

# PROGRAMACION I

TECNICATURA EN PROGRAMACION

**ESTRUCTURA DE ITERACION**

ESTRUCTURA DE  
ITERACION O  
REPETITIVA DEFINIDA  
O  
CONTROLADA POR  
CONTADOR

# OPERADOR INCREMENTAL

**++**

**int CONT;**

**CONT**<sub>(VALOR ACTUAL)</sub> **=** **CONT**<sub>(VALOR ANTERIOR)</sub> **+** **1**;

**CONT** **+=** **1**;

**CONT** **++**;

**++CONT**;

# OPERADOR DECREMENTAL

--

int **CONT**;

**CONT**(VALOR ACTUAL) = **CONT** (VALOR ANTERIOR) - **1**;

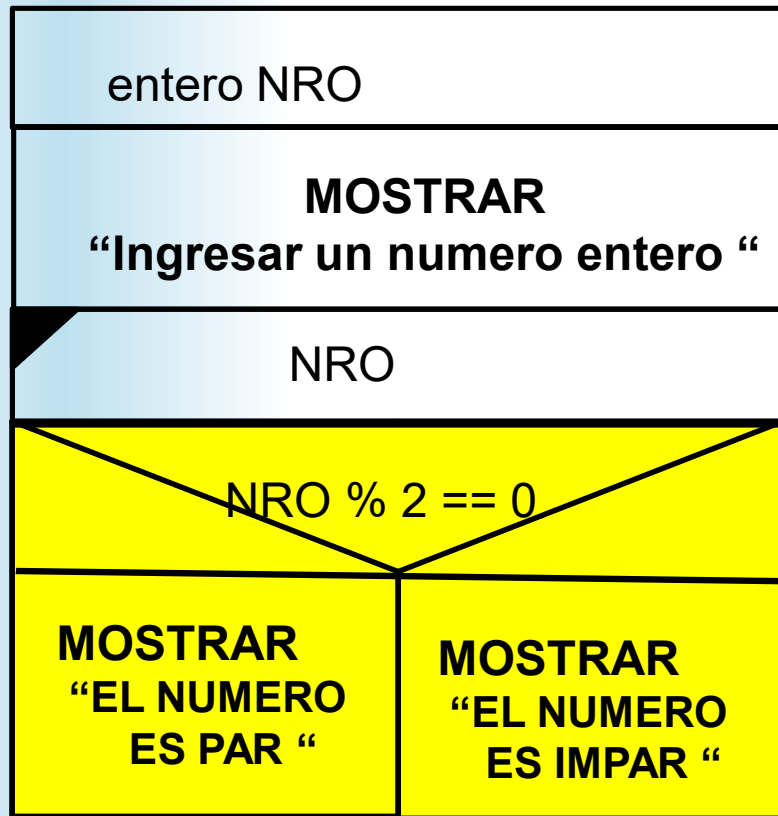
**CONT** - = **1**;

**CONT** --;

--**CONT**;

## EJEMPLO 1

SE INGRESA **UN NRO.** ENTERO. DECIR SI EL NRO. ES PAR O IMPAR



```
#include <stdio.h>
```

```
main()
```

```
{  
  int NRO;
```

```
  printf("\n INGRESAR UN NUMERO ENTERO ");
```

```
  scanf("%d",&NRO);
```

```
  if (NRO % 2 == 0 )
```

