

# Algoritmos

para *solucionar* **Problemas**

- Contesta las siguientes preguntas.

✓ ¿Qué entiendes por algoritmo?

---

---

✓ ¿En qué áreas consideras que se aplican los algoritmos?

---

---

✓ ¿Crees que en la informática son importantes los algoritmos? ¿Por qué?

---

---

# algoritmo



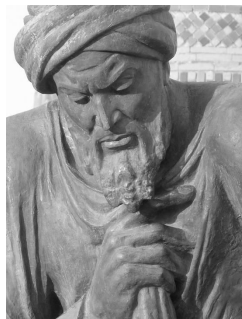
REAL ACADEMIA ESPAÑOLA



ASOCIACIÓN DE ACADEMIAS DE LA  
LENGUA ESPAÑOLA

Quizá del lat. tardío *\*algobarismus*, y este abrev. del ár. clás. *ḥisābu lġubār* 'cálculo mediante cifras arábigas'.

1. m. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
2. m. Método y notación en las distintas formas del cálculo.



Matemático árabe Mohamed Ibn Moussa Al Kow Rizmi, el cual escribió entre los años 800 y 825 su obra Quitab Al Jabr Al Mugabala, donde se recogía el sistema de numeración hindú y el concepto del cero.

Fue Fibonacci, el que tradujo su obra al latín y la inició con las palabras: Algorithmi dicit.

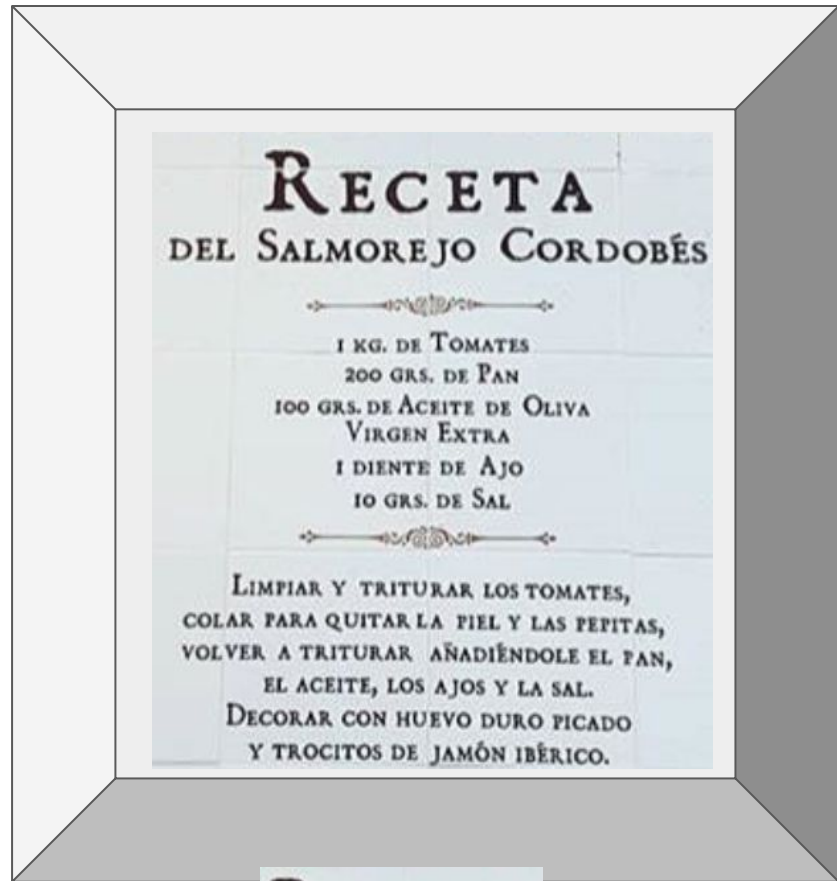
Algoritmo es un método para  
***resolver un problema***  
mediante  
***una serie de pasos definidos, precisos y finitos.***

# Algoritmo



Puede ser representado por:

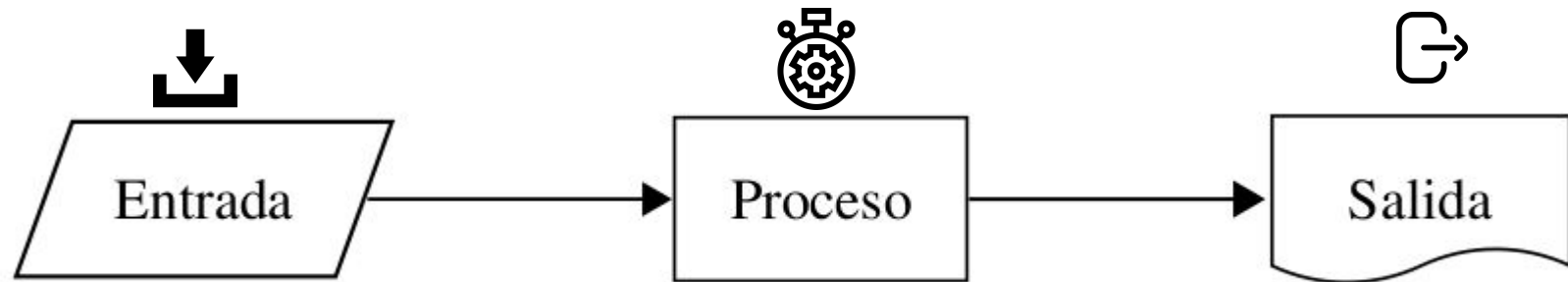
- Lenguaje natural
- Pseudocódigo
- Diagramas de flujo
- Lenguajes de programación



Realizar una **RECETA** es otra forma de implementar un **Algoritmo**.



# Lenguaje natural

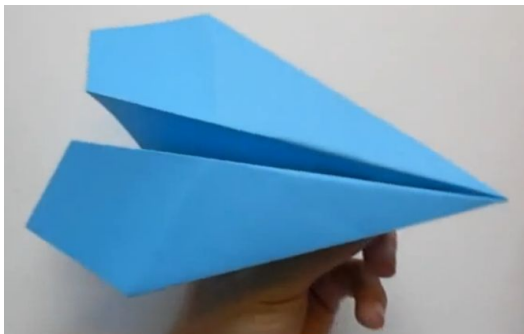
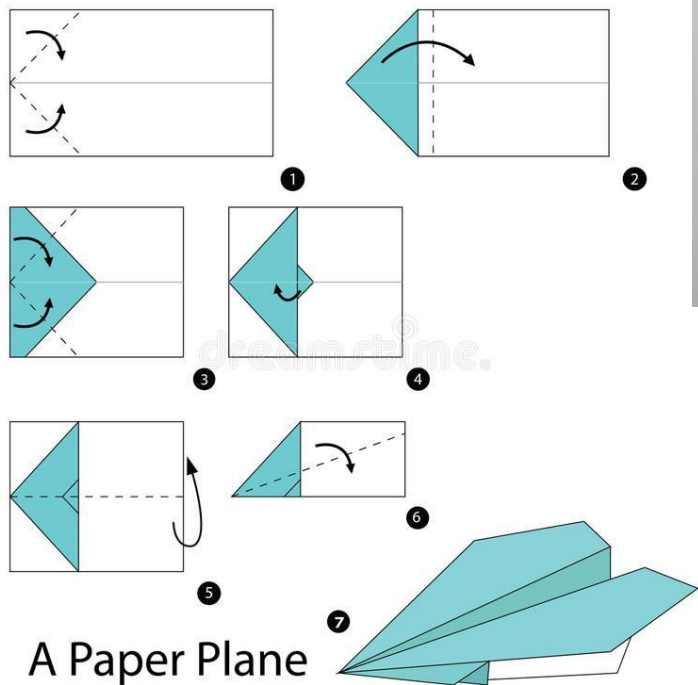
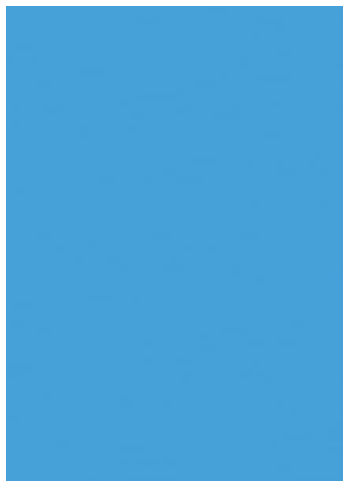
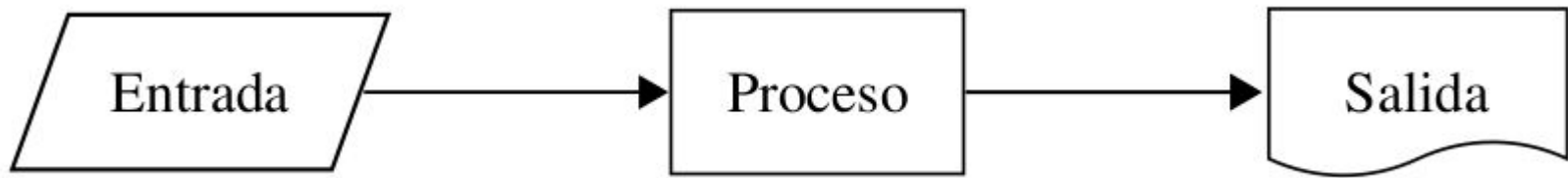


