



Ciclo 1

Semana 4

Estructuras de control: repetición o iteración

Lectura 1 – Estructuras de control de repetición



| Estructuras de control de repetición

Los computadores están especialmente diseñados para realizar una operación o muchas de ellas, de forma repetitiva, por lo que es importante entender la lógica necesaria para desarrollar algoritmos que permitan repetir una o varias acciones un determinado número de veces. Un ejemplo característico es el ingreso de datos desde la consola, donde se le solicitan al usuario un determinado número de datos, cuando ese número es conocido, o hasta que el usuario determine que no tiene más datos, y detenga dicho proceso. Otro ejemplo típico es el proceso de lectura de un archivo, en este caso se leen uno a uno los datos contenidos en él hasta que se alcance el final del archivo y se detenga el proceso.

Las estructuras de repetición se denominan ciclos o bucles y se llama repetición o iteración, al hecho de repetir la ejecución de una secuencia de acciones. Entonces las dos preguntas principales que se deben hacer en el diseño de un bucle son:

1. ¿Cuáles serán las instrucciones que componen el cuerpo del ciclo?, es decir, las que serán repetidas.
2. ¿Cuántas veces se repetirán? Aquí se debe determinar el número de veces, si es conocido, o la condición que permitirá detener el ciclo, en caso contrario.

Si no se tuviera la posibilidad de utilizar ciclos sino lo visto hasta ahora, se tendría la necesidad de repetir en el código las instrucciones tantas veces como se requiera, por ejemplo, si se tuviera que pedir al usuario tres números y obtener su suma, la única opción que se tendría sería de la siguiente manera:

```
algoritmo suma
var
    entero : SUMA, NUMERO
inicio
    SUMA ← 0
    leer(numero)
    SUMA ← SUMA + numero
    leer(numero)
    SUMA ← SUMA + numero
    leer(numero)
fin
```

Figura 1: tomado de Joyanes



Semana 4

Estructuras de control:
repetición o iteración

Para un número pequeño como en el ejemplo, no sería inconveniente, pero si el número fuera grande digamos más de 100, o no se supiera con exactitud cuántos números serán ingresados, la situación se complica bastante. Seguramente se podría llegar a pensar que tal vez habría otra forma más eficiente de hacerlo.

Con el uso de repeticiones, las operaciones se harían de una forma mucho más sencilla. Siguiendo el mismo ejemplo, se podría llegar al siguiente algoritmo:

```
algoritmo suma_numero
var
    entero : N, TOTAL
    real : NUMERO, SUMA
inicio
    leer(N)
    TOTAL ← N
    SUMA ← 0
    mientras TOTAL > 0 hacer
        leer(NUMERO)
        SUMA ← SUMA + NUMERO
        TOTAL ← TOTAL - 1
    fin_mientras
    escribir('La suma de los', N, 'números es', SUMA)
fin
```

Figura 2: tomado de Joyanes

```
N = int(input('Ingrese cuántos números sumará?:'))
TOTAL = N
SUMA = 0
while TOTAL > 0:
    NUMERO = int(input('Ingrese un número a sumar:'))
    SUMA = SUMA + NUMERO
    TOTAL = TOTAL - 1
print('La suma de los ' + str(N) + ' números es: ' + str(SUMA))
```

```
Ingrese cuántos números sumará?:5
Ingrese un número a sumar:3
Ingrese un número a sumar:8
Ingrese un número a sumar:2
Ingrese un número a sumar:6
Ingrese un número a sumar:7
La suma de los 5 números es: 26

Process finished with exit code 0
```

Implementación en Python

En este caso se le solicita al usuario que digite primero la cantidad de números que desea ingresar y con base en ese número se construye una estructura de repetición, para que lo haga ese número de veces. En el cuerpo de la estructura estarán las operaciones que se repetirán, para el caso, leer el número y sumarlo, y la condición deberá detener el ciclo una vez se hayan leído los números, en el ejemplo que la variable total no sea positiva. Como se puede ver, a la variable TOTAL se le asigna el valor N, que es el número de datos que el usuario ingresará, y a la variable SUMA se le asigna el valor inicial de cero (0). Estas dos operaciones más la lectura de N, deben estar por fuera del ciclo puesto que no se deben repetir, es decir, deben hacerse una sola vez. Dentro del cuerpo del ciclo, o sea, las operaciones que se deben repetir, están las operaciones, leer NUMERO, sumar ese NUMERO a la variable SUMA y restar 1 a la variable TOTAL. La instrucción $SUMA \leftarrow SUMA + NUMERO$, se debe entender así: primero se toma el



Semana 4

Estructuras de control:
repetición o iteración

valor que tiene la variable SUMA, luego le suma el valor que tiene la variable NUMERO y por último reescribe o actualiza el valor de SUMA con su nuevo valor. De esa misma forma se debe entender la instrucción $TOTAL \leftarrow TOTAL - 1$, primero toma el valor de TOTAL, luego le resta 1 y por último actualiza la variable TOTAL con su nuevo valor.

La palabra *mientras* nos indica que siempre y cuando la condición sea verdadera, es decir, TOTAL sea positiva o mayor que cero (0), el ciclo se repetirá una vez más, en caso contrario, o sea, cuando la variable TOTAL sea menor o igual a cero (0), el ciclo terminará y se continuará con la instrucción que haya después del *fin_mientras*, en este caso la instrucción escribir, la cual sólo se ejecutaría una sola vez porque ya no se encuentra dentro del cuerpo del *mientras*.

Una vez entendido el pseudocódigo, se debe entonces entrar al lenguaje Python y aprender cómo se debe codificar esta estructura.