FUNDAMENTOS DE INFORMATICA

INGENIERA SILVIA PATRICIA BARDELLI

PROFESORA ING. SILVIA PATRICIA BARDELLI

CLASE NRO 4

Temas:

- Programación estructurada
- Estructura iterativa
- Ciclos
- Contadores y acumuladores

PROFESORA ING. SILVIA PATRICIA BARDELLI

PROBLEMAS REPETITIVOS

Supongamos que se necesita mostrar por pantalla los números enteros del 1 al 5.

Con nuestros conocimientos actuales, ¿cuál sería la forma de resolver ese problema?

PROBLEMAS REPETITIVOS

Ahora que conocemos la estrategia de resolución, vamos a extender el problema para mostrar por pantalla los números enteros del 1 al 100.

¿Resulta adecuada la misma estrategia anterior?

ESTRUCTURA ITERATIVA, CICLO O BUCLE

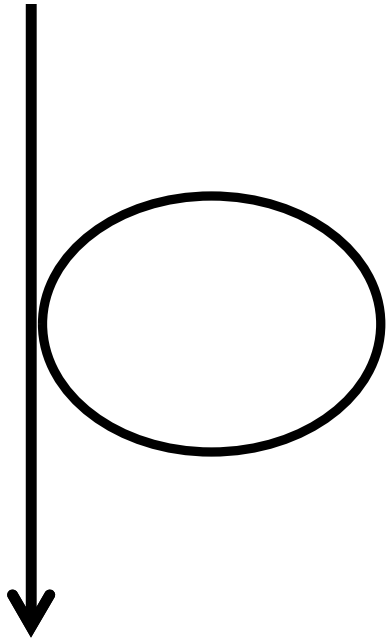
Por eso, además de las estructuras secuencial y alternativa, existe la tercera y última estructura de la Programación Estructurada:

La Estructura Iterativa.

Combinando estas tres estructuras
es posible resolver cualquier problema
informático, por más complejo que sea.

ESTRUCTURA ITERATIVA, CICLO O BUCLE

En esta estructura el programa repetirá una porción de su código una cierta cantidad de veces, y luego seguirá adelante.



ESTRUCTURA ITERATIVA, CICLO O BUCLE

A cada repetición se la denomina ***iteración.***

También necesitaremos una nueva instrucción para poder implementar este tipo de estructura: La instrucción ***while*** (mientras).

INSTRUCCIÓN WHILE

while ***<condición>***:

. . . . . .

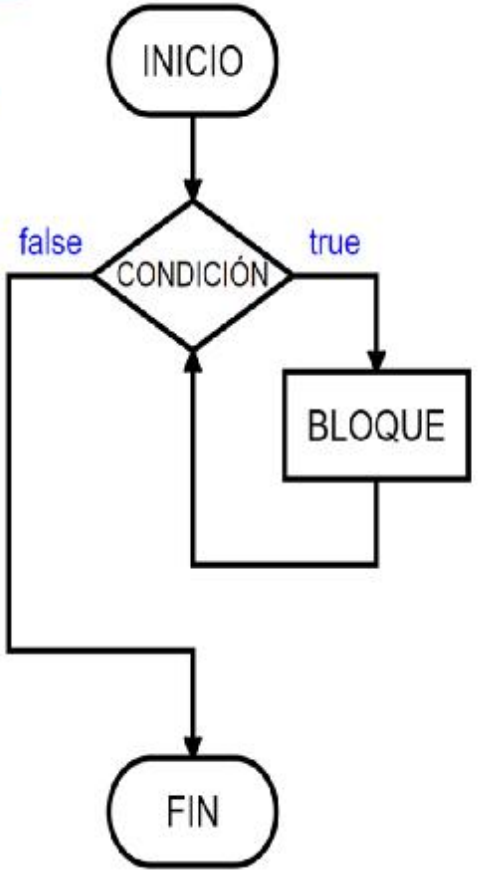
. . . . . .

. . . . . .

