

Fundamentos de Programación

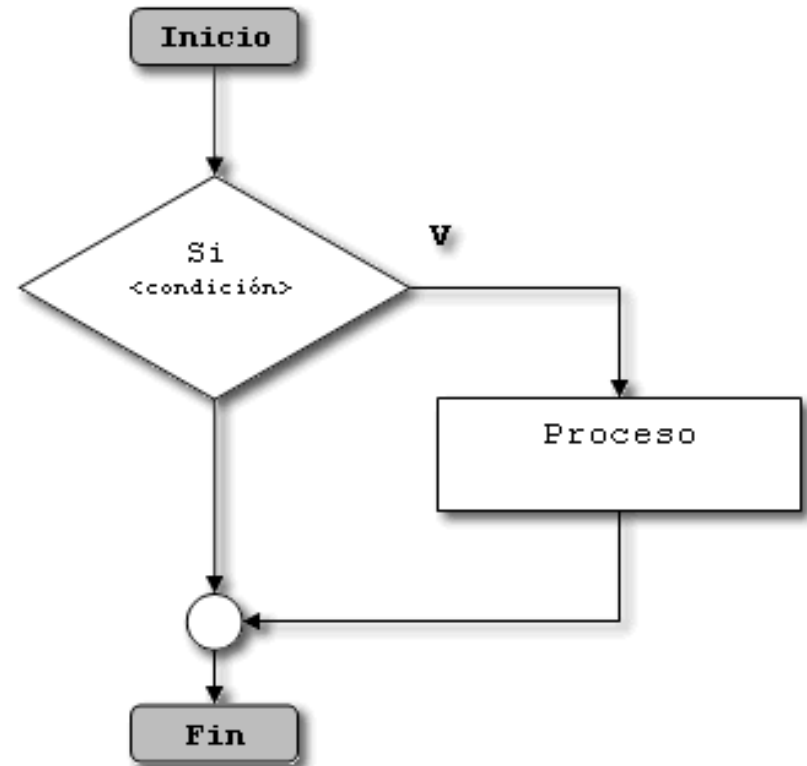
Clase 16

Agenda

- Instrucciones de control
 - Conceptos básicos
 - Operadores relacionales
 - Operadores lógicos
 - Condicionales
 - Simples
 - Encadenados
 - Anidados
 - Compuestos

Instrucciones de control

- Me permiten controlar el flujo de un programa
 - **Condicionales**
 - Ciclos o Iteraciones
 - Instrucciones de salto
- **Condicionales**
 - Cambia el comportamiento de un programa dada una condición



https://elementoscomputaciontec.wikispaces.com/file/view/Diagrama_flujo_condicionales.gif

Conceptos básicos

- Expresiones booleanas
 - Expresiones que utilizan **operadores relacionales** y cuyo resultado es verdadero o falso
- Operadores lógicos
 - Permiten unir expresiones booleanas
- Indentación
 - Uso de espacios o tabuladores para definir un bloque de instrucciones

Operadores Relacionales

- Permiten determinar el valor de verdad para una expresión.

Operador	Símbolo
Es igual que	==
Es diferente de	!=
Es menor que	<
Es mayor que	>
Es menor o igual que	<=
Es mayor o igual que	>=

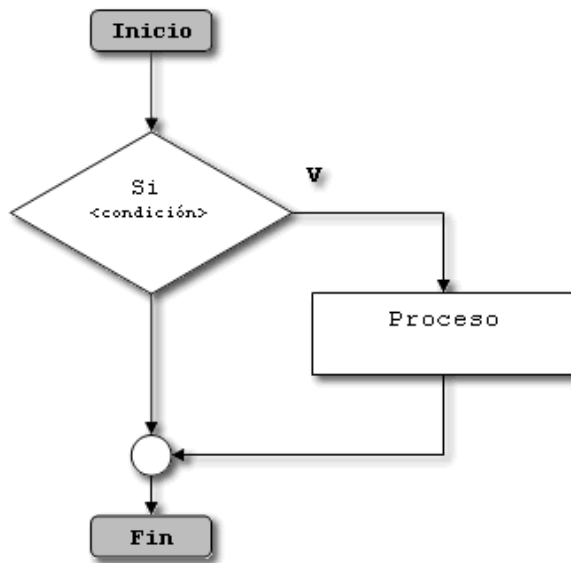
Operadores Lógicos

- Permiten describir situaciones a partir de la composición de varias expresiones relacionales o de atributos de tipo **boolean**

Operador	Símbolo
Y – lógico	&&
O - lógico	
No	!

