

# Estructuras condicionales: IF y SWITCH

# Estructura Básica If

En C, el **if** es una estructura de control fundamental que permite ejecutar un bloque de código sólo si se cumple una condición. También se pueden agregar las sentencias **else if** y **else** para manejar múltiples casos.



# Estructura Básica If

El **if** evalúa una condición dentro de paréntesis **()**.

- Si la condición es verdadera (**1** o cualquier valor distinto de **0**), se ejecuta el bloque de código dentro de **{}**.
- Si la condición es falsa (**0**), el código dentro del **if** se ignora.

```
if (condición) {
```

```
    // Código que se ejecuta si la condición es verdadera
```

```
}
```



# Ejemplo

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

```
int main() {
```

```
    int edad = 18;
```

```
    if (edad >= 18) { // Si edad es 18 o más
```

```
        printf("Eres mayor de edad.\n");
```

```
    }
```

```
    return 0;
```

```
}
```



# If con Else

El **else** se usa cuando queremos ejecutar una acción alternativa si la condición del **if** es falsa.

```
if (condición) {  
    // Código si la condición es verdadera  
} else {  
    // Código si la condición es falsa  
}
```



# Ejemplo

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

```
int main() {
```

```
    int edad = 17;
```

```
    if (edad >= 18) { // Si edad es 18 o más
```

```
        printf("Sos mayor de edad.\n");
```

```
    }else{
```

```
        printf("Sos menor de edad.\n");
```

```
    }
```

```
    return 0;
```

```
}
```



# If con else if (muchas condiciones)

```
if (condición1) {  
    // Código si condición1 es verdadera  
} else if (condición2) {  
    // Código si condición1 es falsa pero condición2 es verdadera  
} else {  
    // Código si ninguna condición anterior es verdadera  
}
```



# Ejemplo

[Ver código en VSC](#)





# Conclusión

- if es una estructura fundamental en C para tomar decisiones en el programa.
- else if permite evaluar múltiples condiciones.
- else maneja el caso cuando ninguna condición previa es verdadera.
- Se pueden combinar con operadores lógicos (&&, ||, !) para condiciones más complejas.
- Se pueden anidar if para evaluar situaciones jerárquicas. Ver ejemplo en VSC



# Switch

El **switch** en C es una estructura de control que permite ejecutar diferentes bloques de código en función del valor de una variable. Es una alternativa más estructurada al uso de múltiples **if-else if** cuando se comparan valores discretos.



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

```
int main() {
```

```
    int opcion = 2;
```

```
    switch (opcion) {
```

```
        case 1:
```

```
            printf("Elegiste la opción 1.\n");
```

```
            break;
```

```
        case 2:
```

```
            printf("Elegiste la opción 2.\n");
```

```
            break;
```

```
        case 3:
```

```
            printf("Elegiste la opción 3.\n");
```

```
            break;
```

```
        default:
```

```
            printf("Opción no válida.\n");
```

```
        }
```

```
        return 0;
```

```
    }
```




# Cuando usar Switch?

Usa **switch** cuando:

- Comparas un solo valor contra constantes (**int**, **char**).
- Tienes muchas opciones (por ejemplo, menús de opciones).
- Quieres mejorar la legibilidad del código.

Usa **if-else** cuando:

- Comparas rangos (**x > 10**, **y <= 5**).
  - Evalúas condiciones booleanas (**&&**, **||**).
  - Necesitas expresiones más dinámicas.
- 

Característica	switch	if-else
Expresión evaluada	Solo permite valores discretos ( <code>int</code> , <code>char</code> , <code>enum</code> )	Permite cualquier expresión booleana ( <code>==</code> , <code>&lt;</code> , <code>&gt;</code> , <code>&amp;&amp;</code> ,
Legibilidad	Más limpio para múltiples opciones de valores fijos.	Más flexible, pero puede ser menos legible con muchas condiciones.
Rendimiento	Puede ser más rápido en compiladores que optimizan con <code>jump tables</code> .	Puede ser más lento si hay muchas comparaciones.
Anidamiento	No permite comparar rangos ni condiciones complejas.	Se puede anidar libremente con operadores lógicos.