

Control de Flujo



Profesora Teresa Tapia Soto



Contenidos

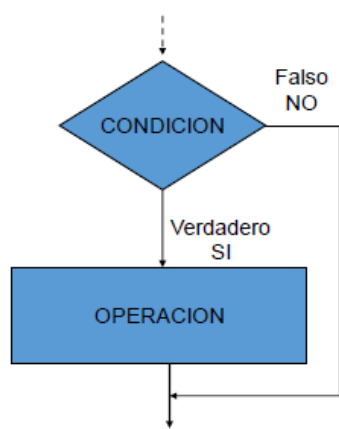
- ❖ Estructuras Selectivas: Simple, Doble y Múltiple
- ❖ Estructuras de Repeticiones: For / While



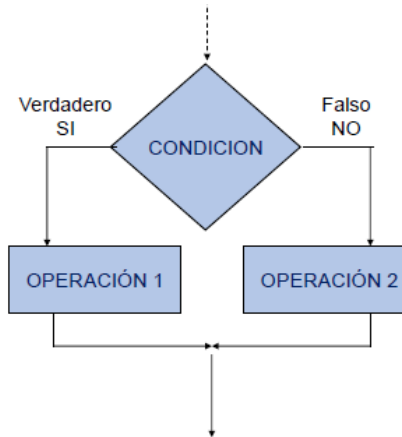
Estructuras Selectivas

Estructuras selectivas: son Formas de controlar un flujo a través de la toma de decisiones y así poder señalar un camino alternativo a seguir.

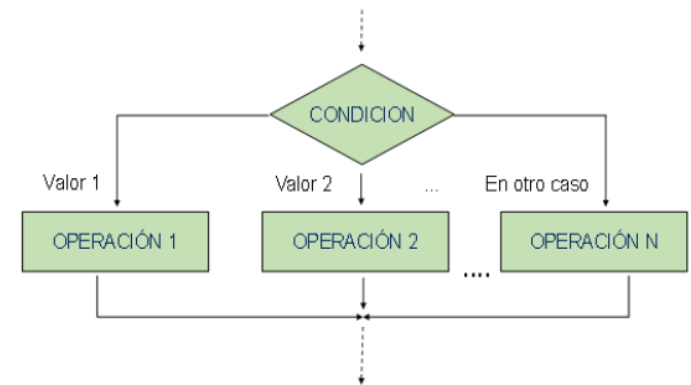
“Esta toma de decisión expresada en un rombo en el diagrama de flujo se basa en la evaluación de una o más condiciones que nos señalaran como alternativa el camino a seguir”.



a) Simple



b) Doble

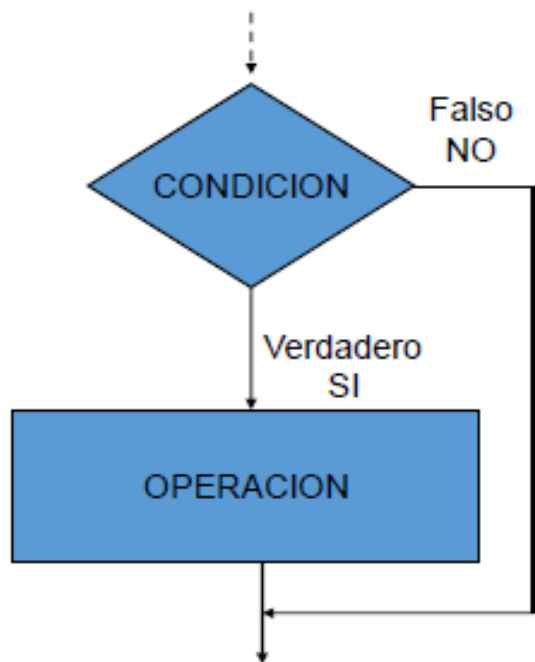


c) Múltiple

Estructura Selectiva Simple

Una **condición** es **cualquier expresión** compuesta por operandos, operadores lógicos y operadores relacionales **que da como resultado un valor booleano (Verdadero o Falso)**.

Evaluar la *condición*, si el resultado es verdadero (*entonces*) ejecuta el conjunto de acciones asociadas a él (operaciones o acciones), caso contrario, si es falso, no hace nada



