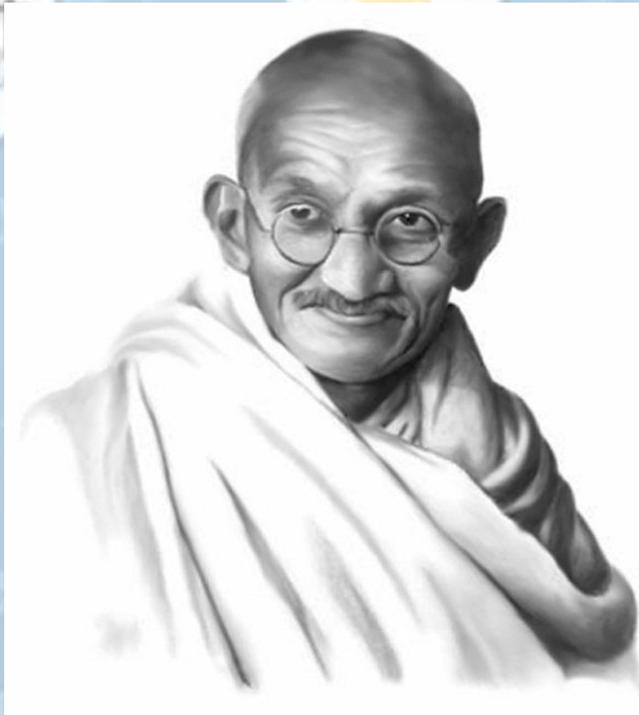


# FUNDAMENTOS DE PROGRAMACIÓN



*“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado.*

*Un esfuerzo total es una victoria completa”.*

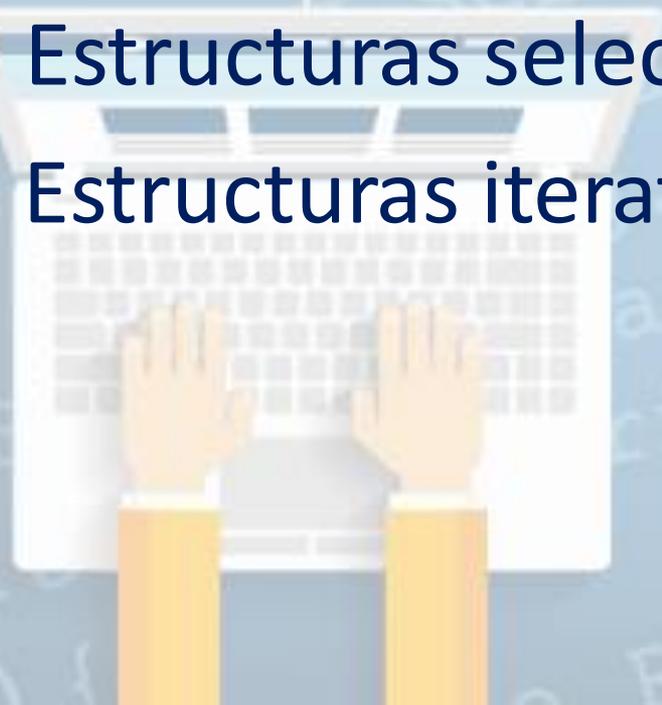
*Mahatma Gandhi.*

# UNIDAD III. CONTROL DE FLUJO

3.1 Estructuras secuenciales

3.2 Estructuras selectivas

3.3 Estructuras iterativas.



# Competencia específica

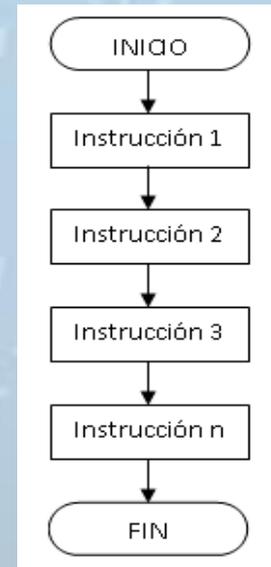
Conoce y aplica las estructuras condicionales y repetitivas de un lenguaje de programación para resolver problemas reales.