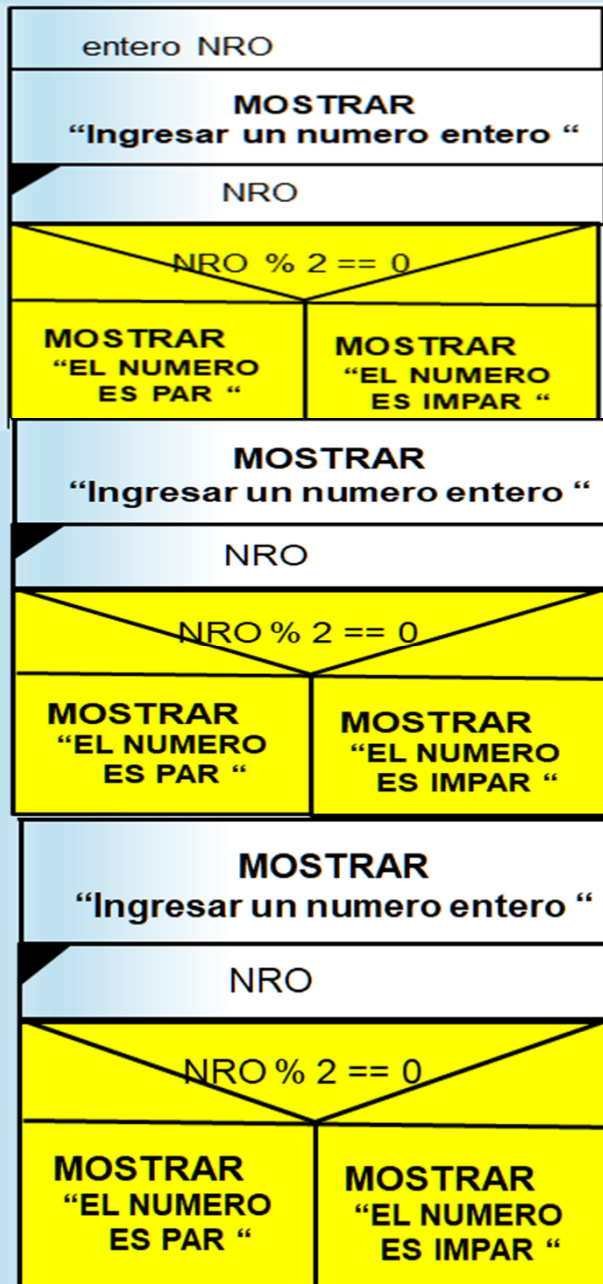
```
    printf ("\n EL NUMERO ES PAR");
```

```
  else
```

```
    printf ("\n EL NUMERO ES IMPAR");
```

```
}
```