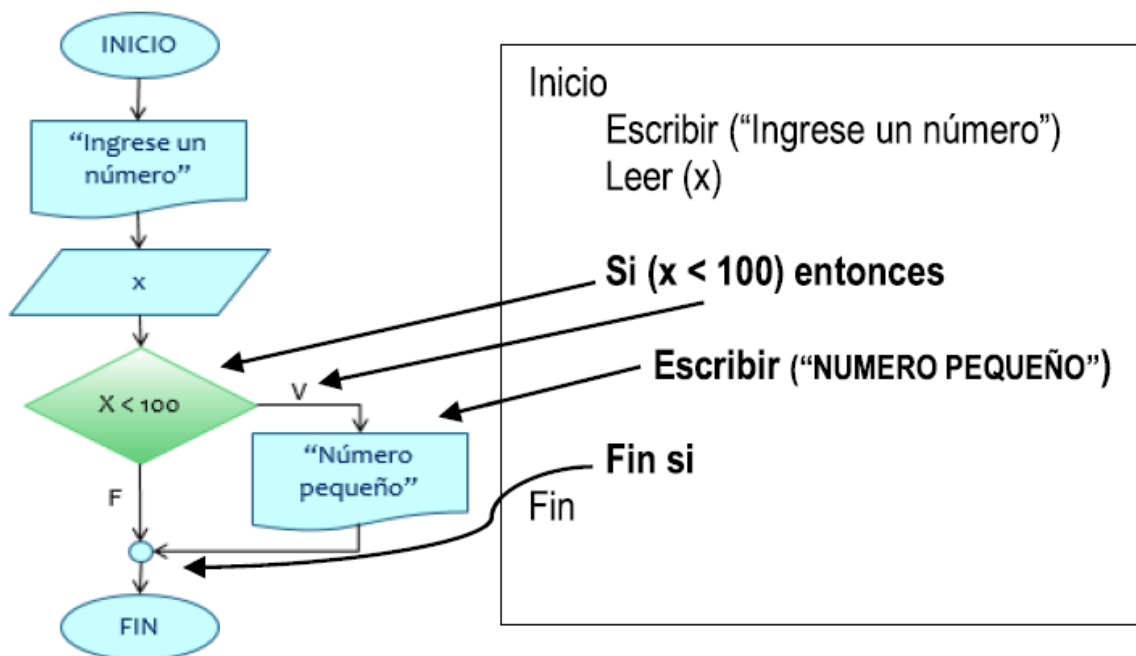
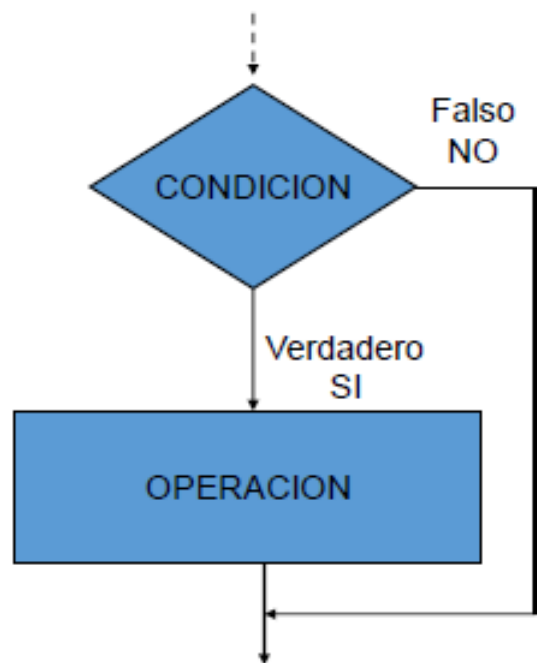
Ejemplos de toma de decisiones al evaluar una condición:

- **SI** la temperatura es 25°
 - compraré un quitasol
- **SI** el promedio de notas es menor a 40:
 - Estudiaré 8 horas diarias
 - Solicitaré ayuda a mi compañero
 - Realizaré un 20% más de ejercicios

Estructura Selectiva Simple

Una **condición** es **cualquier expresión** compuesta por operandos, operadores lógicos y operadores relacionales **que da como resultado un valor booleano (Verdadero o Falso)**.

Evaluar la *condición*, si el resultado es verdadero (*entonces*) ejecuta el conjunto de acciones asociadas a él (operaciones o acciones), caso contrario, si es falso, no hace nada

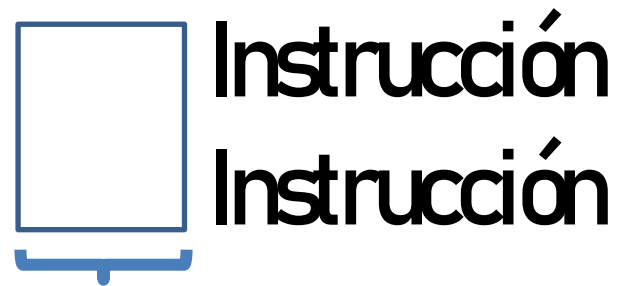


Estructura Selectiva Simple

En **Python** para evaluar la condición usaremos la sentencia **if**

Sintaxi:

