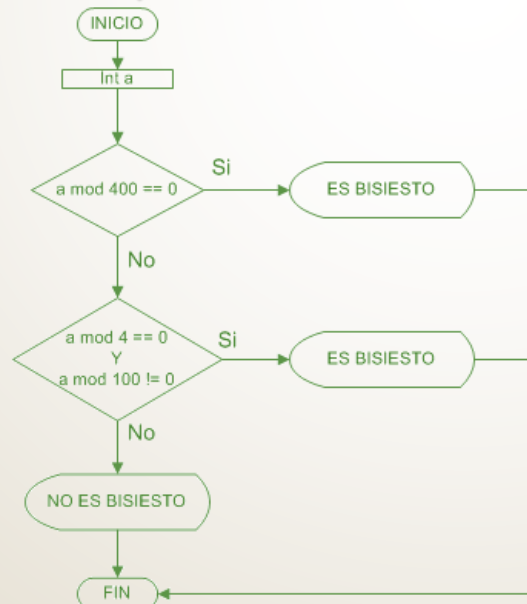


# Programación Estructurada

## FINALIZACIÓN DE BUCLES



# Bucles Infinitos (1)

- ¿Qué ocurre cuando en un programa un conjunto de acciones se repite sin control?
  - Bucles infinitos



## Bucles Infinitos (2)

- Los **bucles infinitos** no alcanzan la condición de finalización y por tanto se repiten indefinidamente.
- Se deben a errores de diseño: incorrecta formulación de la condición de finalización, omisión o errores en las instrucciones que modifican la condición de salida.

```
ciclo<-VERDADERO
MIENTRAS ciclo=VERDADERO HACER
    ESCRIBIR "BUCLE INFINITO"
FIN_MIENTRAS
```

```
contador<-1
MIENTRAS contador < 20 HACER
    ESCRIBIR "BUCLE INFINITO"
    contador<-contador - 1
FIN_MIENTRAS
```

# Criterios de Finalización

- ¿Cuáles son los criterios para finalizar las iteraciones de un bucle?

- Por *Valor Centinela*



- Por *Bandera*



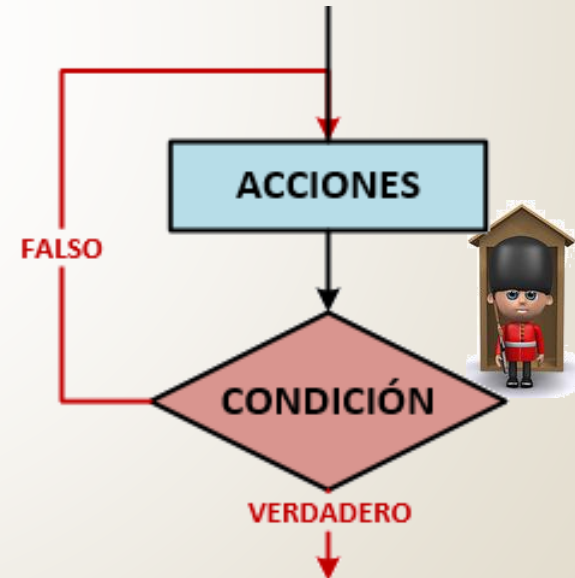
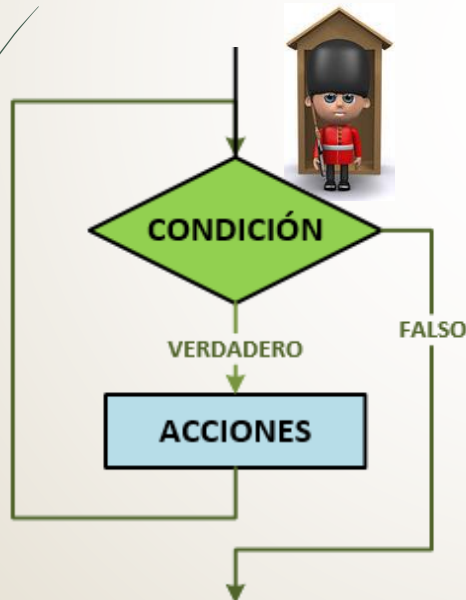
- Por *Contador*