Condicionales Simples

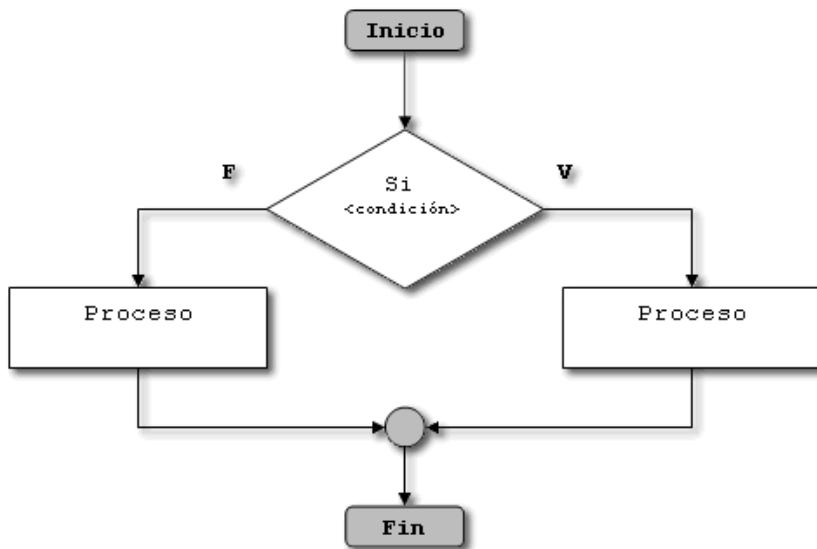
- Alternativa simple
 - Se evalúa una condición
 - Si es **verdadero** se ejecuta el bloque de instrucciones del condicional
 - Si es **falso** no se ejecuta



```
/*  
 * Bloque anterior  
 */  
if(condicion)  
{  
    /*  
     * Bloque del condicional  
     */  
}  
/*  
 * Bloque posterior  
 */
```

