

# LÓGICA Y REPRESENTACIÓN I

Programa de Ingeniería de Sistemas

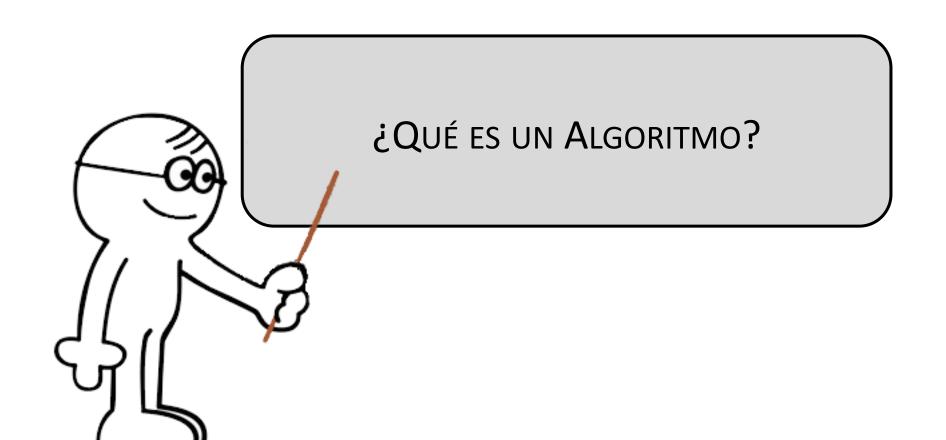
# **C**ONTENIDO



#### ALGORITMOS SECUENCIALES

- ¿Qué es un algoritmo?
- La entrada, el proceso y la salida de un algoritmo
- Formas de escribir un algoritmo
- Elementos de un algoritmo
- Operadores
- Expresiones aritmeticológicas

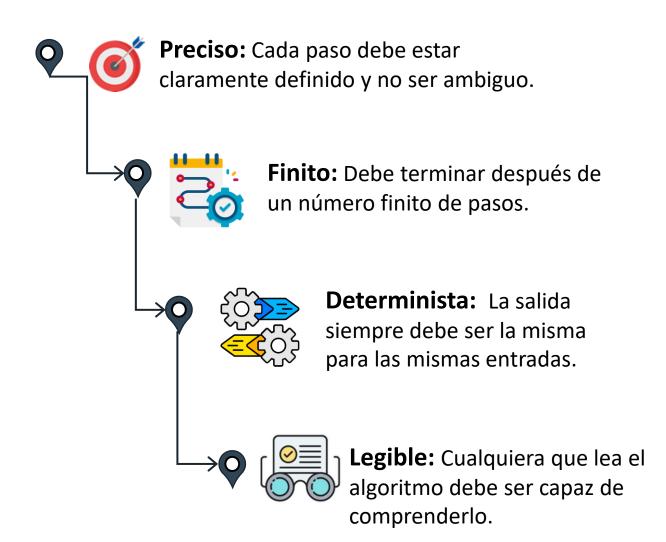






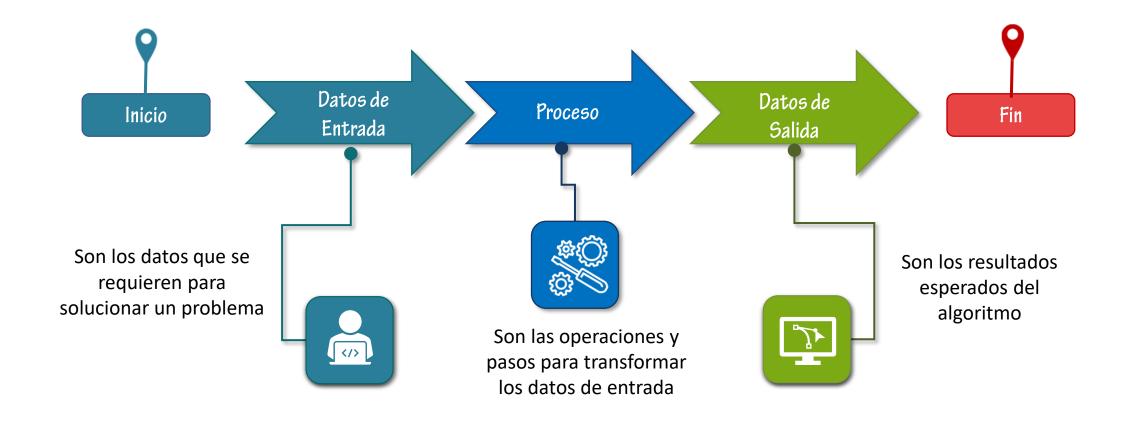
### DEFINICIÓN DE ALGORITMO

Es una secuencia ordenada de instrucciones bien definidas, que se ejecutan en un tiempo finito. Las instrucciones describen el paso a paso para dar solución a un problema.





