

## LÓGICA Y REPRESENTACIÓN I

Programa de Ingeniería de Sistemas

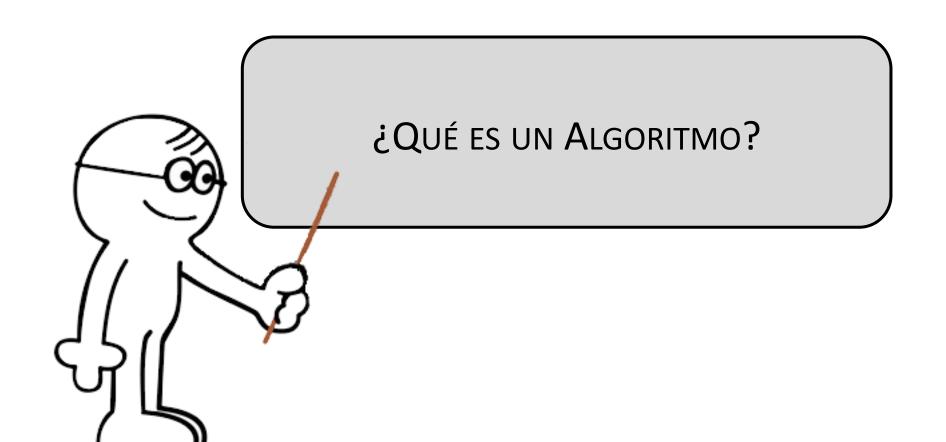
### **C**ONTENIDO



#### 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



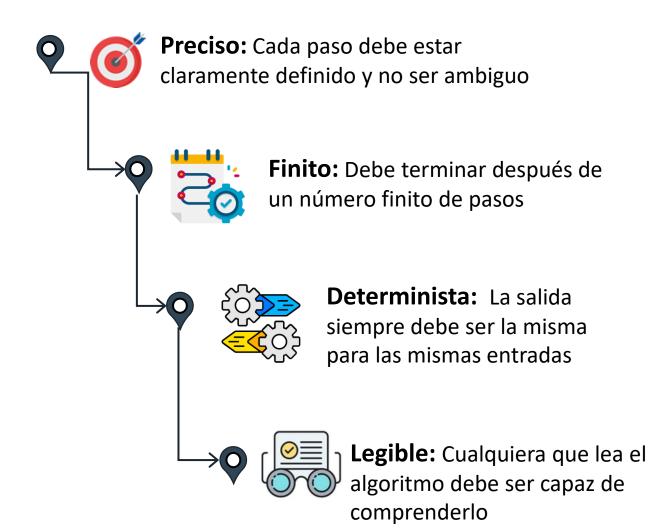




#### DEFINICIÓN DE ALGORITMO

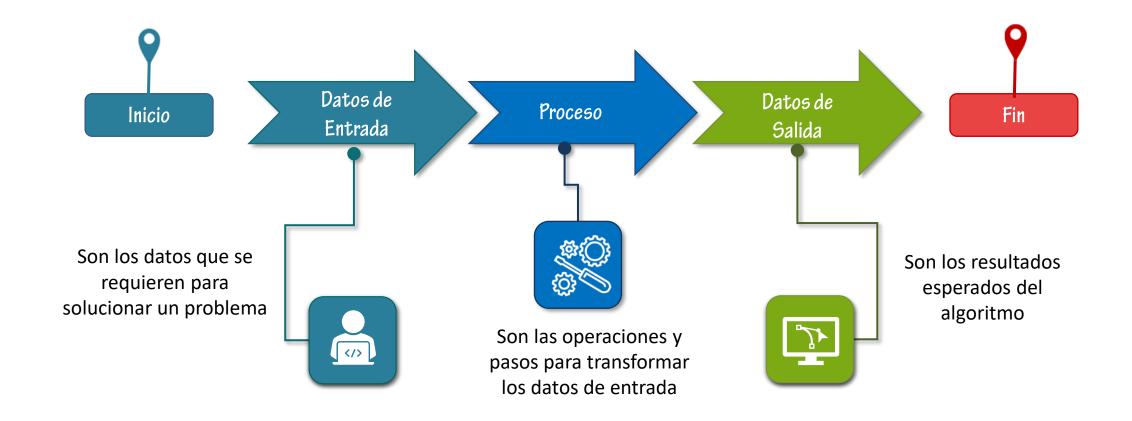
Es una secuencia ordenada de instrucciones bien definidas, que se ejecutan en un tiempo finito. Las instrucciones describen el paso a

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 su lectura y entender la lógica de la solución al problema, una vez se conocen los símbolos.