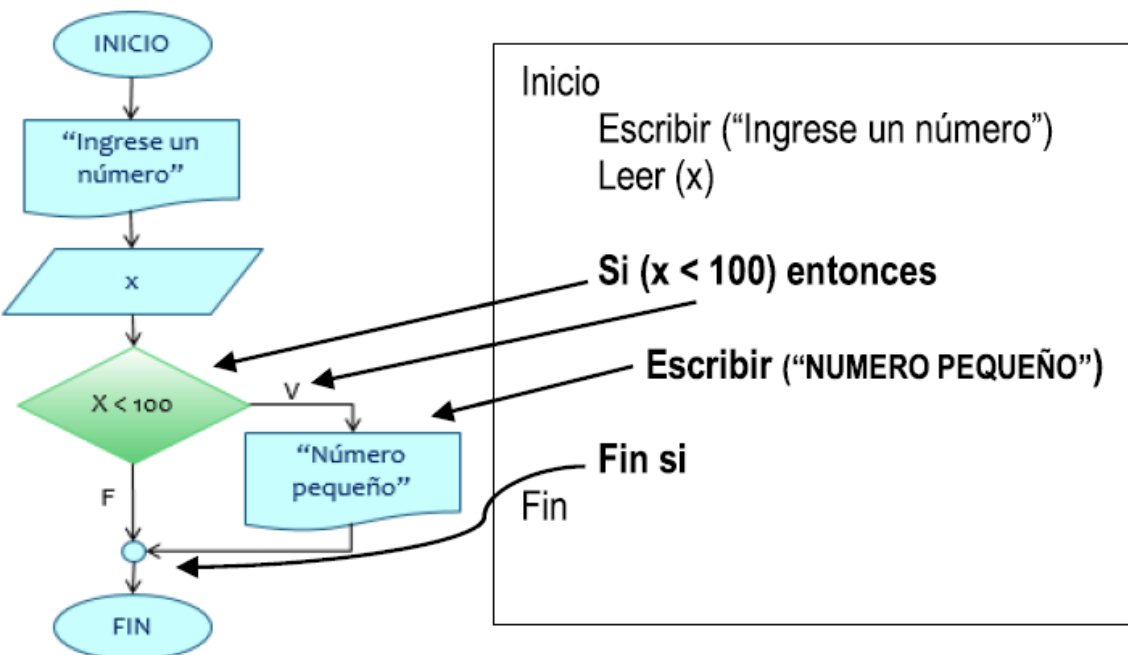
if condición:



Indentación, es equivalente a 4 espacios

- Todas las instrucciones dentro del if que se encuentran indentadas se ejecutaran si la condición es verdadera.
- Python IDLE realiza la indentación automáticamente

Estructura Selectiva Simple



En Python sería:

```
print("Ingrese un Número: ")  
x=int(input())  
if x < 100:  
    print("El número es pequeño")
```

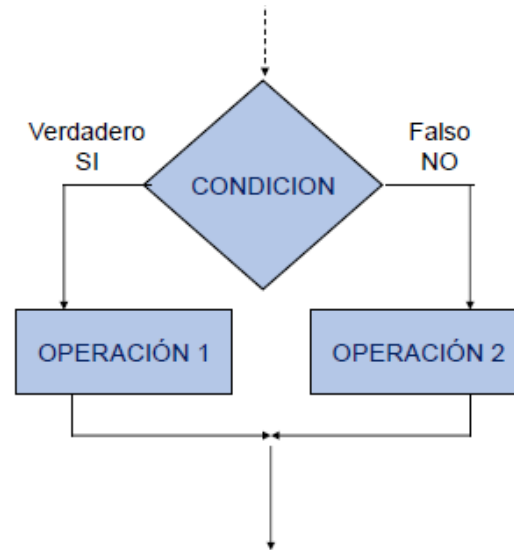
Estructura Selectiva Simple

Ejemplos:

- 1) Dado el sueldo de un trabajador, aplique un aumento del 15% si su sueldo es inferior a \$1000. Imprima el sueldo que percibirá.
- 2) Construya un programa que, dados tres números, muestre el mensaje “IGUALES” si la suma de los dos primeros es igual al otro número.

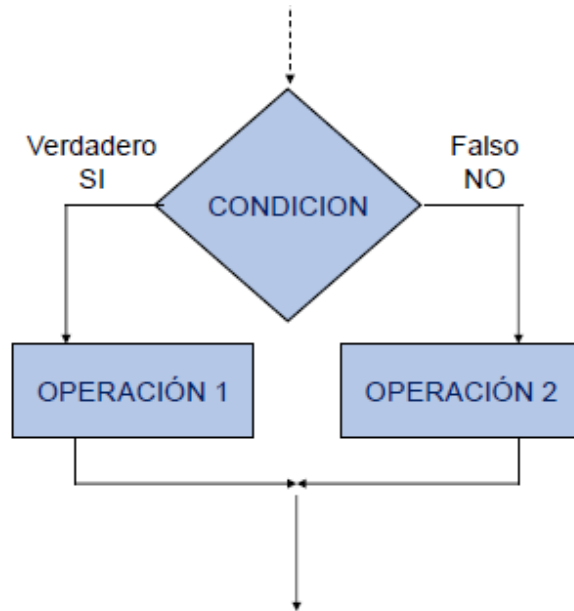


Estructuras Selectivas Doble



b) Doble

Estructura Selectiva Doble



Ejemplos de toma de decisiones al evaluar una condición:

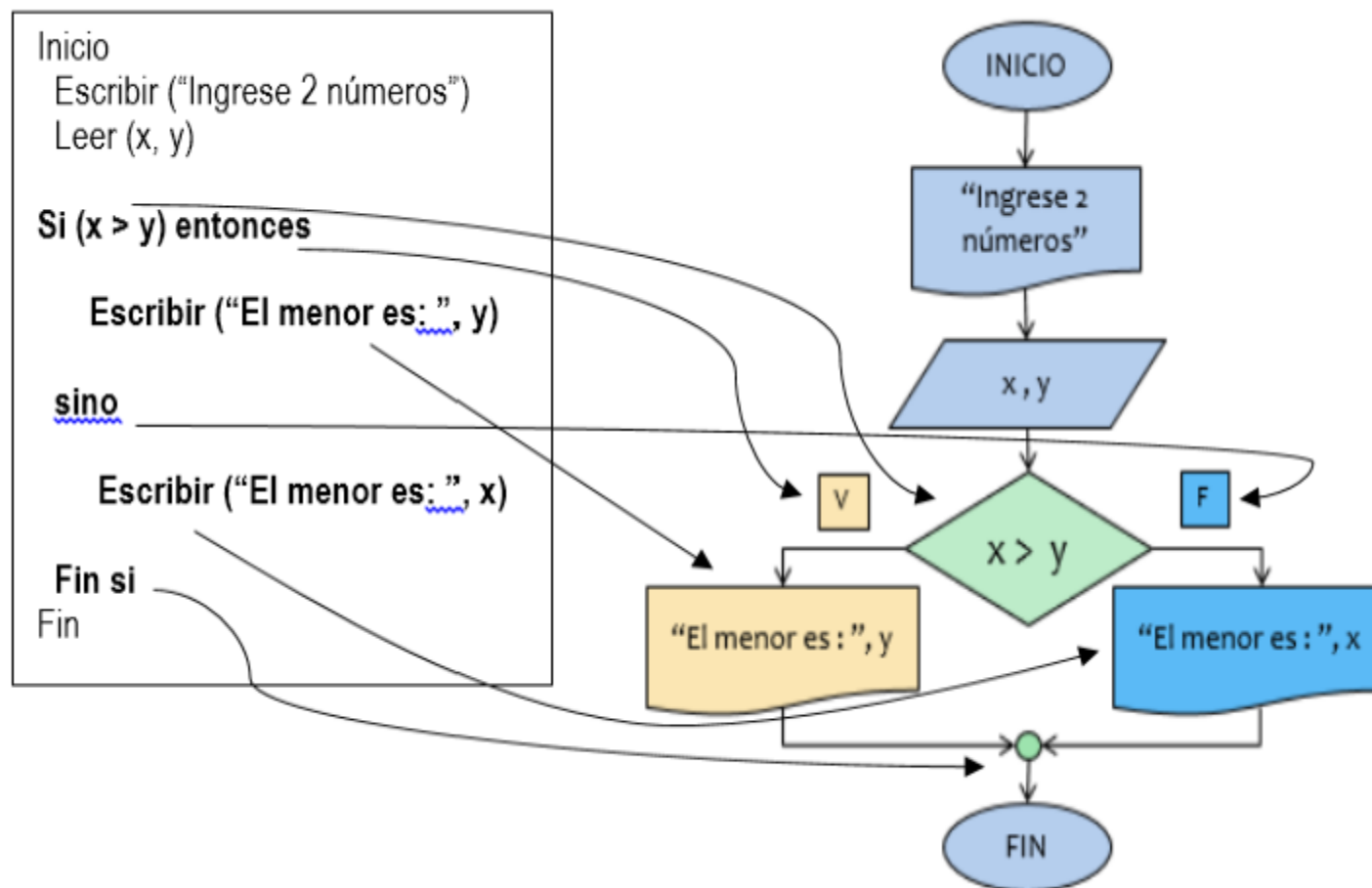
Ejemplo 1:

- **SI** mi calificación es ≥ 40
 - Inscribir nuevas asignaturas
- **SINO**
 - Inscribir curso reprobado

Ejemplo 2:

- **SI** la edad de la persona ≥ 18 :
 - Comprar boleto de avión
- **SINO**
 - Entregar autorización de Padres
 - Indicar Apoderado

Estructura Selectiva Doble



Estructura Selectiva Doble

En **Python** para evaluar la condición usaremos la sentencia **if / else**

Sintaxi:

if condición:

Instrucción

Instrucción

else:

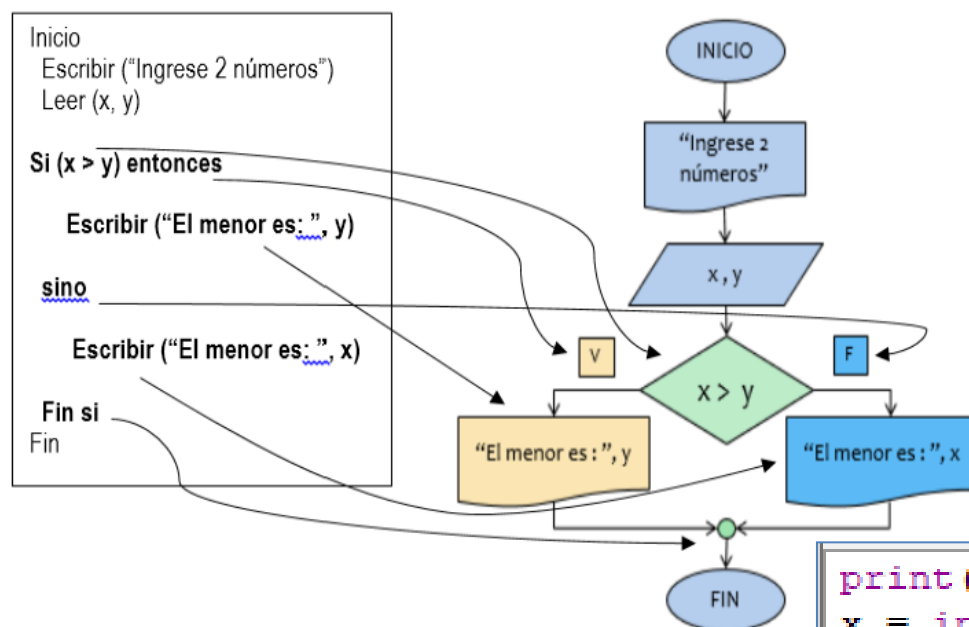
Instrucción

Instrucción

Instrucción #esto se ejecuta fuera de la condición



Estructura Selectiva Doble



En Python sería:

```
print("Ingrese 2 números: ")
x = int(input("Primer Número: "))
y = int(input("Segundo Número: "))
```

```
#NOTA: No olvidarse de las indentación.
if x > y : #Esto esta dentro del IF
    print("El menor es:", y)
else: #Esto esta dentro del ELSE
    print("El menor es:", x)
```

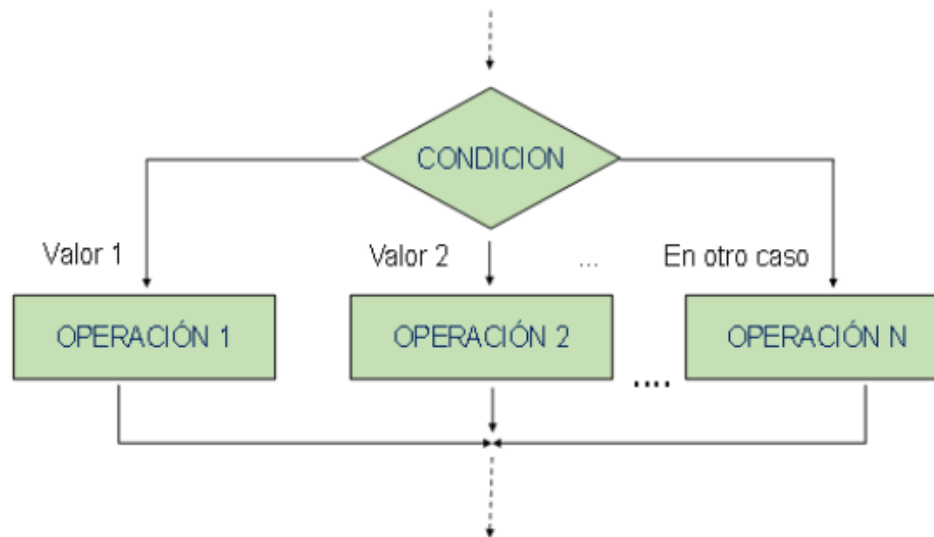
Estructura Selectiva Doble

Desarrollar:

- 1) Dado dos números “a” y “b”, muestre sus valores en orden de menor a mayor.
- 2) Construya un algoritmo que, dados tres números, muestre el mensaje “IGUALES” si la suma de los dos primeros es igual al otro número y el mensaje “DISTINTOS” en caso contrario.



Estructuras Selectivas Múltiple



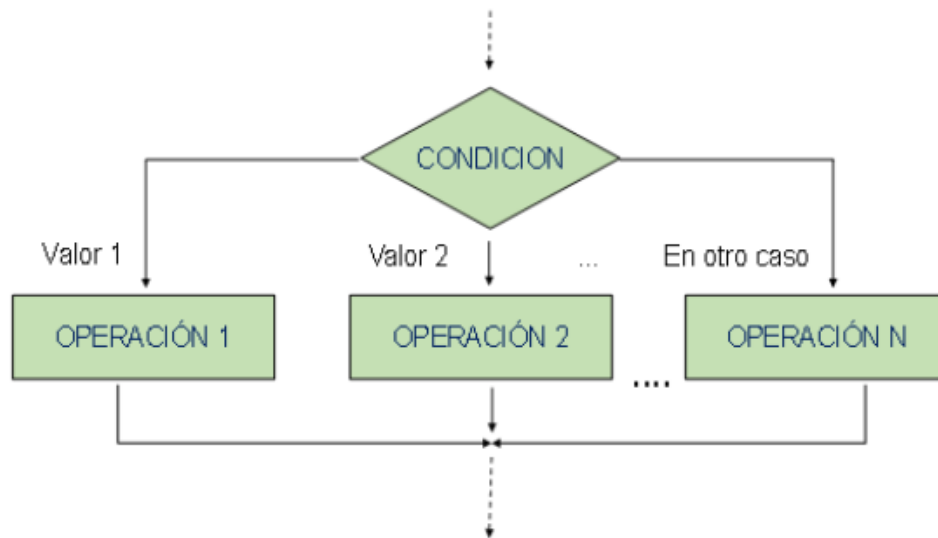
c) Múltiple



Estructura Selectiva Múltiple

Se evalúa una variable **selector** que puede tomar **diversos valores**. Según el valor que la variable selector tome en cada momento se ejecutan las acciones correspondientes al valor.

En realidad, equivale a un **conjunto de condiciones anidadas**. En un lenguaje de programación se le conoce como **Case, Select case o Switch**.