# 3.1 Estructuras secuenciales

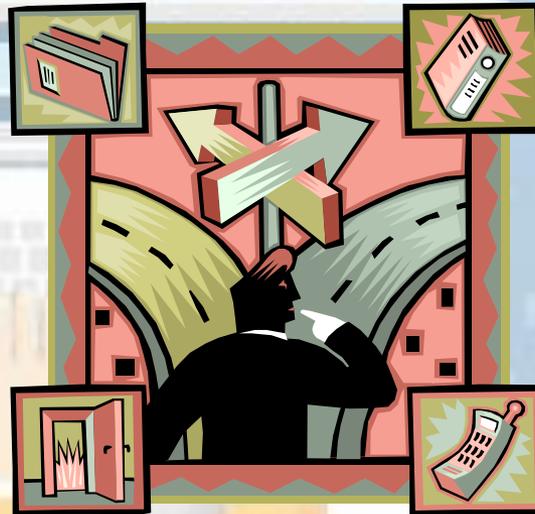
- El orden de ejecución de las sentencias o instrucciones de un programa es secuencial, en el orden en que aparecen escritas en el programa.
- La estructura secuencial está formada por una sucesión de instrucciones, cada una de las cuales están separadas por el carácter punto y coma (;).
- Las instrucciones se suelen agrupar en bloques, definidos cada uno por el carácter llave de apertura ({) para marcar el inicio del mismo, y el carácter llave de cierre (}) para marcar el final.
- Ejemplo:

```
{  
instrucción 1;  
instrucción 2;  
instrucción 3;  
}
```



## 3.2 Estructuras selectivas

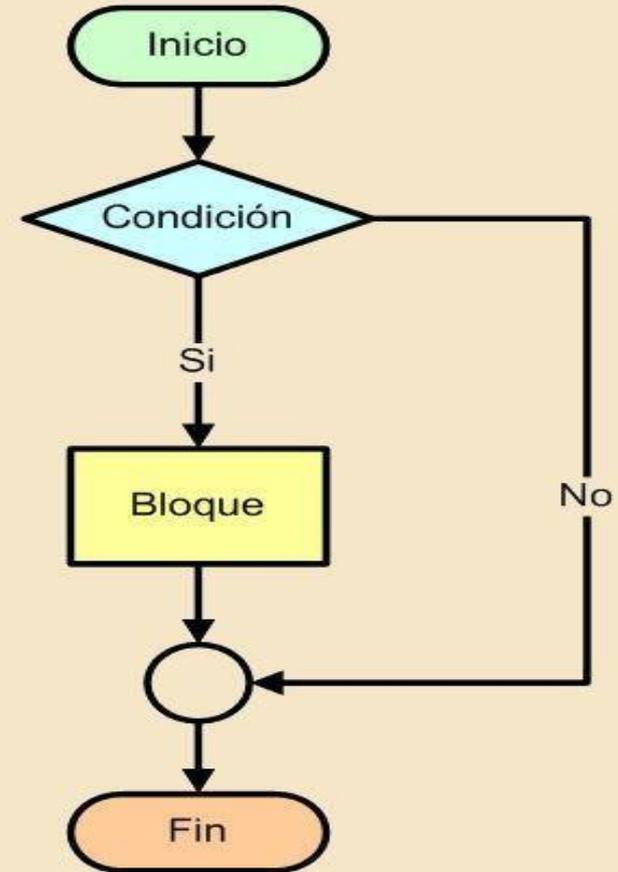
- Las estructuras selectivas sirven para seguir una sola de entre varias líneas de ejecución disponibles.



## 3.2 Estructuras selectivas... selectivas...

### Selectiva simple

- La estructura selectiva simple sirve para seguir una línea de ejecución cuando se cumple una condición.