#### 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.





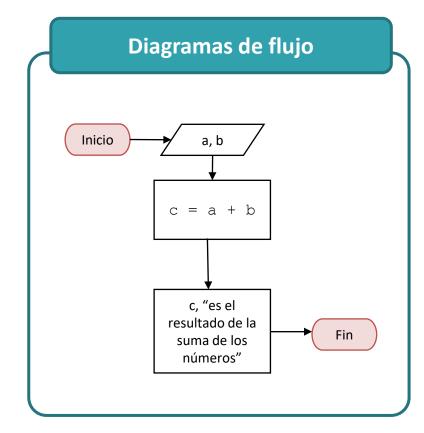
FORMAS DE REPRESENTACIÓN - EJEMPLO: escriba un algoritmo que pida dos números y muestre el resultado de su suma

#### 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

Escribir("Ingrese 2 números")

Leer(a, b)

c = a + b

Escribir(c, "es el resultado de

la suma de los números")

Fin



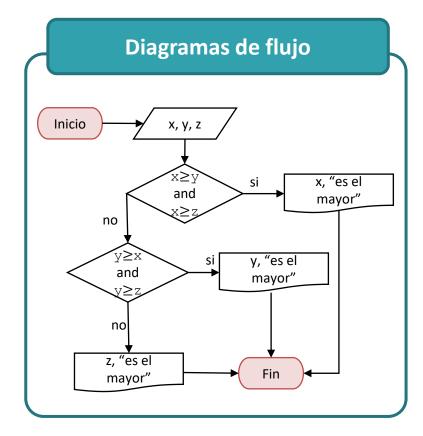
FORMAS DE REPRESENTACIÓN - EJEMPLO: escriba un algoritmo que determine cual es el mayor entre tres números ingresados por teclado

#### 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.
- 3. 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.
- 4. Si y tampoco es el mayor, entonces muestre a z como el número mayor.

Fin



#### Seudocódigo

# 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

Fin si

Fin Si

Inicio

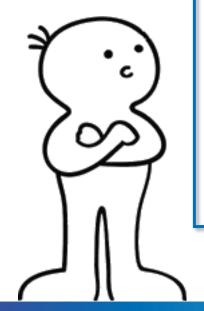


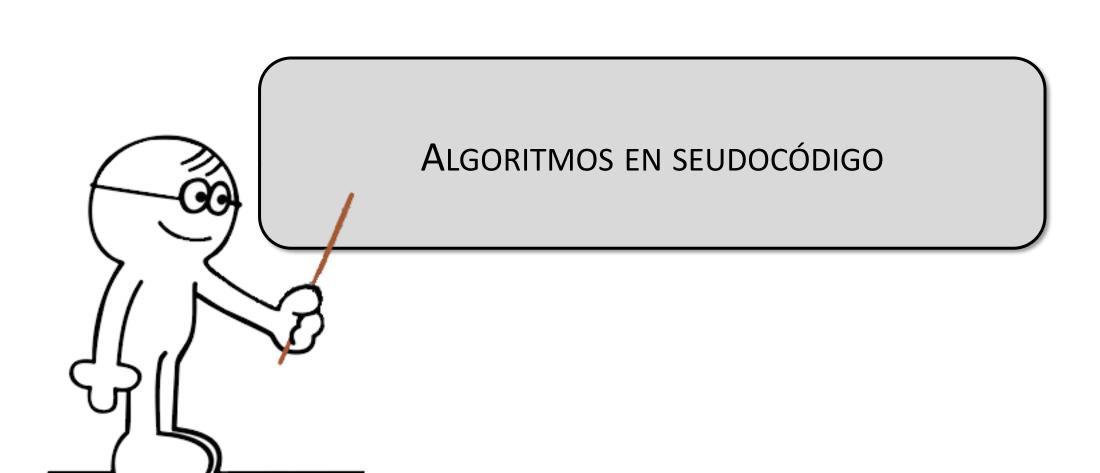


#### **EJERCICIO**

Identifique las entras y las salidas, y 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