INSTRUCCIÓN WHILE

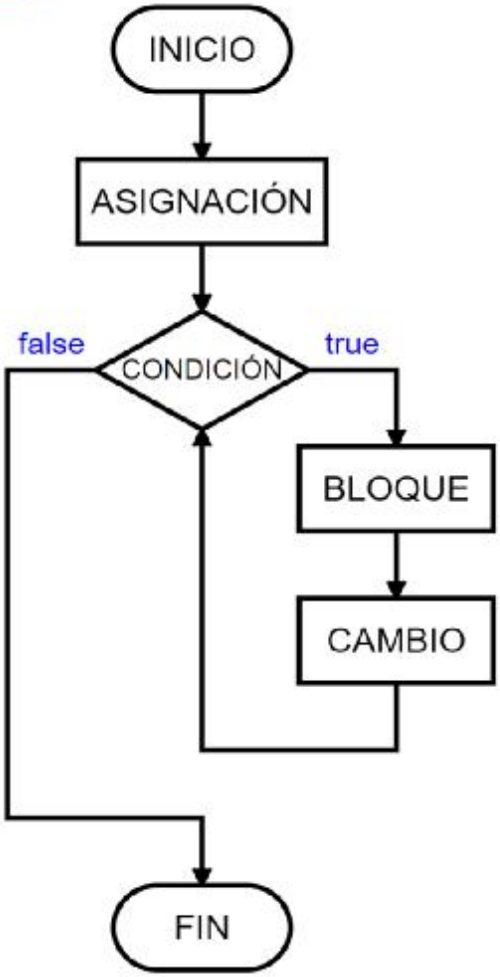
- La condición va seguida del carácter “dos puntos”.
- La sangría o indentación es lo que establece el alcance del ciclo.
- Python recomienda una sangría standard de 4 espacios, sin tabs.
- La sangría debe ser uniforme

DIAGRAMA DE FLUJO DEL CICLO WHILE



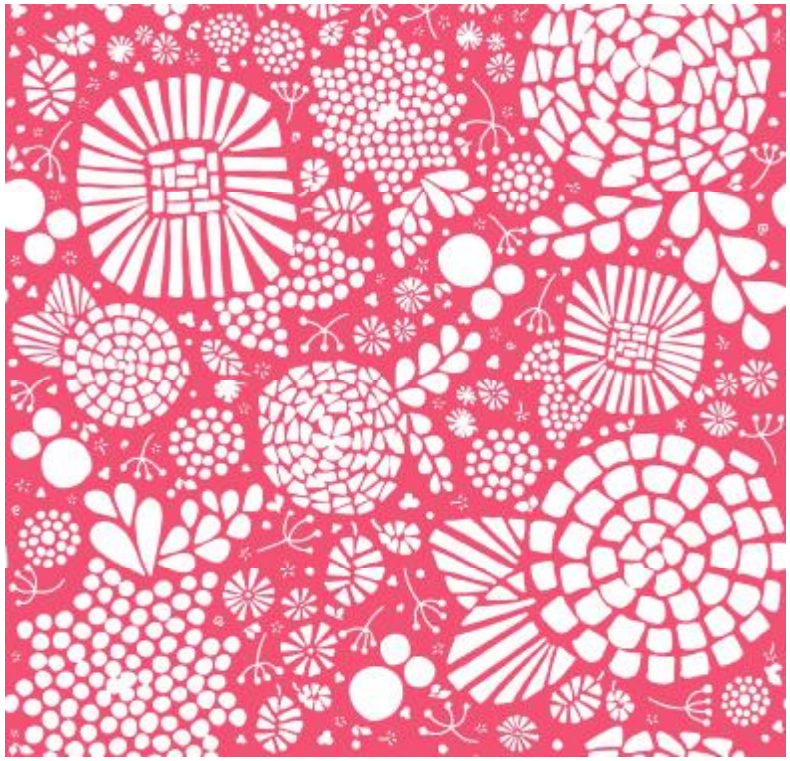
PROFESORA ING. SILVIA PATRICIA BARDELLI

DIAGRAMA DE FLUJO DEL CICLO WHILE CON VARIABLES DE CONTROL



PROFESORA ING. SILVIA PATRICIA BARDELLI

DIFERENCIAS Y SIMILITUDES ENTRE EL IF Y EL WHILE



PROFESORA ING. SILVIA PATRICIA BARDELLI

EJEMPLO 1

Objetivo:

Imprimir los números enteros entre 1 y 100.

¿Qué datos debemos solicitarle al usuario?

EJEMPLO 1

```
# Imprimir los números enteros entre 1 y 100

a = 1
while a <= 100:
    print(a)
    a = a + 1
# Fin del programa
```

DEFINICION CONTADOR

Cuando una variable es modificada en una cantidad fija respecto de su valor anterior, se la denomina ***contador***.