## 3.2 Estructuras selectivas... selectivas...

### Selectiva simple...

#### Sintaxis

##### a) Con una instrucción

```
if (condicion(es)):  
    instrucción
```

Ejemplos :

```
if (x==y):
```

```
    x = y + 2
```

```
if ((a <= b) and (c == d)) :
```

```
    print("ERROR!!")
```



## 3.2 Estructuras selectivas... selectivas... Selectiva simple...

Sintaxis

### b) Con varias instrucciones

if (condición(es)):

instrucción1

instrucción2

...

instrucción n



Ejemplos:

```
if (w != 0):  
    print("Dato")
```

```
if (valor1 >= valor2):
```

```
    r = z %w
```

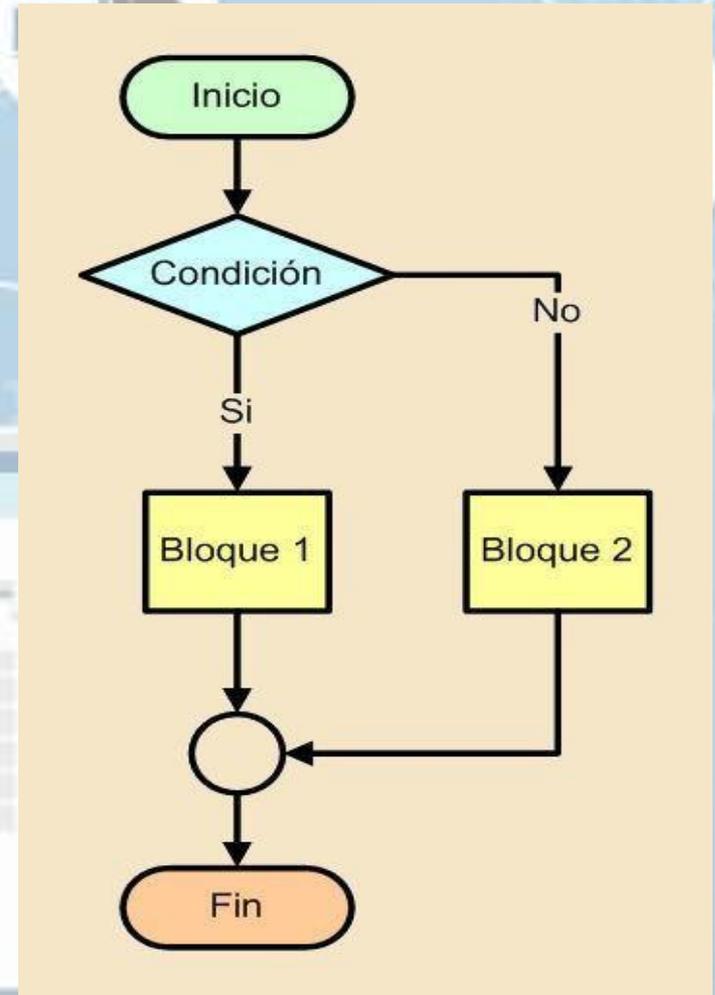
```
    x = y + 3
```



## 3.2 Estructuras selectivas... selectivas...

### Selectiva doble

- La estructura selectiva doble sirve para seguir una línea de ejecución cuando se cumple la condición, o seguir otra línea cuando no se cumple la condición.



## 3.2 Estructuras selectivas... selectivas... Selectiva doble...

### Sintaxis

#### a) Con una instrucción

```
if (condición(es)):  
    instrucción  
else :  
    instrucción
```

#### Ejemplos:

```
if (x==y):  
    x = y + 2  
else:  
    y = r / 2
```

```
if ((a <= b) and (c == d)):  
    print ("ERROR!!")  
else:  
    print ("Datos válidos")
```



## 3.2 Estructuras selectivas...

### selectivas... Selectiva doble...

#### Sintaxis

#### b) Con varias instrucciones

```
if (condición(es)):  
    instrucción1  
    ...  
    instrucción n  
else:  
    instrucción1  
    ...  
    instrucción n
```