**SE INGRESAN 3 NROS. ENTEROS DECIR SI CADA NRO. ES PAR O IMPAR**



primer NRO

**CONTADOR = 1**

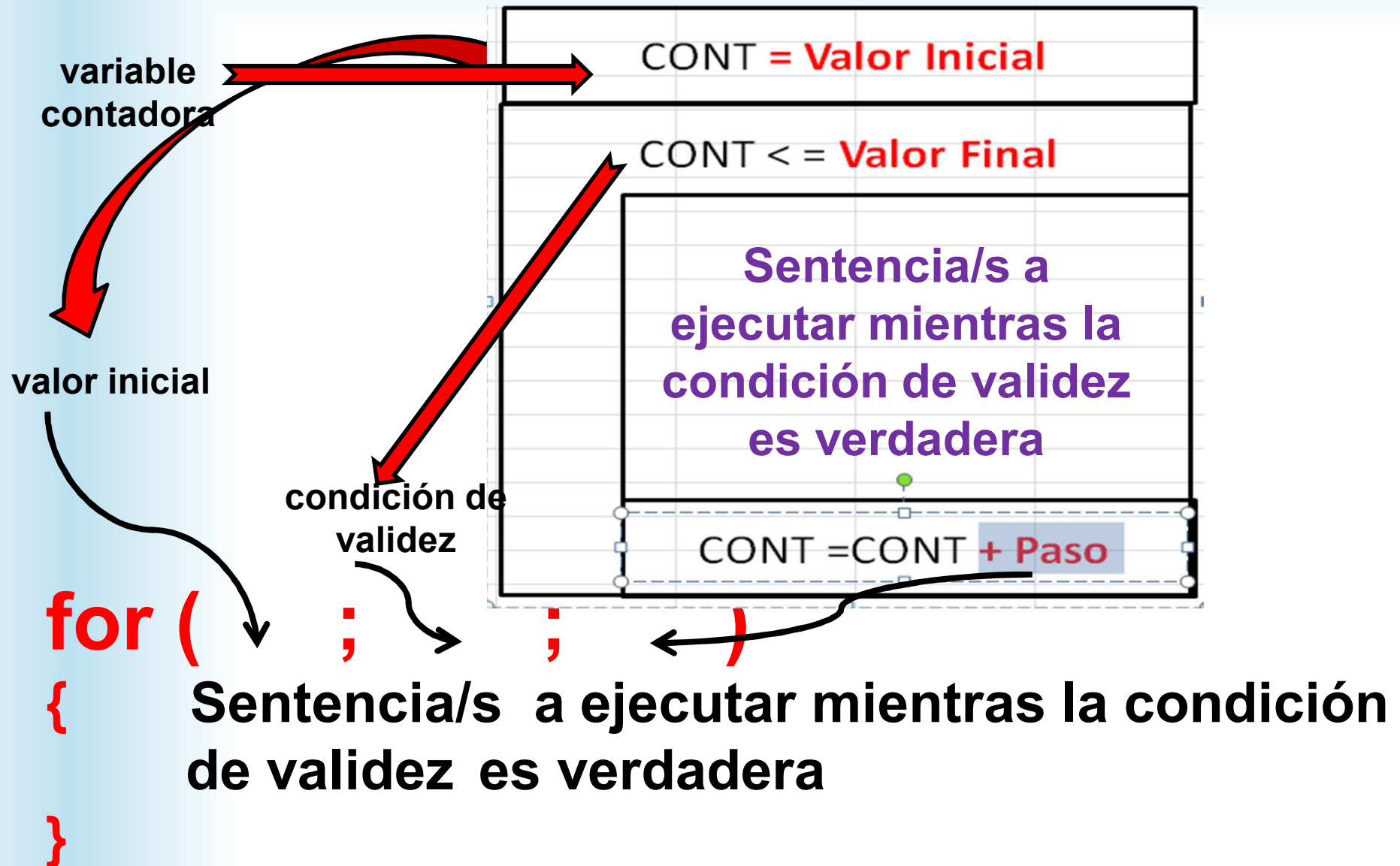
segundo NRO

**CONTADOR = 2**

tercer NRO

**CONTADOR = 3**

# ESTRUCTURA ITERATIVA DEFINIDA



CONT = 1

Inicializa el  
