

Control de Flujo: ciclos

Clase #05

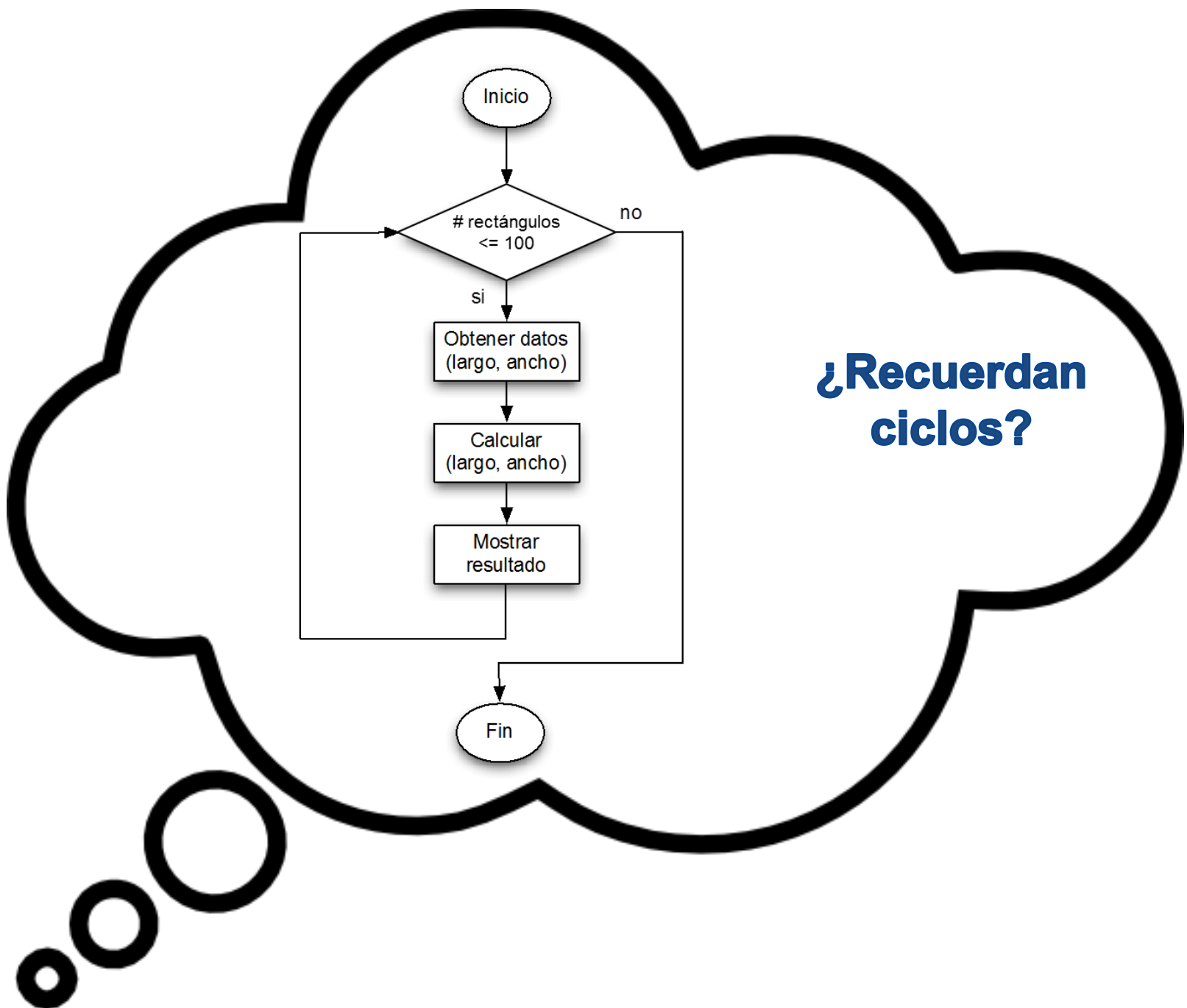
IIC1103 – Introducción a la Programación

Marcos Sepúlveda (marcos@ing.puc.cl)

Veremos hoy ...