# Finalización por Centinela

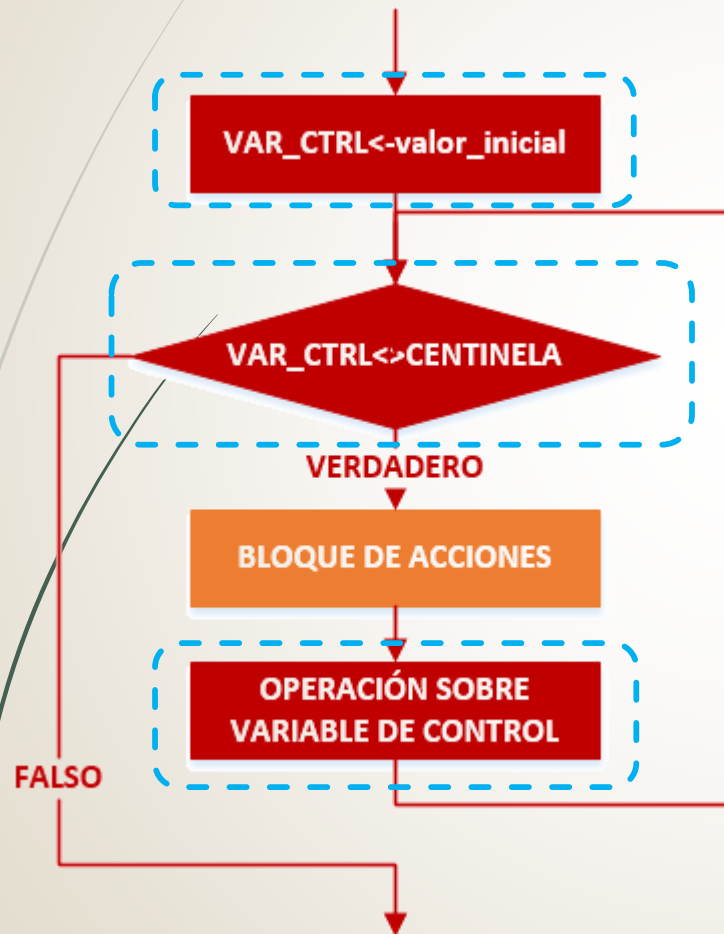
## ► Valor Centinela

- El bucle es controlado por un valor especial denominado centinela. Cuando este valor se presenta entonces el bucle finaliza.





# Mientras: Controlado por Centinela (1)



- **Inicialización:** se asigna el valor inicial a la variable de control del bucle
- **Condición de repetición:** se verifica que la variable de control sea **DISTINTA** al valor centinela.
  - Cond. Verdadera repite
  - Cond. Falsa finaliza
- **Modificación de la variable de control:** se realiza alguna operación que modifica la variable de control.

# Mientras: Controlado por Centinela (2)

- Diseñe un algoritmo que sume valores ingresados por el usuario, hasta que se introduzca un 0. Implemente el bucle de cálculo con estructuras MIENTRAS y utilice la finalización por centinela.

- Inicialización:

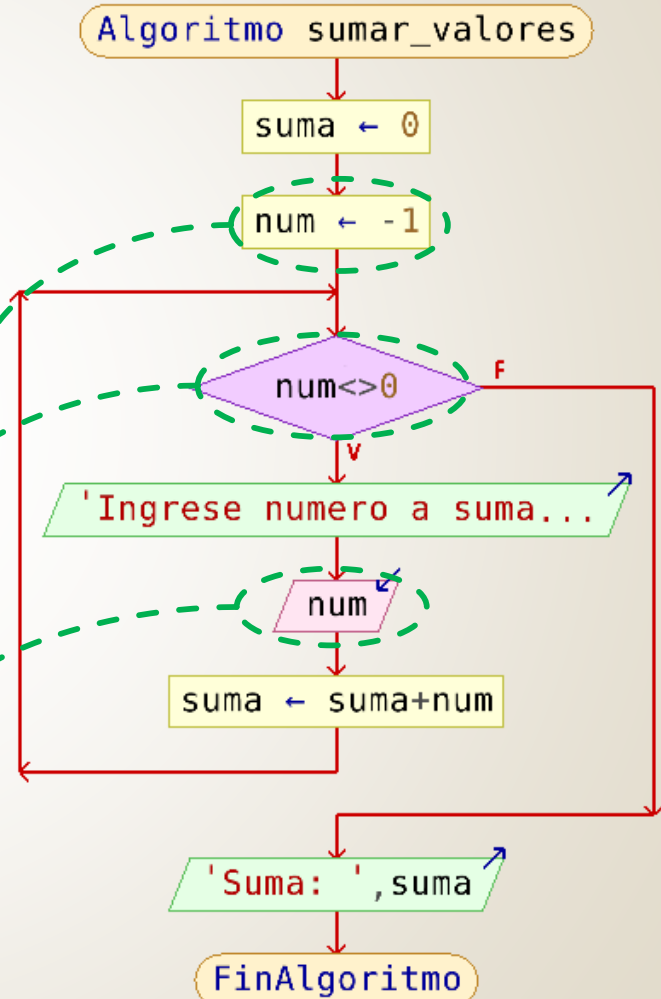
$\text{num} \leftarrow -1$  (arbitrario)

- Condición de repetición:

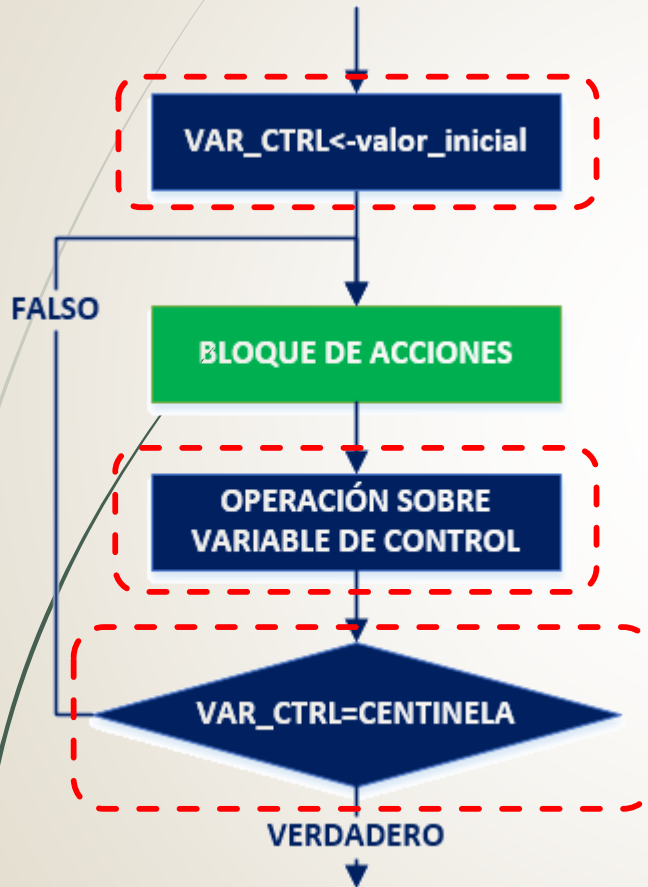
$\text{num} \neq 0$

- Modificación:

LEER num



# Repetir: Controlado por Centinela (1)



- **Inicialización:** se asigna el valor inicial a la variable de control del bucle (OPCIONAL)
- **Modificación de la variable de control:** se realiza alguna operación que modifica la variable de control.
- **Condición de repetición:** se verifica que la variable de control sea **IGUAL** al valor centinela.
  - Cond. Falsa repite
  - Cond. Verdadera finaliza



# Repetir: Controlado por Centinela (2)

- Diseñe un algoritmo que sume valores ingresados por el usuario. Considere que el ingreso finaliza a petición del usuario. Implemente el bucle de cálculo con estructuras REPETIR y utilice la finalización por centinela.

- Inicialización:

$rta \leftarrow ''$  (arbitrario y opcional)

- Modificación:

LEER  $rta$

- Condición de repetición:

$rta = 'n'$  O  $rta = 'N'$



Algoritmo sumar\_valores

$\text{suma} \leftarrow 0$

$rta \leftarrow ''$

'Ingrese numero a suma...'

num

$\text{suma} \leftarrow \text{suma} + \text{num}$

'Mas datos s/n: '