1 KG DE TOMATES  
200 GRs. DE PAN  
100GRs. DE ACEITE DE OLIVA  
VIRGEN EXTRA  
1 DIENTE DE AJO  
10 GRs. DE SAL

LIMPIAR Y TRITURAR LOS TOMATES,  
COLAR PARA QUITAR LA PIEL Y LAS PEPITAS,  
VOLVER A TRITURAR AÑADIÉNDOLE EL PAN,  
EL ACEITE, LOS AJOS Y LA SAL.  
DECORAR CON HUEVO DURO PICADO  
Y TROCITOS DE JAMÓN IBÉRICO.



**RECETA**  
DEL SALMOREJO CORDOBÉS





## Características de un algoritmo

- **Preciso:** estar compuesto de pasos **bien definidos** (no ambiguos) y **ordenados**.
- **Definido:** si se sigue dos veces, se obtiene el mismo resultado cada vez.
- **Finito:** tener un número finito de pasos.

# Ejemplo algoritmo

## Ejemplo

Lavarnos los dientes es un procedimiento que realizamos varias veces al día. Veamos la forma de expresar este procedimiento como un Algoritmo:

1. *Tomar la crema dental*
2. *Destapar la crema dental*
3. *Tomar el cepillo de dientes*
4. *Aplicar crema dental al cepillo*
5. *Tapar la crema dental*
6. *Abrir la llave del lavamanos*
7. *Remojar el cepillo con la crema dental*
8. *Cerrar la llave del lavamanos*
9. *Frotar los dientes con el cepillo*
10. *Abrir la llave del lavamanos*
11. *Enjuagarse la boca*
12. *Enjuagar el cepillo*
13. *Cerrar la llave del lavamanos*
14. *Secarse la cara y las manos con una toalla*