contador (cont =1)

CONT <= 3

verifica la condición  
de validez (CONT <= 3)

**MOSTRAR**  
"Ingresar un numero entero "

NRO

$NRO \% 2 == 0$

**MOSTRAR**  
"EL NUMERO  
ES PAR "

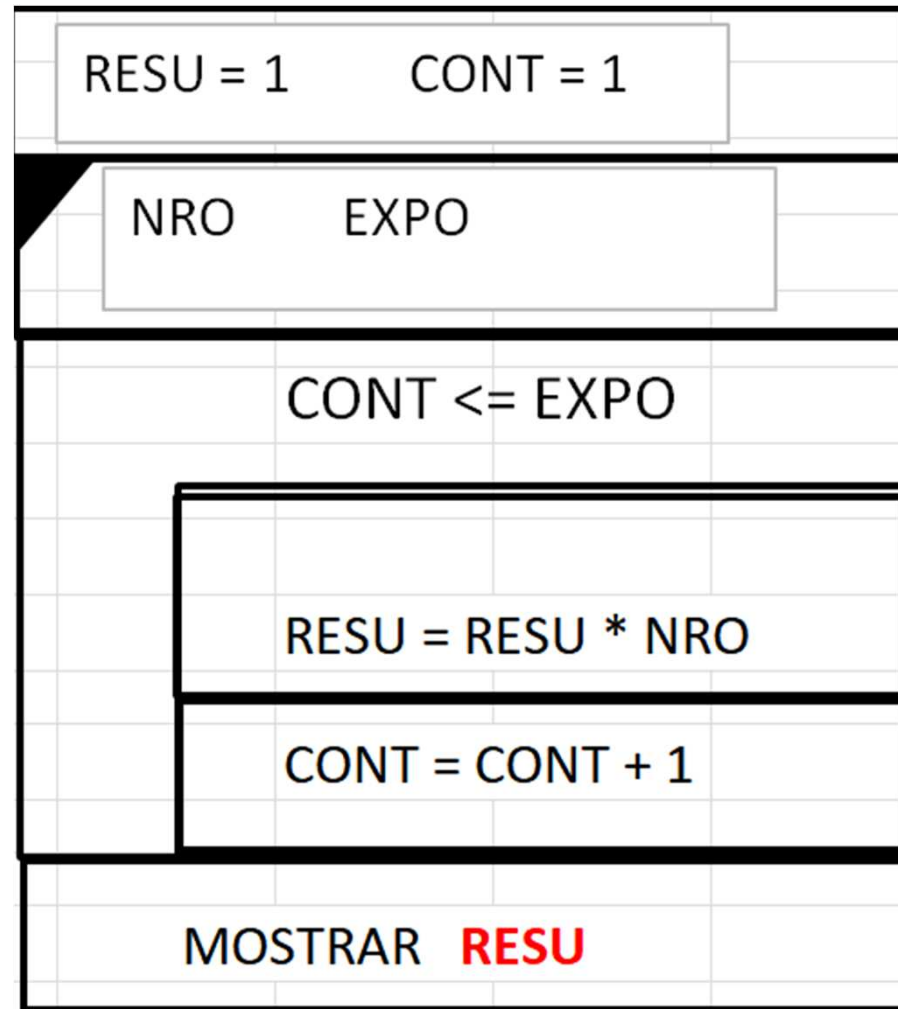
**MOSTRAR**  
"EL NUMERO  
ES IMPAR "