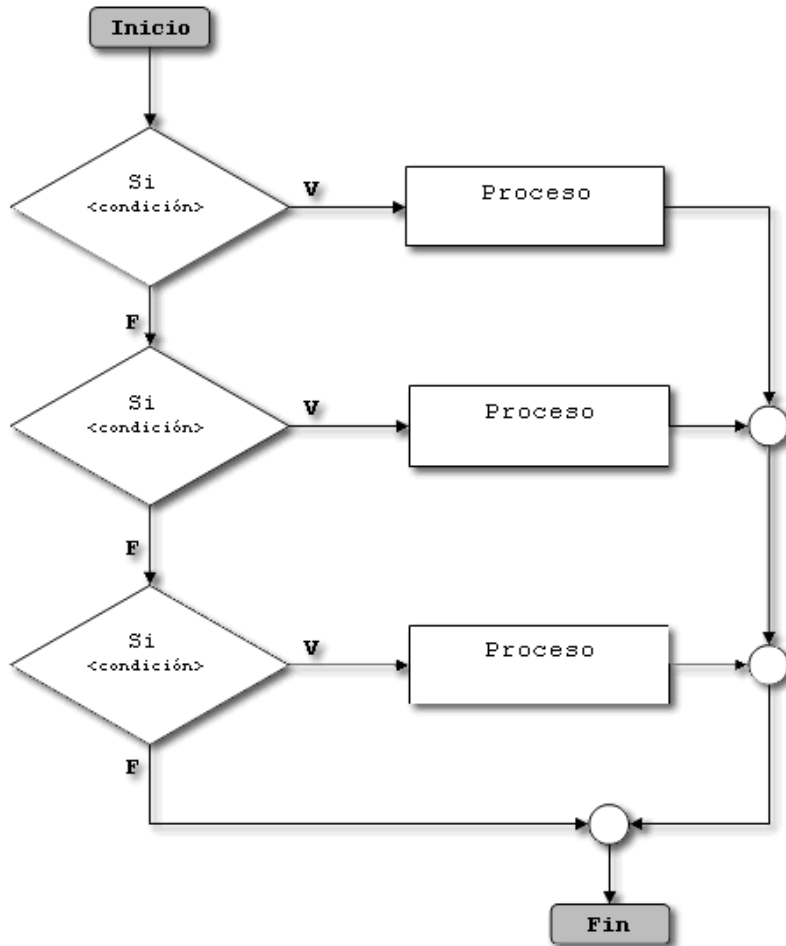
Condicionales Simples

- Alternativa doble
 - Se evalúa una condición
 - Si es **verdadero** se ejecuta un bloque de instrucciones
 - Si es **falso** se ejecuta otro bloque de instrucciones diferente



```
// Bloque anterior  
if(condicion)  
{  
    // Bloque para verdadero  
}  
else  
{  
    // Bloque para falso  
}  
// Bloque posterior
```


Condicionales encadenados



- Se evalúa una primera condición
 - Si es verdadero se ejecuta el bloque de instrucciones
 - Si es falso, se evalúa una segunda condición
 - Si es verdadero, se ejecuta el bloque de instrucciones
 - Si es falso, se evalúa una tercera condición

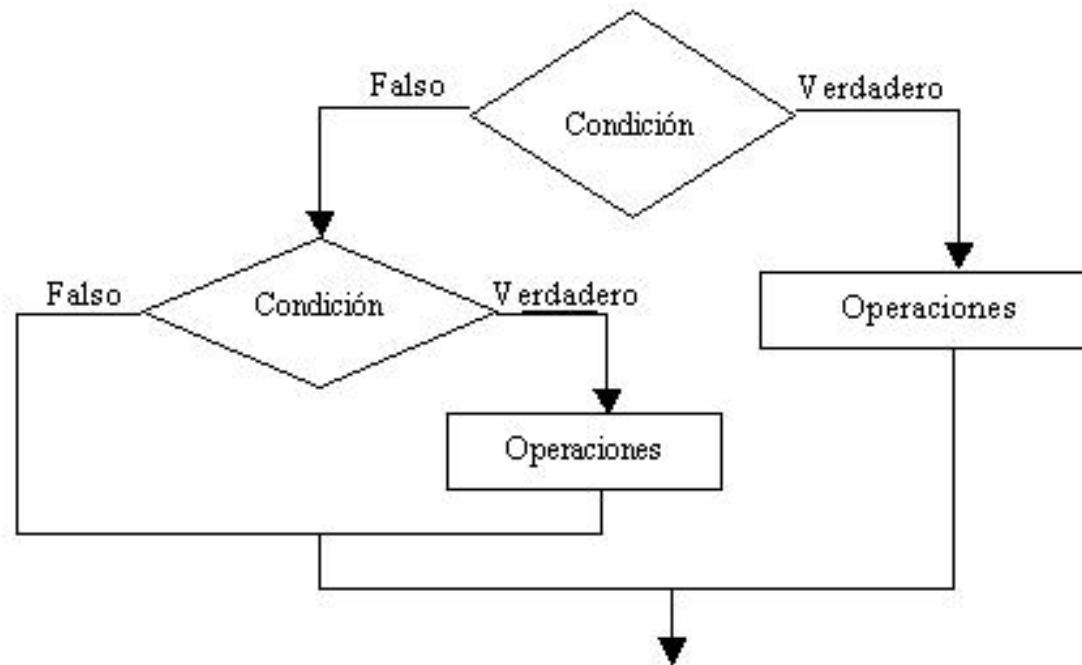
...