rta

$rta = 'n'$  O  $rta = 'N'$

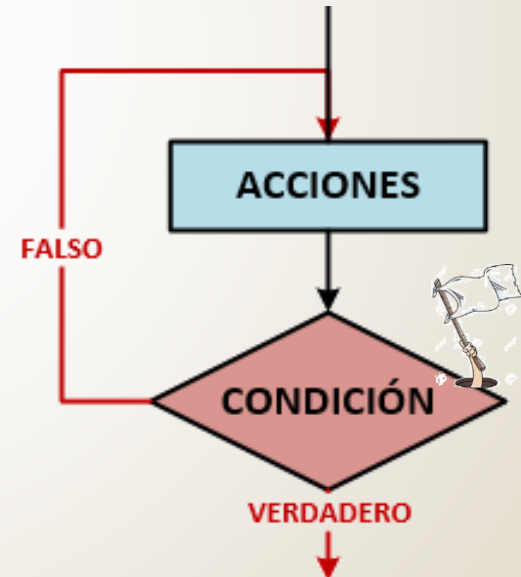
'Suma: ', suma

FinAlgoritmo

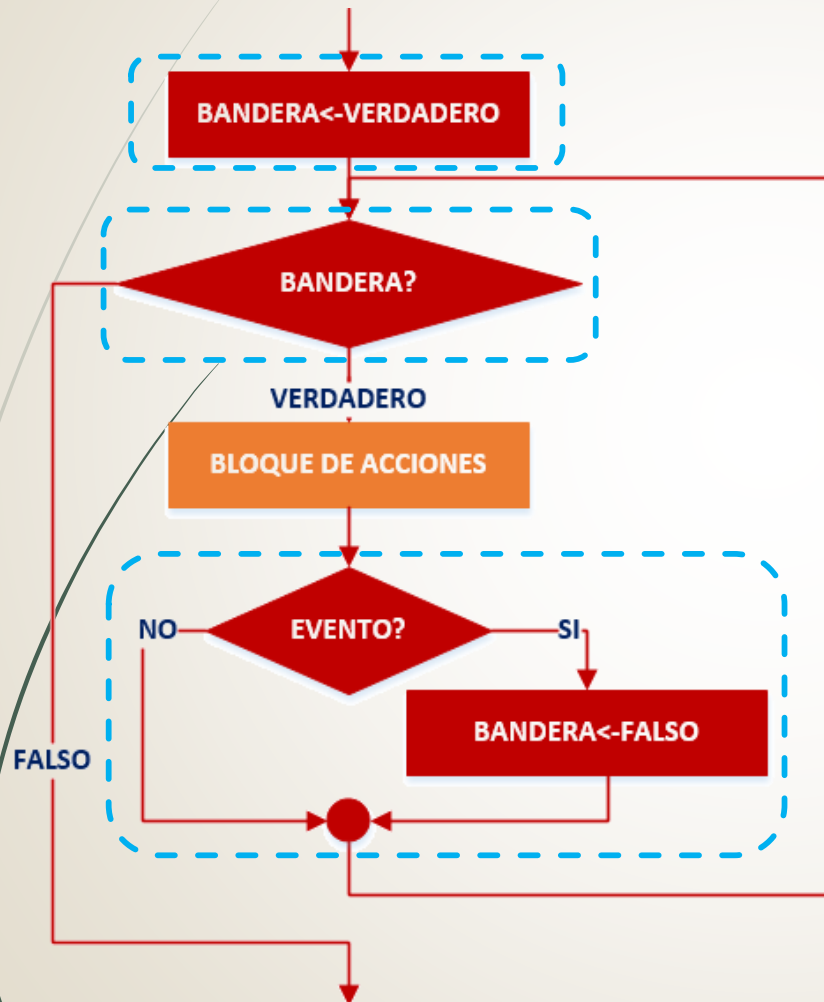
# Finalización por Bandera

## ➤ Bandera

- El bucle es controlado por una variable lógica, denominada bandera, que se utiliza para detectar la ocurrencia de un evento. Cuando éste se produce el bucle finaliza.



# Mientras: Controlado por Bandera (1)



- **Inicialización:** se asigna el valor inicial a la bandera
- **Condición de repetición:** se analiza el valor de la bandera
  - Cond. Verdadera repite
  - Cond. Falsa finaliza
- **Detección de un evento:** se verifica si determinado evento ocurrió o no en el programa. La ocurrencia del evento implica modificar la bandera.

# Mientras: Controlado por Bandera (2)

- Diseñe un algoritmo que calcule, mediante la suma de impares, el cuadrado de un número. Realice el bucle de cálculo con estructuras MIENTRAS y utilice la finalización por bandera.

- Inicialización:

$\text{calcular} \leftarrow \text{num} > 0$

- Condición de repetición:

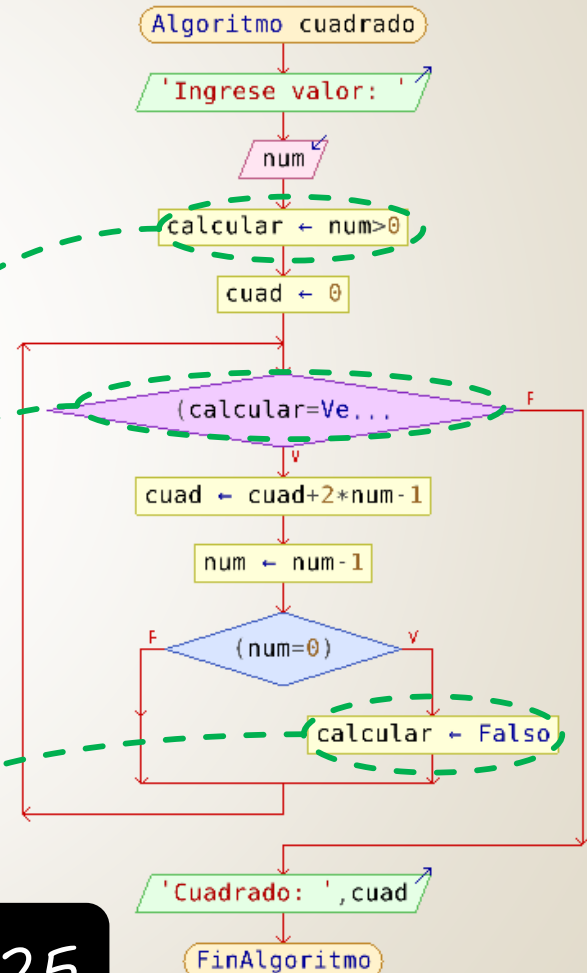
$\text{calcular} = \text{Verdadero}$

- Modificación:

