

# Guión Video Condicional If

Estructuras selectivas:

- **SI-ENTONCES** (Estructura Selectiva Simple)
  1. **si** (condición) **entonces**
  2. ... // Bloque de instrucciones
  3. **fin-si**
- **SI-ENTONCES / SINO** (Estructura Selectiva Compuesta)
  1. **si** (condición) **entonces**
  2. ... // Bloque de instrucciones
  3. **sino**
  4. ... // Bloque de instrucciones
  5. **fin-si**
- **SI MÚLTIPLE** (Estructura Selectiva múltiple)

Expresiones equivalentes

- $p \rightarrow q$
- $p \Rightarrow q$
- $q \Leftarrow p$
- $q \leftarrow p$

Si p entonces q

Si sucede p entonces sucede q

p implica q

p es una condición suficiente para q:

1.

q es una condición necesaria para p:  $q \Leftarrow p$

**Introducción:**

Iniciar en la programación puede parecer agobiante en un principio, en especial sino es

Hola a todos, en este video hablaremos sobre la estructura de control if en programación. Esta estructura es muy importante ya que nos permite tomar decisiones en nuestro código y ejecutar diferentes acciones según las condiciones que se cumplan.

### Explicación de la estructura de control if:

La estructura de control if se compone de una condición y un bloque de código que se ejecuta si la condición es verdadera. Si la condición es falsa, el bloque de código no se ejecuta. La sintaxis básica de un if en la mayoría de los lenguajes de programación es la siguiente:

```
C/C++  
if (condición) {  
    // código a ejecutar si la condición es verdadera  
}
```

### Ejemplo:

Veamos un ejemplo para entenderlo mejor. Supongamos que queremos escribir un programa que determine si un número es par o impar. Para ello, podemos utilizar la estructura de control if de la siguiente manera:

```
Algoritmo mayor_edad  
    Leer edad  
    Si edad >= 18 Entonces  
        Escribir 'Es mayor de edad'  
    SiNo  
        Escribir 'Es menor de edad'  
    FinSi  
FinAlgoritmo
```

```
C/C++  
int numero = 5;  
  
if (numero % 2 == 0) {  
    // si el número es par, se ejecutará este bloque de  
    código  
    std::cout << ("El número es par") << std::endl;  
} else {  
    // si el número es impar, se ejecutará este bloque  
    de código  
    std::cout << ("El número es impar") << std::endl;  
}
```

```
}
```

En este ejemplo, estamos utilizando el operador módulo para obtener el resto de la división entre el número y 2. Si el resto es igual a cero, significa que el número es par y se ejecutará el primer bloque de código. Si el resto es distinto de cero, significa que el número es impar y se ejecutará el segundo bloque de código.

**Conclusión:**

En conclusión, la estructura de control “if” es una herramienta muy útil en programación que nos permite tomar decisiones y ejecutar diferentes acciones según las condiciones que se cumplan. Espero que este video haya sido útil para entender mejor cómo funciona la estructura de control if en programación. ¡Gracias por ver este video!

**Referencias**

[SI – Fundamentos de Programación \(wordpress.com\)](#)