SI <Selector> igual

- <Valor 1>: <Operación 1>
- <Valor 2>: <Operación 2>
- <Valor 3>: <Operación 3>
- <En otro caso>: <Operación n+1>

Nota: Dentro del selector múltiple el término **<En otro caso>** se ejecuta cuando la expresión no toma ninguno de los valores que aparecen antes.

Estructura Selectiva Múltiple

En **Python** no existe SWITCH como en otros lenguajes. La selectiva Múltiple se resuelve con **Diccionarios** (otro tipo de dato que veremos más adelante)

Ejemplo: Ingresar el número de mes e imprime su nombre.

```
switcher = {
    1: "Enero",
    2: "Febrero",
    3: "Marzo",
    4: "Abril",
    5: "Mayo",
    6: "Junio",
    7: "Julio",
    8: "Agosto",
    9: "Septiembre",
    10: "Octubre",
    11: "Noviembre",
    12: "Diciembre" }

argument = int( input("Ingrese número de mes: "))

#para acceder a los elementos se hace uso de la funcion .get del diccionario
# diccionario.get(AlgúnArgumento, mensaje por Defecto)

nombreDeMes = switcher.get(argument, "Mes invalido")
print(nombreDeMes)
```

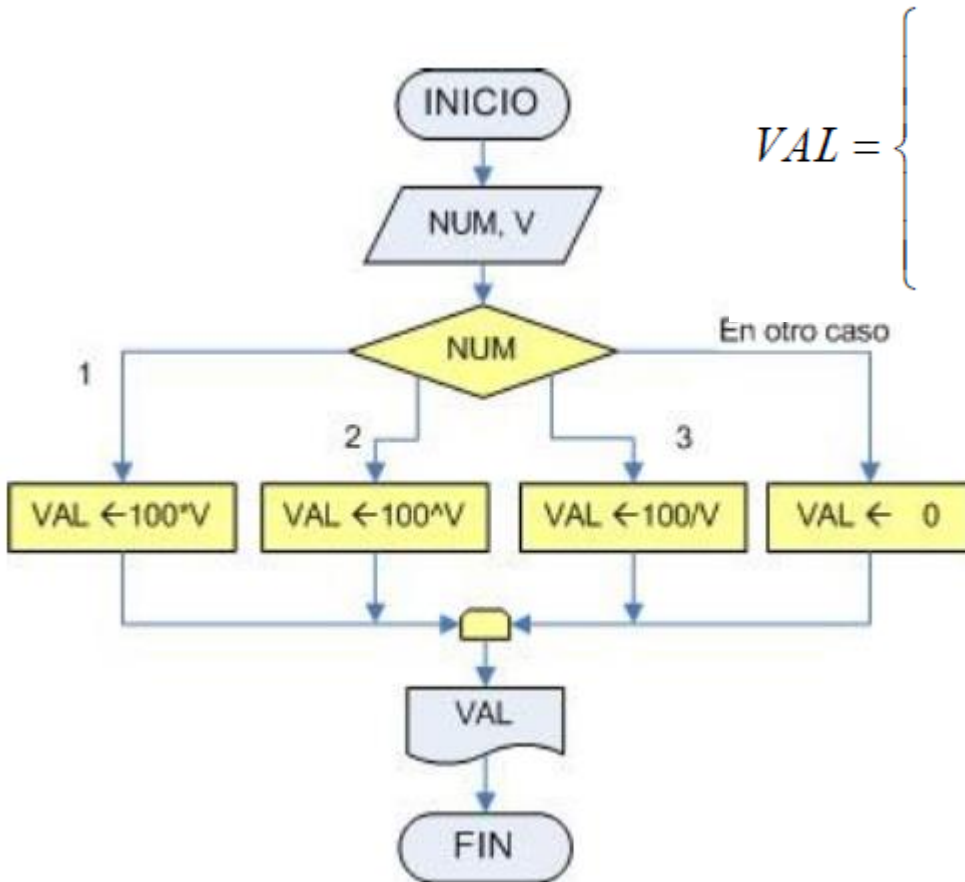
```
Ingrese número de mes: 7
Julio
```



Estructura Selectiva Múltiple

Desarrollar: Calcular la función con los datos entrada NUM y V:

$$VAL = \left\{ \begin{array}{ll} 100 * V & \text{si_NUM} = 1 \\ 100^V & \text{si_NUM} = 2 \\ 100 / V & \text{si_NUM} = 3 \\ 0 & \text{para_cualquier_otro_valor} \end{array} \right\}$$



Estructura Selectiva Múltiple

Para sustituir un **Switch o IF's anidados** se puede hacer uso de palabra reservada **ELIF**

Sintaxi:

if condición:

☐ Instrucción

...