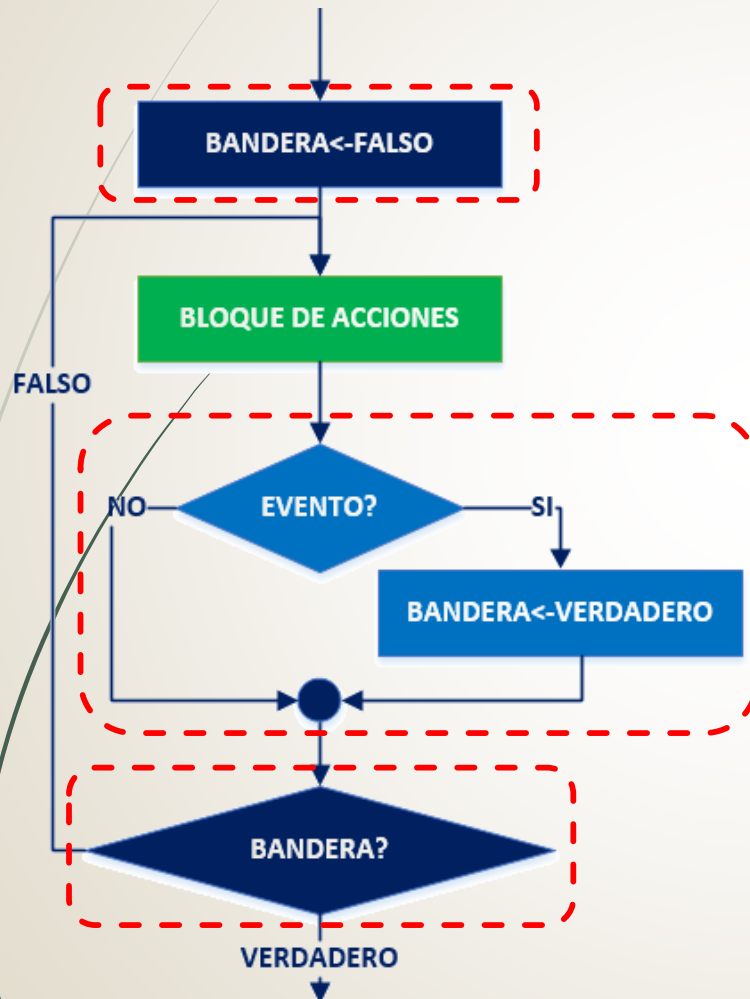
$\text{calcular} \leftarrow \text{Falso}$



$$5^2 = 1 + 3 + 5 + 7 + 9 = 25$$



# Repetir: Controlado por Bandera (1)



- **Inicialización:** se asigna el valor inicial a la bandera
- **Detección de un evento:** se verifica si determinado evento ocurrió o no en el programa. La ocurrencia del evento implica modificar la bandera.
- **Condición de repetición:** se analiza el valor de la bandera
  - Cond. Falsa repite
  - Cond. Verdadera finaliza



# Repetir: Controlado por Bandera (2)

- Diseñe un algoritmo que calcule el cociente entero, mediante restas, de 2 números ingresados por el usuario. Implemente el bucle de cálculo con estructuras REPETIR y utilice la finalización por bandera.

- Inicialización:

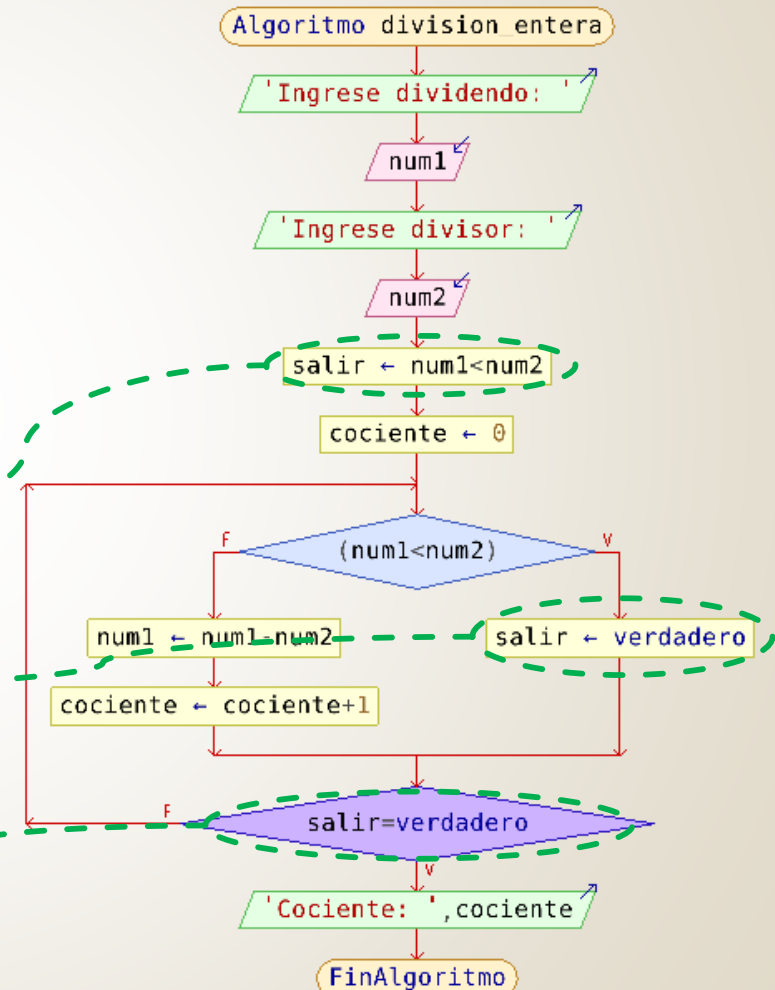
$\text{salir} \leftarrow \text{num1} < \text{num2}$