CONT = CONT + 1

Incrementa el  
contador ( CONT ++)



Ingresar un numero N y un exponente E ambos enteros. Mostrar la potencia E del numero N



# CONTADOR

## CARACTERISTICAS

- Se inicializa antes de todo proceso repetitivo

$$\text{CONT} = 0$$

- Dentro del proceso repetitivo

$$\text{CONT} = \text{CONT} + \text{CONSTANTE}$$

Ej.  $\text{CONT} = \text{CONT} + 1$    o    $\text{CONT} ++$

- AL finalizar el proceso repetitivo, se obtiene el resultado.

# SUMADOR O ACUMULADOR

## CARACTERISTICAS

- Se inicializa antes de todo proceso repetitivo

$$\text{SUM} = 0$$

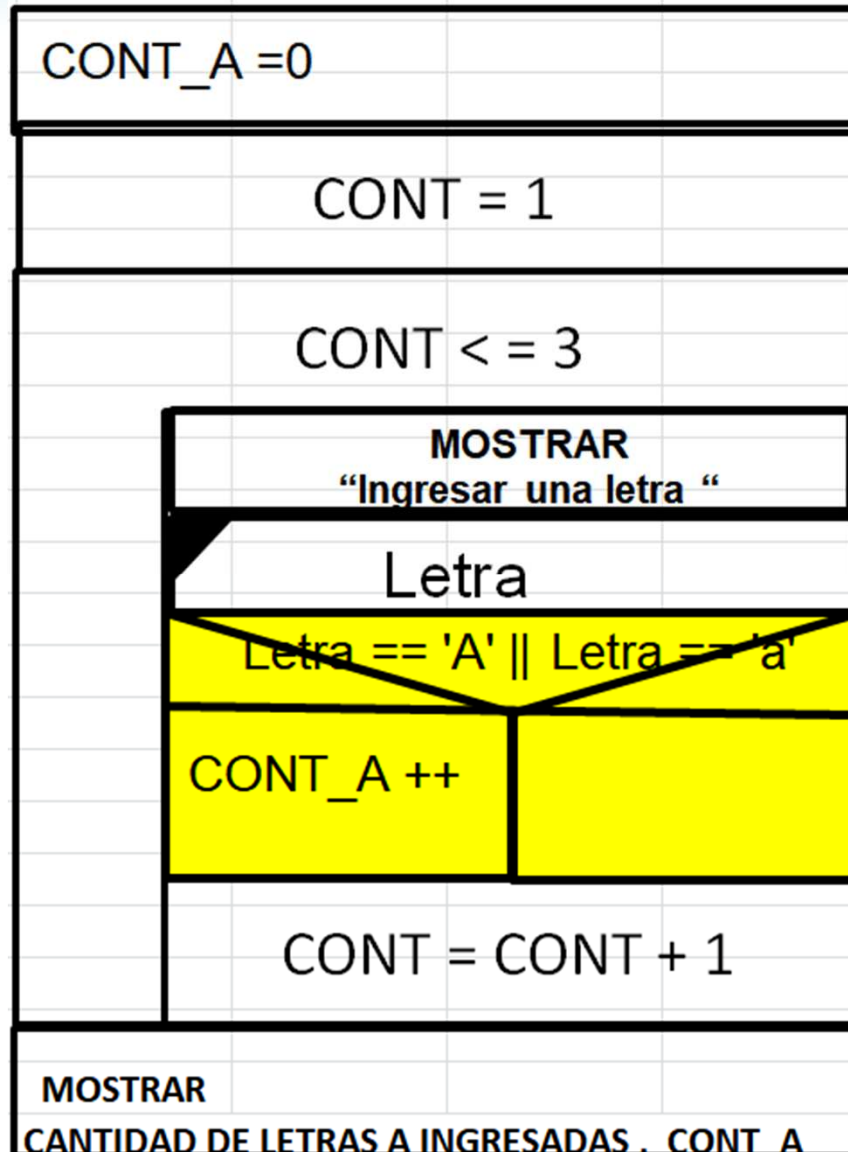
- Dentro del proceso repetitivo

$$\text{SUM} = \text{SUM} + \text{VARIABLE}$$

Ej. **SUM = SUM + NRO**    o    **SUM +=NRO**

- AL finalizar el proceso repetitivo, se obtiene el resultado.

# SE INGRESAN 10 LETRAS. INDICAR CUANTAS LETRAS 'A' o 'a' FUERON INGRESADAS



# CONVERSION DE CARACTERES

**#include <ctype.h>**

- PARA CONVERTIR DE MINUSCULA A MAYUSCULA

carácter = **toupper** (carácter)

- PARA CONVERTIR DE MAYUSCULA A MINUSCULA

carácter = **tolower** (carácter)

## EJERCITACION

- ❑ Suma los siguientes números

$$500 + 450 + 400 + 350 + \dots + 50$$

- ❑ Se ingresa un Nro. y se informa la tabla de multiplicar del 1 al 10.
- ❑ Realizar las tablas de multiplicar de los todos los números del 1 al 10.

# ESTRUCTURA DE ITERACION O REPETICION CONDICIONADA

**SE INGRESAN NUMEROS ENTEROS. FINALIZA CON CERO.**  
**SUMAR LOS NUMEROS INGRESADOS**

### PRUEBA DE ESCRITORIO

**NRO**

**12**

Primer Numero  
ingresado

SE PREGUNTA SI EL NUMERO  
INGRESADO ES DISTINTO DE CERO

**50**

**14**

**-10**

**40**

**0**

SE INGRESAN NROS.  
MIENTRAS NO SEA CERO

ULTIMO Numero  
ingresado

**SUM**

**0**

Valor inicial  
(acumulador)