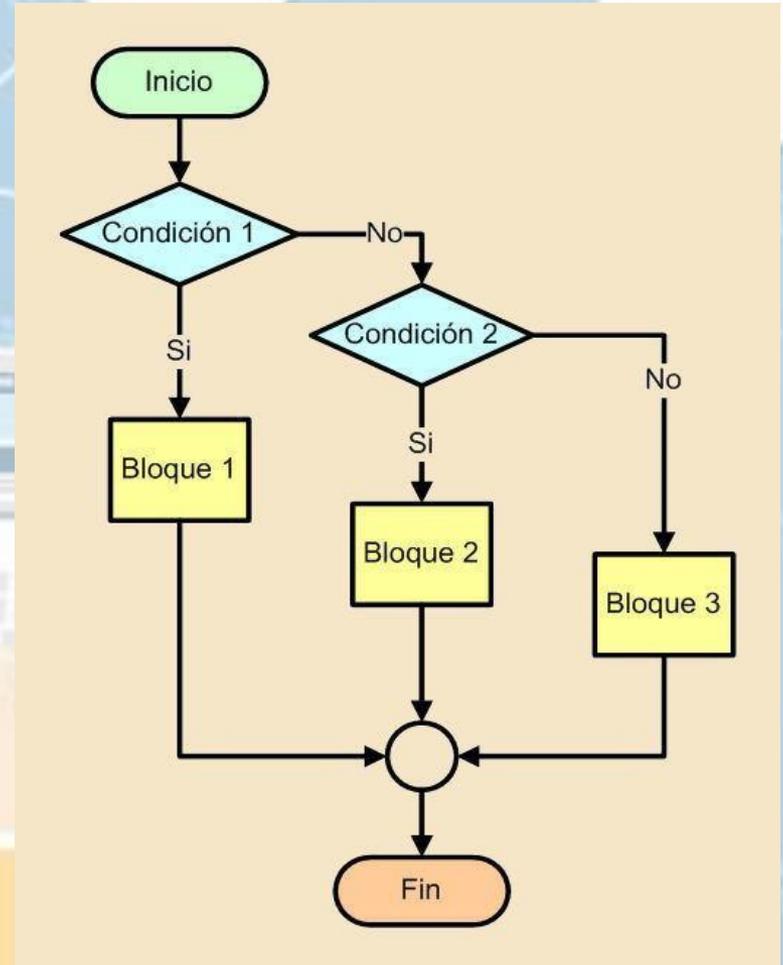
#### Ejemplo:

```
if ((j!=i) or (i>0)):  
    r = z %w  
    print("El resultado es",r)  
else :  
    e = t / 2  
    t = m*2
```

## 3.2 Estructuras selectivas... selectivas...

### Selectiva anidada

- La estructura selectiva anidada es una cadena de estructuras selectivas que se conectan de la parte **else** de la actual con la parte **if** de la siguiente.



## 3.2 Estructuras selectivas... selectivas...

Selectiva anidada...

### Sintaxis

#### a) Con una instrucción

```
if (condición(es)):  
    if (condición(es)):  
        Bloque 1  
    else :  
        Bloque 2  
else :  
    Bloque 3
```

```
if (condición(es)):  
    Bloque 1  
else :  
    if (condición(es)):  
        Bloque 2  
    else :  
        Bloque 3
```

## 3.2 Estructuras selectivas... selectivas...

### Selectiva anidada...

Ejemplos:

```
if (a != b):
```

```
    if (c%2 == 0):
```

```
        print("Valido")
```

```
    else:
```

```
        print("inválido")
```

```
else:
```

```
    z = x + y
```

```
if (k + 2 == 4)
```

```
    z = j % 3;
```

```
else :
```

```
    if (k/2 !=0)
```

```
        s = cadena1
```

```
    else:
```

```
        s = cadena2
```