# Diagrama de flujo

**Símbolo**

**Significado/Función**



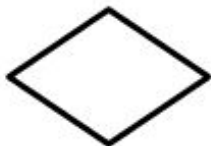
**Inicio/Fin/Conector**



**Proceso**



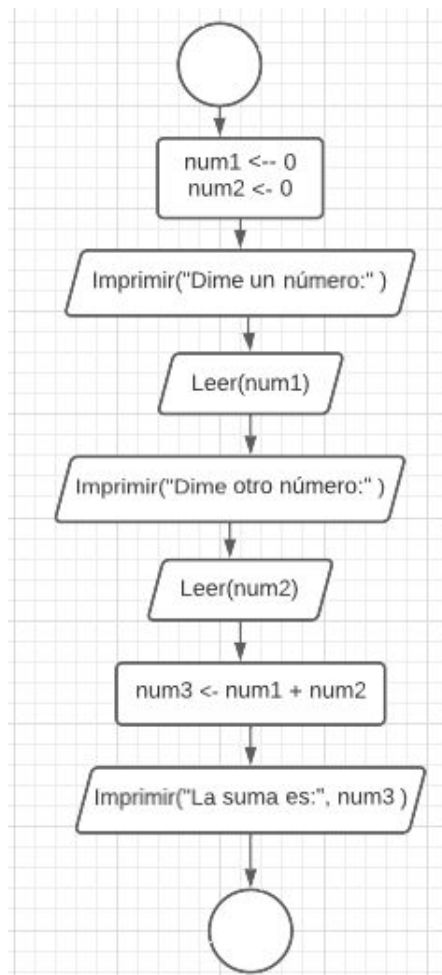
**Entrada/Salida**



**Decisión**



**Flujo de ejecución**



# Pseudocódigo – Estructuras de control

Secuencial	<instrucción_1> <instrucción_2> .... <instrucción_n>
Condicional	<b>Si</b> <condición> <b>entonces</b> <Instrucciones> <b>Sino</b> <Instrucciones>
Condicional múltiple	<b>Según sea</b> <variable> <b>Hacer</b> <b>Caso</b> valor 1: <instrucciones> <b>Caso</b> valor 2: <instrucciones> <b>Caso</b> valor n: <instrucciones> <b>Fin Según</b>
Repetir-hasta	<b>Repetir</b> <instruccione> <b>Hasta que</b> <condicion>
Hacer-mientras	<b>Mientras</b> <condición> <instrucciones> <b>Fin mientras</b>

INICIO

edad: Entero

Escribir “¿Cuál es tu edad?”

Leer edad

Si edad >= 18 entonces

    Escribir “Eres mayor de edad”

Sino

    Escribir “Eres menor de edad”

Escribir “Fin del algoritmo”