Condicionales encadenados

```
// Bloque anterior  
if(condicionUno)  
{  
    // Bloque para condicion uno  
}  
else if(condicionDos)  
{  
    // Bloque para condicion dos  
}  
else if(condicionTres)  
{  
    // Bloque para condicion tres  
}  
// Bloque posterior
```

Condicionales anidados

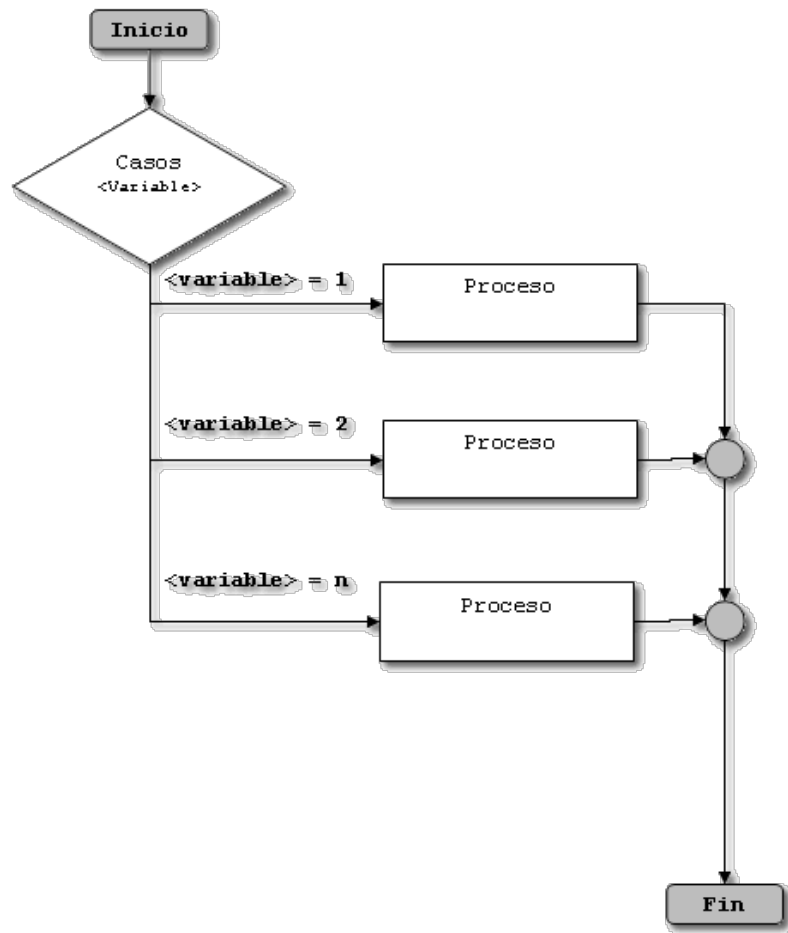
- Los bloques de instrucciones de los condicionales pueden ser a su vez otros condicionales



Condicionales anidados

```
// Bloque anterior
if(condicionUno)
{
    // Bloque para condicion uno verdadero
    if(condicionDos)
    {
        // Bloque para condicion dos verdadero
    }
    else
    {
        // Bloque para condicion dos falso
    }
}
else
{
    // Bloque para condicion uno falso
}
// Bloque posterior
```

Condicionales compuestas



- Se evalúa una expresión que puede tener varias soluciones
- Dependiendo de la solución se ejecuta un bloque de instrucciones
- Las soluciones deben ser de tipo entero

Condicionales compuestas

```
// Bloque anterior
switch(expresion)
{
    case SOLUCION_1 :
        // Bloque para condicion uno
        break;
    case SOLUCION_2 :
        // Bloque para condicion uno
        break;
    case SOLUCION_3 :
        // Bloque para condicion uno
        break;
}
// Bloque posterior
```

¿PREGUNTAS?