## 3.2 Estructuras selectivas... selectivas...

### Selectiva anidada...

Ejemplos:

```
if (w != 0):
```

```
    if ((j!=i) or (i>0)):
```

```
        r = z %w
```

```
        print ("El resultado es",r)
```

```
    else :
```

```
        t = m*2
```

```
        e = t / 2
```

```
else :
```

```
    z =2
```



```
if (c/2 != 0):
```

```
    print("Dato Valido!!)
```

```
else:
```

```
    if ((j!=i) or (i>0)):
```

```
        j = t + y
```

```
        print("El resultado es",j)
```

```
    else:
```

```
        print ("Dato no valido!!")
```

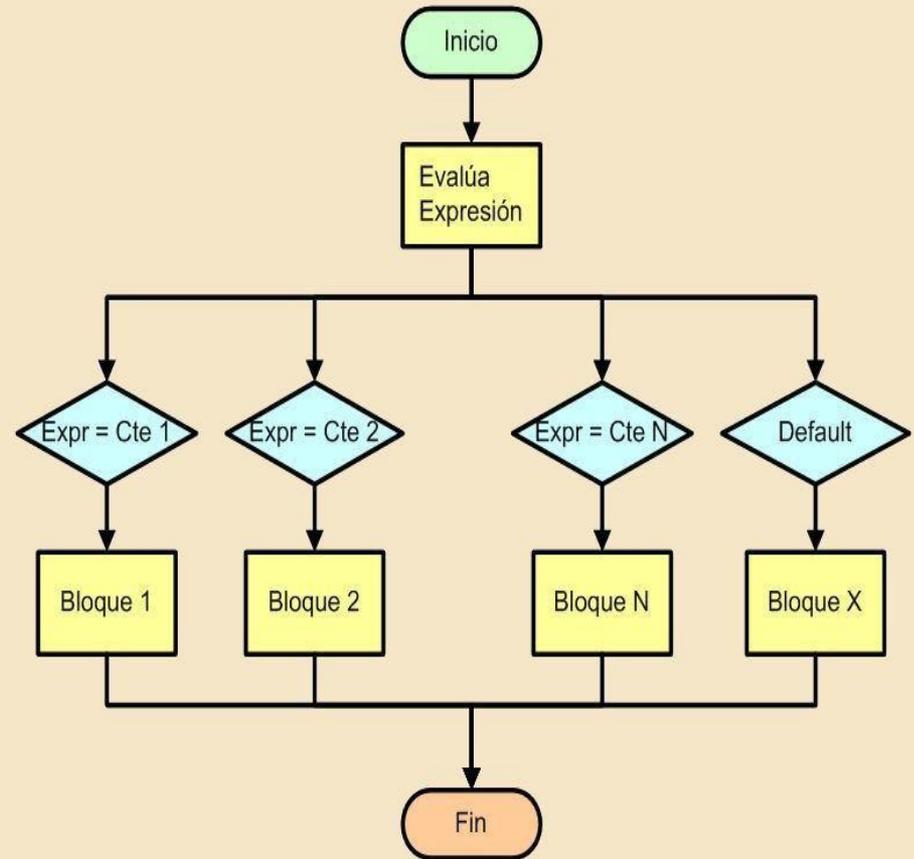


## 3.2 Estructuras selectivas... selectivas...

### Selectiva múltiple

- La estructura selectiva múltiple no existe en Python, pero se puede simular mediante el uso del comando:

***elif (condición):***



## 3.2 Estructuras selectivas... selectivas...

### Selectiva múltiple...

#### Sintaxis

```
if (condición_1):
```

```
    Bloque 1
```

```
elif (condición_2):
```

```
    Bloque 2
```

```
...
```

```
elif (condición_N):
```

```
    Bloque N
```

```
else:
```

