

## Clase 2: Condicionales

Prof. Nicolás Torres

[nicolas.torresr@usm.cl](mailto:nicolas.torresr@usm.cl)

Ingeniería Civil Telemática

Departamento de Electrónica

Universidad Técnica Federico Santa María

## Condiciones

Las condicionales permiten controlar el flujo de los programas, logrando que se ejecute cierta parte del código cuando una condición sea verdadera y un código diferente cuando sea falsa.

Son ramificaciones que modifican la ejecución del código y permiten tomar decisiones con base en condiciones definidas previamente. Por ejemplo:

- Si está lloviendo, entonces llevo paraguas. Sino, no llevo paraguas.
- Si me saco más de un 60 en el control, entonces aprobaré. Si no lo logro, y me saco menos de un 40, reprobaré. En otro caso, me iré al global.

## Operadores Relacionales

Son operadores que **comparan valores**.

El resultado de la operación es **siempre un valor booleano** ( `True` o `False` ).

- igual a ( `==` )
- distinto a ( `!=` )
- mayor que ( `>` )
- menor que ( `<` )
- mayor o igual que ( `>=` )
- menor o igual que ( `<=` )

```
In [1]: 5 == 3
```

```
Out[1]: False
```

```
In [2]: "a" == 'A'
```

```
Out[2]: False
```

```
In [3]: 4 > 7
```

```
Out[3]: False
```

```
In [4]: 4 != 3
```

```
Out[4]: True
```

```
In [5]: 3 < 3
```

```
Out[5]: False
```

## Comparaciones de strings

Los **operadores de comparación** se pueden utilizar sobre strings.

¿Cómo se decide el orden de los caracteres?

### El código ASCII

El código ASCII asocia todos los caracteres imprimibles (letras, números y símbolos) con un número para decidir el orden.

	caracter		caracter
48	"0"	65	"A"
49	"1"	66	"B"
...	...	...	...
61	"="	97	"a"
62	">"	98	"b"
...	...	...	...

```
In [6]: # Las minúsculas son mayores que las mayúsculas. Se ubican después en el código ASCII  
"a" > "A"
```

```
Out[6]: True
```

```
In [7]: "a" < "z"
```

```
Out[7]: True
```

```
In [8]: # La t es menor que la x en el código ASCII  
"auto" < "auxilio"
```

```
Out[8]: True
```

```
In [9]: # La s es mayor que el carácter nulo en el código ASCII  
"casas" > "casa"
```

```
Out[9]: True
```

```
In [10]: # El '1' es menor que el '9' en código ASCII  
'10' < '9'
```

Out[10]: True

## Operadores lógicos

Los operadores lógicos relacionan valores booleanos para tomar decisiones y entregan siempre como resultado un valor booleano.

- conjunción lógica `and`
- disyunción lógica `or`
- negación lógica `not`

```
In [11]: ((3 + 4) < 2) or ((2**4) >= 4)
```

Out[11]: True

```
In [12]: ((3 + 4) < 2) and ((2**4) >= 4)
```

Out[12]: False

## Condicional `if`

El condicional `if` ("si") ejecuta las instrucciones solo si se cumple una condición. Si la condición es falsa, entonces las instrucciones no se realizan.

### Sintaxis:

```
if condición:
    instrucciones
```

```
In [13]: nota = int(input("Ingrese su nota: "))
```

```
if nota >= 55:
    print("Aprobado")
```

Ingrese su nota: 60  
Aprobado

## Condicional `else`

El condicional `else` ("si no"), que funciona como complemento o parte opcional a la sentencia `if`, permite ejecutar instrucciones si la condición anterior no se cumple. Este condicional tiene como prerequisite una sentencia `if` declarada previamente, no puede aparecer por sí mismo.

### Sintaxis:

```
if condición:
    instrucciones si condición es True
else:
    instrucciones si condición es False
```

```
In [14]: nota = int(input("Ingrese su nota: "))
```

```
if nota >= 55:
    print("Aprobado")
else:
    print("Reprobado")
```

Ingrese su nota: 20  
Reprobado

## Condicional elif

El condicional `elif`, también funciona como un complemento o parte opcional a la sentencia `if`, y permite agregar una nueva condición que será evaluada si la anterior no se cumple. Este condicional tiene como prerrequisito una sentencia `if` declarada previamente, no puede aparecer por sí mismo, pero puede aparecer más de una vez dentro del mismo bloque de condiciones.

### Sintaxis:

```
if condición 1:
    instrucciones si condición 1 es True
elif condición 2:
    instrucciones si condición 1 es False y condición 2 es True
else:
    instrucciones si ninguna condición es True
```

In [15]: `nota = int(input("Ingrese su nota: "))`

```
if nota >= 80:
    print("Excelente")
elif nota >= 65:
    print("Aceptable")
elif nota >= 55:
    print("Aprobado")
elif nota >= 45:
    print("Global")
else:
    print("Reprobado")
```

Ingrese su nota: 70  
Aceptable

### Sintaxis General:

```
if <condición1>: #obligatoria
    sentencias a ejecutar si <condición1> es verdadera
elif <condición2>: #opcional
    sentencias a ejecutar si <condición2> es verdadera
elif <condición3>: #opcional
    sentencias a ejecutar si <condición3> es verdadera
...
else: #opcional
    sentencias cuando ninguna condición anterior es verdadera
```

La **Indentación** después de cada instrucción indica que cosas se realizarán si se cumple su respectiva condición (consecuentemente, si **no** se cumplen las condiciones anteriores). Se utiliza el **caracter de tabulación** (TAB, Tab, ⌘).

# Condiciones anidadas

Cuando una sentencia condicional se ubica dentro de otra, se habla de condiciones "anidadas".

```
In [16]: nota = int(input("Ingrese su nota: "))
```

```
if nota >= 55:
    if nota >= 80:
        print("Excelente!!")
    else:
        print("Aprobado.")
else:
    print("Reprobado.")
```

Ingrese su nota: 70

Aprobado.