



Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Computación

Asignatura: Teoría de programación

Ciclo 1 - Unidad: 2

AA 2: Cuadro comparativo entre las estructuras repetitivas

Docente: Ing.

Lisette Geoconda López Faicán

Estudiantes:

Christopher Alexander Pineda Rodas

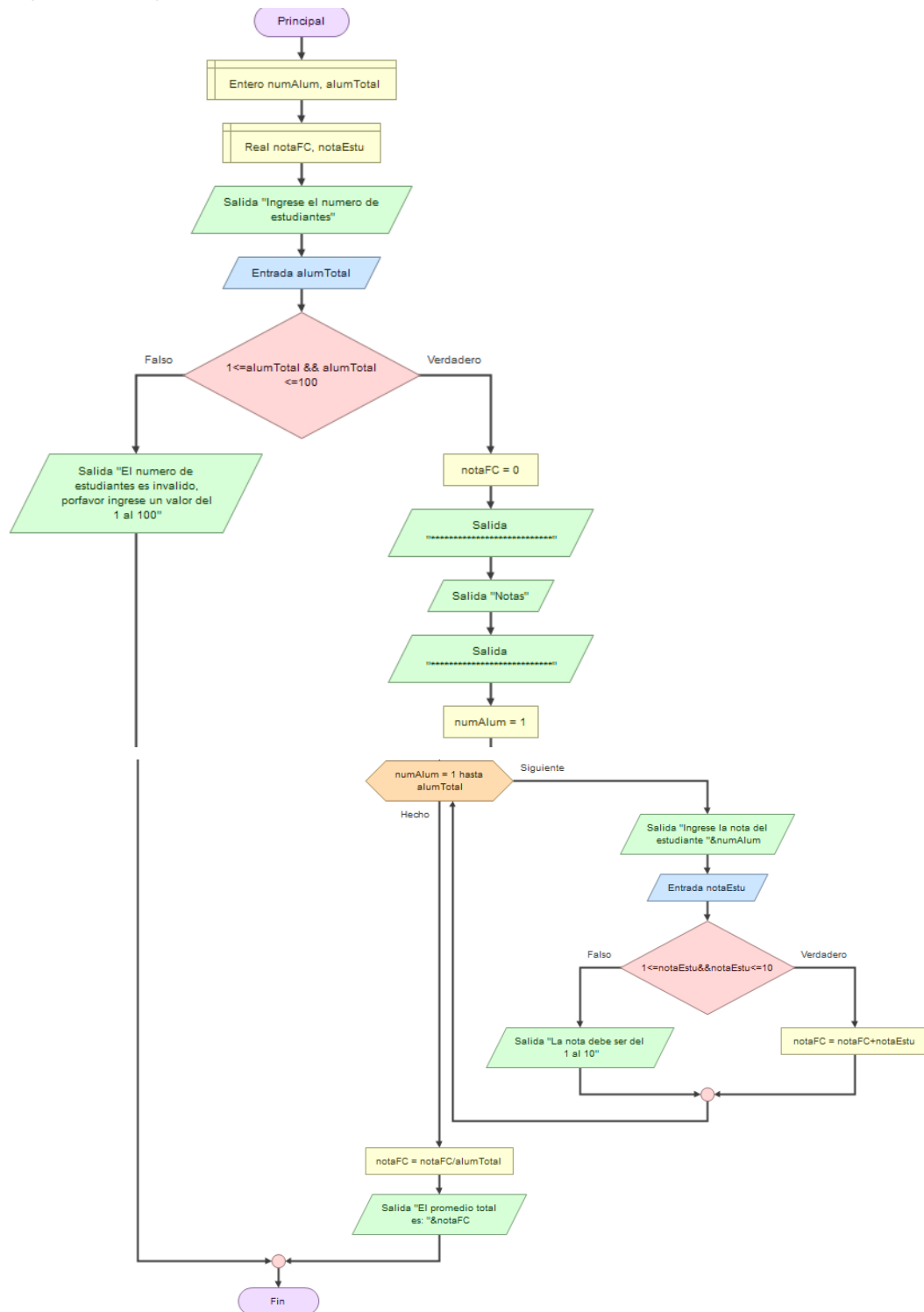
Loja – Ecuador
2025

¿Qué son las estructuras repetitivas?

