



Universidad
Nacional
de Loja

FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES
Carrera de Ingeniería en Sistemas / Carrera Computación

Universidad Nacional de Loja

Asignatura:

Teoría de la Programación

Ciclo / Semestre: 1er

Paralelo: A

Unidad 2

Tema:

Cuadro comparativo entre las estructuras repetitivas.

Docente:

Ing. Lissette López

Estudiante:

Selena Castillo

Período Académico:

Septiembre - Febrero 2026

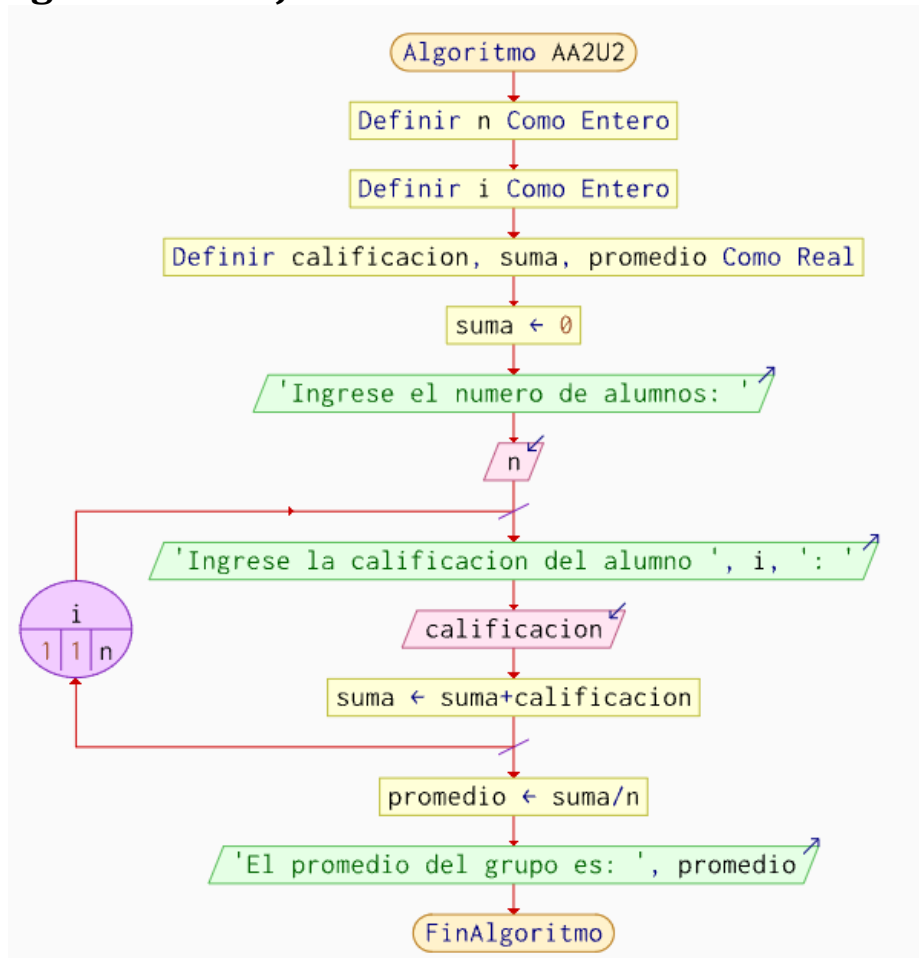
• **Tabla comparativa: Estructuras repetitivas**

	Tipo	Sintaxis general	¿Cuándo se usa?	Características
for	Ciclo con contador	<pre>for (inicio; condición; actualización) { instrucciones; }</pre>	Cuando sabemos exactamente cuántas veces queremos repetir una acción. Ideal para recorrer arreglos, contar iteraciones o ejecutar acciones con un número definido de repeticiones.	<ul style="list-style-type: none"> ○ La inicialización, condición y actualización están en una sola línea. ○ Fácil de leer al controlar el contador en un solo lugar. ○ Si la condición no se cumple desde el inicio, no entra al ciclo.
while	Ciclo de condición	<pre>while (condición) { instrucciones; }</pre>	Cuando no sabemos cuántas veces se repetirá el proceso y depende de una condición que puede cambiar internamente. Ideal para validación de datos y procesos con condición incierta.	<ul style="list-style-type: none"> ○ Evalúa la condición antes de entrar al ciclo. ○ Puede no ejecutarse nada si la condición es falsa al inicio. ○ Control más flexible, pero puede generar ciclos infinitos si no se actualiza la condición.
do...while	Ciclo de ejecución mínima	<pre>do { instrucciones; } while (condición);</pre>	Cuando necesitamos que el código se ejecute al menos una vez antes de verificar la condición. Útil en menús, lecturas iniciales y procesos que requieren una primera ejecución obligatoria.	<ul style="list-style-type: none"> ○ La condición se evalúa después de ejecutar el bloque. ○ Garantiza una ejecución mínima del ciclo. ○ Similar a while, pero la validación es posterior.

- **Planteamiento del ejercicio**

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

- **Diagrama de flujo**



- **Código en C**

```
#include <stdio.h>

int main() {
    int n;
    double calificacion, suma = 0;

    // Pedir número de alumnos
    printf("Ingrese el numero de alumnos: ");
    scanf("%d", &n);
```

```
// Pedir calificaciones
for (int i = 0; i < n; i++) {
    printf("Ingrese la calificacion del alumno %d con un
    solo decimal: ", i + 1);
    scanf("%lf", &calificacion);
    suma += calificacion;
}

// Calcular promedio
double promedio = suma / n;

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

return 0;
}
```

• Conclusiones

- Los bucles facilitan repetir acciones sin escribir el mismo código varias veces, lo que hace que los programas sean más ordenados, cortos y fáciles de entender.
- Cada tipo de bucle tiene su propio uso según la situación
- Aprender a usar bucles ayuda a resolver problemas más rápido y de manera más eficiente.

• Bibliografía

- [1] "Estructuras repetitivas," *Programación Galanista*, Jan. 22, 2013.
<https://galanistaprogramando.wordpress.com/estructuras-repetitivas/>
- [2] J. A. S. Gálvez, "Estructuras de Repetición," *Unidades De Apoyo Para El Aprendizaje*
- CUAED - UNAM. https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/3086/mod_resource/content/1/UAPA-Estructuras-Repeticion/index.html