

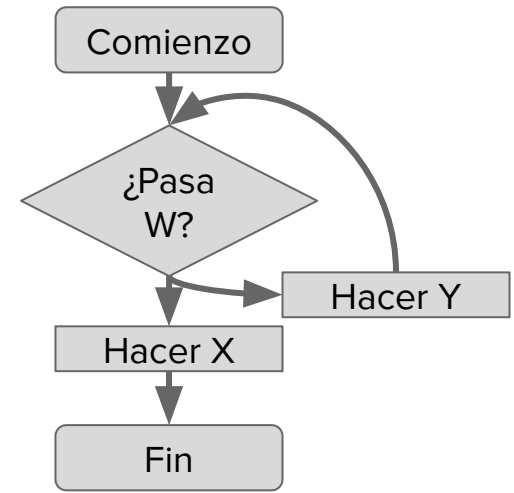
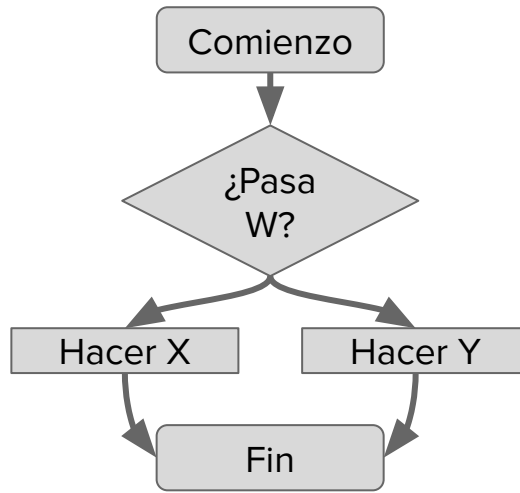
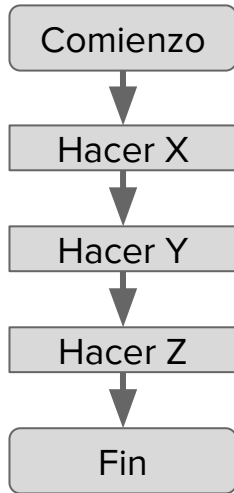
# Flujo de control



Introducción al pensamiento computacional

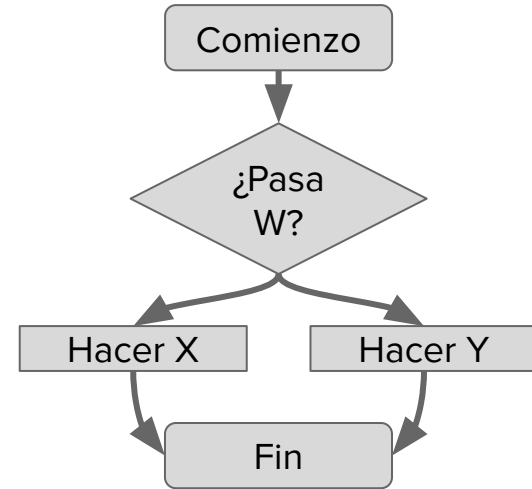
# Flujo de control

Un computador ejecuta un programa de forma secuencial, para poder modificar este comportamiento, hacemos uso de una herramienta llamada flujo de control:



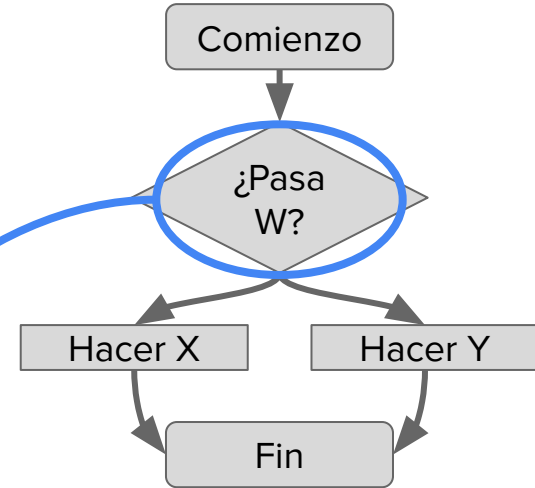
# Flujo de control: condicionales

- Es la forma más sencilla de usar un bloque de condición.
- Nos permite dividir el flujo del programa en dos (o más) flujos.
- Aun así, solo se ejecuta uno de los flujos.



# Flujo de control: condicionales

- Es la forma más sencilla de usar un bloque de condición.
- Nos permite dividir el flujo del programa en dos (o más) flujos.
- Aun así, solo se ejecuta uno de los flujos.



**¿Cómo expresamos esto en un computador?**

# Booleanos

La mayoría de lenguajes de programación cuentan con otro tipo de valores aparte de los números. Se conocen como booleanos (en honor al lógico George Boole):

## Números

- Valores fijos: 0, 1, 2, 3, ...
- Operaciones: +, -, \*, /
- Representan: Cantidades

## Booleanos

- Valores fijos: True, False
- Operaciones: and, or, not
- Representan: Afirmaciones

