

TC1028

Pensamiento Computacional para Ingeniería

## Ciclo: While

Tecnológico de Monterrey



# Sintaxis del while

```
while expr-test:  
    instrucción_1  
    instrucción_2  
    ...  
    instrucción_N
```

Donde **expr-test** es una expresión que evalúa a **verdadero** o **falso**.



# While



- La estructura de código **while** nos permite repetir la ejecución de una secuencia de instrucciones.
- La repetición es controlada por la evaluación de una **condición, mientras esta condición sea verdadera, entonces ejecuta las instrucciones.**

# While



- **Expresión:** es cualquier expresión numérica, relacional o lógica.
- Las instrucciones dentro del **while** se ejecutan en forma repetida, en secuencia de arriba a abajo, **mientras** la **expresión** sea **verdadera** (o distinto de **0**).
- Si la **expresión** se evalúa como **falsa** en la primera ocasión, las instrucciones dentro del **while** **nunca** se ejecutan.

# While



La ejecución de la sentencia **while** sucede así:

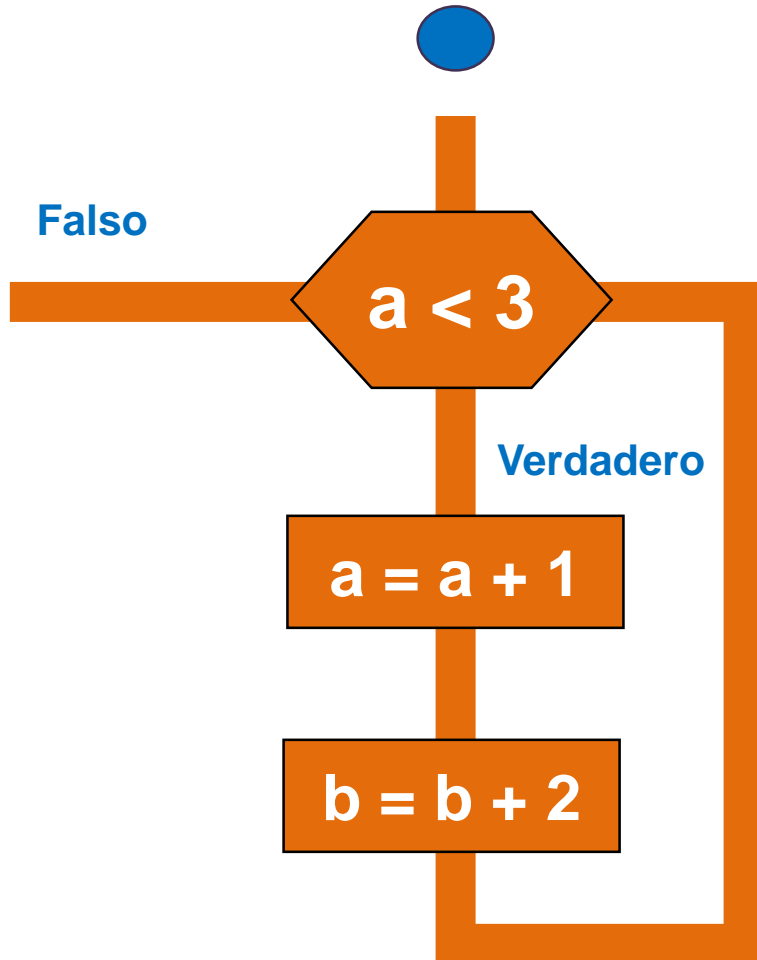
1. Se evalúa la **expresión**.
2. Si el resultado de la **expresión** es **falso**, la(s) instrucción(es) no se ejecuta y se pasa a ejecutar la siguiente instrucción en el programa.
3. Si el resultado de la **expresión** es **verdadero**, se ejecuta la(s) instrucción(es) y el proceso se repite comenzando en el **punto 1**.

# Actividad Grupal

Analizar la ejecución de la siguiente simulación.



# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

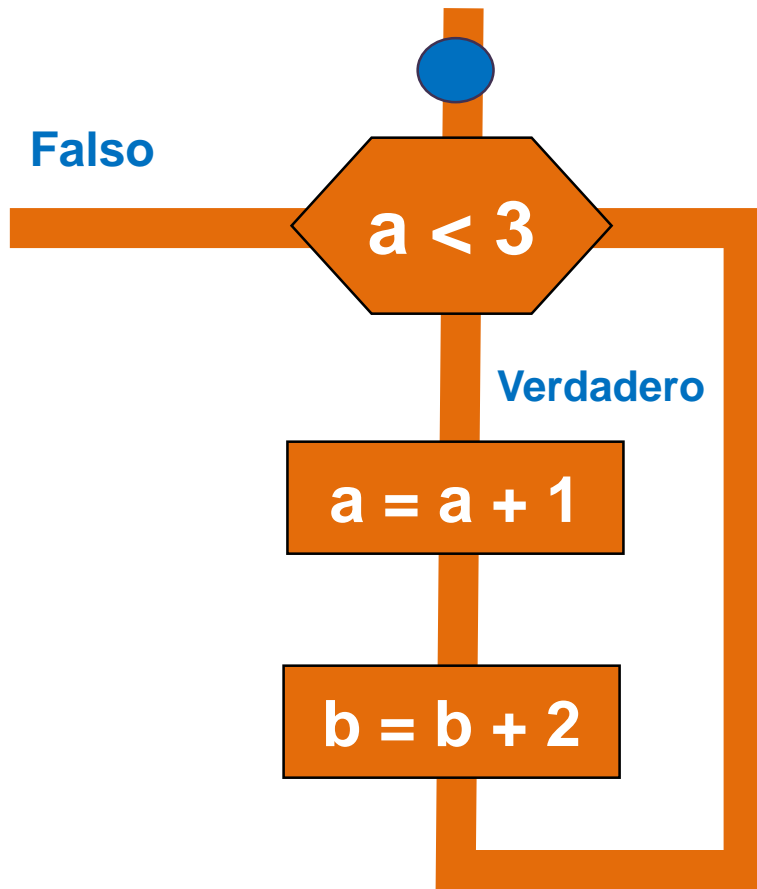
$a = a + 1$

$b = b + 2$

**a =** 0

**b =** 0

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

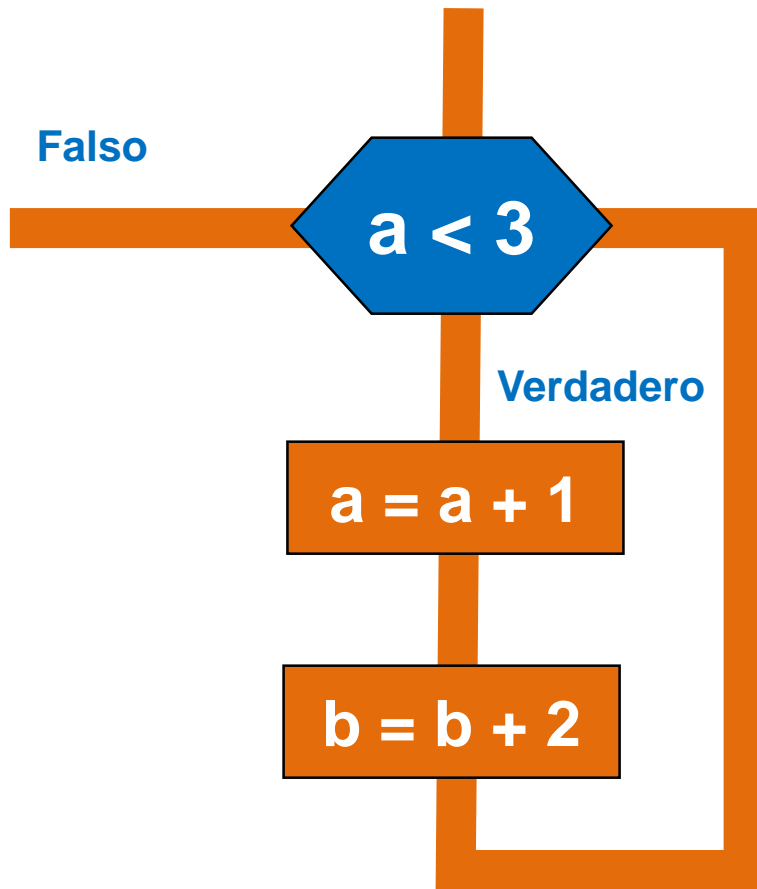
$b = b + 2$

$a =$

$b =$



# Simulación: Entendiendo el while



**a = 0**

**b = 0**

**while a < 3:**

**a = a + 1**

**b = b + 2**

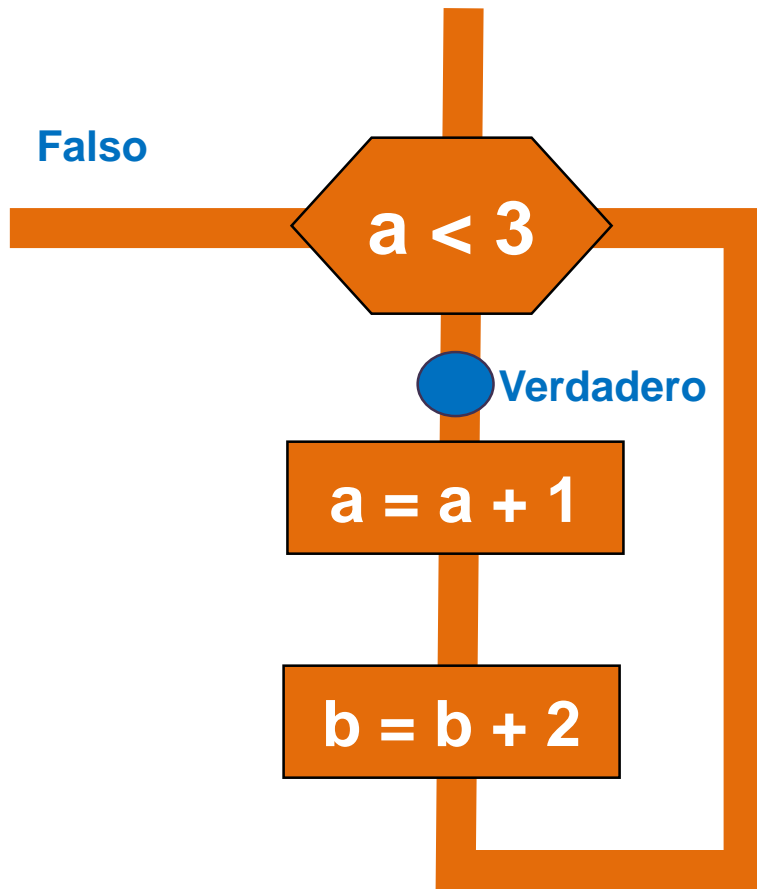
**a =**

**0**

**b =**

**0**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

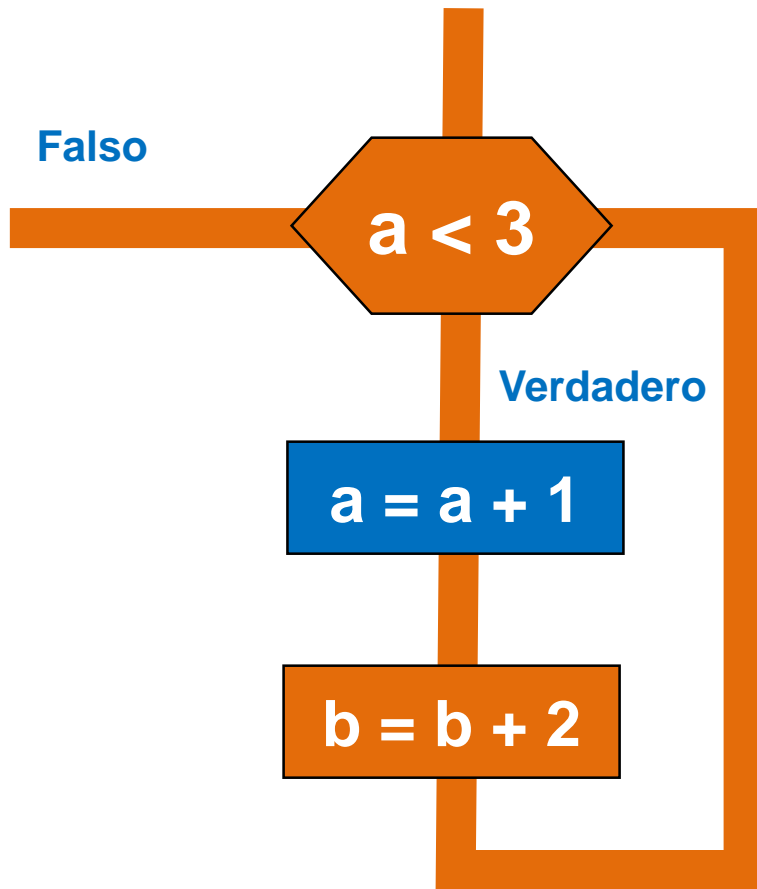
$a = a + 1$

$b = b + 2$

$a =$  0

$b =$  0

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

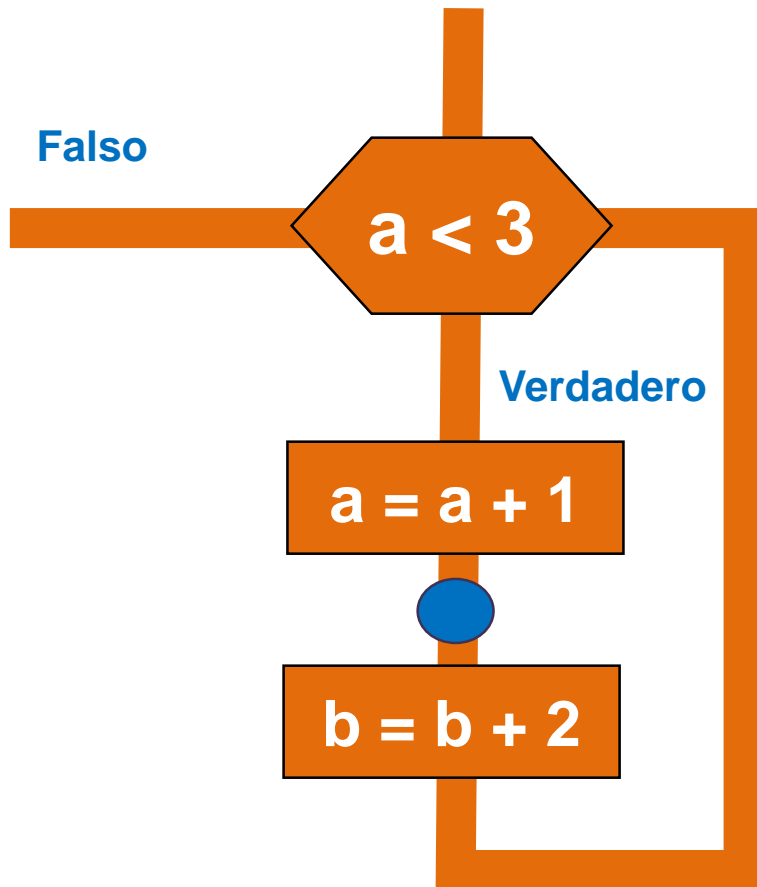
$a = a + 1$

$b = b + 2$

$a =$

$b =$

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

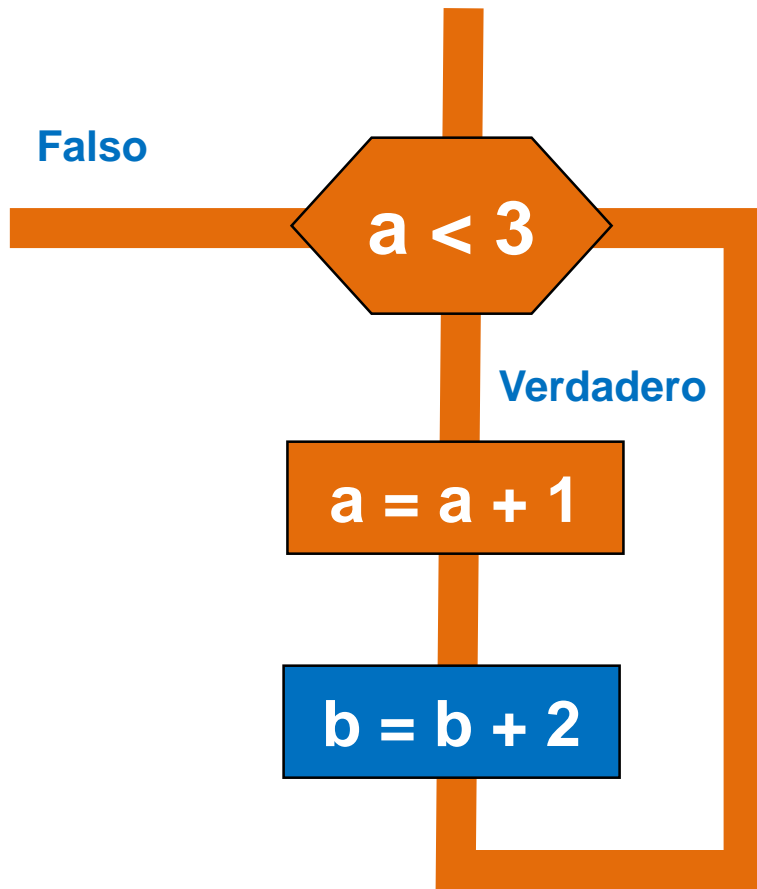
**a =**

**1**

**b =**

**0**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

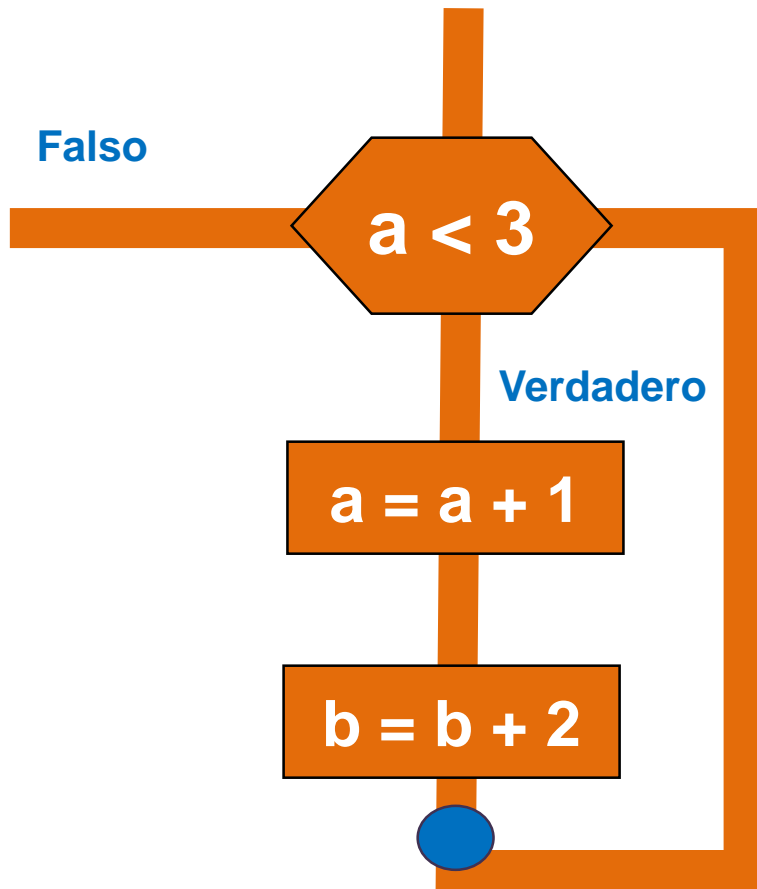
$a = a + 1$

$b = b + 2$

$a =$  1

$b =$  0

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

