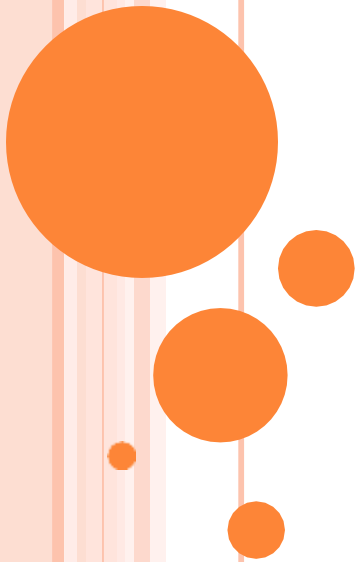


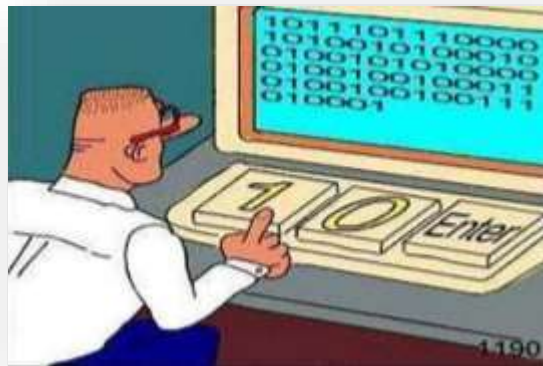
INTRODUCCIÓN A LA PROGRAMACIÓN

Herramientas de análisis
para planificar algoritmos



PSEUDOCÓDIGO

- ▶ El pseudocódigo es un lenguaje de especificación de algoritmos
- ▶ Es una mezcla de lenguaje de programación y de lenguaje natural



```
INICIO
  Edad: Entero
  ESCRIBA "Cual es tu edad?"
  LEA Edad
  Si Edad >=18 entonces
    Escribe "Eres mayor de edad"
  Fin-Si
  ESCRIBA "Fin del algoritmo"
FIN
```

La idea del pseudocódigo consiste en aprovechar la flexibilidad y poder expresivo del lenguaje natural por un lado, y las reglas de composición de los lenguajes de programación de alto nivel por el otro



PSEUDOCÓDIGO: EJEMPLO

```
x ← 5  
mostrar(x)
```

- ▶ La primera línea realiza una asignación:
 - ▶ Del lado izquierdo ponemos el nombre de una variable, en este caso x
 - ▶ Del lado derecho ponemos el valor que queremos asignarle, en este caso le asignamos el valor 5



PSEUDOCÓDIGO: EJEMPLO

$x \leftarrow 5$

mostrar(x)

En general, entonces, una asignación se escribe así:

nombreVariable \leftarrow valor



PSEUDOCÓDIGO: EJEMPLO I

$x \leftarrow 5$

mostrar(x)

- ▶ La segunda línea realiza una salida:
 - ▶ Lo que resulta de ejecutar la línea es que se muestre algo en pantalla
 - ▶ Escribimos `mostrar(...)`, y lo que se muestra es lo que está entre paréntesis `()`
 - ▶ En este caso pusimos `x`, por lo que se va a mostrar 5, que es el valor que se asignó en la línea anterior



EJEMPLO II

- ▶ Cambiaremos el programa para que sea el usuario quien decide el valor de la variable `x`
- ▶ El texto se denomina cadena
- ▶ Las cadenas sirven para mostrar mensajes, textos o símbolos a lo largo del programa, pero sin que interfieran con comandos propios del lenguaje que se esté usando
- ▶ Se escriben entre comillas para distinguirlas de los nombres de variables

“Soy una cadena” “Hola!”



EJEMPLO II

$x \leftarrow \text{ingresar}(\text{"Ingrese un número"})$
mostrar (x)

- ▶ Escribimos `ingresar(...)` , para que se abra una ventana que pida al usuario un valor, y que tenga un mensaje igual a lo que está entre paréntesis ()
- ▶ Cuando la computadora corra el programa asignará en **x** el valor que ingrese el usuario



EJEMPLO III: REALICE UN CÁLCULO

$x \leftarrow \text{ingresar}(\text{"Ingrese un número"})$

$y \leftarrow 2x + 1$

mostrar (y)

¿Qué hace este programa?



ESTRUCTURA DE LOS PROGRAMAS

Entrada de Datos

$x \leftarrow \text{ingresar}(\text{"Ingrese un valor para x:"})$

Proceso

$y \leftarrow 2x + 1$

Salida de Datos

mostrar(y)



EJEMPLO

► Entrada

Pedir al usuario que ingrese dos números enteros (**a** y **b**) diferentes entre sí.

Asignamos a una variable **c**, el valor 0 (cero) y a una variable **op**, el valor “suma”.

► Proceso

Si **a** es menor que **b**, el programa deberá sumar $a+b$

Si **a** es mayor a **b** el programa deberá restar $a-b$ y colocar en la variable **op**, el valor “resta”.

► Salida

La salida del programa será:

“La, **op**, “de”, **a**, “y”, **b**, “es”, **c**

reemplazando **op**, **a**, **b** y **c** por los valores.



EN PYTHON

1. `a = int(input("Ingrese un número entero"))`
2. `b = int(input("Ingrese otro número entero"))`
4. `c = 0`
5. `op = "suma"`
6. `if a < b:`
7. `c = a + b`
8. `else:`
9. `c = a - b`
10. `op = "resta"`
11. `print("La", op, "de", a, "y", b, "es", c)`



QUÉ VERÉ EN LA PANTALLA?

Ingrese un número entero

25

Ingrese otro número entero distinto al anterior

34

La suma de 25 y 34 es 59



cuando **a**
sea mayor
que **b**

cuando **a**
sea menor
que **b**

Ingrese un número entero

83

Ingrese otro número entero distinto al anterior

22

La resta de 83 y 22 es 61