FIN






# Algoritmos

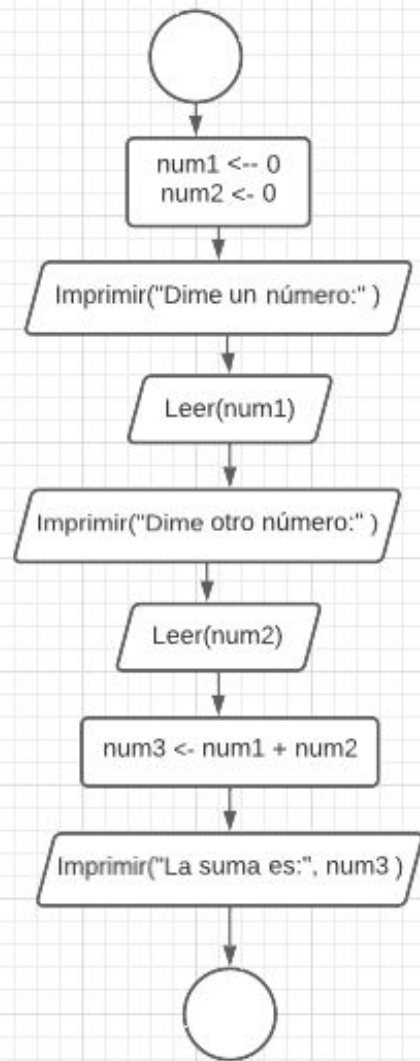
para *solucionar* **Problemas**

Sumar dos números

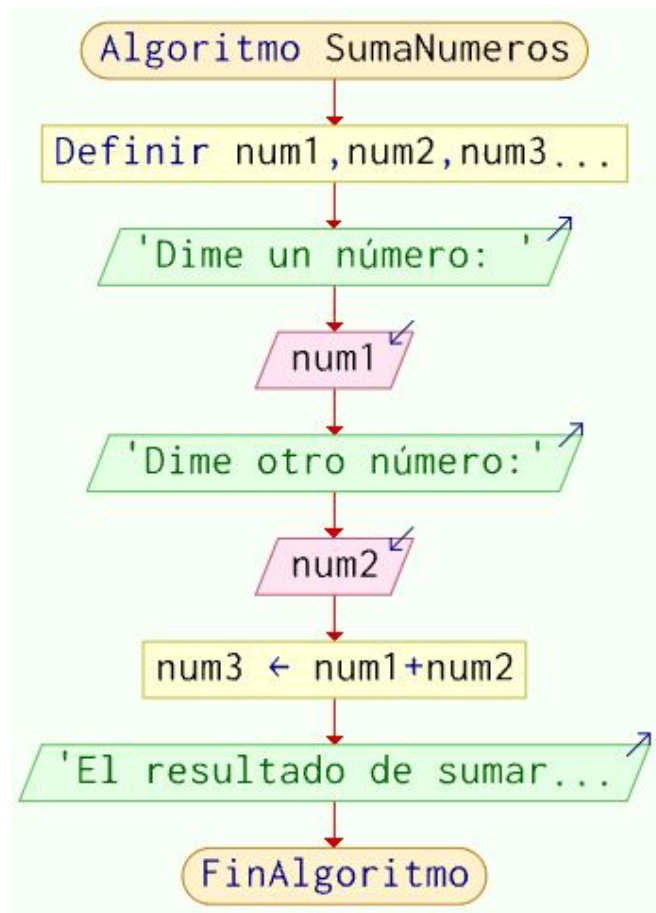


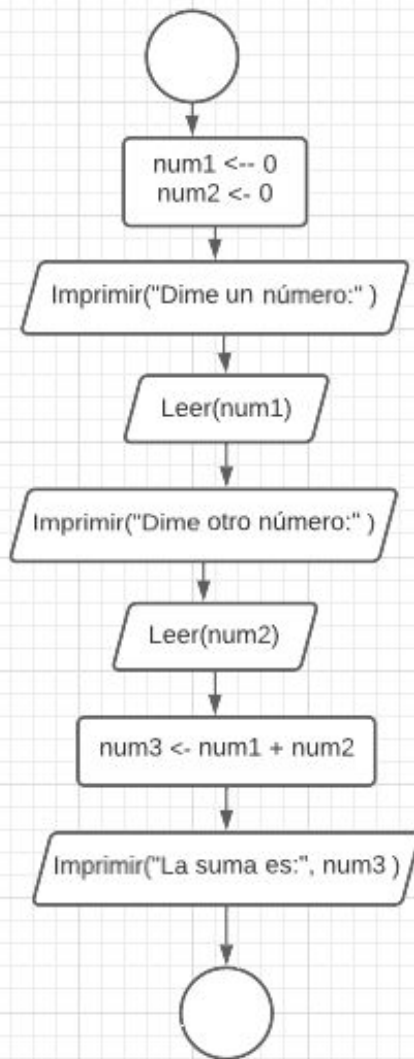
# Diagrama de flujo

Símbolo	Significado/Función
	Inicio/Fin/Conector
	Proceso
	Entrada/Salida
	Decisión
	Flujo de ejecución



```
1 Algoritmo SumaNumeros
2   Definir num1, num2, num3 Como Entero
3   Imprimir "Dime un número: "
4   Leer num1
5   Imprimir "Dime otro número:"
6   Leer num2
7   num3 ← num1 + num2
8   Imprimir "El resultado de sumar es: ", num3
9 FinAlgoritmo
```





```
#Esto es un comentario
```

```
num1 = 0
```

```
num2 = 0
```

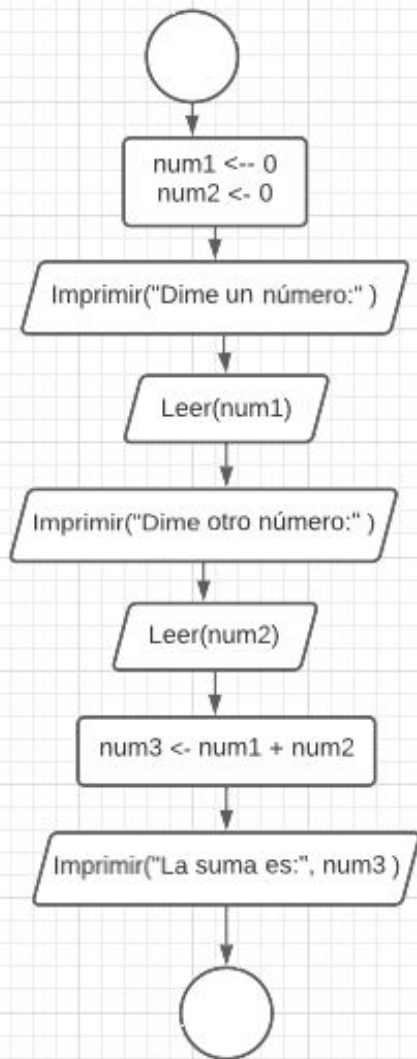
```
num1 = int(input("dime un número: "))
```

```
num2 = int(input("dime otro número: "))
```

```
num3 = num1 + num2
```

```
print("El resultado de sumar es: ", num3)
```

