UNIVERSIDAD NACIONAL DE LOJA

Carrera de computación
2da Unidad
Ing. Lisset López
Estudiante: José Maldonado

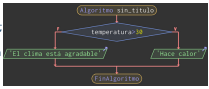
Diferencias entre los tipos de estructuras condicionales

Condicional If :
Es una estructura de control que evalúa una condición y ejecuta un bloque de código si dicha condición es verdadera [1]. Ejemplo:

```
if (edad >= 18){
    printf("Eres mayor de edad ");
}
```

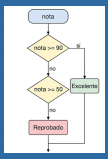
Condicional if-else:
El condicional IF-ELSE es una evolución del IF sencillo, que nos permite añadir un código a ejecutar cuando la condición es false [1]. Ejemplo:

```
if (temperatura > 30) {
    imprimir("Hace calor");
} else {
    imprimir("El clima está agradable");
}
```



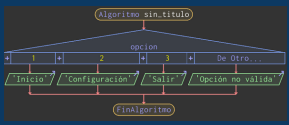
Condicional ife-else if:
Permite evaluar múltiples condiciones y ejecuta diferentes bloques de código en función del resultado de cada condición [1]. Ejemplo:

```
if (nota >= 90)
    printf("Excelente");
else if (nota >= 50)
    printf("Aprobado");
else
    printf("Reprobado");
```



Switch:
Ese permite evaluar una expresión y ejecutar diferentes bloques de código en función del valor de dicha expresión [1]. Ejemplo:

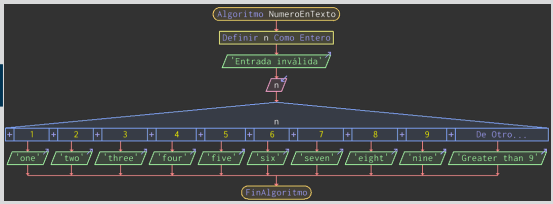
```
switch (opcion) {
    case 1:
        printf("Inicio");
        break;
    case 2:
        printf("Configuración");
        break;
    case 3:
        printf("Salir");
        break;
    default:
        printf("Opción no válida");
}
```



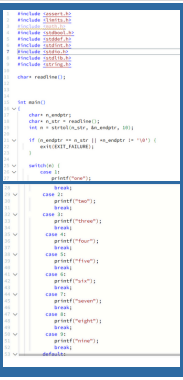
Ejercicio con estructura condicional

Dado un entero positivo que denota , haga lo siguiente:
Si 1 <= n <= 9, escriba la palabra en minúsculas correspondiente al número (p. ej., uno para 1, dos para 2, etc.).
Si n > 9, escriba Mayor que 9.

DIAGRAMA DE FLUJO



CÓDIGO EN LENGUAJE DE PROGRAMACIÓN C



```c
#include <assert.h>
#include <limits.h>
#include <stdio.h>
#include <stdbool.h>
#include <stddef.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char n_endptr;
    char* n_str = readline();
    int n = strtol(n_str, &n_endptr, 10);

    if (n_endptr == n_str || *n_endptr != '\0') {
        exit(EXIT_FAILURE);
    }

    switch(n) {
        case 1:
            printf("one");
            break;
        case 2:
            printf("two");
            break;
        case 3:
            printf("three");
            break;
        case 4:
            printf("four");
            break;
        case 5:
            printf("five");
            break;
        case 6:
            printf("six");
            break;
        case 7:
            printf("seven");
            break;
        case 8:
            printf("eight");
            break;
        case 9:
            printf("nine");
            break;
        default:
            printf("Greater than 9");
    }

    return 0;
}

char* readline() {
    size_t alloc_length = 1024;
    size_t data_length = 0;
    char* data = malloc(alloc_length);

    while (true) {
        char* cursor = data + data_length;
        char* line = fgets(cursor, alloc_length - data_length, stdin);

        if (!line) { break; }

        data_length += strlen(cursor);

        if (data_length < alloc_length - 1 || data[data_length - 1] == '\n') { break; }

        size_t new_length = alloc_length << 1;
        data = realloc(data, new_length);

        if (!data) { break; }

        alloc_length = new_length;
    }

    if (data[data_length - 1] == '\n') {
        data[data_length - 1] = '\0';
    }

    data = realloc(data, data_length);

    return data;
}
```

Line: 59 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

CONCLUSIONES

Las estructuras condicionales son esenciales en la programación porque permiten que un algoritmo tome decisiones y actúe según diferentes situaciones. Al analizar opciones como if, if–else, if–elseif y switch, se comprende que cada una ofrece un nivel distinto de control y flexibilidad para resolver problemas. Representarlas mediante diagramas de flujo y aplicarlas en código demuestra cómo estas estructuras organizan el pensamiento lógico y permiten crear programas más eficientes, adaptables y capaces de responder correctamente a las necesidades del usuario.

REFERENCIAS

[1] L. Llamas, «Qué son los condicionales», Luis Llamas. Accedido: 23 de noviembre de 2025. [En línea]. Disponible en: https://www.luisllamas.es/programacion-condicionales/

Declaración de uso de IA:
Para la elaboración de esta tarea se utilizó apoyo de herramientas de inteligencia artificial (ChatGPT) únicamente con fines de redacción, aclaración conceptual y organización de ideas.