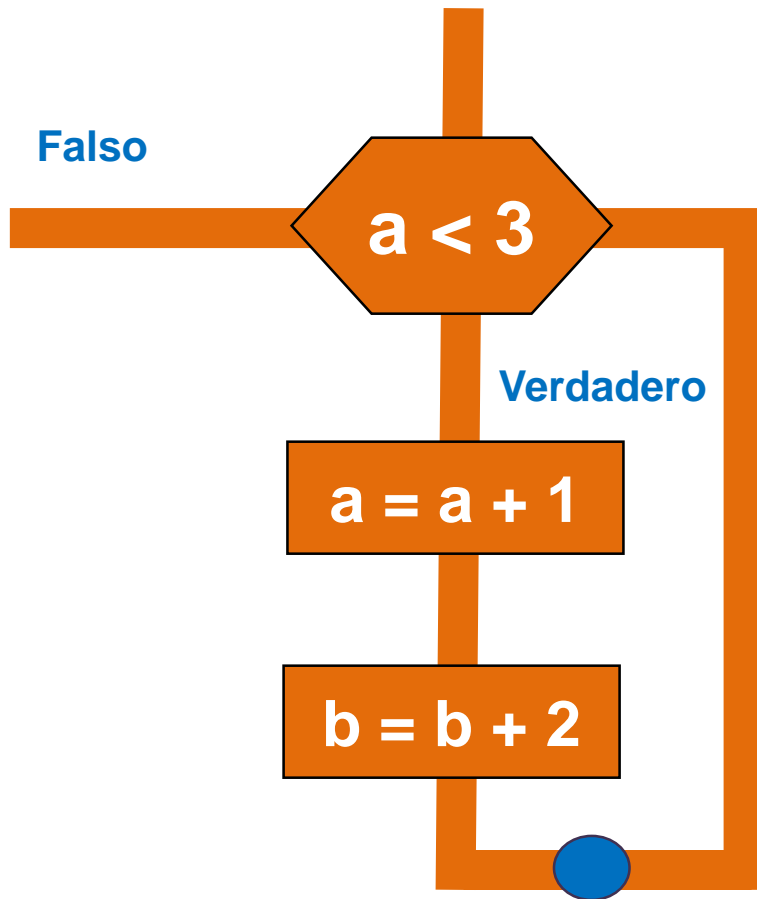
$a =$

1

$b =$

2

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

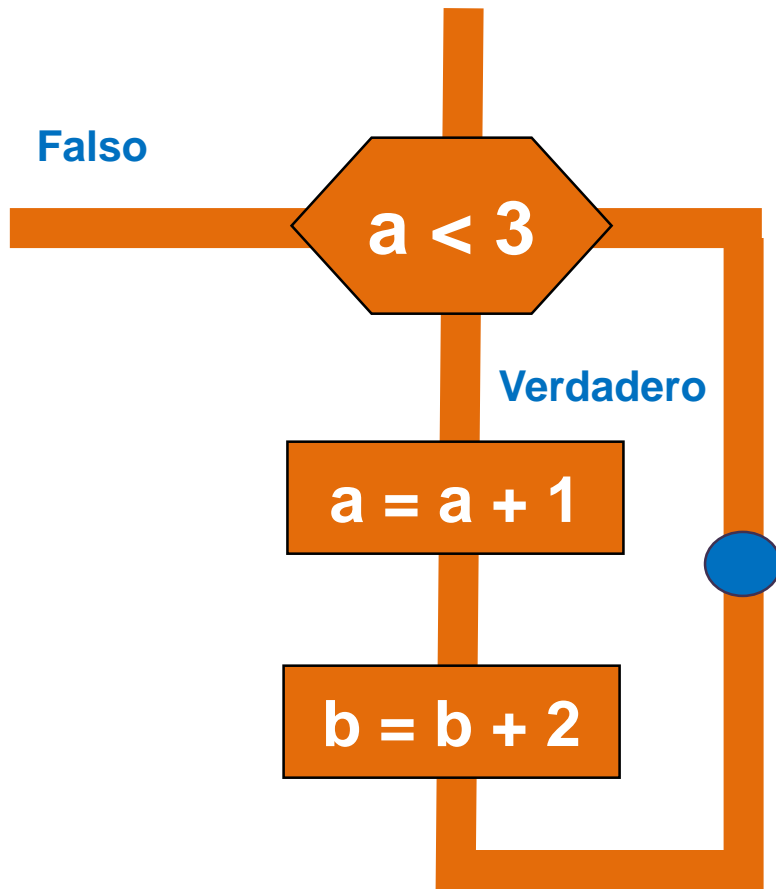
$a = a + 1$

$b = b + 2$

$a =$  1

$b =$  2

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

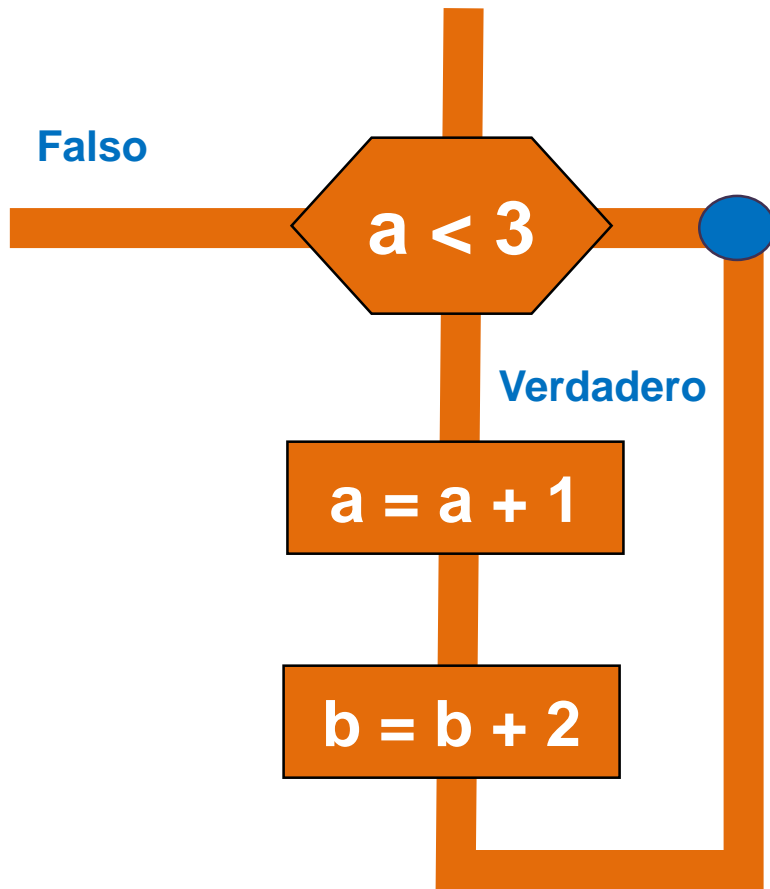
$b = b + 2$

$a =$  1

$b =$  2



# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

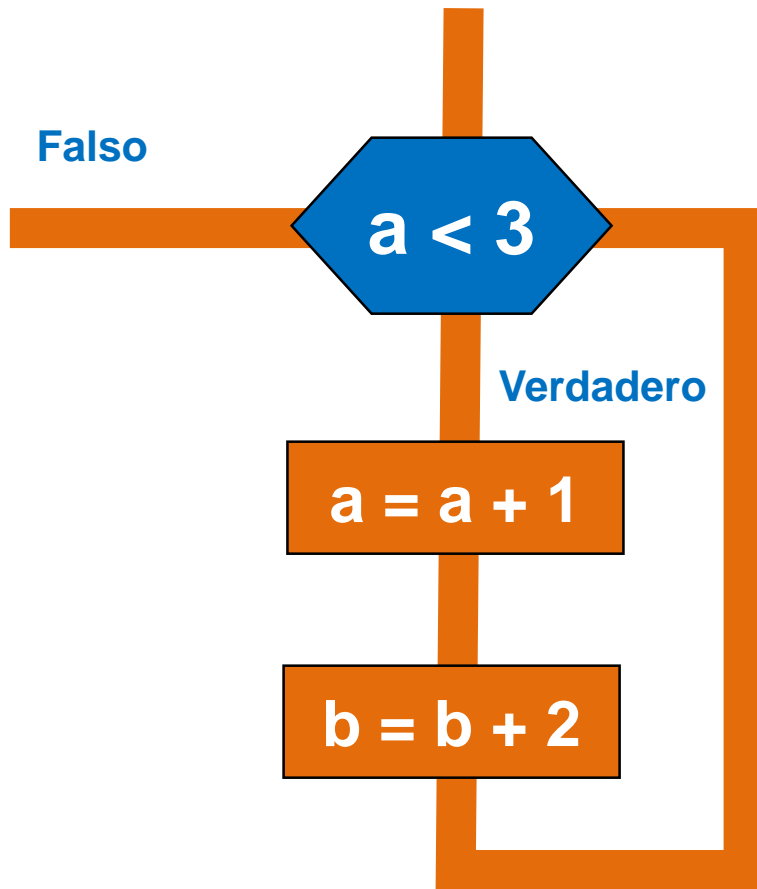
**a =**

**1**

**b =**

**2**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

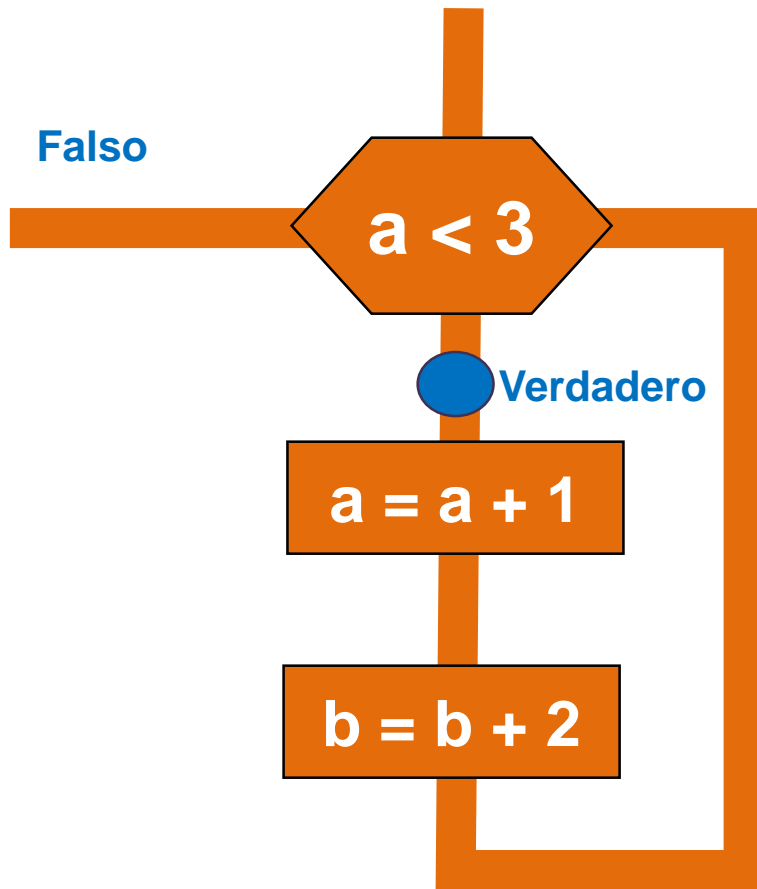
$a = a + 1$

$b = b + 2$

$a =$  1

$b =$  2

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

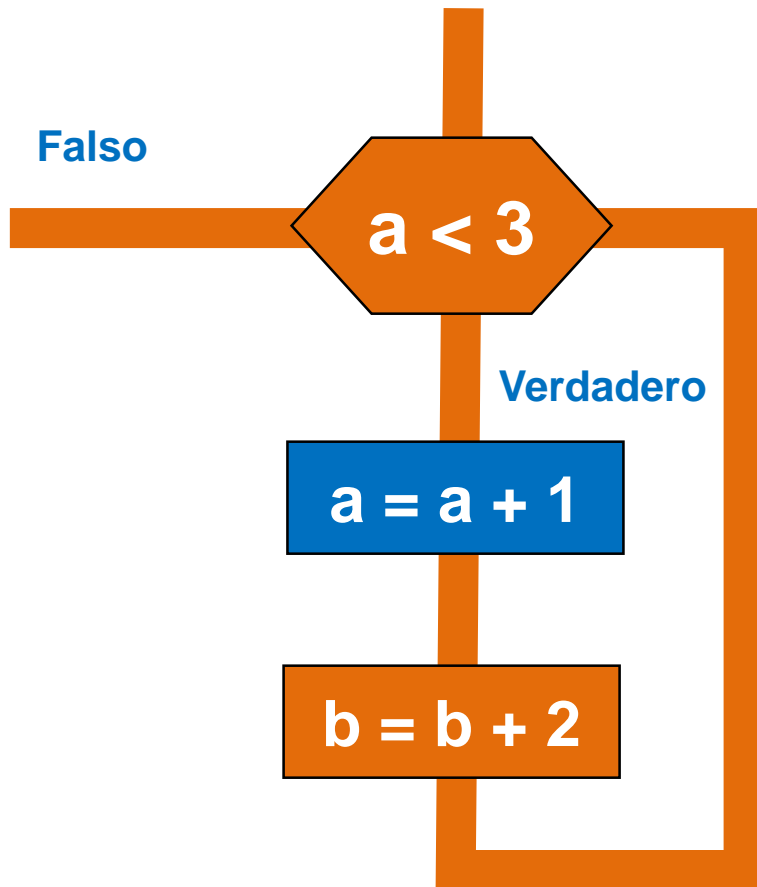
$a = a + 1$

$b = b + 2$

$a =$  1

$b =$  2

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

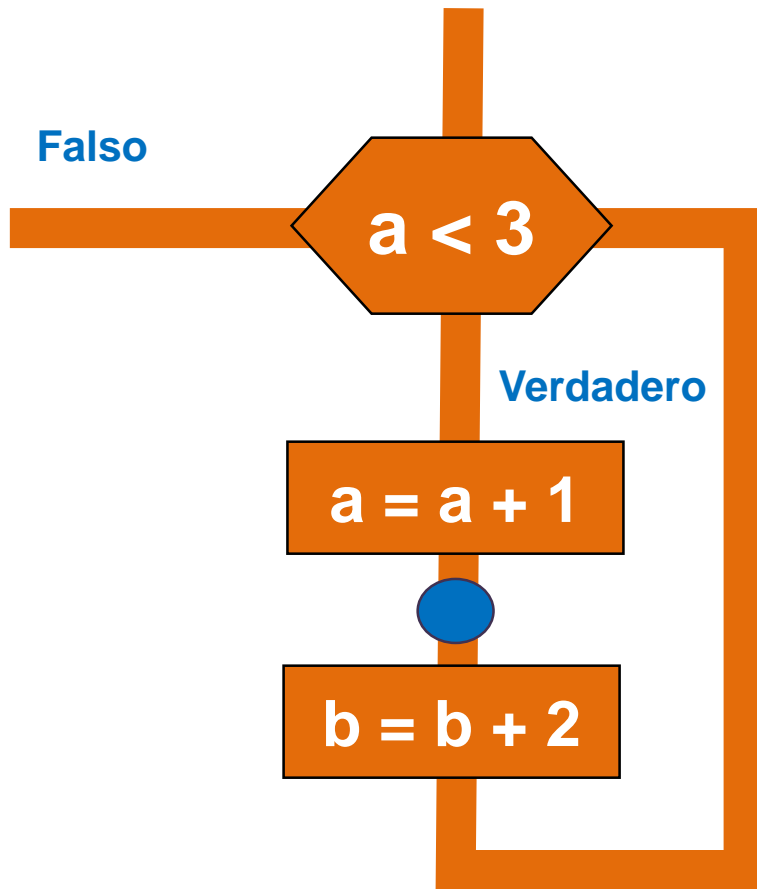
**a =**

**1**

**b =**

**2**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

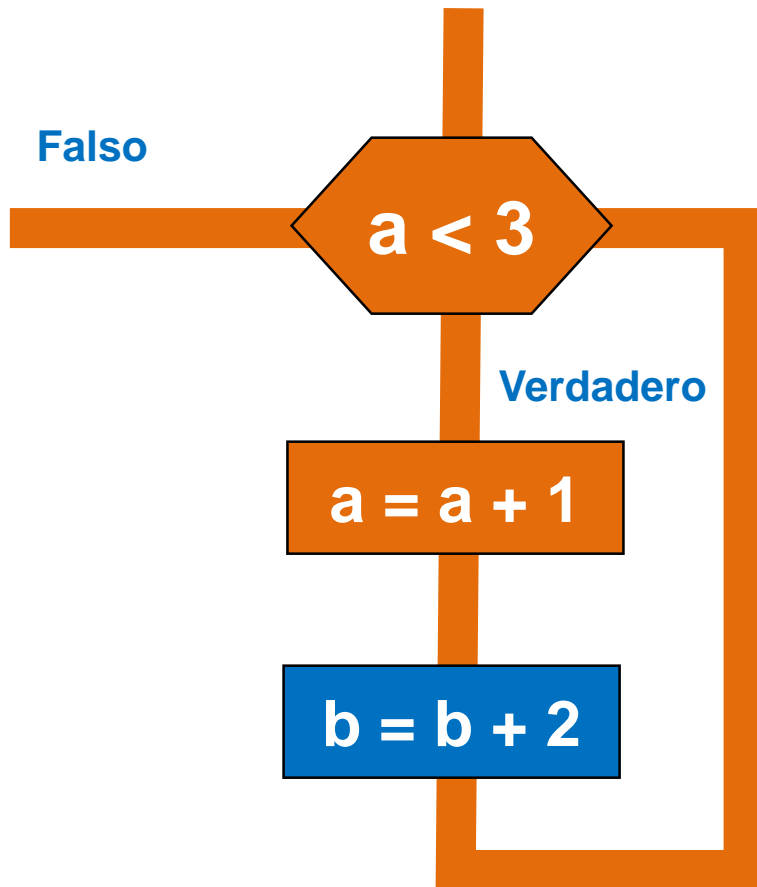
**a =**

**2**

**b =**

**2**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

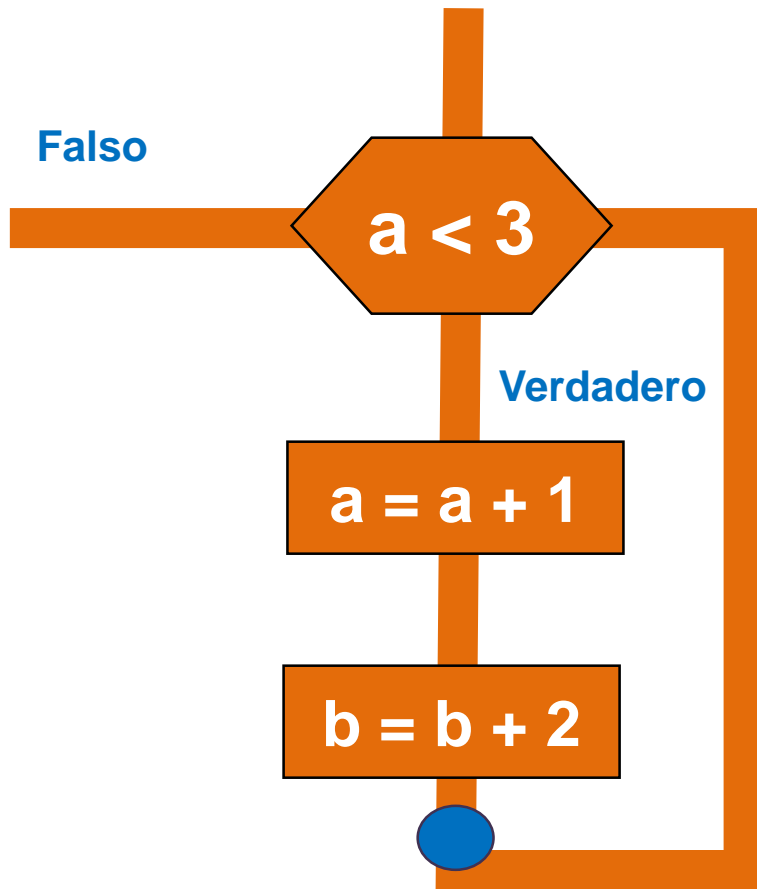