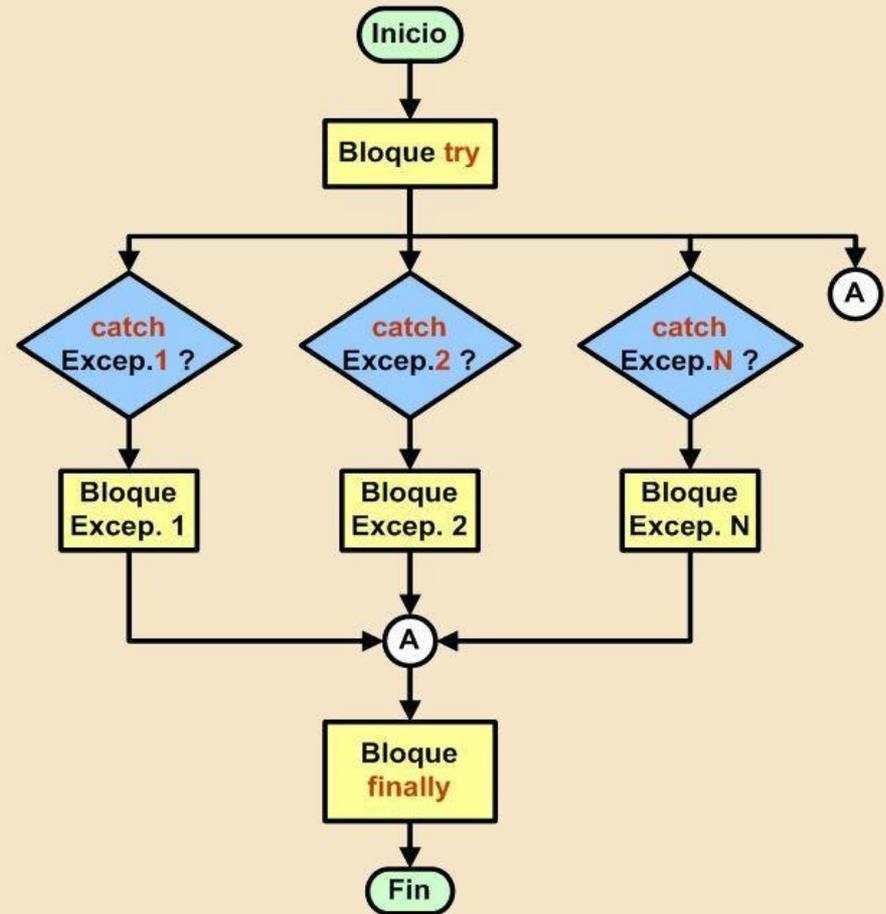
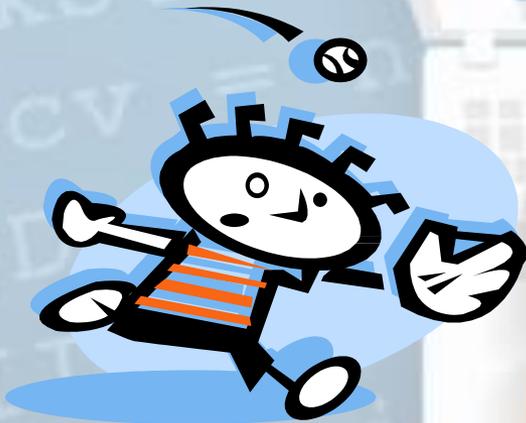
```
    Bloque X
```



## 3.2 Estructuras selectivas... selectivas...

### Selectiva intenta

- La estructura intenta (try/except) se utiliza para atrapar excepciones.



## 3.2 Estructuras selectivas...

selectivas...

Selectiva intenta...

### Sintaxis

**try:**

// Bloque de instrucciones  
que pueden generar una o  
más  
// situaciones excepcionales.

**except** Excepción 1:

// Bloque alternativo para el caso  
de existir la Excepción 1

**except** excepción n

// Bloque alternativo para el  
caso de existir la Excepción N

**Try es el bloque de código donde se prevé que se genere una excepción. Es como si decir "intenta estas sentencias y observa si se produce una excepción".**

## 3.2 Estructuras selectivas... selectivas...

### Selectiva intenta...

Ejemplo:

```
#excepciones
```

```
print("Bienvenido a este programa")
```

```
Num = 0
```

```
try:
```

```
    Num = int(input("Dame un numero entero: "))
```

```
    R = Num
```

```
except ValueError:
```

```
    print("Eso no fue un numero")
```

```
except ZeroDivisionError:
```

```
    print("Se dividio por 0")
```





## 3.3 Estructuras Iterativas

### Concepto

↗ Sirven para formar bucles o ciclos en los que se ejecuta repetidamente un bloque de instrucciones.