**PARTES DE UN ALGORITMO:** independiente de la forma de representar un algoritmo, éste tiene la siguiente estructura general:





FORMAS DE REPRESENTACIÓN: existen diferentes formas para representar un algoritmo.

### Lenguaje natural

Se caracterizan por ser muy simples de leer puesto que usa el lenguaje común para describir textualmente el paso a paso de las actividades que se deben realizar para resolver el problema.

### Diagramas de flujo

Utilizan símbolos gráficos para representar los pasos que solucionan el problema. Cada símbolo tiene un significado específico, lo que facilita la lectura y el entendimiento lógico de la solución.



### Seudocódigo

El pseudocódigo utiliza una mezcla de palabras clave en un formato similar al de un lenguaje de programación, pero sin seguir una sintaxis estricta.





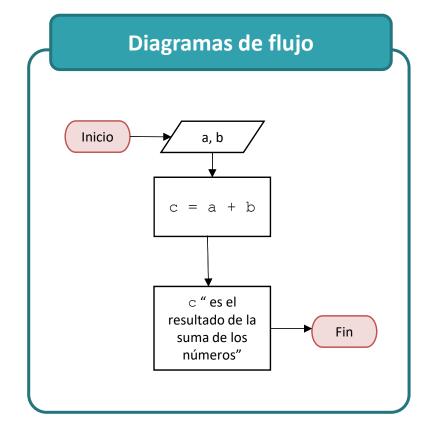
FORMAS DE REPRESENTACIÓN - EJEMPLO: escriba un algoritmo que pida dos números y muestre el resultado de su suma.

### Lenguaje natural

#### Inicio

- 1. Tome dos números y llámelos a y b.
- 2. Llame c al resultado de sumar a y b.
- 3. Muestre c como el resultado de la suma entre a y b.

Fin



### Seudocódigo

#### Inicio

Fin



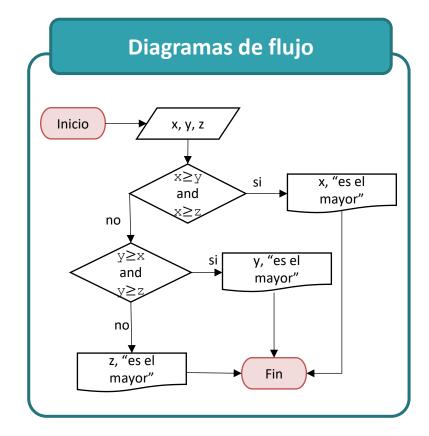
FORMAS DE REPRESENTACIÓN - EJEMPLO: escriba un algoritmo que determine cuál es el mayor entre tres números ingresados por teclado.

#### Lenguaje natural

#### Inicio

- 1. Tome tres números y llámelos x, y e z.
- 2. Compare x con y e z. Si x mayor o igual que y e z, muéstrelo como el número mayor.
- Si x no es el mayor, entonces compare y con x e z. Si y es mayor o igual que x y z, muéstrelo como el número mayor.
- Si y tampoco es el mayor, entonces muestre a z como el número mayor.

Fin



#### Seudocódigo

#### Inicio

```
Escribir("Ingrese 3 números")
Leer(x,y,z)
Si (x>=y and x>=z) Entonces
    Escribir(x, "es el mayor")
Sino
    Si (y>=x and y>=z) Entonces
        Escribir(y, "es el mayor")
    Sino
        Escribir(z, "es el mayor")
    Fin_Si
Fin_si
Fin
```

