

# Condicionales

- ▼ Una estructura condicional, permite ejecutar una o un grupo de sentencias en función del valor de una *condición*.
- ▼ La estructura general de un condicional es SI ... ENTONCES, SI ... ENTONCES ... SINO y SEGÚN.
- ▼ Se pueden combinar múltiples condiciones mediante los operadores lógicos.
- ▼ Las condiciones siempre son evaluadas como Verdaderas o Falsas.
- ▼ Siempre debe existir un lado verdadero y puede o no existir una lado falso.
- ▼ Es considerada uno de los pilares de la programación estructurada (junto con las estructuras de repetición)

# Condición simple

- ▼ Es utilizada cuando en un algoritmo se quiere ejecutar cierta acción o conjunto de acciones en base al resultado de una condición:

```
if (condicion){  
    Accion_1;  
    Accion_2;  
    Accion_3;  
}
```

## Ejemplo

```
if (totalCompra > 5000){  
    totalCompra = totalCompra * 0.95;  
}
```

# Condición simple

- En ocasiones se desea que se ejecuten acciones tanto cuando se cumple la condición como cuando no se cumple:

```
if (condicion){  
    accion_1;  
    accion_2;  
}  
else{  
    accion_3;  
    accion_4;  
}
```

## Ejemplo

```
if (cUnidadesCompradas > 6){  
    cout << "No puede comprar más de seis tickets";  
}  
else{  
    total = cUnidadesCompradas * precioTicket;  
    cout << "Total compra: $ " << total;  
}
```

# Condición compuesta

- Una condición compuesta es la que contiene más de una condición simple a evaluar. Estas condiciones deben combinarse mediante el uso de operadores lógicos.

## Operadores lógicos

Los operadores lógicos que utilizaremos son:

Nombre	Operador	Uso	
AND	&&	if (A && B)	Si se cumple A y se cumple B
OR		if (A    B)	Si se cumple A o se cumple B
NOT	!	if (!A)	Si no se cumple A

# Operador lógico - AND

## Ejemplo

```
if (notaExamen >=4 && notaExamen <= 6){  
    cout << "El examen se encuentra aprobado pero no promocionado.";  
}
```

- La condición se evalúa como verdadera si se cumple que la nota del examen sea mayor o igual que 4 y que la nota del examen sea menor o igual que 6.

VARIABLE_1	&&	VARIABLE_2	RESULTADO
V	&&	V	V
V	&&	F	F
F	&&	V	F
F	&&	F	F

# Operador lógico - OR

## Ejemplo

```
if (primerParcial < 4 || segundoParcial < 4){  
    cout << "No puede promocionar la materia."  
}
```

- La condición se evalúa como verdadera si se cumple que la nota del primer parcial sea menor que 4 o que la nota del segundo parcial sea menor que 4.

VARIABLE_1		VARIABLE_2	RESULTADO
V		V	V
V		F	V
F		V	V
F		F	F

# Operador lógico - NOT

## Ejemplo

```
if (!(primerParcial < 4)){  
    cout << "El examen está aprobado."  
}
```

- La condición se evalúa como lo contrario al resultado de la condición de si la nota del primer parcial es menor que cuatro.

!	VARIABLE_2	RESULTADO
!	V	F
!	F	V

# Condiciones anidadas

- ▼ La estructura condicional puede anidarse, esto significa que pueden agregarse estructuras condicionales dentro de otras.

## Ejemplo

```
if (parcial >= 7){  
    cout << "El examen está promocionado."  
}  
else{  
    if(parcial >= 4){  
        cout << "El examen está aprobado."  
    }  
    else{  
        cout << "El examen está desaprobado."  
    }  
}
```



# Explicar el funcionamiento del siguiente código

```
if (parcial >= 7){  
    cout << "El examen está promocionado.";  
}  
if(parcial >= 4){  
    cout << "El examen está aprobado.";  
}  
else{  
    cout << "El examen está desaprobado.";  
}
```

# Condición múltiple

- ▼ Es utilizada cuando se evalúa una variable entre distintos casos y se ejecuta las acciones a realizar dependiendo del caso seleccionado.

```
switch (variable){  
    case 1:  
        accion_1;  
        accion_2;  
        break;  
    case 2:  
        accion_3;  
        accion_4;  
        break;  
    default:  
        accion_5;  
        break;  
}
```

# Condición múltiple

## Ejemplo con valores enteros

```
switch (categoria){  
    case 1:  
        total = cant * 5;  
        break;  
    case 2:  
        total = cant * 9.5;  
        break;  
}
```

## Ejemplo con caracteres

```
switch (formaPago){  
    case 'e':  
    case 'E':  
        total = total * 0.9;  
        break;  
    case 't':  
    case 'T':  
        total = total * 1.15;  
        break;  
}
```