

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

Carrera:
Computación

Asignatura:
Teoría de la
Programación

Nombre completo:
Joaquin Moscol
Castillo

Ciclo:1er Ciclo
Unidad:1ra Unidad

Tema: Estructuras algorítmicas de control



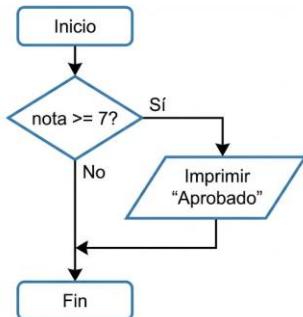
1. Estructura Condicional Simple (if)

- **Descripción:** Es la estructura de control más básica. Evalúa una condición lógica; si esta es verdadera (true), ejecuta las instrucciones dentro del bloque. Si es falsa, el programa continúa sin hacer nada.
- **Código en C:**

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

```
int main() {
    if (nota >= 7) {
        printf("Aprobado");
    }
}
```

- Diagrama de Flujo:



2. Estructura Condicional Doble (if-else)

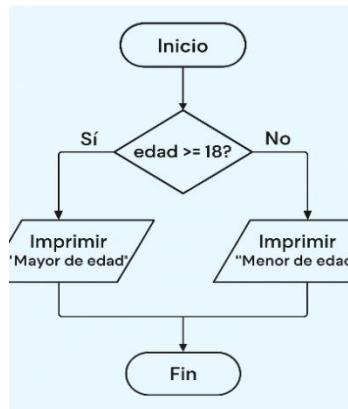
- **Descripción:** Permite bifurcar el flujo del programa en dos caminos excluyentes. Si la condición se cumple, realiza una acción; de lo contrario (else), realiza una acción diferente.

- **Código en C:**

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

```
int main() {
    if (edad >= 18) {
        printf("Mayor de edad");
    } else {
        printf("Menor de edad");
    }
}
```

- Diagrama de Flujo:





UNL

Universidad
Nacional
de Loja
1859

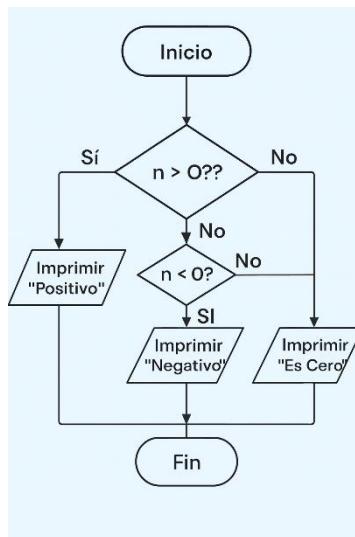
3. Estructura Condicional Múltiple o Escalonada (if-else if)

- **Descripción:** Se utiliza cuando existen más de dos posibles resultados. Evalúa condiciones de forma secuencial (en escalera). Si la primera es falsa, prueba la segunda, y así sucesivamente hasta encontrar una verdadera o llegar al else final por defecto.
- **Código en C:**

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

```
int main() {
    if (numero > 0) {
        printf("Positivo");
    } else if (numero < 0) {
        printf("Negativo");
    } else {
        printf("Cero");
    }
```

- Diagrama de Flujo:



EJERCICIO PROPUESTO EN HACKERRANK

Título del Problema: Conditional Statements in C

Planteamiento:

Dado un número entero positivo n, el programa debe realizar lo siguiente:

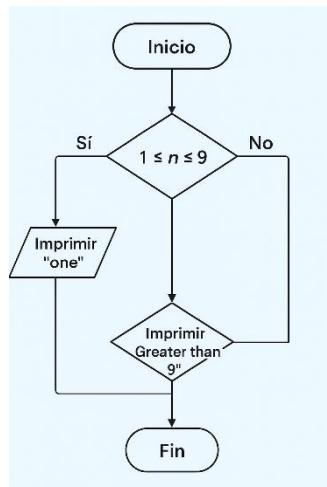
1. Si $1 \leq n \leq 9$, imprimir el nombre del número en inglés y en minúsculas (ej: "one", "two", "three"...).
2. Si $n > 9$, imprimir la frase "Greater than 9".