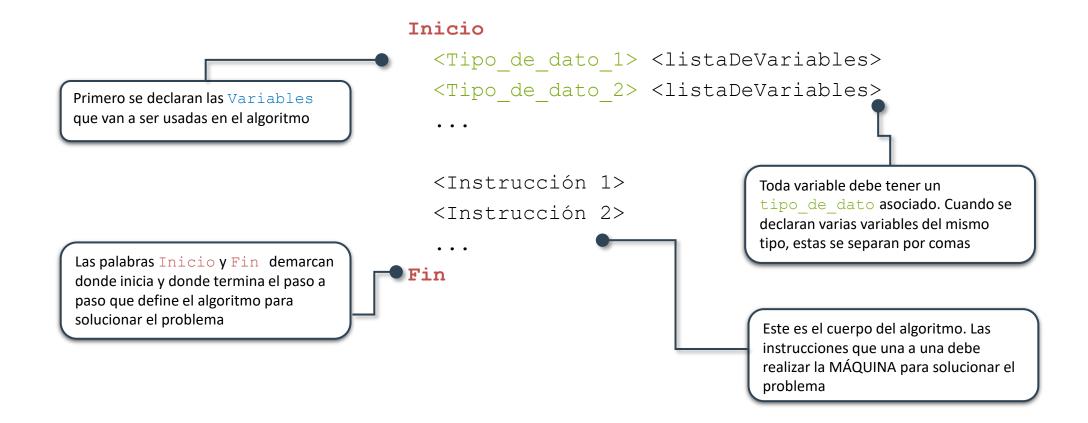
ALGORITMOS EN SEUDOCÓDIGO: El seudocódigo se usa principalmente para escribir algoritmos cuantitativos en los que las entradas siguen un proceso lógico de transformación para obtener una salida. Este lenguaje es un lenguaje intermedio entre el lenguaje natural (bien sea en español o al inglés) y los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

#### Ventajas:

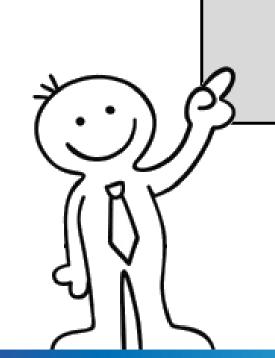
- El pseudocódigo es más estructurado y formal que el lenguaje natural por lo que proporciona una forma más clara y precisa de expresar los pasos del algoritmo, evitando ambigüedades.
- Facilita la comprensión de la lógica de la solución del problema sin estar atado a una implementación en un lenguaje de programación específico
- A diferencia de los diagramas de flujo, el pseudocódigo 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 comentarios son sentencias que permiten explicar las líneas de código de un algoritmo o programa. Los comentarios son ignorados, por lo que no se consideran instrucciones que la máquina debe ejecutar. Dependiendo del lenguaje, pueden existir dos tipos de comentarios:
  - Comentarios de una línea: son aquellos se indica su inicio con el símbolo # y terminan al final de la línea donde se ponen. Se utilizan para explicar de manera corta una instrucción.
    - # Esto es un comentario, esta línea será ignorada
  - Comentarios varias líneas: son aquellos que inician con los símbolos """ y terminan con los símbolos """. Se usan describir o explicar el funcionamiento de un bloque de código.