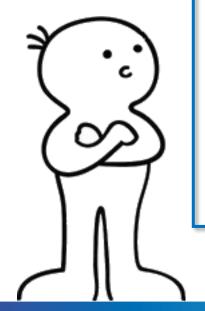


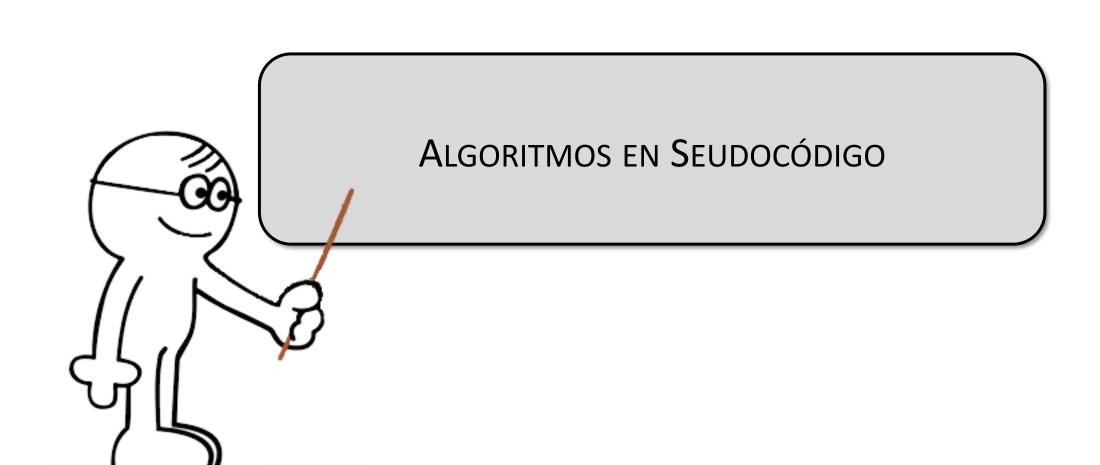


### **EJERCICIO**

Identifique las entradas y las salidas. Escriba el algoritmo, en lenguaje natural, de los siguientes problemas:

- Lavarnos las manos
- Cambiar un bombillo
- Preparar una libra de arroz
- Llegar en moto desde la UdeA al éxito de Robledo







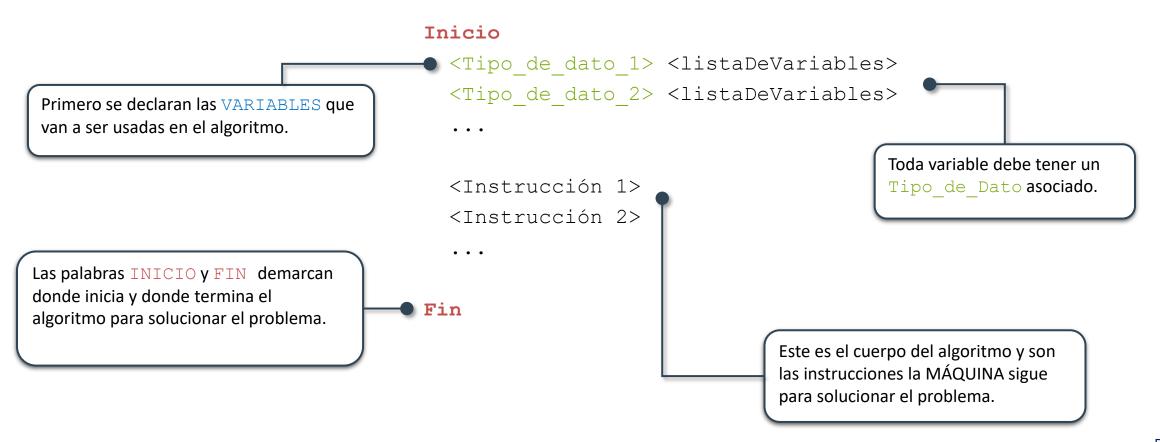
ALGORITMOS EN SEUDOCÓDIGO: El seudocódigo se usa principalmente para escribir algoritmos cuantitativos en los que las entradas siguen un proceso lógico de transformación para obtener una salida. Este es un lenguaje intermedio entre el lenguaje natural (bien sea en español o al inglés) y los lenguajes de programación, por lo tiene un conjunto de instrucciones predefinidas (o palabras clave).

#### Ventajas:

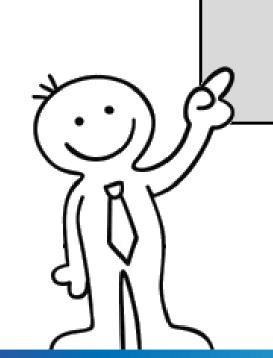
- El seudocódigo es más estructurado y formal que el lenguaje natural por lo que proporciona una forma más clara y precisa de expresar los pasos del algoritmo, evitando ambigüedades.
- Facilita la comprensión de la lógica de la solución del problema sin estar sujeto a una implementación en un lenguaje de programación específico.
- A diferencia de los diagramas de flujo, el seudocódigo puede traducirse fácilmente a diferentes lenguajes de programación.