ATENCIÓN

¿Qué ocurre si olvidamos la línea que incrementa el contador?

```
a = 1
while a <= 100:
    print(a)
    a = a + 1
# Fin del programa
```

ATENCIÓN

Eso se conoce como

ciclo infinito,

y es uno de los peores errores que puede cometer un programador.

EJEMPLO 2

Objetivo:

Imprimir los números pares entre 1 y 100

¿Qué modificaciones tendremos que hacerle al programa anterior?

EJEMPLO 2

```
# Imprimir los números pares entre 1 y 100

a = 2
while a <= 100:
    print(a)
    a = a + 2
# Fin del programa
```

INCREMENTAR UNA VARIABLE ES TAN COMÚN QUE PYTHON PROVEE UNA SINTAXIS ABREVIADA PARA ELLO:

```
>>> contador = 0
>>> contador += 1
>>> contador
1
>>> contador += 1
>>> contador
2
>>> n = 2
>>> n += 5
>>> n
7
```

`contador += 1` *es una abreviación de* `contador = contador + 1`.

El valor del incremento no tiene que ser 1 necesariamente:

TAMBIÉN EXISTEN LAS ABREVIACIONES *=, -=, /=, Y %=:

```
>>> n = 2
>>> n *= 5
>>> n
10
>>> n -= 4
>>> n
6
>>> n /= 2
>>> n
3
>>> n %= 2
>>> n
1
```

EJEMPLO 3:

EL SIGUIENTE EJEMPLO PIDE UN NÚMERO POSITIVO AL USUARIO UNA Y OTRA VEZ HASTA QUE EL USUARIO LO HAGA CORRECTAMENTE:

```
numero = int(input("Escriba un número positivo: "))
while numero < 0:
    print("¡Ha escrito un número negativo! Inténtelo de nuevo")
    numero = int(input("Escriba un número positivo: "))
print("Gracias por su colaboración")
```

- *Escriba un número positivo: -4*
- *¡Ha escrito un número negativo! Inténtelo de nuevo*
- *Escriba un número positivo: -8*
- *¡Ha escrito un número negativo! Inténtelo de nuevo*
- *Escriba un número positivo: 9*
- *Gracias por su colaboración*

EJEMPLO 4

Objetivo:

Leer un conjunto de números enteros
e imprimir su promedio. El fin de los datos
se indica ingresando el valor -1.

¿Qué tendremos que hacer con los valores ingresados?

EJEMPLO 4

```
suma = 0
cant = 0
n = int(input("Ingrese un número o -1 para terminar: "))
while n != -1:
    suma = suma + n
    cant = cant + 1
    n = int(input("Ingrese un número o -1 para terminar: "))
if cant != 0:
    prom = suma/cant
    print("El promedio es", prom)
else:
    print("No se ingresaron valores")
```

IMPORTANTE

§ Los promedios siempre deben calcularse con decimales.

§ Es imprescindible evitar errores de división por cero.

DEFINICIÓN ACUMULADOR

Cuando una variable es modificada en una cantidad cambiante respecto de su valor anterior, se la denomina ***acumulador***.

CONTADORES Y ACUMULADORES

Para poder trabajar con estas estructuras es muy importante comprender el concepto de:

CONTADOR	Contador = contador + constante
ACUMULADOR	acumulador = acumulador + variable

EJEMPLO 5

Objetivo:

Leer un conjunto de números enteros e imprimir el mayor. El fin de los datos se indica con -1.

¿Cómo podemos proceder para hallar el máximo?

EJEMPLO 5

```
n = int(input("Ingrese un número o -1 para terminar: "))
mayor = n
while n != -1:
    if n > mayor:
        mayor = n
    n = int(input("Ingrese un número o -1 para terminar: "))
print("El mayor es", mayor)
```

EJEMPLO 5

¿Qué ocurre si movemos el 2º input?

```
n = int(input("Ingrese un número o -1 para terminar: "))
mayor = n
while n != -1:
    if n > mayor:
        mayor = n
    n = int(input("Ingrese un número o -1 para terminar: "))
print("El mayor es", mayor)
```

EJEMPLO 5

¿Qué ocurre si movemos el 2º input?

```
n = int(input("Ingrese un número o -1 para terminar: "))
mayor = n
while n != -1:
    if n > mayor:
        mayor = n
        n = int(input("Ingrese un número o -1 para terminar: "))
print("El mayor es", mayor)
```

EJEMPLO 6

Leer notas de 35 alumnos de una clase, establecidas entre 0 y 10. Se desea desarrollar el pseudocódigo, diagrama de flujo y lenguaje Python para un programa que determine la nota promedio.