### Iteraciones

↗ Número de veces que se ejecuta el ciclo.

### Variable de control del ciclo

↗ Es la que controla el número de iteraciones de una estructura de repetición.

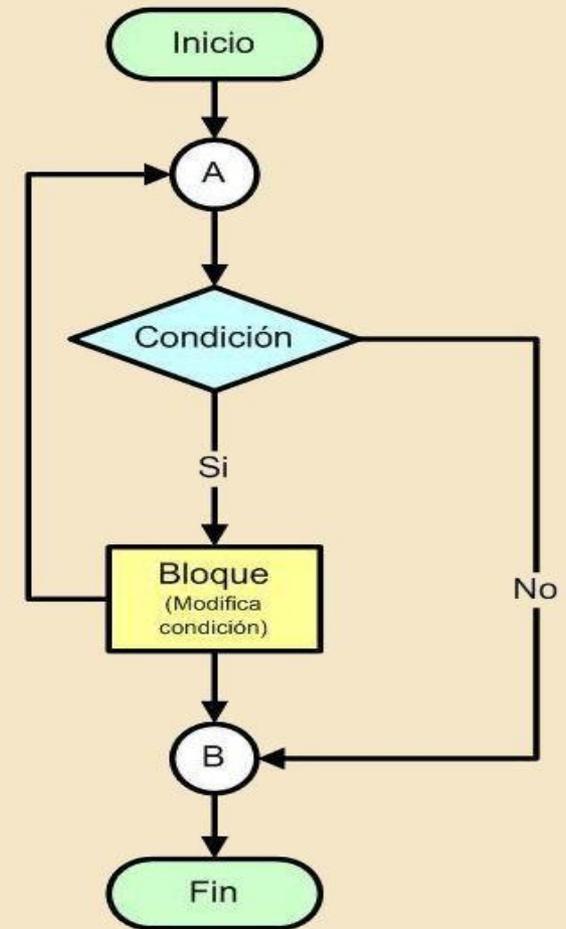




## Repetir mientras (while).

Funcionamiento:

1. Se evalúa una condición.
2. Si es verdadera, se ejecutara un bloque de instrucciones, en el cual debe existir una instrucción que modifique la condición, de lo contrario, se ejecutará un ciclo infinito (loop).
3. Si es **falsa**, el bloque de instrucciones no se ejecuta y finaliza la ejecución.





## Repetir mientras (while)...

Ejemplo 1:

```
a = 1
```

```
b = 5
```

```
while (a < b):
```

```
    println (" a=",a)
```

```
    a+=1
```

Sintaxis:

```
while( condición ):
```

```
    Bloque
```





mientras (while)...

Ejemplo 2

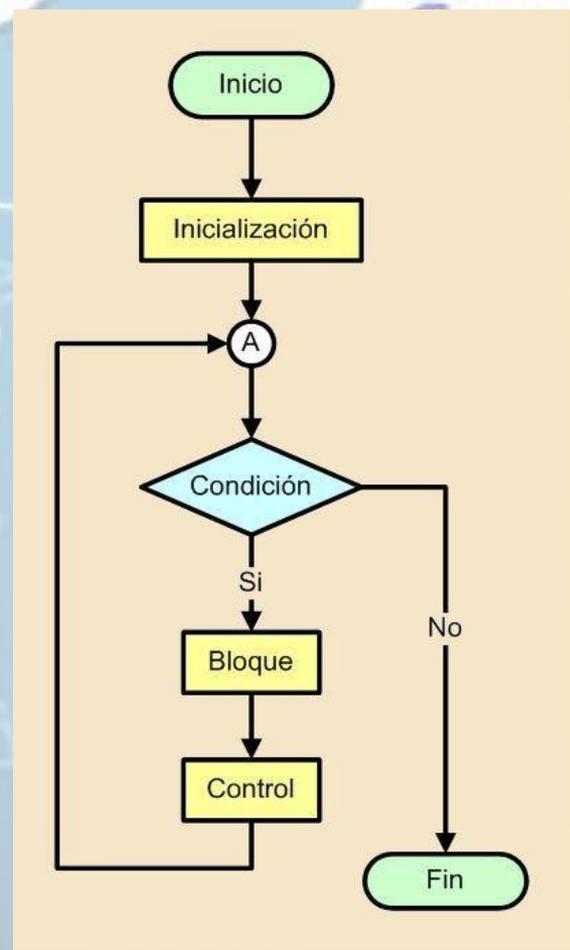
```
boolean x = true;
print ("Teclea solo numeros enteros y un 0 para salir")
while (x== true):
    num = int(input())
    if (num == 0):
        x = false;
    elif ((num % 2)== 0):
        print (num," es un numero par")
    else:
        print (num, " es un numero impar")
```