UNL

Universidad
Nacional
de Loja
1859

Diagrama de Flujo de la Solución:



Código Implementado (Lenguaje C):

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    getchar();

    if (n == 1) { printf("one"); }
    else if (n == 2) { printf("two"); }
    else if (n == 3) { printf("three"); }
    else if (n == 4) { printf("four"); }
    else if (n == 5) { printf("five"); }
    else if (n == 6) { printf("six"); }
    else if (n == 7) { printf("seven"); }
    else if (n == 8) { printf("eight"); }
    else if (n == 9) { printf("nine"); }
    else {
        printf("Greater than 9");
    }

    return 0;
}
```



UNL
1859

Universidad
Nacional
de Loja

INFOGRAFIA

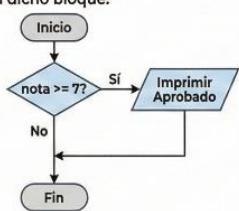
DIFERENCIAS ENTRE LOS TIPOS DE ESTRUCTURAS CONDICIONALES

1. INVESTIGACIÓN: ESTRUCTURAS CONDICIONALES

A. Estructura Condicional Simple (if)

Es la estructura de control más básica. Evalúa una condición lógica; si es verdadero, ejecuta un bloque de instrucciones. Si es falso, el programa salta dicho bloque.

```
if (nota >= 7) {
    printf("Aprobado");
}
```



B. Estructura Condicional Doble (if-else)

Permite definir dos caminos de ejecución. Si la condición es verdadera, se ejecuta el bloque **if**; si es falsa, se ejecuta el bloque **else**.

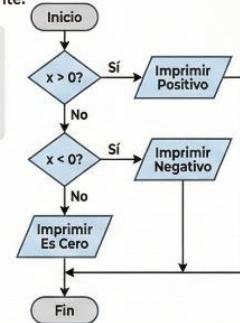
```
if (edad >= 18) {
    printf("Mayor de edad");
} else {
    printf("Menor de edad");
}
```



C. Estructura Condicional Múltiple (if-else if)

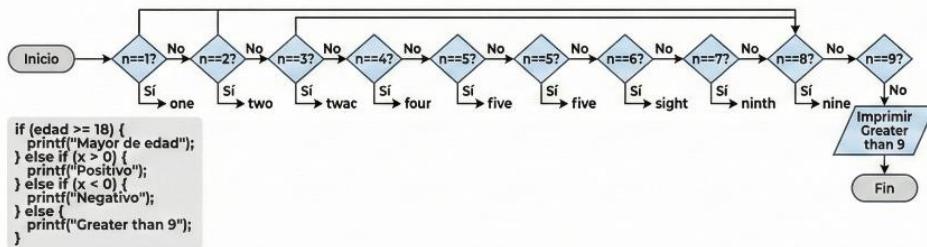
Se utiliza para evaluar múltiples condiciones en secuencia. El programa prueba la primera condición; si falla, prueba la siguiente, y así sucesivamente.

```
if (x > 0) {
    printf("Positivo");
} else if (x < 0) {
    printf("Negativo");
} else {
    printf("Es Cero");
}
```



2. EJERCICIO PRÁCTICO (HACKERRANK): Conditional Statements in C

Enunciado: Dado un entero positivo n, si $1 \leq n \leq 9$ imprimir nombre en inglés; si $n > 9$ imprimir "Greater than 9".



CONCLUSIONES

El estudio y aplicación de las estructuras condicionales es fundamental para el desarrollo de software, ya que permite dotar a los programas de capacidad de decisión lógica. A través de la resolución del ejercicio "Conditional Statements in C", se evidenció que el uso de estructuras anidadas (if-else if) es la estrategia más eficiente para manejar múltiples casos específicos de forma ordenada. A diferencia de las estructuras simples, este enfoque asegura que solo se ejecute el bloque de código correspondiente a la condición verdadera, optimizando así el flujo de ejecución del algoritmo y facilitando la legibilidad del código para futuros mantenimientos.

BIBLIOGRAFÍA

- [1] H. M. Deitel y P. J. Deitel, C How to Program, 9th ed. Pearson, 2021.
- [2] B. W. Kernighan y D. M. Ritchie, The C Programming Language, 2nd ed. Pearson, 2020.
- [3] GeeksforGeeks. "Decision Making in C (if, if..else, Nested if, if-else-if)".
<https://www.geeksforgeeks.org/decision-making-c-cpp/>

3. CONCLUSIONES

Las estructuras condicionales son el núcleo de la toma de decisiones... el uso de estructuras escalonadas (if-else if) es indispensable para manejar múltiples escenarios de forma eficiente y legible.

4. BIBLIOGRAFÍA

- [1] Ennetro Centenuns vez larón de contios. Ila proninonc nrtivo la ervid alematicos documento de 2000, nrtihcmcdxhig30.
- [2] Referencias: Dibodas ro btolog de akzurino de manisionales, salrular leamitutros en las reserco condolossefureue de Condrenso Tümland.
- [3] Lema Enmhethan, Spaffla. Soniu: etci ieneear estructura condicionales.. programai de estructura condicionales Universilal, 2019, 20-210; 385-35.