$a = a + 1$

$b = b + 2$

$a =$  2

$b =$  2

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

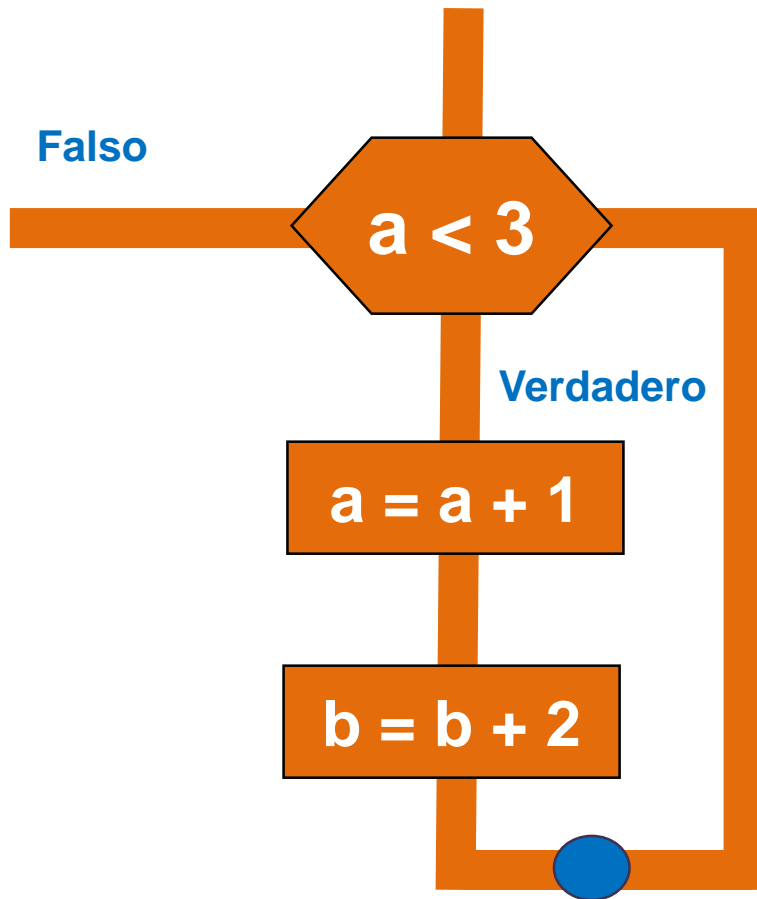
$a = a + 1$

$b = b + 2$

$a =$  2

$b =$  4

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

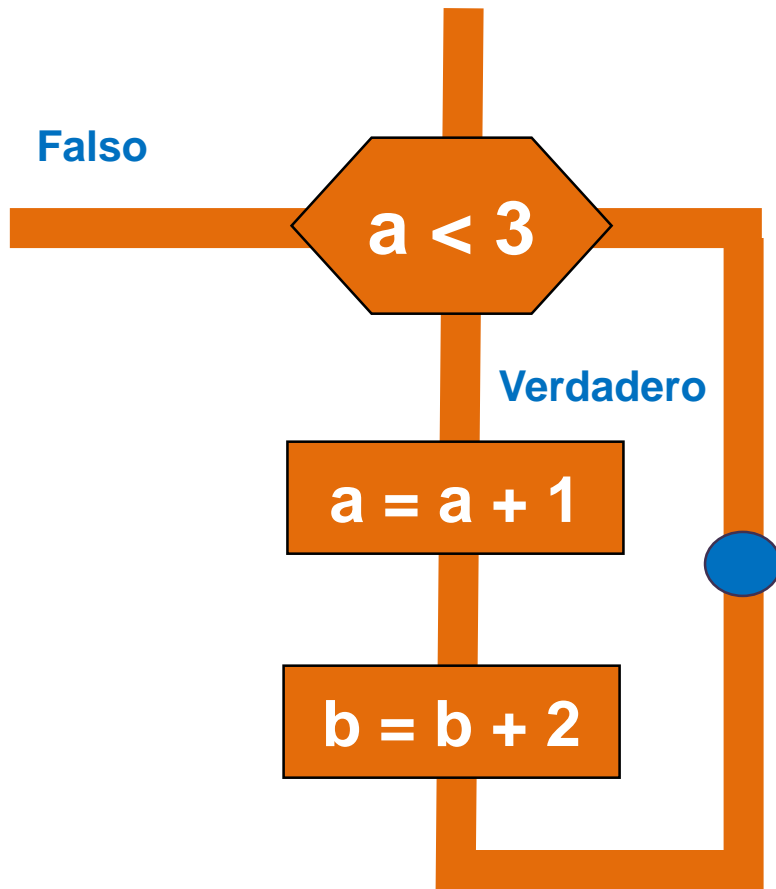
$b = b + 2$

$a =$  2

$b =$  4



# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

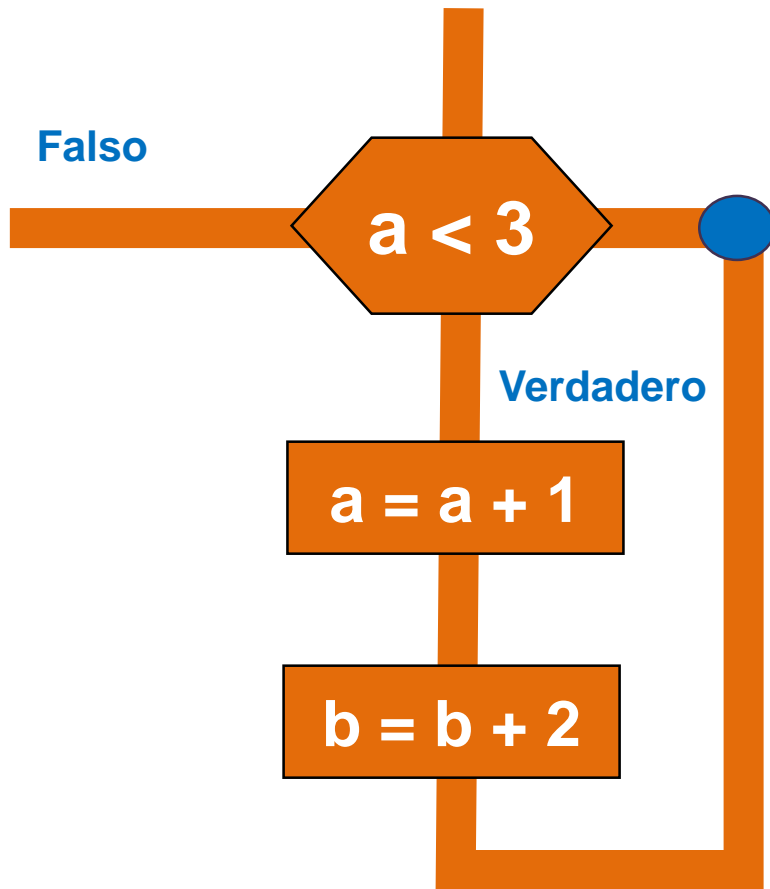
$a = a + 1$

$b = b + 2$

$a =$  2

$b =$  4

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

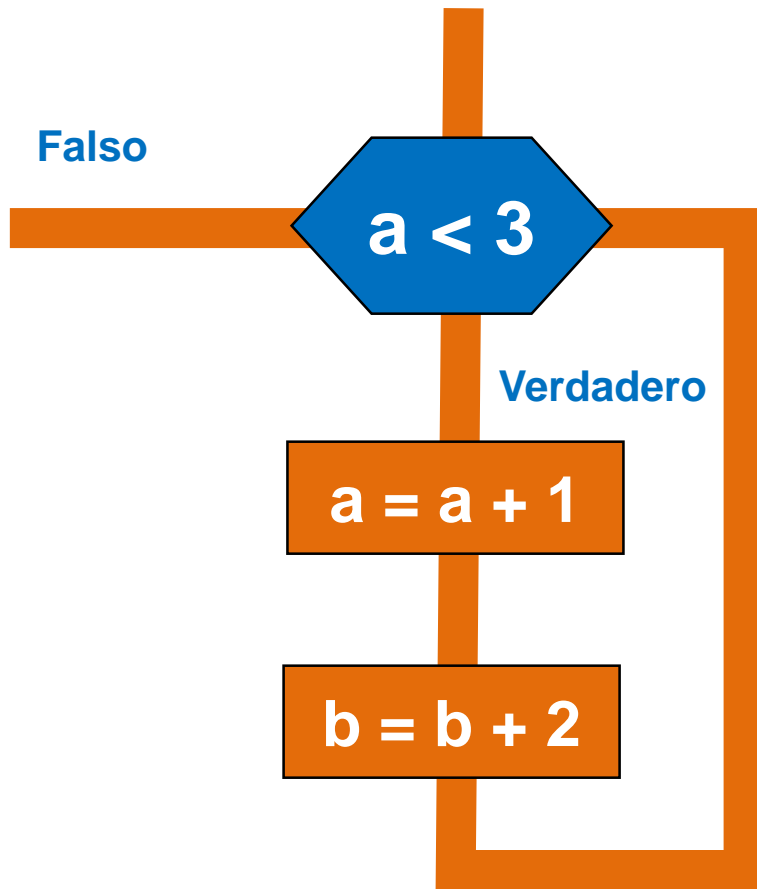
**a =**

**2**

**b =**

**4**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

$b = b + 2$

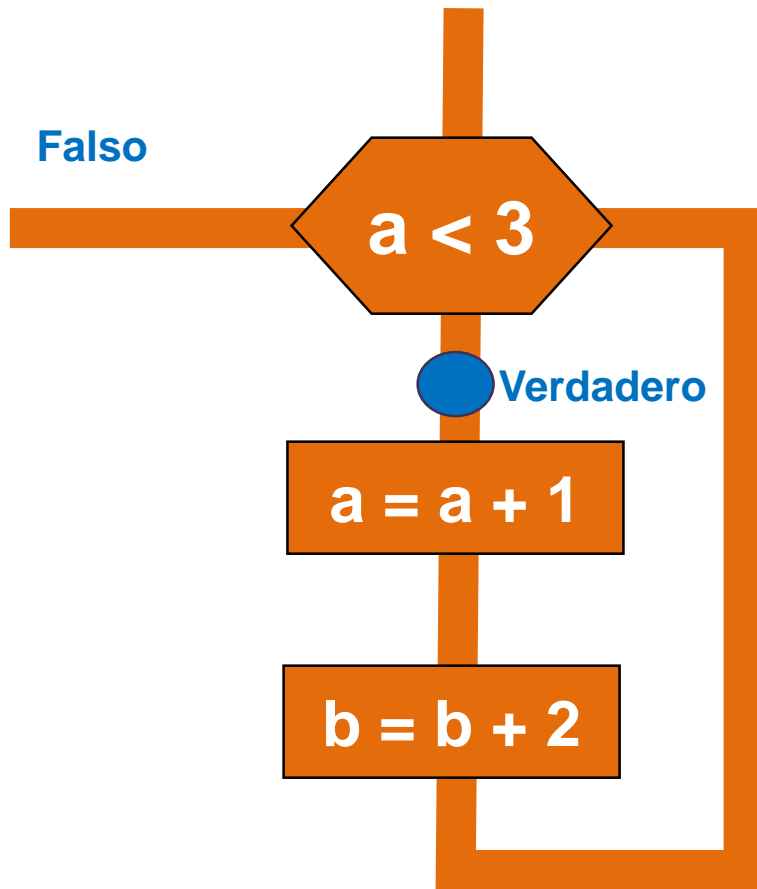
**a =**

**2**

**b =**

**4**

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

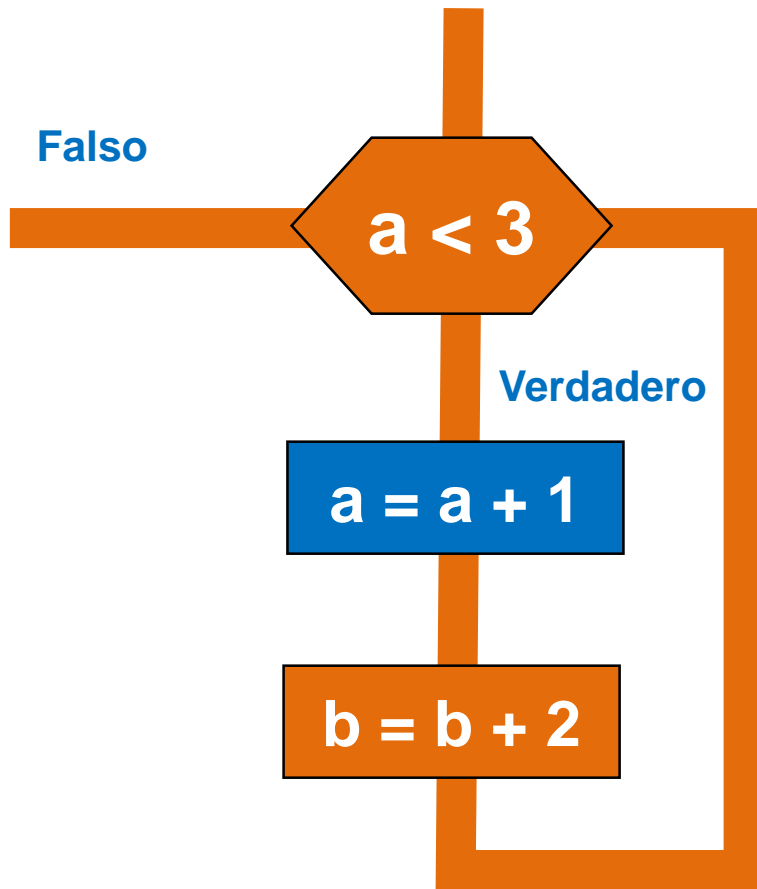
$a = a + 1$

$b = b + 2$

$a =$  2

$b =$  4

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

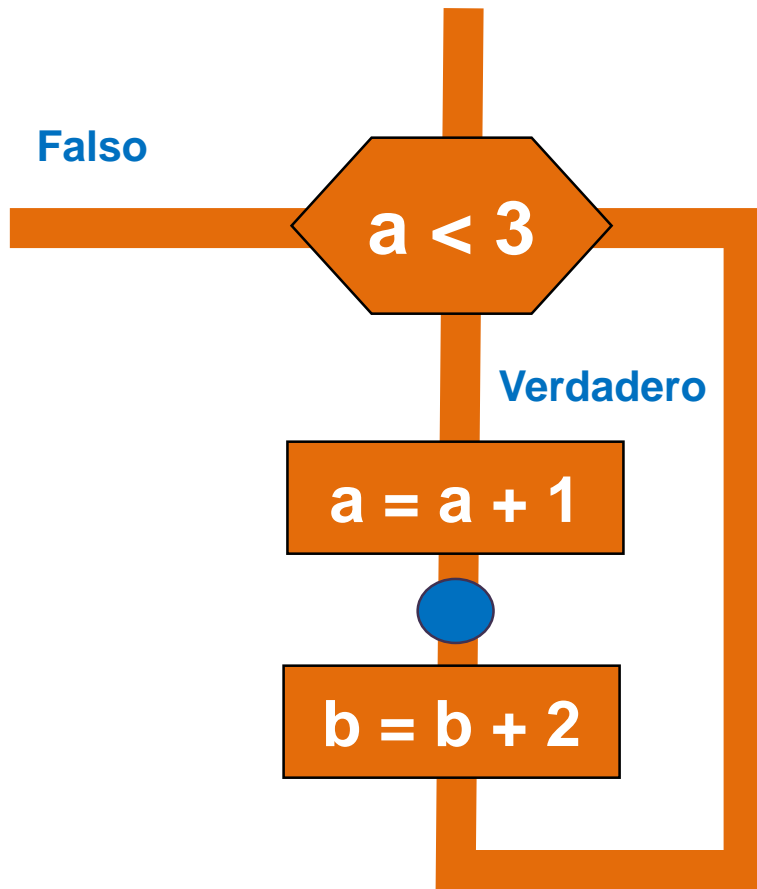
$a = a + 1$

$b = b + 2$