**12**

**62**

**76**

**66**

**106**

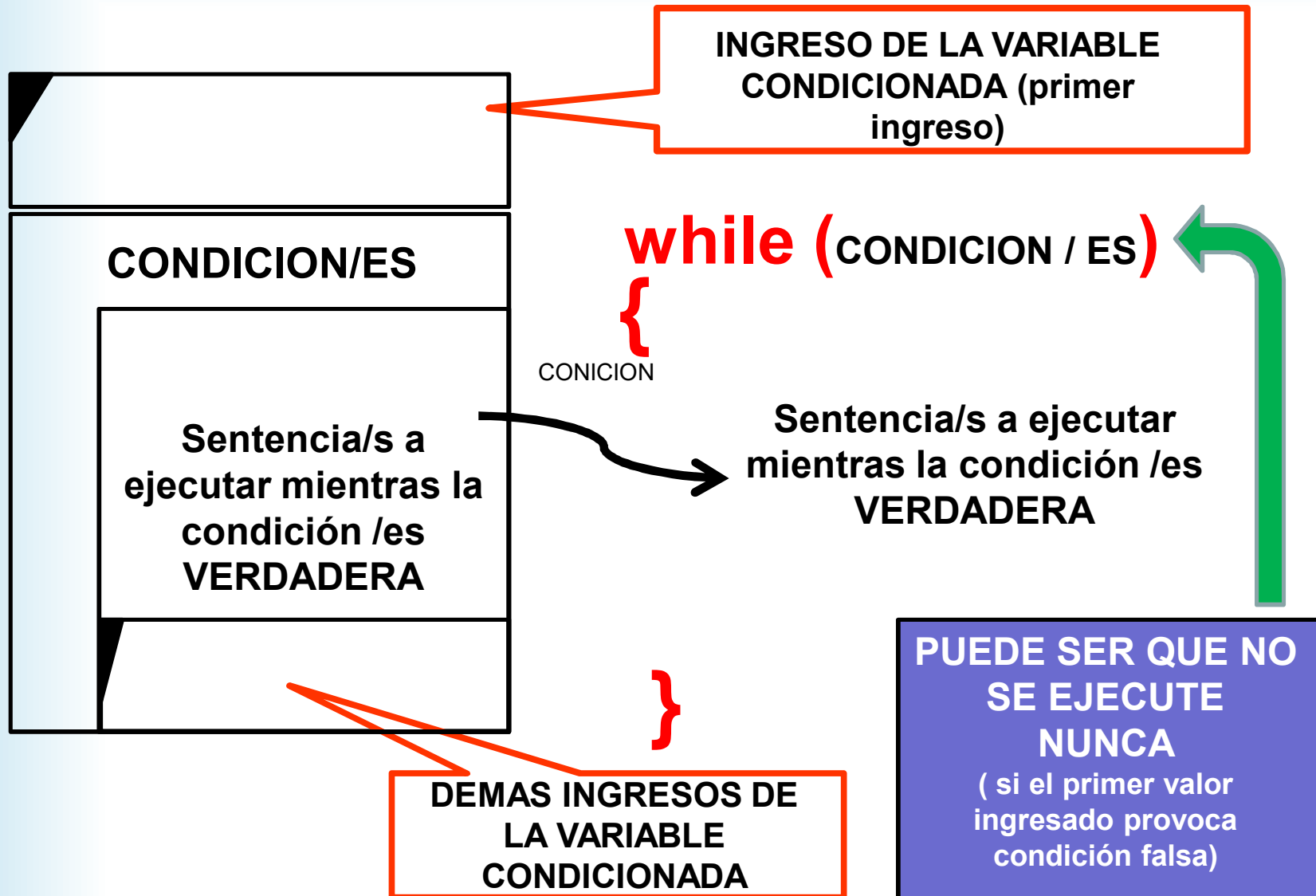
SE ACUMULA LOS  
VALORES DE LOS  
NUMEROS INGRESADOS