# Operadores booleanos: Not

El operador **not** representa la **negación** de una afirmación

x	not x
True	False
False	True

# Operadores booleanos: And

El operador **and** representa una afirmación que es cierta sólo si **ambas** afirmaciones son ciertas

x	y	x and y
True	True	True
True	False	False
False	True	False
False	False	False

# Operadores booleanos: Or

El operador **or** representa una afirmación que es cierta si **al menos una** de las afirmaciones es cierta

x	y	x or y
True	True	True
True	False	True
False	True	True
False	False	False

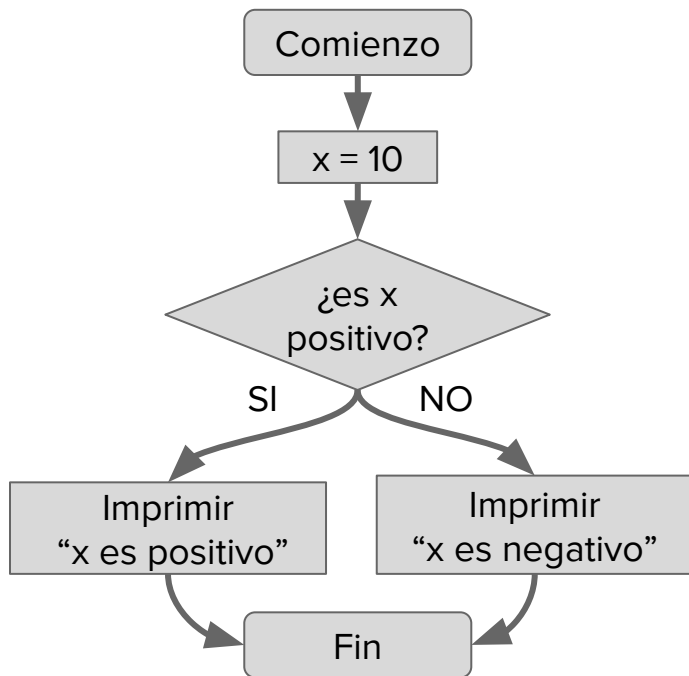


# Construyendo afirmaciones

Además de las operaciones aritméticas básicas, un computador es capaz de comparar números de distintas formas:

- x es mayor a y:  $x > y$
- x es menor a y:  $x < y$
- x es mayor o igual a y:  $x \geq y$
- x es menor o igual a y:  $x \leq y$
- x es igual a y:  $x == y$
- x es distinto a y:  $x \neq y$

# Un programa sencillo con condicionales



```
x = 10
```

```
if (x > 0):
```

```
    print(x, "es positivo")
```

```
else:
```

```
    print(x, "es negativo")
```

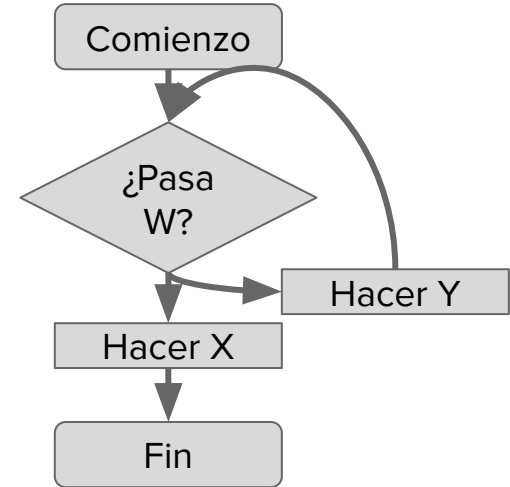
# Ejercicio

Escriban un programa que a partir de una nota entre 0 y 5 imprima si van pasando o no

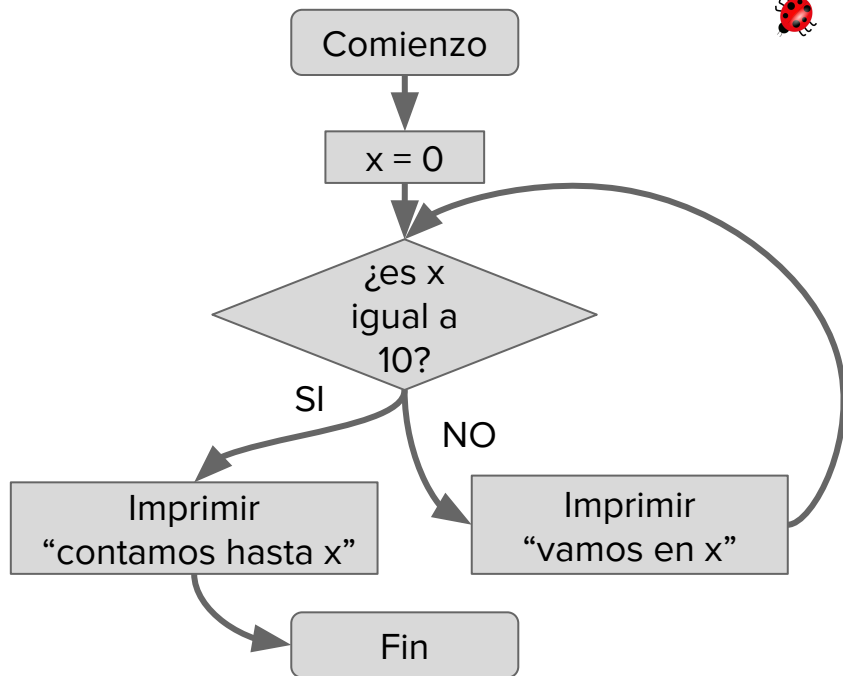
---

# Flujo de control: ciclos

Nos permite repetir una acción  
mientras se cumpla una condición



# Un programa sencillo con ciclos



```
x = 0
```

```
while (x != 10):
```

```
    print("vamos en", x)
```

```
    print("contamos hasta", x)
```

# Ejercicio

Escriban un programa que imprima  
la suma de todos los números entre  
1 y 100

---



# **Tortugas!**

# Tortugas

- Ir adelante x píxeles: `t.forward(x)`
- girar a la derecha x grados: `t.right(x)`
- girar a la izquierda x grados: `t.left(x)`
- levantar el lápiz: `t.penup()`
- bajar el lápiz: `t.pendown()`
- volver a casa: `t.home()`
- limpiar todo: `t.clear()`



# Ejercicio

Escriban un algoritmo para pintar un  
cuadrado

---