$a =$  2

$b =$  4

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

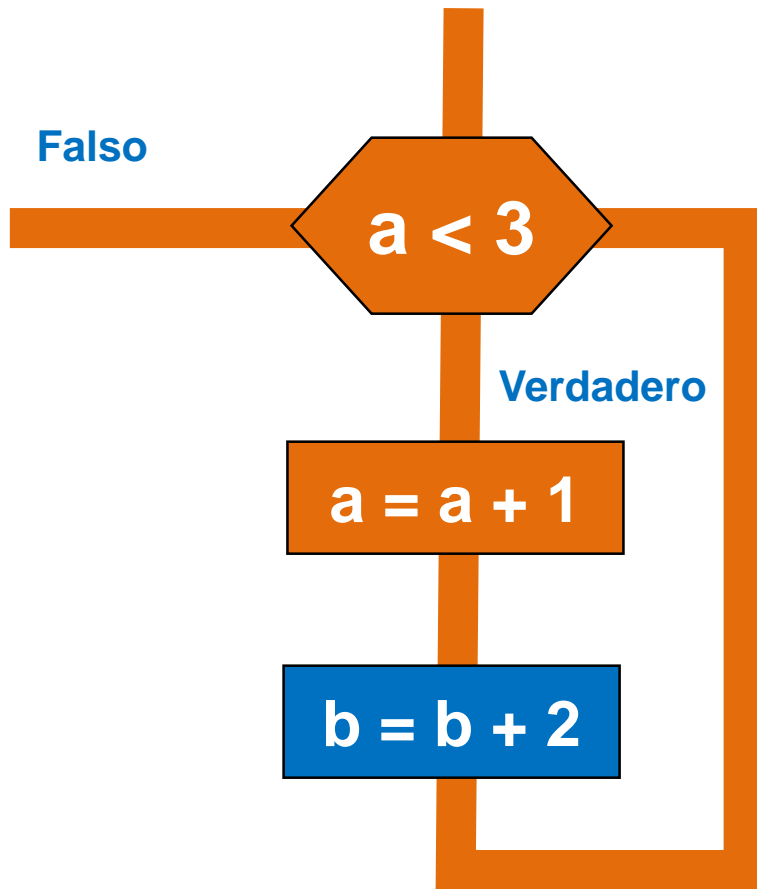
$a = a + 1$

$b = b + 2$

$a =$  3

$b =$  4

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

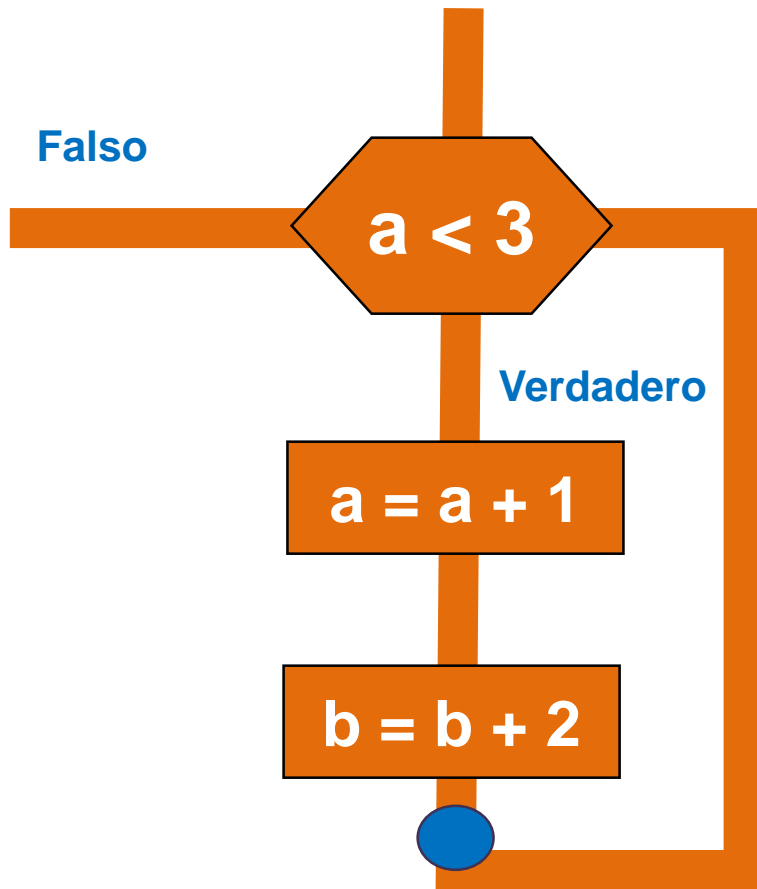
$a = a + 1$

$b = b + 2$

**a =** 3

**b =** 4

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

$a = a + 1$

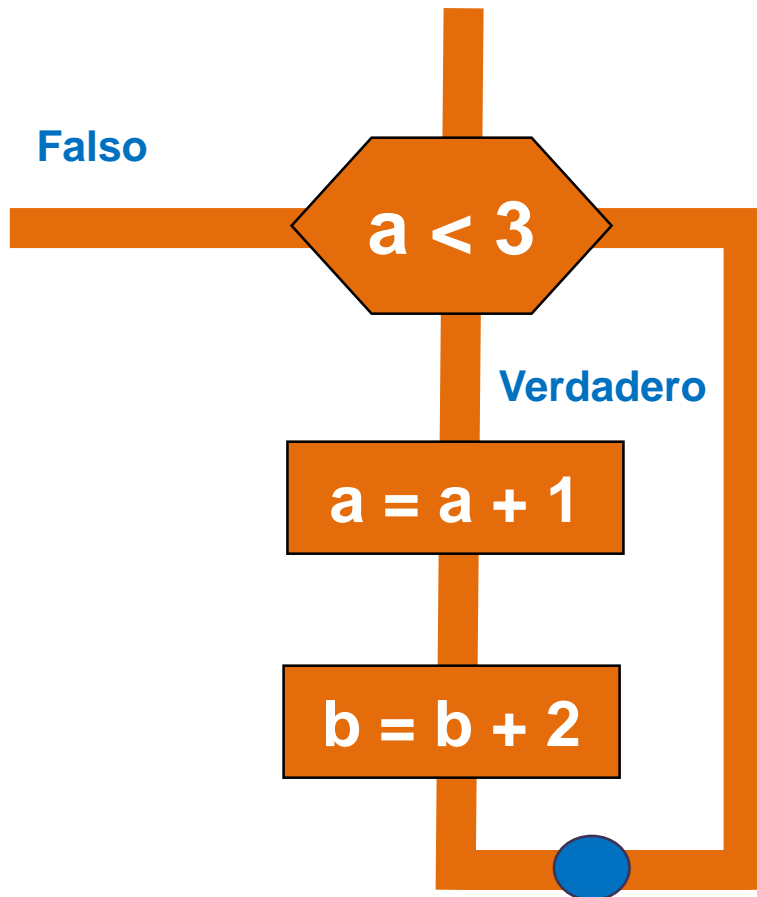
$b = b + 2$

$a =$  3

$b =$  6



# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

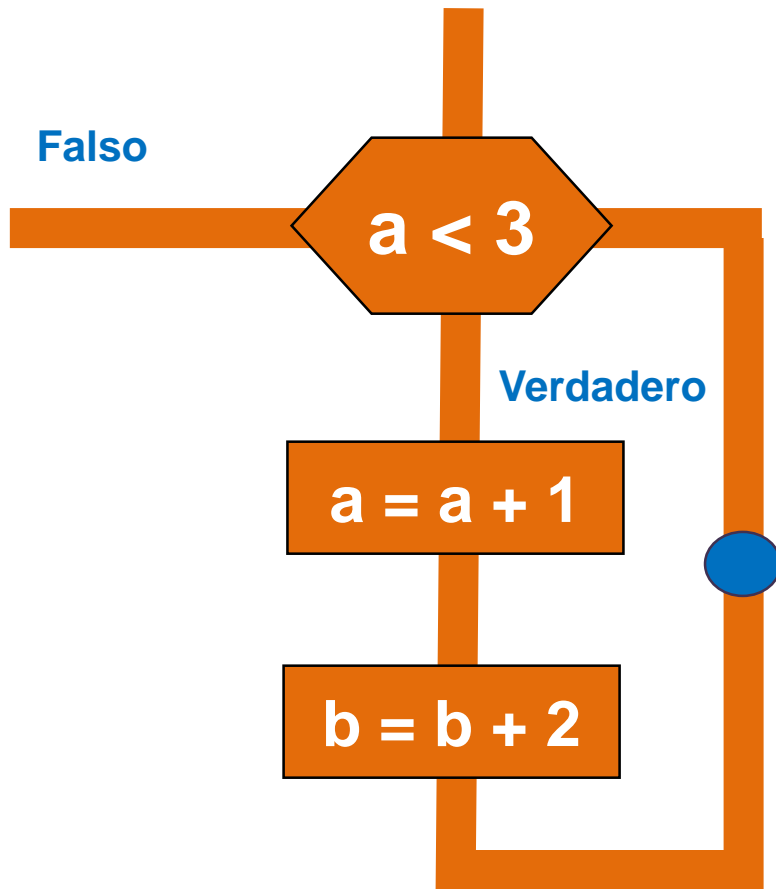
$a = a + 1$

$b = b + 2$

**a =** 3

**b =** 6

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

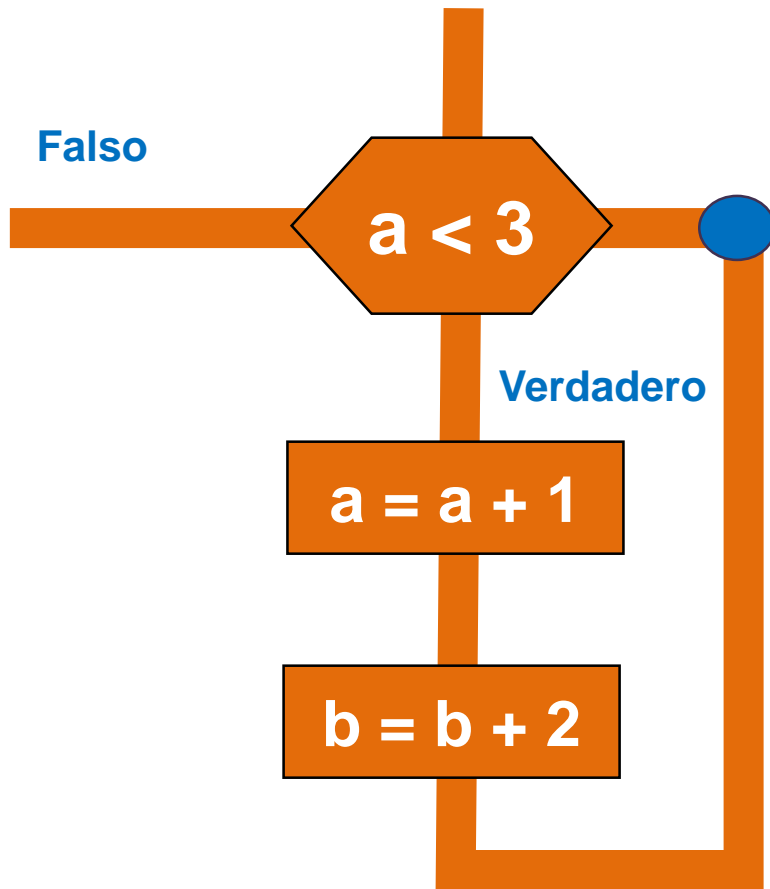
$a = a + 1$

$b = b + 2$

**a =** 3

**b =** 6

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

**while**  $a < 3$ :

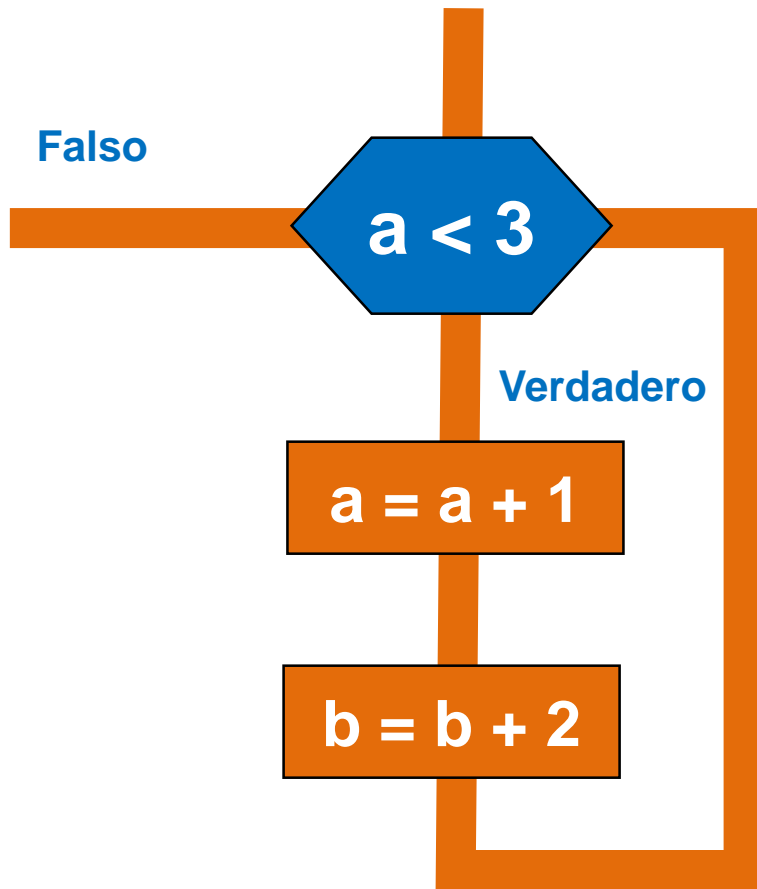
$a = a + 1$

$b = b + 2$

**a =** 3

**b =** 6

# Simulación: Entendiendo el while



**a = 0**

**b = 0**

**while a < 3:**

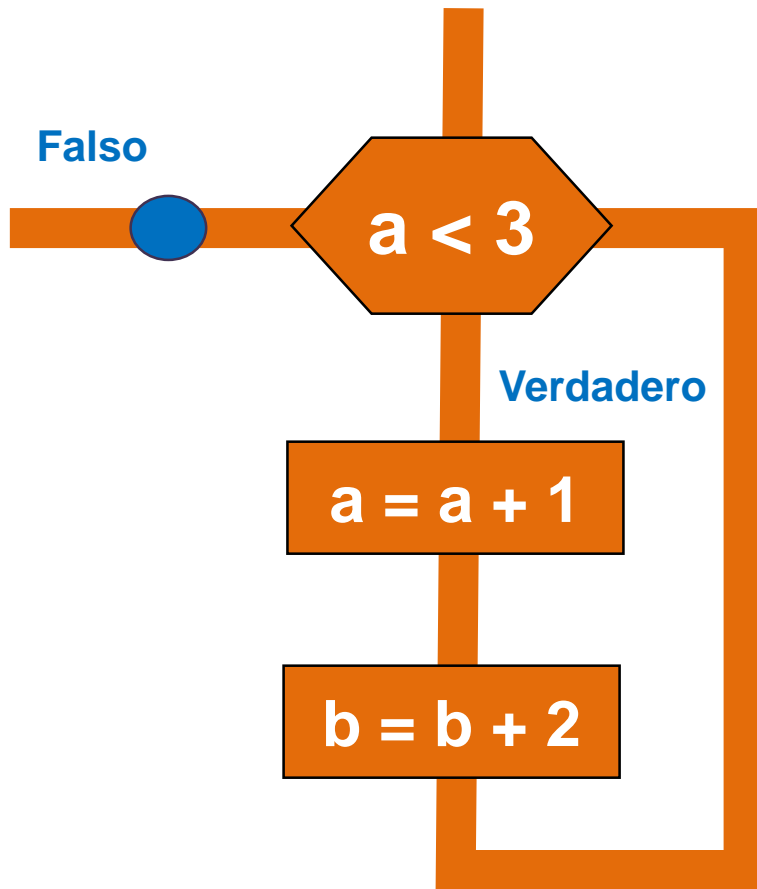