#### **ESTRUCTURA DE UN ALGORITMO EN SEUDOCÓDIGO:**



# ELEMENTOS DE UN ALGORITMO EN SEUDOCÓDIGO



### **ELEMENTOS DE UN ALGORITMO: COMENTARIOS**



- Los <u>COMENTARIOS</u> son sentencias que permiten explicar las líneas de código de un programa. Los comentarios siempre son ignorados por el compilador o interprete del lenguaje. Dependiendo del lenguaje, pueden existir dos tipos de comentarios:
  - **Comentarios de una línea:** son aquellos que inician con el símbolo #. Estos comentarios se utilizan para explicar de manera corta una instrucción.
    - # Esto es un comentario por lo que el intérprete o compilador lo ignora
  - Comentarios varias líneas: son aquellos que inician con los símbolos """ y terminan con los símbolos """. Se usan para describir o explicar el funcionamiento de un bloque de código.

111111

Este es un comentario de varias líneas, lo que quiere decir que todo lo que está entre estos dos símbolos es ignorado.

 $\Pi\Pi\Pi\Pi$ 

### **ELEMENTOS DE UN ALGORITMO: COMENTARIOS**



#### EJEMPLO DE USO DE COMENTARIOS

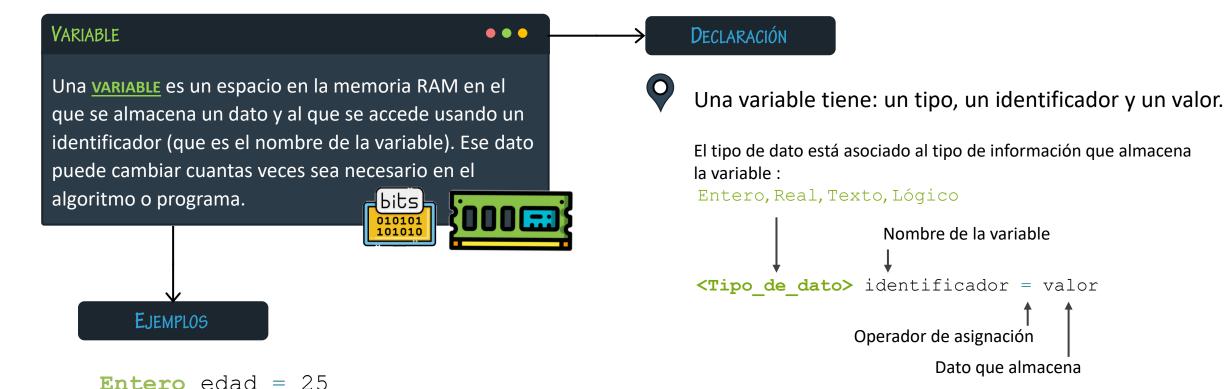
```
# Este algoritmo pide dos números al usuario, los suma y muestra el resultado en la pantalla
Inicio
  #Se declaran las variables requeridas para el programa
 Real a, b, c
  #Esta muestra un mensaje en la pantalla pidiendo los dos números al usuario
  Escribir ("Ingrese los números a sumar: ")
  #Esta línea almacena en las variables a y b los valores digitados
 Leer (a, b)
 #Esta línea suma los valores en a y b y guarda el resultado en c
 c = a + b
 Escribir ("La suma de los 2 números ingresados es ", c)
Fin
```

### **ELEMENTOS DE UN ALGORITMO: VARIABLES**

Real peso = 59.5

Texto nombre = "Carlos"





### **ELEMENTOS DE UN ALGORITMO: CONSTANTES**



#### CONSTANTES

Una <u>constante</u> es una ubicación de la memoria RAM que tiene un identificador y en el que se almacena un valor que NO cambiará durante la ejecución del programa.





Una constante, igual que una variable tiene: un tipo, un identificador y un valor. Sin embargo, el <u>identificador de una constante se escribe en mayúsculas sostenidas</u>.

Tipo de información que almacena (en lenguajes tipados):

Entero, Real, Texto, Lógico





En <u>Python</u>, y otros lenguajes, las constantes no existen, en su lugar se usan variables con identificadores en mayúsculas. Es decir, que si en un programa nos encontramos con un identificador en mayúsculas sabremos que no debe ser alterado porque es una constante.