Las estructuras son un conjunto de instrucciones se ejecuten un cierto número finito de veces, por ejemplo, escribir algo en pantalla cierta cantidad de veces, mover un objeto de un punto a otro cierta cantidad de pasos, o hacer una operación matemática cierta cantidad de veces. Se les llama bucle o ciclo a todo proceso que se repite cierto número de veces dentro de un pseudocódigo o un programa y las estructuras repetitivas nos permiten hacerlo de forma sencilla.

Tipo de Estructura	Estructura (Sintaxis Genérica)	Uso Principal
for (Bucle Determinado)	for (inicialización; condición; actualización) { // Bloque de código a repetir }	Se utiliza idealmente cuando se conoce de antemano el número exacto de repeticiones que se van a realizar, o cuando se necesita iterar sobre una secuencia (como un arreglo o lista). Es el bucle más compacto para la iteración controlada.
while (Bucle Condicional)	while (condición) { // Bloque de código a repetir }	Se utiliza cuando el número de repeticiones es indeterminado y depende de que se cumpla una condición específica. El bucle se repite <i>mientras</i> la condición sea verdadera. La condición se evalúa antes de la primera ejecución.
do...while (Bucle Condicional con Garantía)	do { // Bloque de código a repetir } while (condición);	Similar a while , se usa cuando el número de repeticiones es indeterminado y depende de una condición. La característica clave es que el bloque de código se ejecuta al menos una vez , ya que la condición se evalúa después de la primera ejecución.

Diagrama de Flujo:



Codigo en C de OmegaUp:

```
code cases
Promedio-UP.c
Tamaño de fuente 12px
1 #include <stdio.h>
2 #include <stdint.h>
3
4 int main() {
5     int numAlum, alumTotal;
6     double notaFC, notasEstu;
7
8     printf("Ingrese la cantidad de alumnos del ciclo\n");
9     if (scanf("%i", &alumTotal) != 1) {
10         printf("Error al leer la cantidad de alumnos\n");
11         return 1;
12     }
13
14     if (alumTotal>1 && alumTotal<100 ) {
15         printf("*****\n");
16         printf("Notas\n");
17         printf("*****\n");
18         for(numAlum=1;numAlum<=alumTotal; numAlum++){
19             do{
20
21                 printf("Ingrese la nota del alumno %i\n", numAlum);
22                 if (scanf("%lf", &notasEstu) != 1) {
23                     printf("Error al leer la nota\n");
24                     return 1;
25                 }
26
27                 printf("El estudiante %i tiene %lf\n", numAlum, notasEstu);
28                 while (notasEstu>10 || notasEstu<0);
29                 notaFC=notaFC+notasEstu;
30
31             }
32             notaFC=notaFC/alumTotal;
33             printf("%.21f\n", notaFC);
34         }
35     }else{
36         printf("Ingrese un valor valido del 1 al 100\n");
37     }
38 }
39
40
41
42
43
```

Resultado

```
statement_001.out diff statement_001.err
1+Ingrese la cantidad de alumnos del ciclo
2+*****
3+Notas
4+*****
5+Ingrese la nota del alumno 1
6+El estudiante 1 tiene 9.200000
7+Ingrese la nota del alumno 2
8+El estudiante 2 tiene 7.300000
9+Ingrese la nota del alumno 3
10+El estudiante 3 tiene 8.500000
11+Ingrese la nota del alumno 4
12+El estudiante 4 tiene 10.000000
13+Ingrese la nota del alumno 5
14+El estudiante 5 tiene 4.900000
15 7.98
16+
```

Conclusión:

El uso de las estructuras repetitivas son muy importantes ya que ayudan a automatizar la clasificación de grandes cantidades de datos además de que son la razón por la existen algoritmos complejos.