**a = a + 1**

**b = b + 2**

**a =** 3

**b =** 6

# Simulación: Entendiendo el while



$a = 0$

$b = 0$

while  $a < 3$ :

$a = a + 1$

$b = b + 2$

$a =$  3

$b =$  6

# Actividad Grupal

Analizar los siguientes segmentos de código e interpretar el valor final de las variables solicitadas.



# Actividad Grupal



```
x = 5
```

```
while x >= 5:
```

```
    print(" Hola a todos ")
```

¿Qué hace el siguiente código?

# Actividad Grupal



**$x = 10$**

**while  $x > 0$ :**

**$x = x - 1$**

**¿Cuántas veces se ejecutará este ciclo?**

**¿Cuál será el valor final de  $x$ ?**



# Actividad Grupal



```
x=0
```

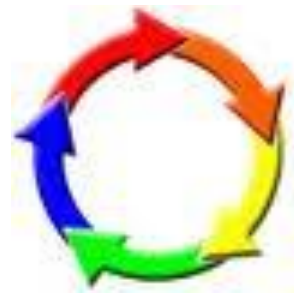
```
while x > 0:
```

```
    print(" Hasta luego ")
```

¿Qué hace el siguiente código?

¿Por qué?

# Funciones con while



- Una gran parte del código de Python que incluye estructuras **while**, lleva a cabo el control del ciclo a partir de llevar una enumeración a través de una variable que llamaremos **contador**.