### **ELEMENTOS DE UN ALGORITMO: BUENAS PRÁCTICAS**



#### BUENAS PRÁCTICAS

Las buenas prácticas ayudan a escribir código más legible, mantenible y eficiente. Además, que facilita a otros desarrolladores la lectura y seguimiento de este.



Hay diferentes notaciones para establecer los nombres de las variables, use siempre la misma notación.

#### Notación camelCase:

Texto nombreCompleto
Real salarioNeto

#### Notación snake\_case:

Texto nombre\_completo
Real salario neto



Usar nombres autodescriptivos, los nombres de las variables deben indicar claramente qué almacenan:

Entero edad
Real peso
Texto nombre asignatura



**Evite** usar variables de una sola letra, son difíciles de encontrar y no indican lo que almacenan:

Entero a, b, c, d



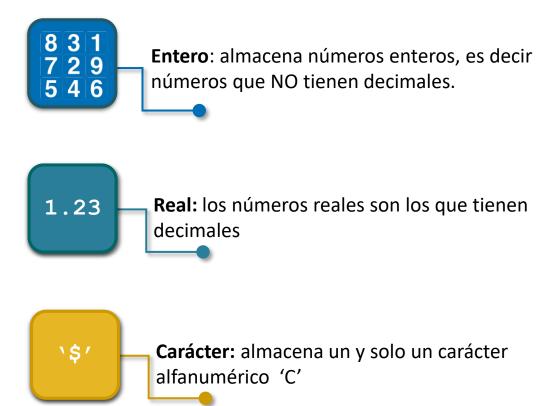
**NO USE** mayúsculas para las variables, estas están reservadas para las constantes:

Entero NOMBRE

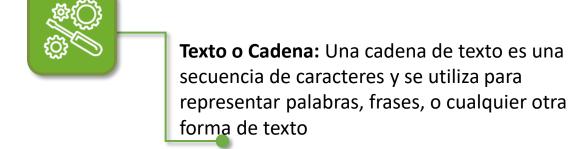
### **ELEMENTOS DE UN ALGORITMO: TIPOS DE DATOS**



En los lenguajes de programación tipados, las variables y las constantes deben tener un tipo de dato definido, lo que determina qué valores se pueden almacenar en dicha variable o constante. Algunos tipos de datos son:









### OPERADORES ARITMÉTICOS

Este tipo de operador nos permite operar variables con valores numéricos y su resultado es un valor numérico. Es decir, este tipo de operadores recibe dos números y su resultado es otro número.

Operadores Aritméticos			
Operador	Símbolo	Ejemplo	En Python
Suma	+	х + у	х + у
Resta	_	х - у	х - у
Multiplicación	*	х * у	х * у
División	/	х / у	х / у
Residuo o Módulo	Mod	x mod y	х % у
Potencia	^	х ^ у	х ** у
División entera	div	x div y	х // у



#### OPERADORES RELACIONALES

Este tipo de operadores nos permiten comparar valores o variables numéricas, pero su resultado es un valor lógico (True o False).

Operadores Relacionales			
Operador	Símbolo	Ejemplo	En Python
¿Son iguales a y b?	==	a == b	a == b
¿Son distintos a y b?	<> , !=	a <> b	a != b
¿Es a menor que b?	<	a < b	a < b
¿Es a mayor que b?	>	a > b	a > b
¿Es a menor o igual que b?	<=	a <= b	a <= b
¿Es a mayor o igual que b?	>=	a >= b	a >= b



### OPERADORES LÓGICOS

Estos operadores reciben valores lógicos y su resultado es un valor lógico.

Operadores Lógicos			
Operador	Símbolo	Ejemplo	En Python
Negación	not, !, ¬	not p	not p
Conjunción	and, ∧, &&	p and q	p and q
Disyunción	or, V,	p or q	porq



### OPERADORES ENTRADA/SALIDA

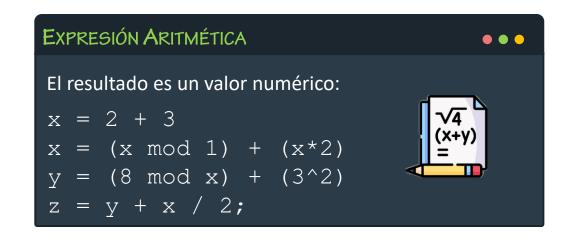
Estos operadores son los que nos permiten mostrar información en pantalla o leer datos que se ingresan por teclado.