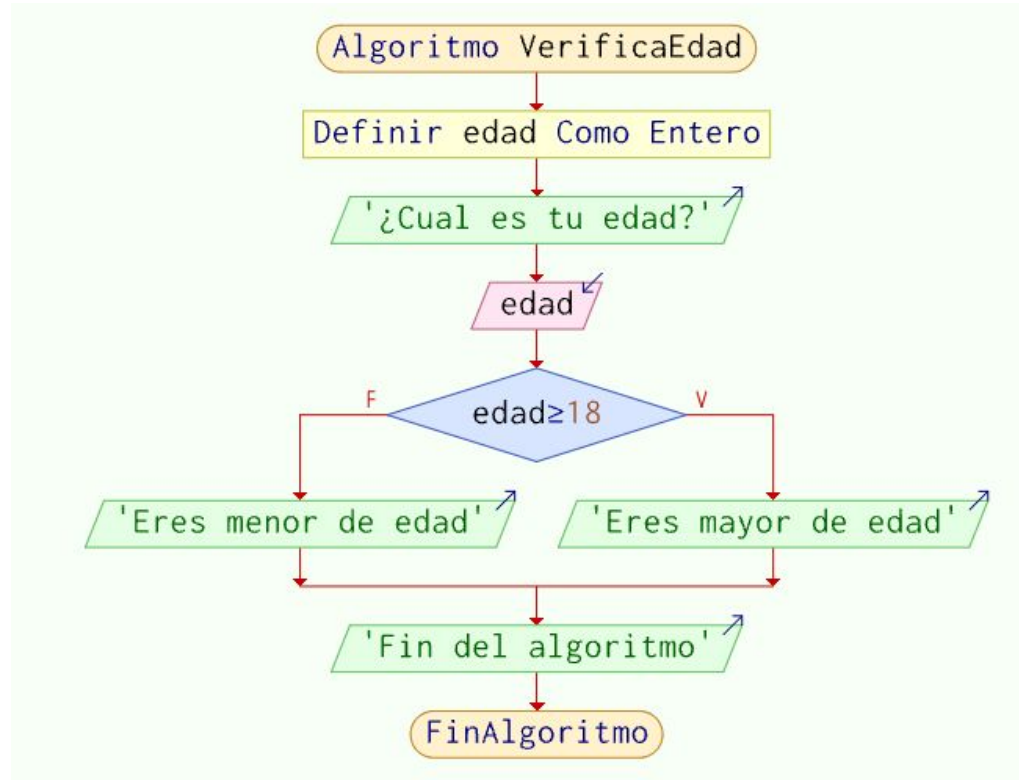




# SCRATCH








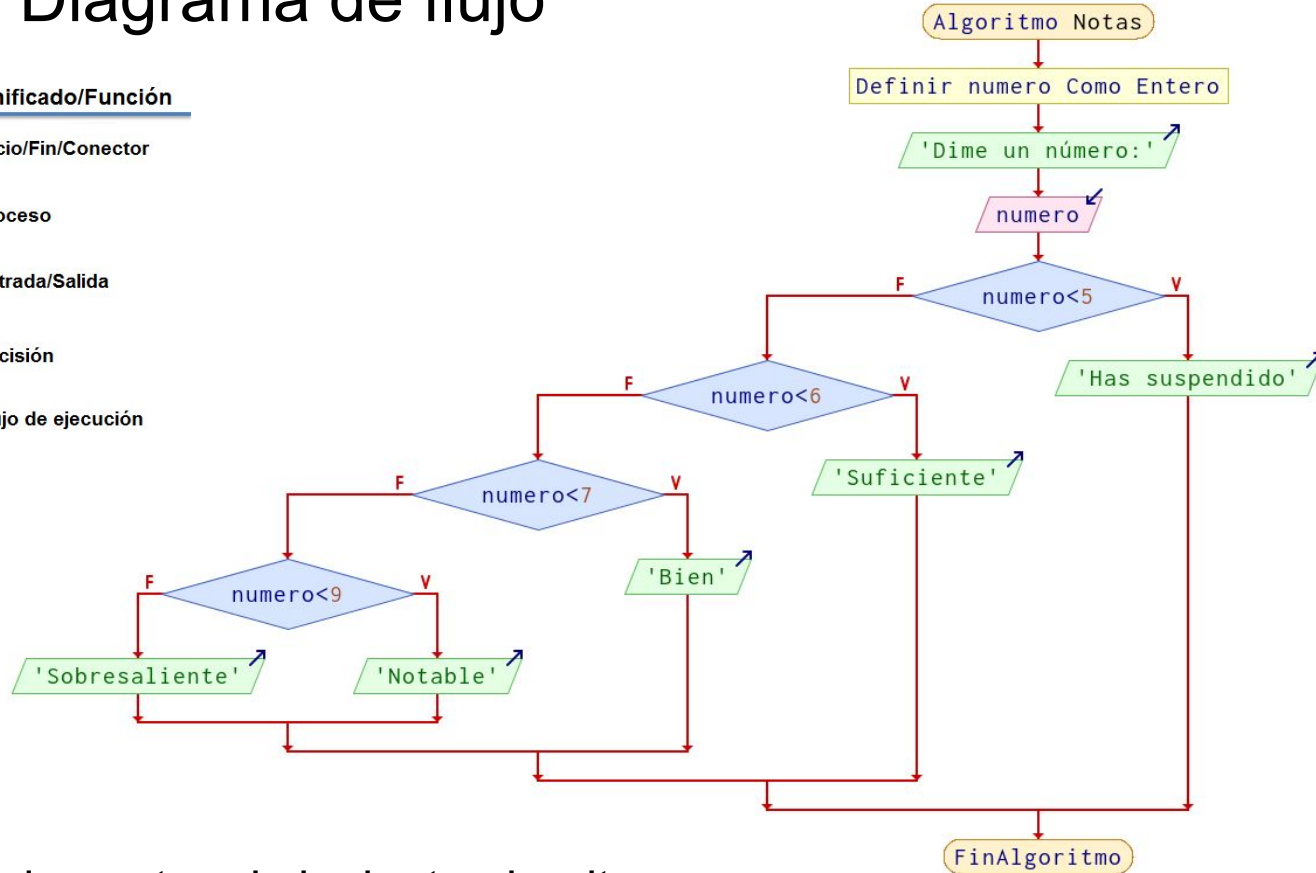
```
1 Algoritmo VerificaEdad
2   Definir edad Como Entero
3   Imprimir "¿Cual es tu edad?"
4   Leer edad
5   Si edad ≥ 18 Entonces
6       Imprimir "Eres mayor de edad"
7   SiNo
8       Imprimir "Eres menor de edad"
9   FinSi
10  Imprimir "Fin del algoritmo"
11 FinAlgoritmo
```