- Modificación:

$\text{salir} \leftarrow \text{Verdadero}$

- Condición de repetición:

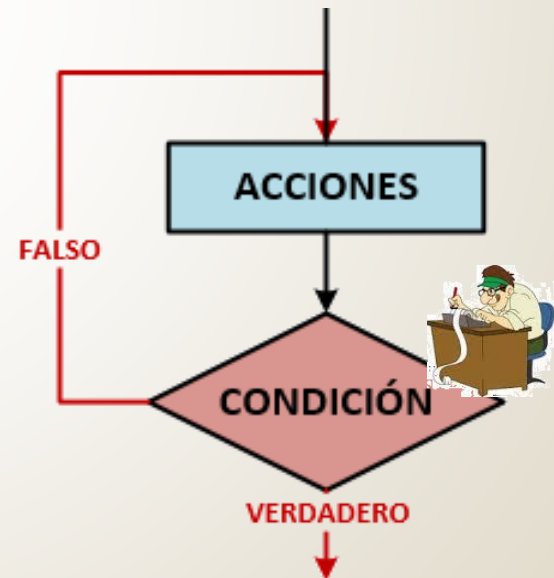
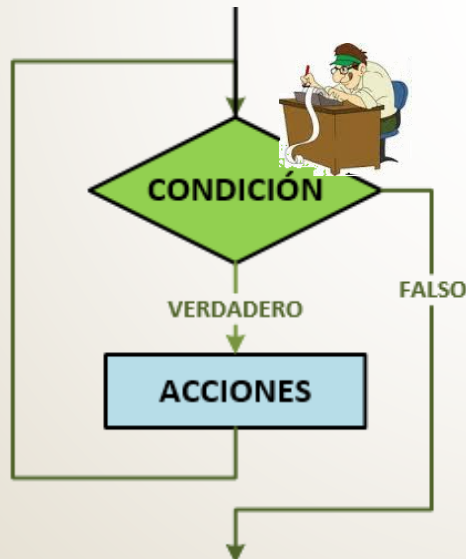
$\text{salir} = \text{Verdadero}$



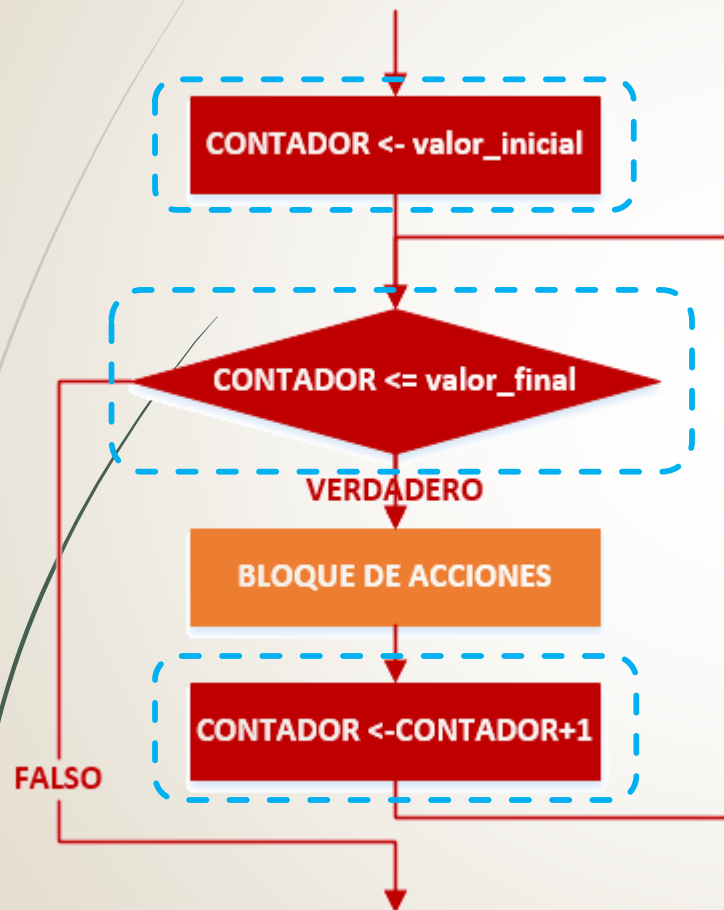
# Finalización por Contador

## ➤ Contador

- El bucle es controlado por una variable numérica, denominada contador, que se utiliza para contar la cantidad de repeticiones u ocurrencias del bucle. El bucle finaliza al completar la cuenta.