(for).

Es la más versátil de las estructuras de iteración, ya que permite la declaración de variables dentro de su estructura.





## Repetir desde (for)...

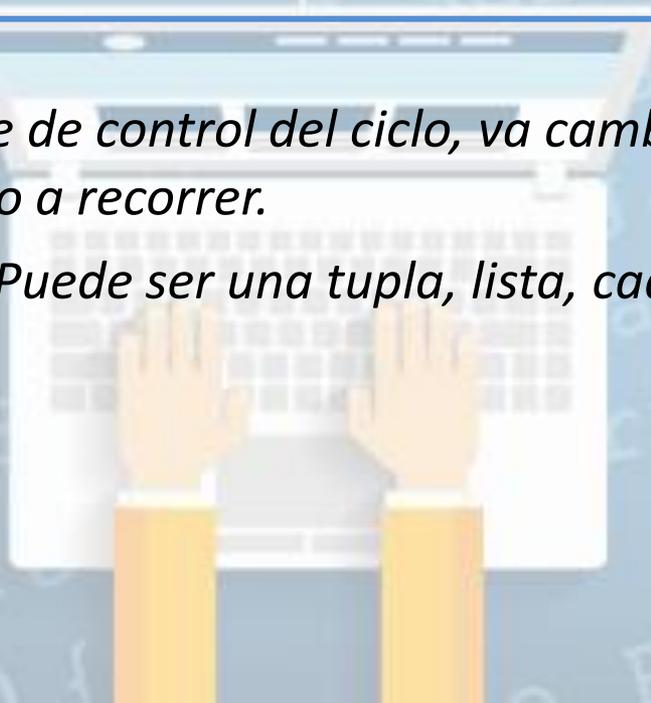
Sintaxis:

**for** variable **in** Elemento\_a\_recorrer:  
Bloque de instrucciones

donde:

**Variable:** Es la variable de control del ciclo, va cambiando de acuerdo al contenido del elemento a recorrer.

**Elemento a Recorrer:** Puede ser una tupla, lista, cadena de texto, rango, etc.





## Repetir desde (for)...

### Ejemplo 1:

```
for i in [1,2,3]:  
    print("Soy la línea ",i)
```

### Ejemplo 2:

```
calif=0  
prom=0  
for i in range(30):  
    print("Teclea la calificación del  
alumno = "+ str(i))  
    calif = int(input(">>: "))  
    prom+= calif  
print("El promedio del grupo es  
= "+ str(prom /30))
```



# Ciclos anidados

Ejemplos:

```
for i in range(10):  
    for j in range(5):  
        print("j = ", j)  
    print("i = ", i)
```

```
while (x < t):  
    for a in range(5,0,-1):  
        m = m*2;  
    print("m = ", m)  
    x+=1
```





# FUNDAMENTOS DE PROGRAMACIÓN

---



*“Dime y lo olvido,  
enséñame y lo recuerdo,  
involúcrame y lo  
aprendo”.*

*Benjamín Franklin*