# Diagrama de flujo

Símbolo	Significado/Función
	Inicio/Fin/Conector
	Proceso
	Entrada/Salida
	Decisión
	Flujo de ejecución



Actividad: Implementar el siguiente algoritmo

# Programa

Un **programa** es una implementación de un **Algoritmo**.

Para realizar un **programa** necesito un ordenador.



y un lenguaje de programación.



# Fases producción de un Programa (objeto Software)

1. **Análisis** 

2. **Diseño** 





3. **Programación** 

4. **Pruebas** 

# Aprender lenguaje de programación

La principal razón para que las personas aprendan lenguajes de programación es utilizar la computadora como una herramienta de resolución de problemas.

La resolución de un problema exige al menos los siguientes pasos:

1. Definición y **análisis** del problema 
2. **Diseño** del algoritmo o método para resolverlo 
3. *Transformación* del algoritmo en un programa 
4. Ejecución y validación del programa 

# Características de los lenguajes de programación

- *Traducción y ejecución*: todos los lenguajes de programación que no son lenguaje máquina ha de traducirse al lenguaje de la máquina. Al programa escrito en un lenguaje de programación se le denomina código o programa fuente.
  - Dos formas puras de traducción del lenguaje de programación a código máquina.
    - **Compilación**: todo el código fuente se traduce a código objeto.  
Ejemplo C++
    - **Interpretación**: consiste en traducir y ejecutar una a una cada instrucción del programa fuente.  
Ejemplo Python

# Características de los lenguajes de programación

- Sintaxis: son reglas de formación de todas las estructuras de un programa, desde cómo construir el esqueleto de un programa entero a cómo está formada una instrucción.
- Semántica: la semántica asigna un significado a cada tipo de construcción de un lenguaje de programación

## Semántica vs Léxico:

Este ejemplo muestra cómo la semántica depende del tipo de datos con los que estás trabajando, y cómo operaciones idénticas a nivel sintáctico y léxico pueden tener diferentes significados semánticos dependiendo del contexto.