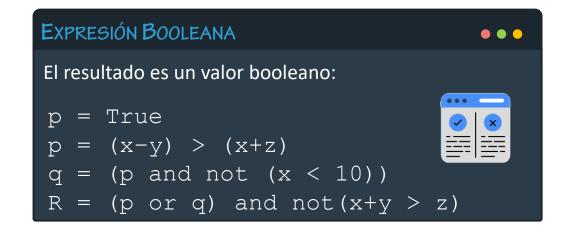
	Operadores de Entrada y Salida		
Operador	Símbolo	Ejemplo	Descripción
Escritura	Escribir, Imprimir, Mostrar	Escribir("Ingrese un número: ")	El operador de escritura se utiliza para mostrar un mensaje en la pantalla
Lectura	Leer, Lea	Leer (x)	El operador de lectura se utiliza para guardar un valor ingresado por el usuario en una variable determinada



#### EXPRESIONES ARITMETICOLÓGICAS

Una expresión se compone de un conjunto de operadores y operandos. Los operadores nos permiten realizar operaciones aritméticas, relacionales o lógicas entre variables y/o constantes. Veamos dos tipos de expresiones:







### **EXPRESIONES ARITMETICOLÓGICAS**

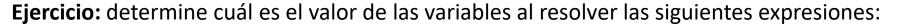
Para el desarrollo de expresiones aritmeticológicas se debe considerar la precedencia entre tipos de operadores y

entre operadores.

	Operador	Tipo de Operador
	()	Paréntesis
	^	Potencia
OPERADORES ARITMÉTICOS	* / DIV MOD	Multiplicativos
	+ -	Aditivos
OPERADORES RELACIONALES	< <= > >=	Relacionales
	== !=	Igualdad
	!	Negación
OPERADORES BOOLEANOS –	&&	AND Lógico
	П	OR Lógico
Operador de Asignación —	=	Asignación







$$x = 5 + 3 * 11$$
  
 $y = (x - 17) - 11 * 2$   
 $x = -3 * 5 + x / 4 * 6$ 

$$z = (-3 * 5 + 2) / (4 - 5)$$

$$y = 8 ^ (3^(-1)) + x div 5 - 4$$

$$p = 5 - 2 * x > y + 4 \mod 2 * 2$$

$$q = not p and x != y or y > z$$

$$r = 'a' == 'A' \text{ or } 4 * 2 ^ (-2) != 2 - 1$$

$$p = 3 + 2 == 5$$
 and  $p == 'p'$ 





#### EXPRESIONES ARITMETICOLÓGICAS

Determine cuales son los valores de las variables en las siguientes expresiones:

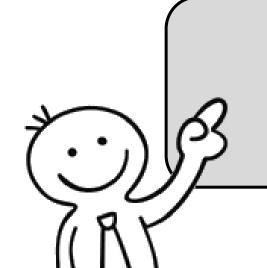
$$c = 5 * 4 + 8 / 2 - 1$$
 $m = 4 * (5 + 2) * 3 / (5 - 1)$ 
 $b = 12 + (3 * 4 * (7 - 5) / 6) - 1$ 

¿Cuál es la expresión algorítmica que representa las siguientes fórmulas matemáticas?

$$z = x^2 + y^2$$

$$y = \frac{x^2}{a^3 - 2a + \sqrt{b^3}}$$

$$w = \left(\frac{(a+b)}{a+\frac{c^2}{d+e}}\right)^3$$



# ALGORITMOS SECUENCIALES EJEMPLOS







**EJEMPLO:** La empresa encuestadora XYZ requiere de un programa que pida a una persona su nombre, año de nacimiento y peso. Después de ingresar estos datos, el programa debe mostrar el nombre de la persona, su edad (aproximada) y su peso.



#### ANÁLISIS DEL PROBLEMA

Problema: La empresa encuestadora XYZ requiere de un programa que le solicite a una persona su nombre, la edad y el peso, y muestre su nombre , la edad y el peso en pantalla.