Biografías:

[1] **Programación Galanista,** "Estructuras Repetitivas," <https://www.google.com/search?q=galanistaprogramando.wordpress.com>, [En línea]. Consultado: Dic. 7, 2025. Disponible en: <https://galanistaprogramando.wordpress.com/estructuras-repetitivas/>.

Uso de IA:

Prompt:

Me ayudas en como puedo evitar que me salga esto al ejecutar un codigo: Main.c: In function 'main': Main.c:9:3: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result] 9 | scanf("%i", &alumTotal); | ^~~~~~ Main.c:22:9: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result] 22 | scanf("%lf", ¬asEstu); | ^~~~~~ [2025-12-07T20:44:33Z INFO omegajail::sys] KernelSiginfo { si_signo: 17, si_errno: 0, si_code: 1, __pad: 0, si_pid: 2, si_uid: 1001, si_status: 0, si_utime: 0, si_stime: 0 }

Este es mi codigo:

```
#include <stdio.h>
#include <stdint.h>
int main() {
    int numAlum, alumTotal;
    double notaFC, notasEstu;

    printf("Ingrese la cantidad de alumnos del ciclo\n");
    scanf("%i", &alumTotal);
    if (alumTotal>1 && alumTotal<100 ) {
        printf("*****\n");
        printf("Notas\n"); printf("*****\n");
        for(numAlum=1;numAlum<=alumTotal; numAlum++){
            do{ printf("Ingrese la nota del alumno %i\n", numAlum);
                scanf("%lf", &notasEstu); }
            while (notasEstu>10 || notasEstu<0);
            notaFC=notaFC+notasEstu;
        }
        notaFC=notaFC/alumTotal;
        printf("El promedio del ciclo es de: %.2lf\n", notaFC);
    }
    else{
        printf("Ingrese un valor valido del 1 al 100\n");
    }
    return 0;
}
```

Respuesta de la IA:

Eso que te aparece **no es un error**, es solo una **advertencia (warning)** del compilador. Te explico de forma clara:

✓ ¿Por qué sale ese warning?

`scanf()` devuelve un valor indicando si la lectura fue exitosa.

El compilador te advierte que **no estás comprobando ese valor**, por eso dice:

warning: ignoring return value of 'scanf'

✓ Cómo quitar correctamente ese warning

Debes verificar el valor de retorno de `scanf()`.

Usa este código corregido:

```
#include <stdio.h>
#include <stdint.h>

int main() {
    int numAlum, alumTotal;
    double notaFC = 0.0, notasEstu;

    printf("Ingrese la cantidad de alumnos del ciclo\n");
    if (scanf("%i", &alumTotal) != 1) {
        printf("Error al leer la cantidad de alumnos\n");
        return 1;
    }

    if (alumTotal > 1 && alumTotal < 100) {
        printf("*****\n");
        printf("Notas\n");
        printf("*****\n");

        for (numAlum = 1; numAlum <= alumTotal; numAlum++) {

            do {
                printf("Ingrese la nota del alumno %i\n", numAlum);
                if (scanf("%lf", &notasEstu) != 1) {
                    printf("Error al leer la nota\n");
                    return 1;
                }
            } while (notasEstu < 0 || notasEstu > 10);
        }
    }
}
```

```
        }
    } while (notasEstu > 10 || notasEstu < 0);

    notaFC += notasEstu;
}

notaFC = notaFC / alumTotal;
printf("El promedio del ciclo es de: %.2lf\n", notaFC);

} else {
    printf("Ingrese un valor valido del 1 al 100\n");
}

return 0;
}
```

✓ Ese mensaje final que ves:

```
KernelSiginfo { ... }
```

No es error de tu código. Es solo información del **entorno donde estás compilando/ejecutando**.

En resumen

- ✓ Tu programa funciona
- ✓ Son solo advertencias
- ✓ Ya te dejé el código limpio y profesional

Si quieres, también puedo enseñarte cómo evitar que el programa se quede en bucle cuando metes letras en vez de números.