

# Estructuras Condicionales

## 1. Investigación de estructuras condicionales

### 01. If simple

- Evalúa una condición booleana [1].
- Si la condición es verdadera, ejecuta un bloque de código; si no, no hace nada y continúa.
- Es la forma más básica de decisión [2].
- Ejemplo (en C/C++):

```
if (x > 0) {  
    printf("x es positivo\n");  
}
```

### 02. If - else

- Extiende el if simple para manejar dos caminos: uno si la condición es verdadera, otro si es falsa [1].
- Muy útil cuando hay que decidir entre dos opciones mutuamente excluyentes [2].
- Ejemplo:

```
if (x % 2 == 0) {  
    printf("x es par\n");  
} else {  
    printf("x es impar\n");  
}
```

### 03. If-else if

- Permite evaluar múltiples condiciones en secuencia [1].
- Primero se verifica el if; si es falso, pasa al else if; puede haber varios else if; al final puede opcionalmente haber un else para el caso por defecto [3].
- Es útil para manejar varios casos distintos.
- Ejemplo:

```
if (nota >= 90) {  
    printf("Calificación: A\n");  
} else if (nota >= 80) {  
    printf("Calificación: B\n");  
} else if (nota >= 70) {  
    printf("Calificación: C\n");  
} else if (nota >= 60) {  
    printf("Calificación: D\n");  
} else {  
    printf("Calificación: F\n");  
}
```

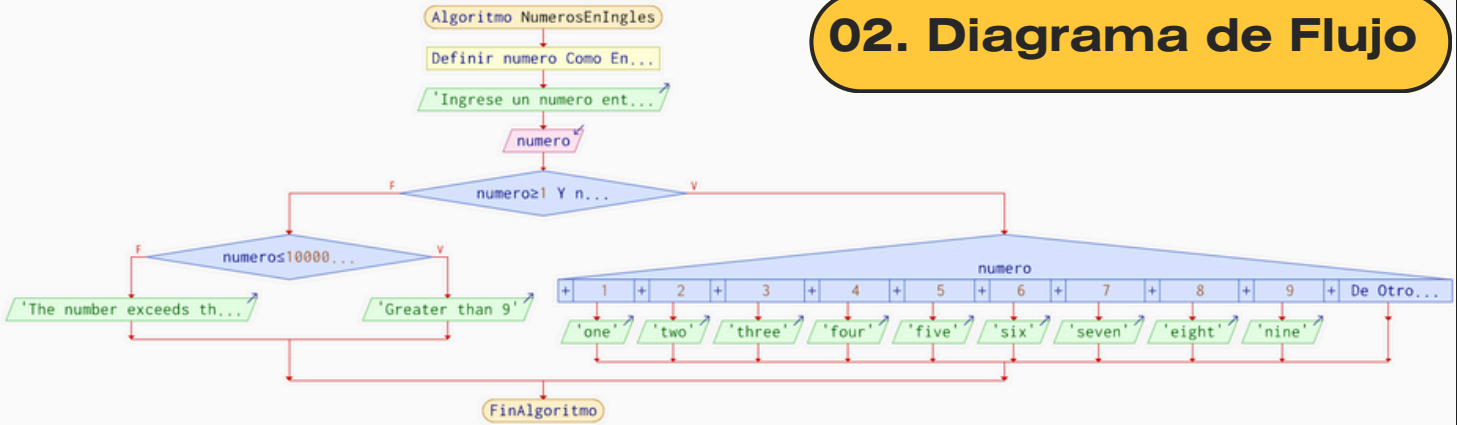
## 2. Ejercicio - de la plataforma HackerRank

Dado un entero positivo que denota  $n$ , realice lo siguiente:

- Si  $1 \leq n \leq 9$ , escriba la palabra inglés en minúsculas correspondiente al número.
- Si  $n > 9$ , escriba Mayor que 9 en inglés.
- Entrada: La primera línea contiene un solo entero.
- Restricción:  $1 \leq n \leq 10$  elevado a la 9
- Salida: Si  $1 \leq n \leq 9$ , escriba la palabra en minúsculas correspondiente al número (p. ej., uno para 1, dos para 2, etc.); de lo contrario, escriba Mayor que 9.

### 01. Planteamiento del Problema

### 02. Diagrama de Flujo



### 03. Código en C

```
1 #include <stdio.h>
2
3 int main (){
4     long long int numero;
5
6     printf("Ingrese un numero entero positivo\n");
7     scanf("%lld", &numero);
8
9     if(numero>=1 && numero<=9){
10         switch (numero)
11         {
12             case 1:
13                 printf("one");
14                 break;
15             case 2:
16                 printf("two");
17                 break;
18             case 3:
19                 printf("three");
20                 break;
21             case 4:
22                 printf("four");
23                 break;
24             case 5:
25                 printf("five");
26                 break;
27             case 6:
28                 printf("six");
29                 break;
30             case 7:
31                 printf("seven");
32                 break;
33             case 8:
34                 printf("eight");
35                 break;
36             case 9:
37                 printf("nine");
38                 break;
39         }
40     } else if (numero <= 1000000000){
41         printf("Greater than 9");
42     } else {
43         printf("The number exceeds the allowed limit (10^9)");
44     }
45     return 0;
46 }
```

## Conclusión

Se concluye que el dominio de las estructuras condicionales resulta imprescindible en la programación, pues constituyen uno de los pilares fundamentales de todo desarrollo de software. Estas estructuras permiten la toma de decisiones en función de diversos factores, lo que las convierte en un recurso esencial no solo en el ámbito computacional, sino también como reflejo de los procesos de decisión en la vida cotidiana.

## Bibliografía

[1] J. C. Rodríguez Casas, «Condicionales anidadas», 2021, Accedido: 21 de noviembre de 2025. [En línea]. Disponible en: <https://repositorio.usam.ac.cr/xmlui/handle/11506/localhost/xmlui/handle/11506/1783>

[2] «Conditional Statements in Programming | Definition, Types, Best Practices», GeeksforGeeks. 2024. Accedido: 21 de noviembre de 2025. [En línea]. Disponible en: <https://www.geeksforgeeks.org/dsa/conditional-statements-in-programming/>

[3] BillWagner, «instrucciones if y switch: seleccione una ruta de acceso de código que se va a ejecutar. - C# reference», 2024. Accedido: 21 de noviembre de 2025. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/statements/selection-statements>