



universidad
cenfotec_
La U de la informática

Principios de Programación.

Estructuras de construcción.
Estructura iterativa for.

Objetivos

- Comprender el uso de la estructuras iterativas for.

Repaso: estructura iterativa

Una **estructura iterativa** es una estructura de control que permite ejecutar una instrucción o un bloque de instrucciones múltiples veces.

Estas estructuras pueden ser llamadas también bucles o ciclos.

Al igual que en una estructura de selección, las acciones se ejecutarán de acuerdo a una condición dada.

Repaso: Tipos de estructuras iterativas

Incondicional con contador.

- La condición de estas estructuras depende de una variable **contador** de tipo **numérica**.
- Antes de iniciar la ejecución del ciclo es posible determinar **cuántas** veces se ejecutarán las instrucciones.
- En cada ejecución del ciclo se incrementa (o decrementa) el valor de la variable **contador**.

Condicional con bandera.

- La condición depende de un valor o una variable llamada **bandera**.
- **No** se conoce de antemano cuántas veces se ejecutará el ciclo debido a que en cada ejecución se evalúa la condición.
- En cada ejecución del ciclo se debe **modificar** uno de los valores que eventualmente harán que la condición ya no se cumpla.

Repaso: Acumuladores y contadores

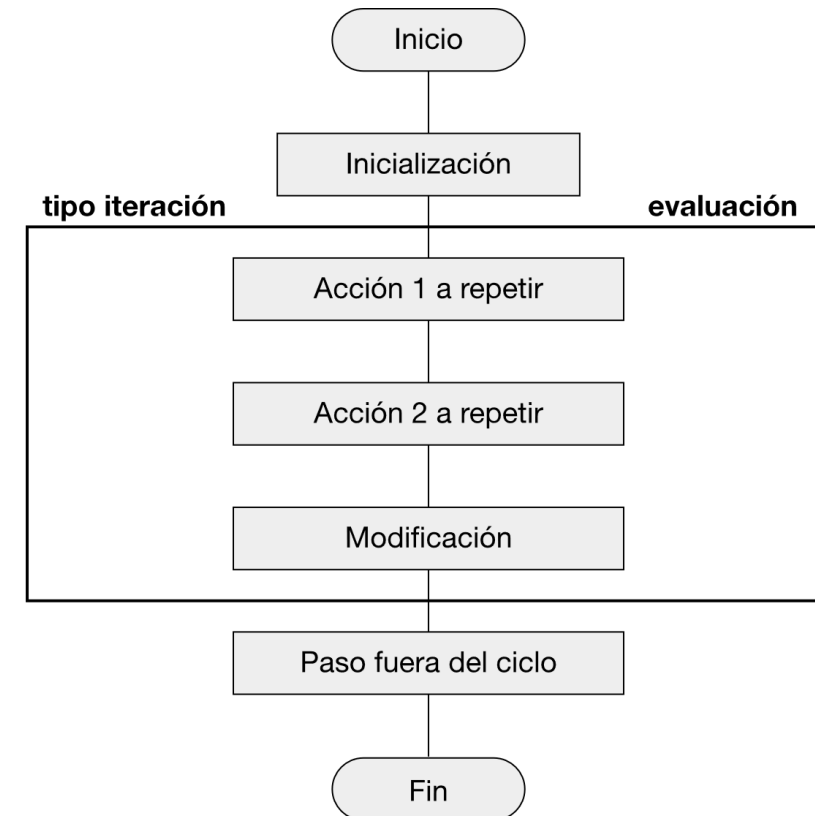
Durante la ejecución de las estructuras iterativas se presenta la necesidad de almacenar valores que se obtendrán durante las ejecuciones del ciclo. Esto se logra mediante variables **acumuladoras** y **contadoras**.

- **Variable acumuladora:** Es una variable que **acumula** los valores obtenidos en las iteraciones. Una vez se concluye con la ejecución del ciclo, esta variable contiene el resultado total de los valores obtenidos en las repeticiones.
 - **Ejemplos:** La suma total del salario de 5 empleados, la suma del porcentaje de las tareas realizadas por un estudiante.
- **Variable contadora:** Esta variable tiene la responsabilidad de **contar** la cantidad de iteraciones realizadas por la estructura iterativa. Tiene un funcionamiento similar a una variable acumuladora, ya que por cada ejecución acumula un nuevo valor.
 - **Ejemplos:** Panel que muestra la posición del cliente que está siendo atendido en un banco, aplicación para contar la cantidad de personas dentro de un local.

Repaso: Representación en diagrama

Las estructuras iterativas están compuestas por los siguientes pasos:

1. Inicialización.
2. Evaluación.
3. Cuerpo del ciclo.
4. Modificación.



Ciclo for

Ciclos for

La palabra clave for es utilizada para generar ciclos que iteran sobre una colección.

Por ejemplo, considere el siguiente problema:

"Se desean imprimir todos los números que se encuentren entre el 0 y el 7"

Palabra reservada: range

La función range () devuelve una secuencia de números, comenzando desde 0 de forma predeterminada, se incrementa en 1 en 1 y termina en el número especificado.

Observe a continuación la implementación:

imprimir_valores.py

```
1 limite = 7
2 for numero in range(limite):
3     print(numero)
```

```
/usr/local/bin/python3 "/Users/pabs_laptop/Desktop/Progra
a ejercicios/semana 8/ejemplo_for.py"
(base) pabs_laptop@MacBook-Pro Progra ejercicios % /usr/
local/bin/python3 "/Users/pabs_laptop/Desktop/Progra eje
rcicios/semana 8/ejemplo_for.py"
0
1
2
3
4
5
6
```

Tablas de multiplicar

Observe el siguiente ejercicio:

"Realice un programa que solicite al usuario un número e imprima las tablas de multiplicar de todos los números que se encuentren hasta el número dado"

Tablas de multiplicar

tablas_multiplicar.py

```
1     limite = int(input('Ingrese el número deseado: '))
2     for numero in range(limite):
3         print(f'--- Tabla del {numero} ---')
4         print(f'{numero} x 0 = {numero * 0}')
5         print(f'{numero} x 1 = {numero * 1}')
6         print(f'{numero} x 2 = {numero * 2}')
7         print(f'{numero} x 3 = {numero * 3}')
8         print(f'{numero} x 4 = {numero * 4}')
9         print(f'{numero} x 5 = {numero * 5}')
10        print(f'{numero} x 6 = {numero * 6}')
11        print(f'{numero} x 7 = {numero * 7}')
12        print(f'{numero} x 8 = {numero * 8}')
13        print(f'{numero} x 9 = {numero * 9}')
14        print(f'{numero} x 10 = {numero * 10}')
15        print('--- --- --- --- ---')
```

```
Ingrese el número deseado: 2
--- Tabla del 0 ---
0 x 0 = 0
0 x 1 = 0
0 x 2 = 0
0 x 3 = 0
0 x 4 = 0
0 x 5 = 0
0 x 6 = 0
0 x 7 = 0
0 x 8 = 0
0 x 9 = 0
0 x 10 = 0
--- --- --- --- ---
--- Tabla del 1 ---
1 x 0 = 0
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
--- --- --- --- ---
```



universidad
cenfotec_

La U de la informática