#### **CLIENTE Y USUARIO**

•



#### **ENTIDADES DEL MUNDO**



#### **REQUERIMIENTOS FUNCIONALES**



### ENTRADAS, SALIDAS Y EL PROCESO

-





### ANÁLISIS DEL PROBLEMA

Cliente	La empresa XYZ
Usuario	Las personas que serán encuestadas
Requerimientos Funcionales	El sistema debe permitir:  Mostrar en la pantalla el nombre, la edad y el peso de una persona
Entidades del Mundo	■ La persona, de la que se requieren el nombre, año de naciemiento y el peso
Entradas	<ul> <li>El nombre de la persona que es una cadena de texto</li> <li>El año de nacimiento que es un número entero</li> <li>El peso de la persona que es un número real</li> </ul>
Salidas	<ul> <li>Un mensaje que muestra el nombre y la edad aproximada de la persona</li> </ul>
Proceso	<ol> <li>Se piden los datos de entrada (nombre, año de nacimiento y peso de la persona)</li> <li>Se calcula la edad aproximada</li> <li>Se muestra un mensaje en la pantalla con el nombre, la edad y peso de la persona</li> </ol>



#### DISEÑO DE LA SOLUCIÓN

```
# Este algoritmo pide el nombre, la edad y el peso de una persona y los muestra en la pantalla
Inicio
 Texto nombre
 Entero anno nacimiento, edad
  Real peso
  # 1. Se piden los datos de entrada al usuario
  Escribir ("Ingrese el nombre, año de nacimiento y peso de la persona: ")
 Leer (nombre, anno nacimiento, peso)
  # 2. Calcula la edad
  edad = 2024 - anno nacimiento
  # 3. Se muestra en la pantalla los datos
 Escribir ("Te llamas ", nombre, " y tienes ", edad, " años y dices que pesas ", peso, " Kg.")
Fin
```







**EJEMPLO:** como el profesor de geometría quiere ayudar a sus estudiantes de una manera práctica con una app para la clase, le ha pedido a usted que para empezar desarrolle un programa que le permita calcular el área y el perímetro de un rectángulo.



#### ANÁLISIS DEL PROBLEMA

Problema: El profesor de geometría quiere ayudar a sus estudiantes de una manera práctica con una app para la clase, por eso le ha pedido a usted que desarrolle un programa que le permita a los estudiantes calcular el área y el perímetro de un rectángulo.

#### **CLIENTE Y USUARIO**



#### **ENTIDADES DEL MUNDO**

- .
- \_



#### **REQUERIMIENTOS FUNCIONALES**



### ENTRADAS, SALIDAS Y EL PROCESO

- \_





#### ANÁLISIS DEL PROBLEMA

Cliente	El profesor de geometría
Usuario	Los estudiantes de geometría
Requerimientos Funcionales	El sistema debe permitir:  Calcular el perímetro de un rectángulo  Calcular el área de un rectángulo
Entidades del Mundo	■ El Rectángulo, del que se requieren la base y la altura
Entradas	Base y altura del rectángulo, ambos son números reales
Salidas	<ul><li>El área del rectángulo</li><li>El perímetro del rectángulo</li></ul>
Proceso	<ul> <li>Se solicita al usuario la base y la altura del rectángulo</li> <li>Se calcula el área con la fórmula: area = base * altura</li> <li>Se calcula el perímetro con la fórmula: perimetro = 2*base + 2*altura</li> <li>Se muestra en la pantalla el área y el perímetro calculados</li> </ul>



#### DISEÑO DE LA SOLUCIÓN

#Este algoritmo calcula el perímetro y el área de un rectángulo con base en su base y altura Inicio Real base, altura, area, perimetro #Se piden los datos de entrada al usuario Escribir ("Ingrese la base y la altura del rectángulo: ") Leer (base, altura) #Se calcula el área y el perímetro area = base \* altura perimetro = 2\*base + 2\*altura #Se muestra el resultado Escribir ("El área del rectángulo es ", area, " y el perímetro es ", perimetro) Fin







**EJEMPLO:** Juanito requiere un algoritmo que al ingresar una hora en el formato hh:mm:ss muestre cuál será la hora pasadas 2 horas, 37 minutos y 28 segundos.



#### ANÁLISIS DEL PROBLEMA

Problema: Juanito requiere un programa en la Web que, al ingresar una hora en el formato hh:mm:ss, muestre cuál será la hora pasadas 2 horas, 37 minutos y 28 segundos.