SOLUCIÓN EJEMPLO 6: SEUDOCÓDIGO

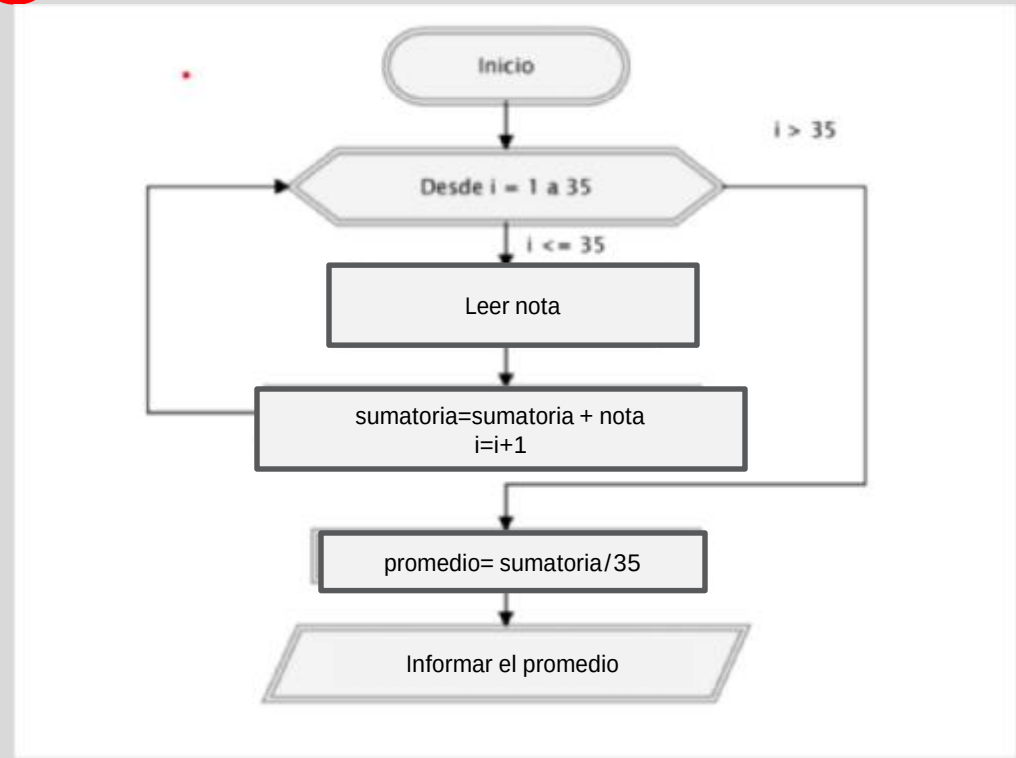
1. Inicio [Algoritmo de la nota promedio]
2. Desde i = 1 hasta 35 Hacer

        Leer Nota

        Sumatorio = Sumatorio + Nota

   Siguiente
3. Promedio = Sumatorio / 35
4. Mostrar “La nota promedio de la clase es”, Promedio
5. Fin

EJEMPLO 6: DIAGRAMA DE FLUJO



EJEMPLO 6: RESOLUCIÓN EN LENGUAJE PYTHON

```
i=1
sum=0
while i<36:
    nota=int(input("Ingrese una nota"))
    sum=sum+nota
    i=i+1
print ("El promedio de los 35 alumnos es ,",sum/35)
```

EJEMPLO 7: IMPRIMIR LA TABLA DE MULTIPLICAR SEUDOCÓDIGO

```
# Se supone que las tablas llegan hasta el 10
 LIMITE = 10
 # Comenzar en 1
 contador = 1
 while contador <= LIMITE:
     resultado = contador * numero
     imprimir (numero, contador, resultado))
     # Incrementar contador para no caer en ciclo infinito
     contador = contador + 1
```

EJEMPLO 7: RESOLUCIÓN EN LENGUAJE PYTHON

```
numero=int(input("Ingrese un numero "))
i=1
LIMITE=10
while i <=LIMITE:
    resultado=i*numero
    print(numero,"x",i,"=", resultado)
    i=i+1
```

Ejercitación

- Práctica 4 Estructura Iterativa
- Práctica 5: Ejercicios Integradores