# Mientras: Controlado por Contador (1)



- **Inicialización:** se asigna el valor inicial al contador
- **Condición de repetición:** se verifica que el contador se encuentre entre el valor\_inicial y valor\_final
  - Cond. Verdadera repite
  - Cond. Falsa finaliza
- **Modificación del contador:** se incrementa o decrementa el valor del contador.

# Mientras: Controlado por Contador (2)

- Diseñe un algoritmo que calcule el factorial de un número ingresado por el usuario. Implemente el bucle de cálculo con estructuras MIENTRAS y utilice la finalización por contador.

- Inicialización:

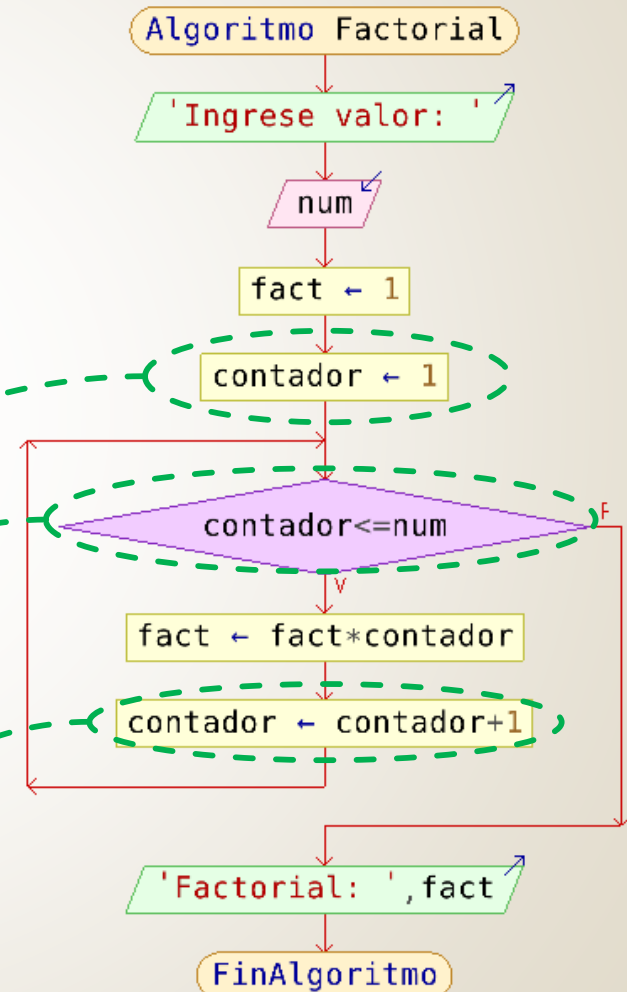
contador  $\leftarrow$  1

- Condición de repetición:

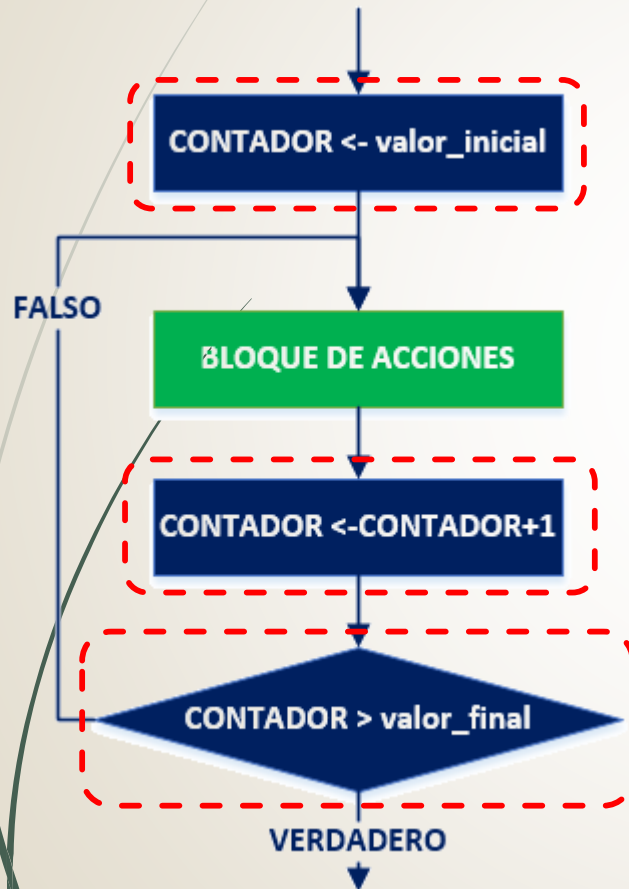
contador  $\leq$  num

- Modificación:

contador  $\leftarrow$  contador + 1



# Repetir: Controlado por Contador (1)



- **Inicialización:** se asigna el valor inicial al contador
- **Modificación del contador:** se incrementa o decrementa el valor del contador.
- **Condición de repetición:** se verifica que el contador no supere el valor\_final
  - Cond. Falsa repite
  - Cond. Verdadera finaliza



# Repetir: Controlado por Contador (2)

- Diseñe un algoritmo que calcule el producto, mediante sumas, de 2 números ingresados por el usuario. Implemente el bucle de cálculo con estructuras REPETIR y use la finalización por contador.

