



# Programación

---

ESTRUCTURAS DE CONTROL

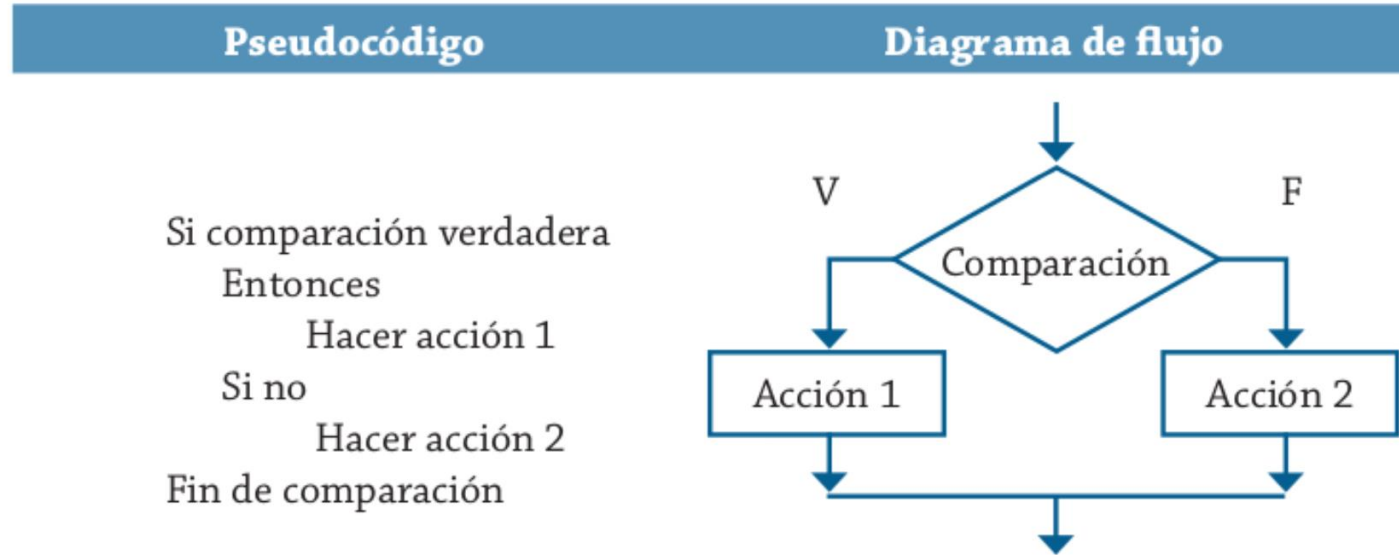
# Control de flujo

---

- **Permiten modificar el flujo de ejecución de las instrucciones**
- **Estructuras básicas**
  - *Secuencia*
    - Las instrucciones se suceden una tras otra
  - *Decisión*
    - En función de uno o varios valores, se toma una bifurcación u otra en el flujo del programa
  - *Repetición*
    - Se repite N veces una parte del flujo del programa
    - N depende de diferentes factores

# Estructuras de decisión

- **Estructura SI simple**



- **Comparación**

- Entre variables, literales, resultados de operaciones y/o llamadas a función
- Da lugar a dos posibles valores: *TRUE* o *FALSE*

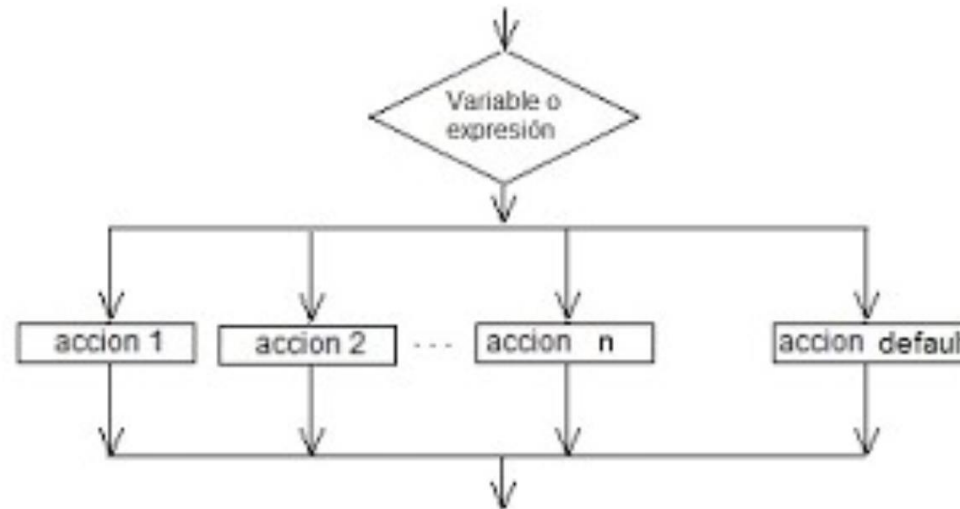
# Estructuras de control

- **Estructura SI múltiple**

## Pseudocódigo

```
segun ( selector ){  
  caso c1:  
    accion1  
  caso c2:  
    accion2  
  .  
  .  
  .  
  caso cn:  
    accionn  
  defecto:  
    acciondefecto
```

## Diagrama de flujo



- **Comparación**

- Cada caso (c1, c2...Cn) es un valor posible de la comparación
- Cada valor lleva a una acción posible
- El valor por defecto se usa en caso de no encontrar ninguna otra coincidencia

# Operadores de comparación

---

- **Igual que...**

12 igual a 12 → TRUE

- **Mayor/menor que...**

12 menor que 15 → TRUE

12 menor 12 → FALSE

- **Mayor/ menor o igual que...**

12 mayor o igual que 15 → FALSE

12 mayor o igual que 12 → TRUE

- **Distinto a ...**

12 distinto que 12 → FALSE

"Manzana" es distinto de "pera" → TRUE

Cuando comparamos cadenas de texto, el criterio para decidir qué valor es mayor, suele ser el orden alfabético **ASCII**.