 $11\,11\,11$ 

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

1111111

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

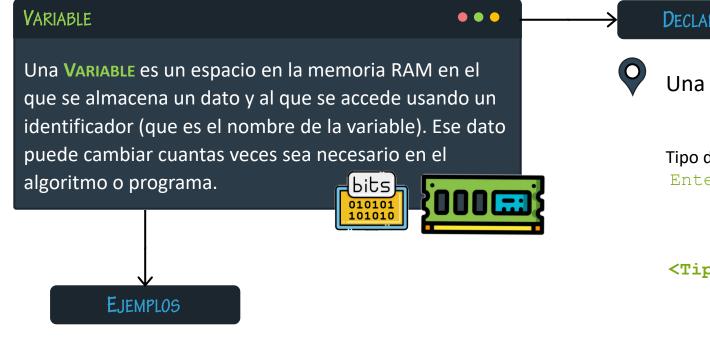


#### EJEMPLO DE USO DE COMENTARIOS

```
# 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**





Entero edad = 25
Real peso = 59.5
Texto nombre = "Carlos"

#### DECLARACIÓN

Una variable tiene: un tipo, un identificador y un valor:

Tipo de información que almacena (en lenguajes tipados):

Entero, Real, Texto, Lógico



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



#### CONSTANTES

Una Constante es una ubicación de la memoria RAM que tiene un identificador y en el que se almacena un valor que NO cambiará durante todo el algoritmo.



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

#### DECLARACIÓN



Una constante, igual que una variable tiene: un tipo, un identificador y un valor, pero el identificador se escribe en mayúsculas:

Tipo de información que almacena (en lenguajes tipados):

Entero, Real, Texto, Lógico



### **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 el mismo tipo:

#### 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:

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**

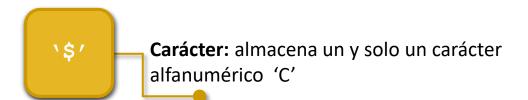


En los lenguajes de programación tipados, las variables y las constantes deben tener un tipo de dato definido, lo que determina qué valores se pueden almacenar en dicha variable o constante. Algunos tipos de datos son:

Entero: almacena números enteros, es decir números que NO tienen decimales.



1.23 Real: los números reales son los que tienen decimales





**Texto o Cadena:** Una cadena de texto es una secuencia de caracteres. Puede contener cero o más caracteres y se utiliza para representar palabras, frases, o cualquier otra forma de texto dobles



#### 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	p or q



#### OPERADORES ENTRADA/SALIDA

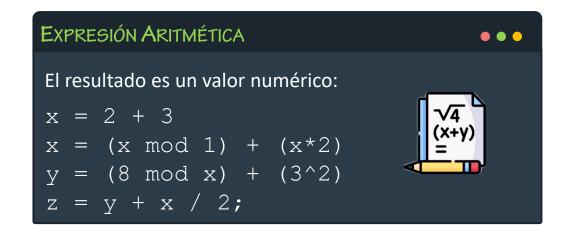
Estos operadores son los que nos permiten mostrar información en pantalla o leer datos que se ingresan por teclado.

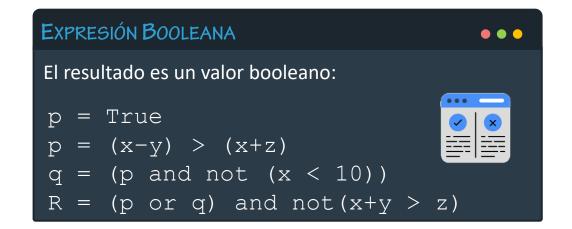
	Operadores de Entrada y Salida			
Operador	Símbolo	Ejemplo	Descripción	
Escritura	Escribir, Imprimir, Mostrar	Escribir("Ingrese un número: ")	El operador de escritura se utiliza para mostrar un mensaje en la pantalla	
Lectura	Leer, Lea	Leer (x)	El operador de lectura se utiliza para guardar un valor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

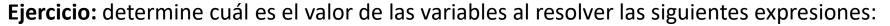
Para el desarrollo de expresiones aritmeticológicas se debe considerar la precedencia entre tipos de operadores y

entre operadores.

		Operador	Tipo de Operador
		()	Paréntesis
OPERADORES ARITMÉTICOS		^	Potencia
	OPERADORES ARITMÉTICOS	* / DIV MOD	Multiplicativos
		+ -	Aditivos
OPERADORES RELACIONALES	OPERADORES RELACIONALES	< <= > >=	Relacionales
		== !=	Igualdad
OPERADORES BOOLEANOS		!	Negación
	OPERADORES BOOLEANOS –	&&	AND Lógico
		_ 11	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:

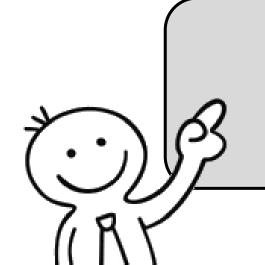
$$c = 5 * 4 + 8 / 2 - 1$$
 $m = 4 * (5 + 2) * 3 / (5 - 1)$ 
 $b = 12 + (3 * 4 * (7 - 5) / 6) - 1$ 

¿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 le pida a una persona que será encuestada el nombre, la edad y el peso y muestre los datos ingresados en la pantalla..



#### ANÁLISIS DEL PROBLEMA

Problema: La empresa encuestadora XYZ requiere de un programa que pida le pida a una persona que será encuestada el nombre, la edad y el peso y muestre los datos ingresados en la 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 los datos ingresados de una persona que será encuestada
Entidades del Mundo  La Persona, de la que se requieren el nombre, la edad y el peso	
Entradas	<ul> <li>El nombre de la persona que es una cadena de texto</li> <li>La edad de la persona que es un número entero</li> <li>El peso de la persona que es un número real</li> </ul>
Salidas	■ Un mensaje que muestra el nombre, edad y peso ingresados para una persona
Proceso	<ul> <li>Se piden los datos de entrada (nombre, edad y peso de la persona)</li> <li>Se muestra un mensaje en la pantalla con los datos ingresados</li> </ul>



#### 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 edad
   Real peso

#Se piden los datos de entrada al usuario
   Escribir ("Ingrese el nombre, edad y peso de la persona: ")
   Leer (nombre, edad, peso)

#Se muestra en la pantalla los datos ingresados
   Escribir ("Te llamas ", nombre, " 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

- .









**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**



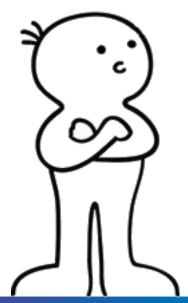
#### ENTRADAS, SALIDAS Y EL PROCESO

.









**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

-





#### **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