#### **CLIENTE Y USUARIO**



#### **ENTIDADES DEL MUNDO**



#### **REQUERIMIENTOS FUNCIONALES**



### ENTRADAS, SALIDAS Y EL PROCESO

-





### ANÁLISIS DEL PROBLEMA

CLIENTE	Juanito
Usuario	Juanito
REQUERIMIENTOS FUNCIONALES	El sistema debe permitir:  Registrar la hora en el formato hh:mm:ss  Mostrar la hora que corresponde a pasadas 2 horas, 37 minutos y 28 segundos
ENTIDADES DEL MUNDO	■ La Hora, de la cual se debe almacenar las horas, minutos y segundos
ENTRADAS	<ul> <li>Horas (hh), que es un número entero entre 0 y 23</li> <li>Minutos (mm) que es un número entero entre 0 y 59</li> <li>Segundos (ss) que un número entre 0 y 59</li> </ul>
SALIDA	■ Una nueva hora
PROCESO	<ul> <li>Se debe pedir la hora en el formato hh:mm:ss</li> <li>Se suma a los segundos el valor de 28 teniendo presente que, si el resultado es mayor a 59, se reinician los segundos y se aumentan los minutos en 1.</li> <li>Se suma a los minutos el valor de 37 y el acarreo de la operación anterior, teniendo presente que, si el resultado es mayor a 59, se reinician los minutos y se aumenta las horas en 1.</li> <li>Se suma a las horas 2 y el acarreo de la operación anterior teniendo presente que, si el resultado es mayor 23 se deben reiniciar las horas.</li> </ul>

Dianositiva # 39

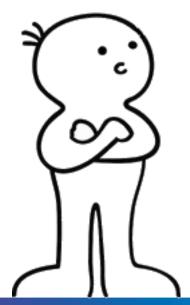


#### DISEÑO DE LA SOLUCIÓN

```
# Este algoritmo calcula una nueva hora para una hora de base
Inicio
  Entero hh, mm, ss, newHH, newMM, newSS, acarreo
  #Se piden los datos de entrada al usuario
  Escribir ("Ingrese la hora en formato HH:MM:SS: ")
  Leer (hh, mm, ss)
  newSS = (ss + 28) \mod 60
  acarreoMM = (ss + 28) div 60 #Acarreo que dejan los segundos
  newMM = (mm + acarreoMM + 37) MOD 60
  acarreoHH = (mm + acarreoMM + 37) DIV 60 #Acarreo que dejan los minutos
  newHH = (hh + acarreoHH + 2) MOD 24
  Escribir ("La nueva hora es ", newHH, ":", newMM, ":", newSS)
Fin
```







**EJEMPLO:** El gerente de la empresa "*Working*" requiere que se desarrolle una aplicación que permita calcular el nuevo salario de un empleado sabiendo que este se incrementa en un 25%.



#### ANÁLISIS DEL PROBLEMA

Problema: El gerente de la empresa "Working" requiere que se desarrolle una aplicación que permita calcular el nuevo salario de un empleado sabiendo que este se incrementa en un 25%.

#### **CLIENTE Y USUARIO**



#### **ENTIDADES DEL MUNDO**

.



#### **REQUERIMIENTOS FUNCIONALES**



### ENTRADAS, SALIDAS Y EL PROCESO

-





### ANÁLISIS DEL PROBLEMA

CLIENTE	La empresa " <i>Working</i> "
USUARIO	El departamento de talento humano
REQUERIMIENTOS FUNCIONALES	El sistema debe permitir:  Calcular y mostrar el salario actualizado de un empleado
ENTIDADES DEL MUNDO	■ Empleado, la cual hace referencia al empleado de la empresa "Working" y del cual se requiere para el problema sólo el salario
ENTRADAS	El salario del empleado que es un número real
SALIDA	El nuevo salario del empleado
PROCESO	<ul> <li>Se pide el salario inicial del empleado</li> <li>Se calcula el nuevo salario así: salario_nuevo = salario_actual * 1.25</li> <li>Se muestra en la pantalla el nuevo salario calculado</li> </ul>



#### DISEÑO DE LA SOLUCIÓN

#Este algoritmo calcula el nuevo salario de un empleado con base en su salario actual

### Inicio

```
#Se piden los datos de entrada al usuario
Escribir ("Ingrese el salario actual del empleado: ")
Leer (salario_actual)

salario_nuevo = salario_actual + salario_actual*0,25

Escribir ("El nuevo salario del empleado es ",salario_nuevo)
Fin
```



#### **EJERCICIO:**

La empresa "*Empleos por Montón*" requiere de un programa para calcular el salario semanal de sus empleados sabiendo que se paga a \$ 22000 por hora. La empresa hace los descuentos de ley que corresponden al 8% por salud, 4% por pensión y 10% por retención en la fuente.

Haga un algoritmo que pida el número de horas que un empleado trabajó en la semana y con base en estas calcule y muestre el salario bruto, el valor de cada descuento aplicado y el salario neto.





# LÓGICA Y REPRESENTACIÓN I

Programa de Ingeniería de Sistemas