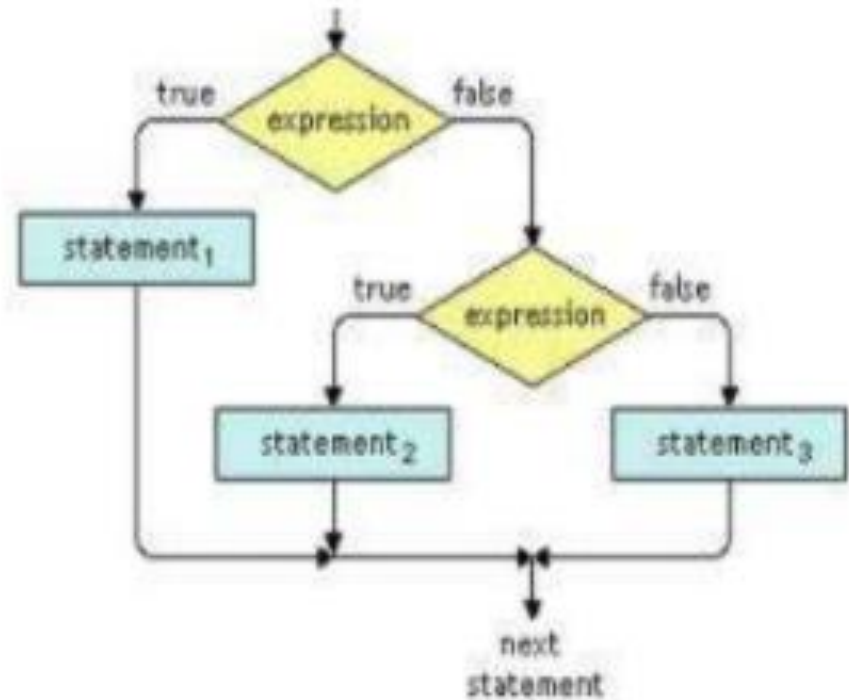
elif condición2:

☐ Instrucción

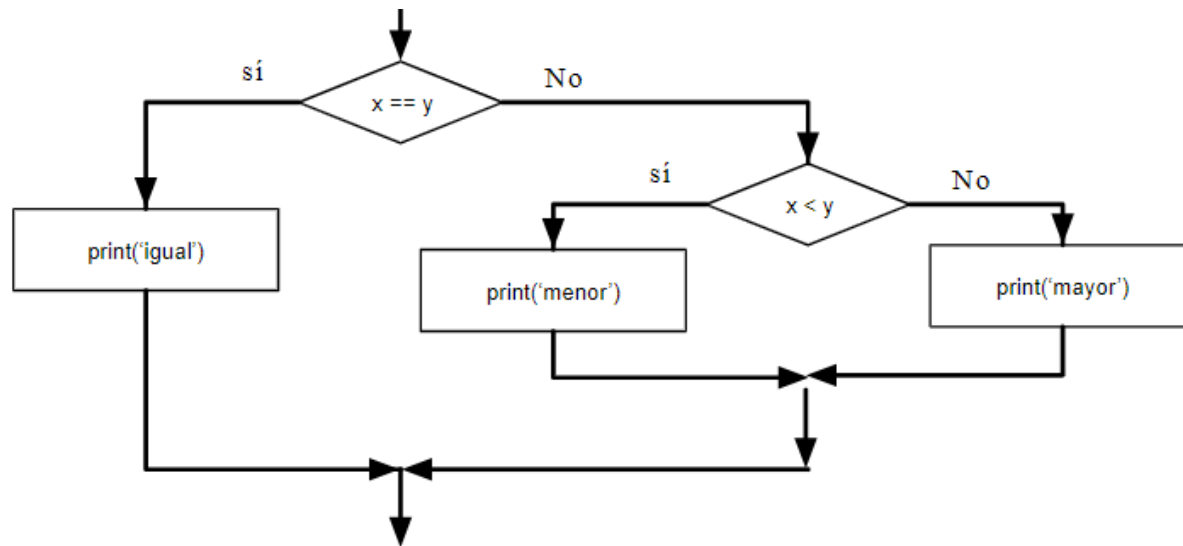
...

else:

☐ Instrucción



Estructura Selectiva Múltiple



```
x=input("Ingresar x : ")
y=input("Ingresar y : ")

if x == y:
    print("Los números son iguales")
elif x < y:
    print(" x es menor a y ")
else:
    print("y es mayor a x ")
```

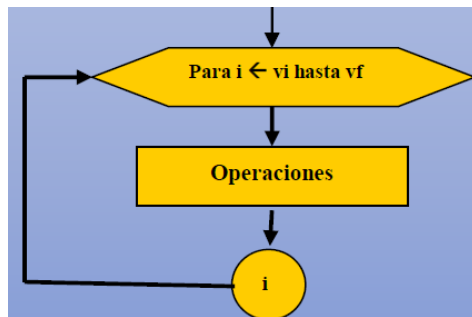
```
Ingresar x : 2
Ingresar y : 4
x es menor a y
```

Estructuras Repeticiones

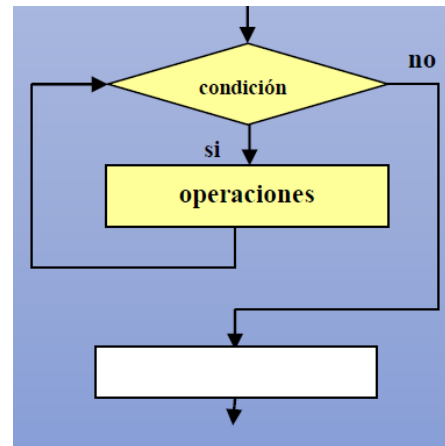
Las estructuras **que repiten una secuencia de instrucciones** un número determinado de veces se denominan **bucles**. En algunos algoritmos podemos establecer a priori que el bucle se repetirá un número de veces y en otros no.