- Por ejemplo, si queremos repetir una determinada instrucción 10 veces, es necesario definir una variable que vaya contando en qué pasada del ciclo se encuentra.

# Actividad Grupal

Escribir una **función** en Python que imprima  
10 veces **“hola a todos”** con su respectivo  
pseudo-algoritmo



# Algoritmo y Pseudo-algoritmo

## Algoritmo

Es el método o pasos utilizados para solucionar un problema o describir un proceso. El método usado es independiente de la computadora o plataforma.

## Pseudo-Algoritmo

Es la descripción de un proceso mediante la utilización de estructuras de un lenguaje de programación y sentencias informales del lenguaje (español).



# Pseudo-Algoritmo: Imprime 10

**funcion** **imprime10**

contador = 0

**mientras** contador < 10

**escribir**("hola a todos")

contador = contador + 1



**Solución**

# Función imprime 10

```
def imprime10:
```

```
    contador=0
```

```
    while contador < 10:
```

```
        print (" Hola a todos ")
```

```
        contador+= 1  #contador=contador+1
```

```
imprime10()
```



**Solución**

# Actividad Grupal

Escribe la función **f1** que recibe un valor entero **mayor o igual a 1** y regresa el resultado de la siguiente serie...

$$f1(n) = 1 + 2 + 3 + 4 + \dots + n$$



# Pseudo-Algoritmo : Función f1

funcion **f1** (n)

contador = 1

acumulador = 0

**mientras** contador <= n

    acumulador = acumulador + contador

    contador= contador + 1

**regresar** acumulador



**Solución**



# Función f1

```
def f1 (n):
```

```
    acumulador=0
```

```
    contador=1
```

```
    while contador <= n:
```

```
        acumulador=acumulador + contador
```

```
        contador = contador + 1
```

```
    return acumulador
```



## Solución

# Función f1

```
num = int(input("Dame un numero entero: "))  
if num >= 1:  
    res = f1(num)  
    print("f1(%d) = %d" % (num, res))  
else:  
    print("Número inválido")
```



**Solución**

# Actividad individual

```
x = 8
```

```
y = 3
```

```
while y <= 16 :
```

```
    x = x+ 1
```

```
    y = y+ 2
```

```
print(x)
```

```
print(y)
```

¿Qué aparece en la pantalla si se ejecuta el siguiente código de Python?