Control de flujo:

- ▶ Instrucción ***while***



Iteración: *while*

► Sintaxis

```
while condicion:  
    instruccion  
    ...
```

- El conjunto de instrucciones se repite mientras la ***condición*** es ***True***
- Nota: las instrucciones que estén dentro del ***while*** deben estar indentadas.

Iteración: *while*

- Por ejemplo, el siguiente programa escribe los números del 1 al 10:

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

“Mientras *i* sea menor o igual a 10”



```
>>>
```

```
1
2
3
4
5
6
7
8
9
10
```

Iteración: *while*

► *ciclo infinito*

- Si la condición del ciclo se cumple siempre, el ciclo no terminará nunca de ejecutarse y tendremos lo que se denomina un ciclo infinito.

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

Nos olvidamos de modificar la variable *i* dentro del ciclo

```
i=1
while i>0:
    print(i)
    i += 1
```

Se ha escrito una condición que se cumplirá siempre y el programa imprimirá números indefinidamente



Ejemplo – área de 5 rectángulos

```
print("Este programa calcula el área de 5 rectángulos")
```

```
i = 1
```

```
while i<=5:
```

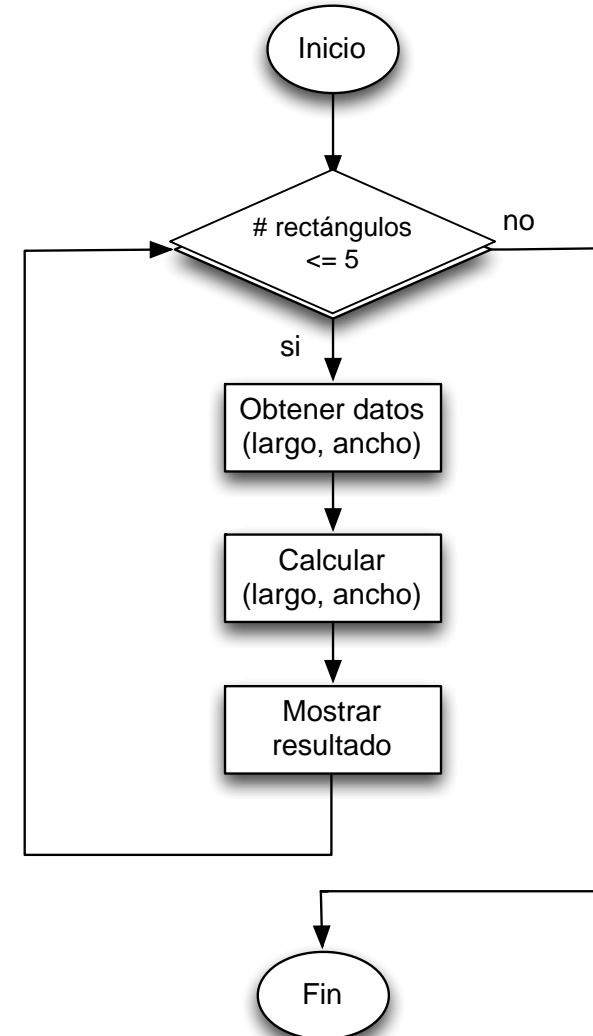
```
    largo = float(input("Ingrese largo: "))
```

```
    ancho = float(input("Ingrese ancho: "))
```

```
    area = largo * ancho
```

```
    print("El área del rectángulo es: " + str(area))
```

```
    i += 1
```



Ejemplo – productoria

```
print("Productoria de primeros 20 números naturales")
```

```
producto = 1
```

```
i = 1
```

```
while i<=20:
```

```
    producto *= i
```

```
    i += 1
```

```
print("productoria: " + str(producto))
```

$$\prod_1^{20} n$$

```
>>>
```

```
Productoria de primeros 20 números naturales
```

```
productoria: 2432902008176640000
```


Ejemplo: potencias

- Desplegar las doce primeras potencias de dos: $2^0, 2^1, 2^2, 2^3, \dots, 2^{10}, 2^{11}$

```
i=0
while i<12:
    print(2 ** i)
    i += 1
```

```
>>>
1
2
4
8
16
32
64
128
256
512
1024
2048
```