Estructuras:

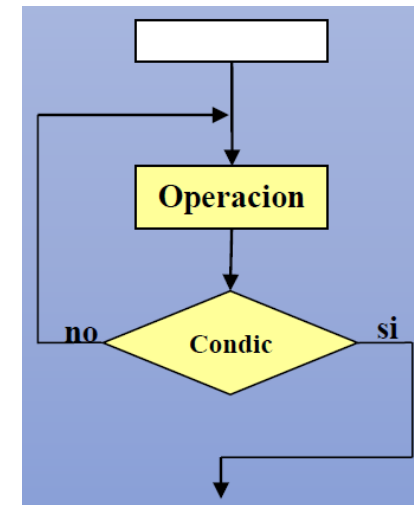
- PARA o DESDE
- MIENTRAS y REPETIR.



a) for



b) While



c) Repetir

Estructura de repetición: for

Un ciclo **for** recorre cada uno de los elementos(iterador) de una **colección (lista, tuplas, cadenas)**, o cualquier objeto iterable.

Sintaxi:

for <iterador> **in** <colección>:

 Instrucción

...

```
for caracter in "Hola mundo":  
    print(caracter)
```

H
o
l
a

m
u
n
d
o

Estructura de repetición: for y range()

Ejemplo: Imprimir los números del 1 al 4

```
for x in range(5):  
    print("x: ", x)
```

range: es una función que crea una lista de acuerdo a sus parámetros.

#si tiene un solo parámetro: de cero hasta menor que el numero
range(10) #Crea una lista desde 0 hasta 9: [0,1,2,3,4,5,6,7,8,9]

#range(inicio, fin)
range(1, 10) #Crea una lista desde 1 hasta 9: [1,2,3,4,5,6,7,8,9]

#range(inicio, fin, incremento/decremento)
range(1, 10, 2) #Crea una lista desde 1 a 9 de 2 en 2: [1,3,5,7,9]
#NOTA: El último elemento no se toma en cuenta. Si es 10, se cuenta hasta 9



Estructura de repetición: for y range()

Ejemplos de range()

<code>range(9)</code>	0, 1, 2, 3, 4, 5, 6, 7, 8
<code>range(3, 13)</code>	3, 4, 5, 6, 7, 8, 9, 10, 11, 12
<code>range(3, 13, 2)</code>	3, 5, 7, 9, 11
<code>range(11, 4)</code>	ningún valor
<code>range(11, 4, -1)</code>	11, 10, 9, 8, 7, 6, 5



Estructura de repetición: for y break

Break: Sentencia que permite Salir de un Bucle

```
>>> for i in range(10):  
    print(i)  
    if (i==4):  
        break
```

```
0  
1  
2  
3  
4
```



Estructura de repetición: for y continue

continue: Permite detener la iteración actual y saltar a la siguiente

```
>>> for i in range(10):  
    if i==5:  
        continue  
    print(i)
```

```
0  
1  
2  
3  
4  
6  
7  
8  
9
```



Estructura de repetición: While

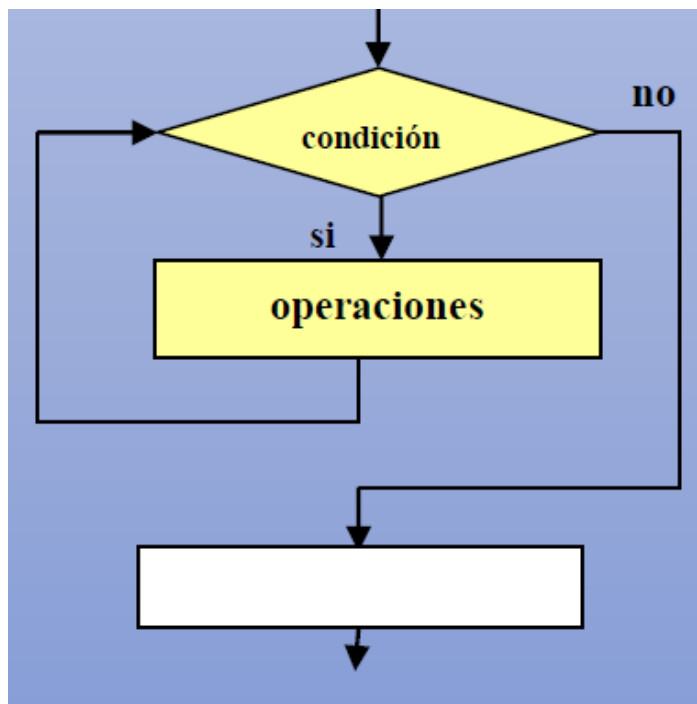
La condición del bucle se **evalúa al principio, antes de entrar en él**. Si la condición es verdadera, comenzamos a ejecutar las acciones del bucle y después de la última volvemos a preguntar por la condición.

Sintaxi:

while condición:

Instrucción

...



```
>>> i=1
>>> while i<10:
>>>     print(i)
>>>     i=i+1
```

```
1
2
3
4
5
6
7
8
9
>>> |
```

Estructura de repetición: While

break y continue: también tienen el mismo comportamiento que presentan en la sentencia for



Control de Flujo



Profesora Teresa Tapia Soto