# Operadores de comparación

- Operan sobre valores **TRUE** o **FALSE**
- Se rigen por el álgebra de Boole
- Tipos

Y LÓGICA (AND)		
A	B	RES.
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

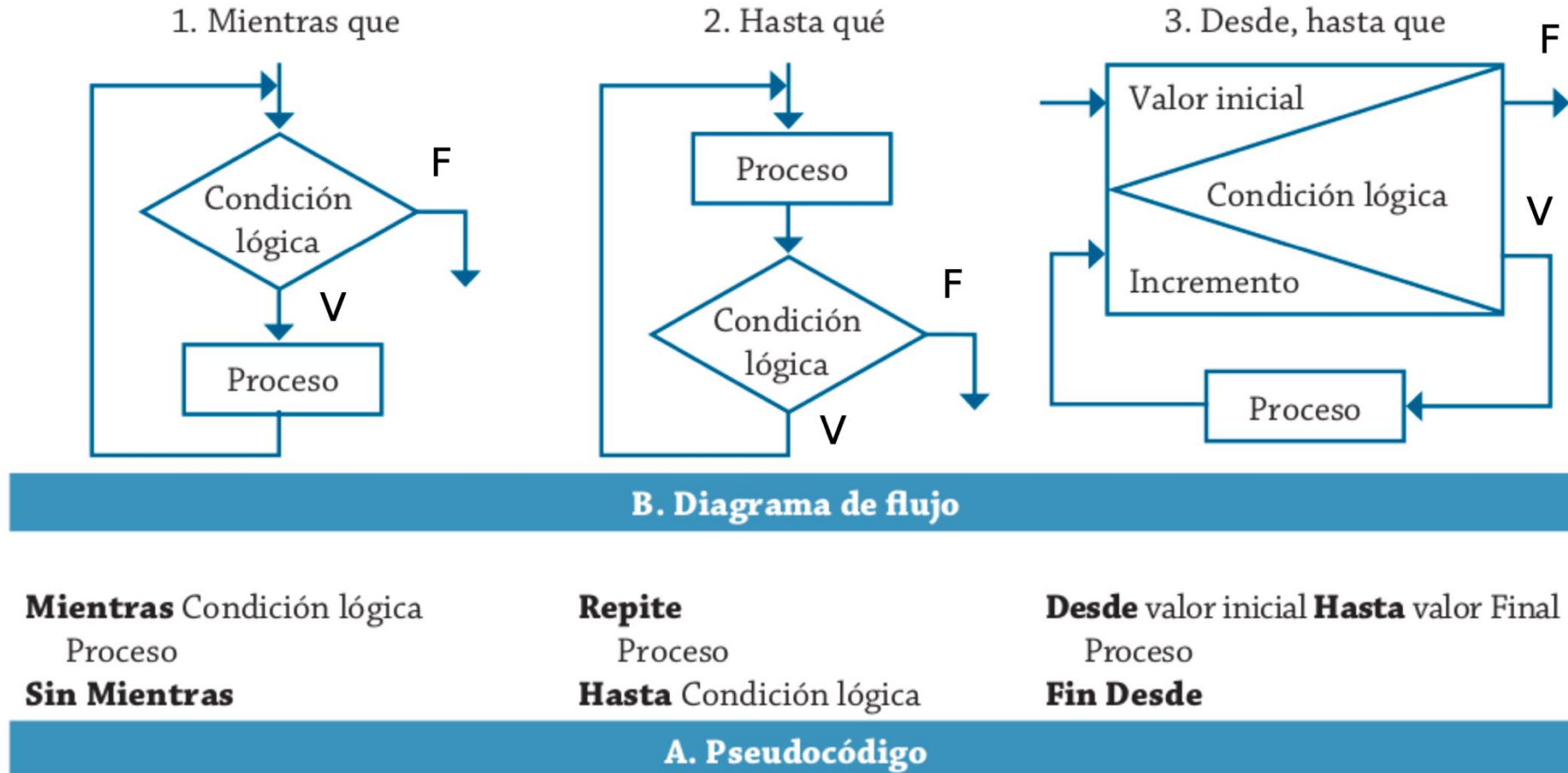
O LÓGICA (OR)		
A	B	RES.
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

NEGACIÓN LÓGICA (NOT)	
A	RES.
FALSE	TRUE
TRUE	FALSE

- Ejemplo:

*(12 mayor que 10) Y ("Hola" distinto de "adiós") → TRUE*

# Estructuras de repetición



# Estructuras de repetición

---

- **Permiten ahorra código repetitivo**
- **ITERACIÓN → cada una de las repeticiones**
- **La repetición siempre depende de una condición**
  - En cada iteración se vuelve a comprobar la condición
- ***CUIDADO:* puede dar lugar a un *bucle infinito* si la condición nunca se llega a cumplir, y producir efectos impredecibles...**
- **En muchos casos, hay implicada una variable “contador” de iteraciones en la comparación**





