

### Diagrama de Flujo:



## Codificación en C:

```
#include <stdio.h>

int main() {

    int n;
    double calificacion, suma = 0, promedio;

    printf("Ingrese la cantidad de alumnos: ");
    scanf("%d", &n);

    suma = 0;

    for (int i = 0; i < n; i++) {
        printf("Ingrese la calificacion del alumno %d: ", i + 1);
        scanf("%lf", &calificacion);
        suma += calificacion;
    }

    promedio = suma / n;

    printf("El promedio del grupo es: %.2f\n", promedio);

    return 0;
}
```

## Conclusiones

Las estructuras repetitivas constituyen uno de los pilares fundamentales de la programación, ya que permiten automatizar tareas, reducir la redundancia de código y controlar el flujo de ejecución de manera eficiente. Comprender la diferencia entre ciclos definidos y no definidos es esencial para resolver problemas reales donde la cantidad de repeticiones puede variar según la situación.

El uso adecuado de estructuras como for, while y do...while facilita la construcción de algoritmos más robustos, dinámicos y escalables. Además, dominar estos ciclos permite al programador crear programas que interactúan con el usuario, procesan datos en volumen y toman decisiones iterativas. En síntesis, el aprendizaje de estas estructuras es clave para desarrollar pensamiento lógico-computacional y avanzar hacia algoritmos más complejos.

## Bibliografía

[1] M. Goin, Caminando junto al Lenguaje C. Río Negro, Argentina: Editorial

UNRN, 2022. [Online]. Available:

[https://editorial.unrn.edu.ar/index.php/catalogo/346/view\\_bl/62/lecturas-de-catedra/26/caminando-junto-al-lenguajec?tab=getmybooksTab&is\\_show\\_data=1](https://editorial.unrn.edu.ar/index.php/catalogo/346/view_bl/62/lecturas-de-catedra/26/caminando-junto-al-lenguajec?tab=getmybooksTab&is_show_data=1)

[2] J. E. Guerra Salazar, M. V. Ramos Valencia, and G. E. Vallejo Vallejo,

Programando en C desde la práctica: problemas resueltos. Puerto Madero:

Puerto Madero Editorial, 2023. [Online]. Available:

<https://dialnet.unirioja.es/servlet/libro?codigo=933288>

## Declaración de Uso de la IA

Para la elaboración de este informe se empleó la asistencia de una herramienta de inteligencia artificial (ChatGPT) exclusivamente con fines educativos, para apoyar en la redacción, organización y generación de ejemplos. El contenido ha sido revisado, comprendido y adaptado por la estudiante.