Ejemplo: potencias

- ▶ Desplegar las doce primeras potencias de dos: $2^0, 2^1, 2^2, 2^3, \dots, 2^{10}, 2^{11}$

```
i=0
while i<12:
    print(2 ** i, end=" ")
    i += 1
```

```
>>>
1 2 4 8 16 32 64 128 256 512 1024 2048
```

Ejemplo: potencias

- Desplegar las potencias de dos que son inferiores a 500:

```
i=0
while 2**i<500:
    print(2 ** i)
    i += 1
```

```
>>>
1
2
4
8
16
32
64
128
256
```

Ejemplo: promedio de edad de invitados a una fiesta

- Construir un programa que le permita a un usuario calcular la edad promedio de los invitados a una fiesta. Dado que no se sabe cuántos son los invitados, construir un algoritmo que continúe agregando personas mientras el usuario lo desee, para luego calcular la edad promedio de los invitados que asistieron.

```
sumaEdad = 0
numeroPersonas = 0

print("Ingrese la edad de sus invitados (coloqué 0 para terminar)")

edad = int(input())
while (edad > 0):
    sumaEdad += edad
    numeroPersonas += 1
    edad = int(input())

if numeroPersonas > 0:
    promedioEdad = sumaEdad / numeroPersonas
    print("Tienes " + str(numeroPersonas) + " invitados")
    print("Su promedio de edad es: " + str(promedioEdad) + " años")
else:
    print("Que fiesta más fome; ningún invitado!")
```

Ejemplo: Suma los dígitos de un número positivo

- ▶ Construya un programa en Python que pida al usuario que ingrese un número positivo, luego sume los dígitos de dicho número y despliegue el resultado en la pantalla.
- ▶ Ejemplo: **12345** \rightarrow **1 + 2 + 3 + 4 + 5 = 15**
- ▶ Ayuda:
Nos sirven los operadores aritméticos **//** y **%**

¿Y por qué?



Ejemplo: Suma los dígitos de un número positivo

- ▶ Se necesita descomponer el número para poder sumar sus dígitos.
- ▶ El número mágico que nos sirve es el **10** ... ¿y por qué?
- ▶ Porque si al número ingresado le aplicamos **% 10**, obtenemos el último dígito



$$\begin{array}{r} 12345 : 10 = 1234 \\ 5 \end{array}$$

Ejemplo: Suma los dígitos de un número positivo

$$\begin{array}{r} 12345 : 10 = 1234 \\ 5 \end{array}$$

$$\begin{array}{r} 1234 : 10 = 123 \\ 4 \end{array}$$

$$\begin{array}{r} 123 : 10 = 12 \\ 3 \end{array}$$

$$\begin{array}{r} 12 : 10 = 1 \\ 2 \end{array}$$

Y:

$$5 + 4 + 3 + 2 + 1 = 15$$

$$\begin{array}{r} 1 : 10 = 0 \\ 1 \end{array}$$

paramos cuando la división entera es 0

Ejemplo: Suma los dígitos de un número positivo

```
print("Suma los dígitos de un número positivo")

print("Ingresa el número:", end=" ")
numero = int(input())

suma = 0
ladoIzquierdo = numero

while (ladoIzquierdo > 0):
    digitoDerecha = ladoIzquierdo % 10
    ladoIzquierdo = ladoIzquierdo // 10
    print(ladoIzquierdo, digitoDerecha)
    suma += digitoDerecha

print("La suma de los dígitos de " + str(numero) + " es " + str(suma))
```


Ejercicios propuestos

- ▶ Solicitar al usuario que ingrese N números. Al finalizar el programa, se debe mostrar la suma de todos los números. El programa termina cuando $N=0$.
- ▶ Solicitar al usuario que ingrese series de 2 números, sumarlos y desplegarlos mientras la suma sea distinta de cero. Al finalizar el programa mostrar la suma total.
- ▶ Mostrar los múltiplos de 3 a partir del número 15 hasta el 30.
- ▶ Mostrar los múltiplos de 3 y 5 simultáneamente menores de 100.