- Inicialización:

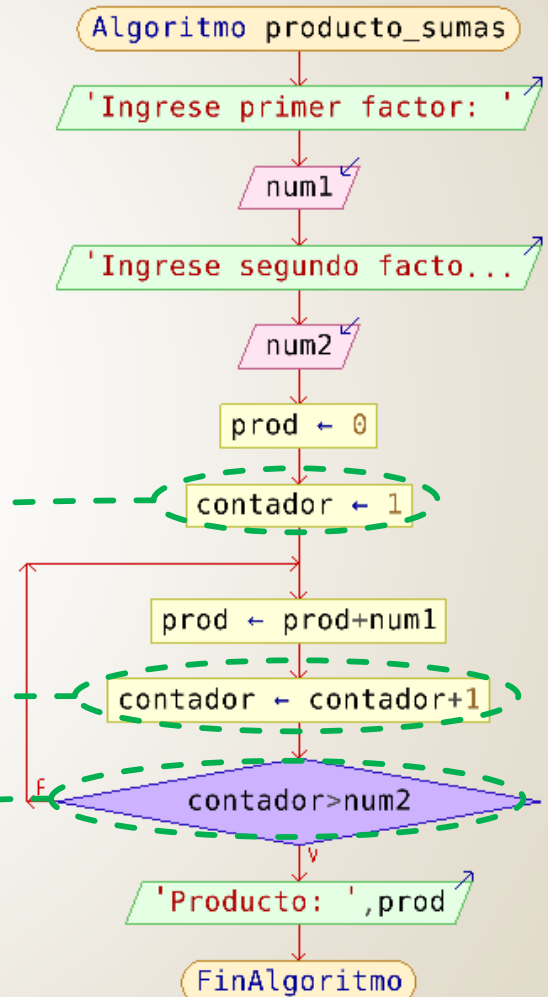
$\text{contador} \leftarrow 1$

- Modificación:

$\text{contador} \leftarrow \text{contador} + 1$

- Condición de repetición:

$\text{contador} > \text{num2}$



# Ejemplos (1)



- Diseñe un algoritmo que, mediante productos sucesivos, calcule la potencia de un número A elevado a B, siendo ambos ingresados por el usuario. Utilice estructuras MIENTRAS para el cálculo y el criterio de finalización por VALOR CENTINELA para controlar las repeticiones.
- Diseñe un algoritmo que, mediante productos sucesivos, calcule la potencia de un número A elevado a B, siendo ambos ingresados por el usuario. Utilice estructuras REPETIR para el cálculo y el criterio de finalización BANDERA para controlar las repeticiones.
- Diseñe un algoritmo que, mediante productos sucesivos, calcule la potencia de un número A elevado a B, siendo ambos ingresados por el usuario. Utilice el criterio de finalización por CONTADOR para controlar las repeticiones.

## Ejemplos (2)

- Diseñe un algoritmo que determine si un número ingresado por el usuario es primo o no.

Un número natural es primo si sólo es divisible por 1 y por sí mismo.

Por ejemplo:

2 es primo? Divisible por 1 y 2

7 es primo? Divisible por 1 y 7

9 es primo? Divisible por 1, 3 y 9

21 es primo? Divisible por 1, 3, 7, 21

23 es primo? Divisible por 1 y 23

...

## Ejemplos (3)

- Diseñe un algoritmo que determine si un número ingresado por el usuario es primo o no.

Análisis del número 7

Para  $7 / 2$  se obtiene resto 1

Para  $7 / 3$  se obtiene resto 1

Para  $7 / 4$  se obtiene resto 3

Para  $7 / 5$  se obtiene resto 2

Para  $7 / 6$  se obtiene resto 1

Sólo es  
divisible  
por 1 y 7



## Ejemplos (4)

- Diseñe un algoritmo que determine si un número ingresado por el usuario es primo o no.

Análisis del número 15

Para  $15 / 2$  se obtiene resto 1

Para  $15 / 3$  se obtiene resto 0

Para  $15 / 4$  se obtiene resto 3

Para  $15 / 5$  se obtiene resto 0

...

Para  $15 / 14$  se obtiene resto 1

Es  
divisible  
por 1, 3,  
5 y 15