RESULTADO  
DE LA SUMA

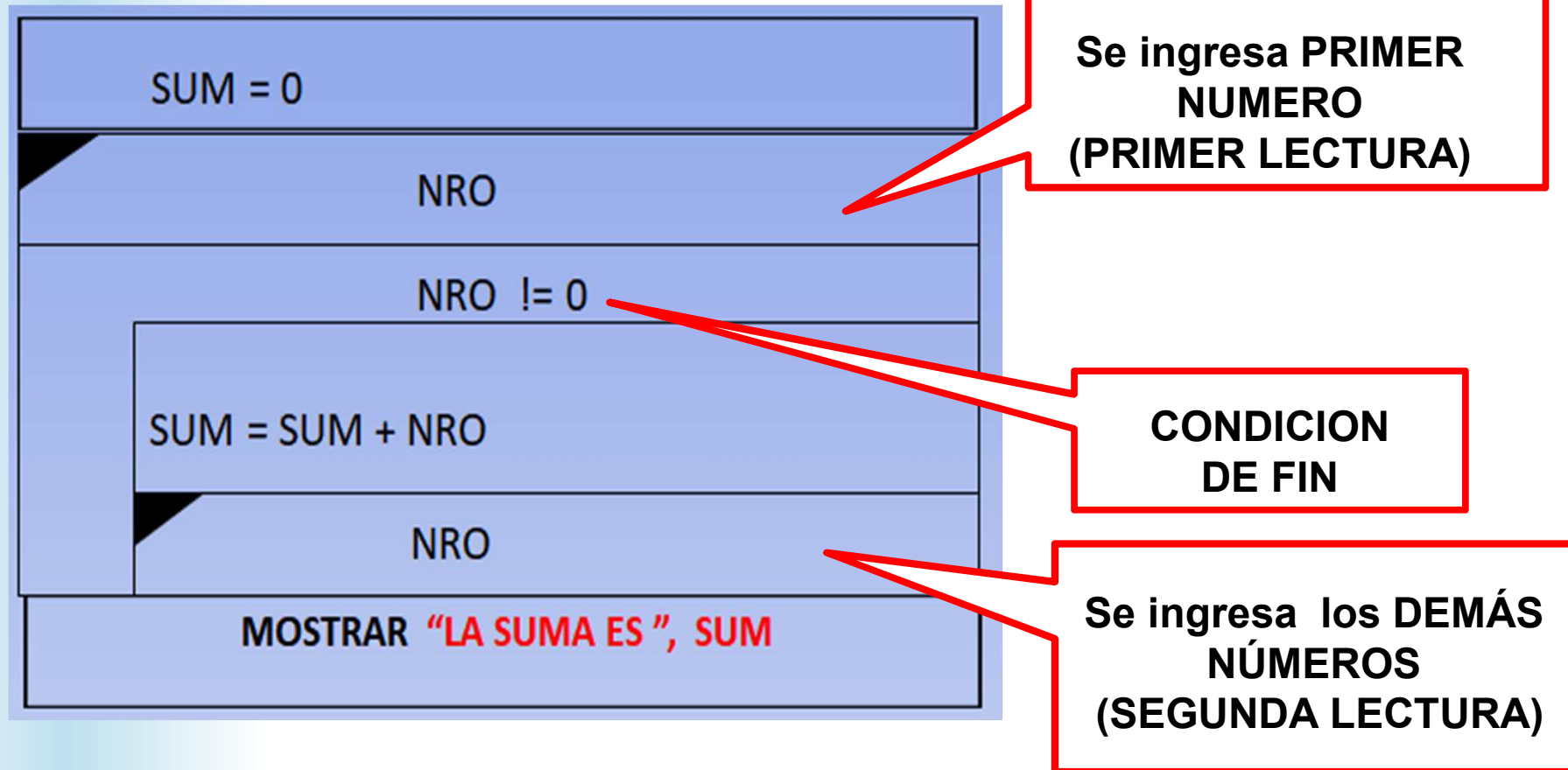
**CONDICION DE FIN**



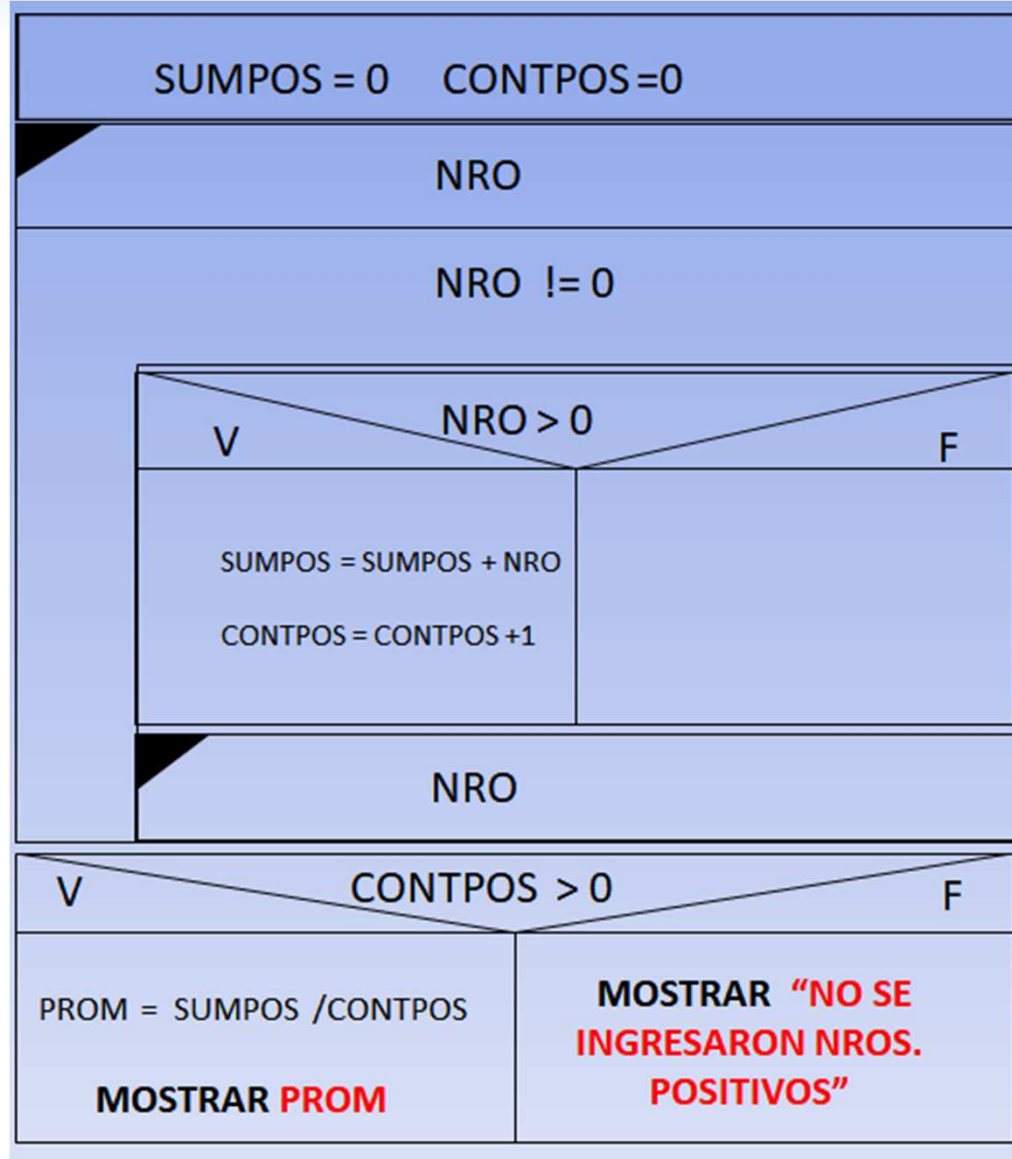
# ESTRUCTURA ITERATIVA CONDICIONADA



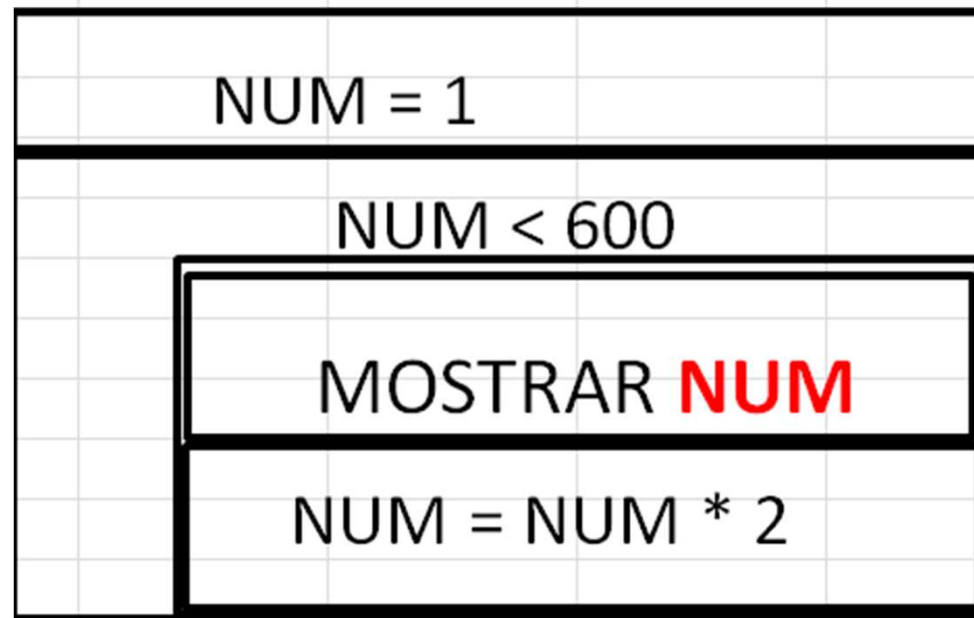
# SUMAR NUMEROS. FINALIZA CON NUMERO CERO



**Se ingresan números. Determinar el promedio de numeros positivos. El ingreso finaliza con numero igual a cero.**



**Confeccionar un programa que calcule e informe los valores de las potencias de 2 cuyo resultado sean menores que 600 ( 1 2 4 ....)**



## EJERCITACION

1. Se ingresan Nros. enteros. Finaliza con cero o negativo. Determinar el promedio de los números ingresados
2. Se ingresan pares de números enteros. Termina con el par (0,0). Indicar la cantidad de pares iguales ingresados.
3. Se ingresan números enteros. Termina con el ingreso de un numero cuyo último dígito es 9. Determinar la cantidad de números de 3 cifras ingresados.



# HACER LA EJERCITACION