### # Fragmento 1

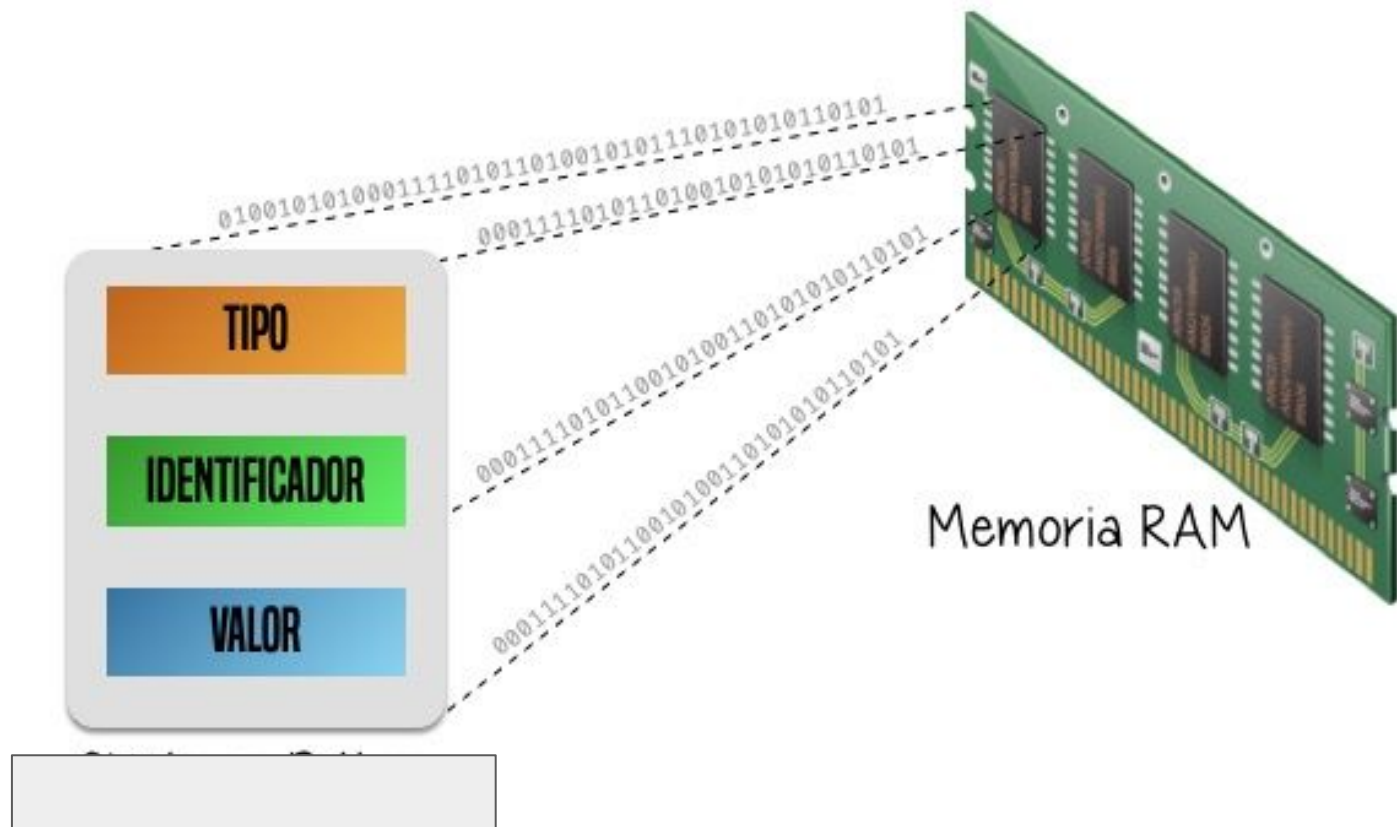
```
a = 5
b = 10
c = a + b
print(c) # Imprime 15
```

### # Fragmento 2

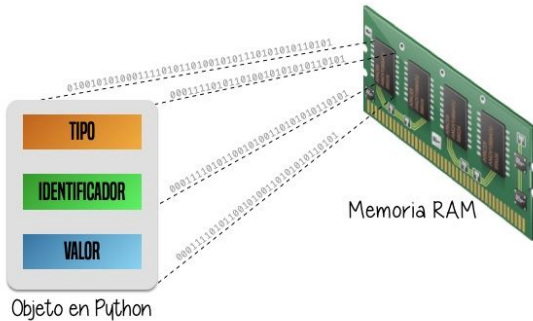
```
a = "5"
b = "10"
c = a + b
print(c) # Imprime "510"
```



# Variables



# Variables

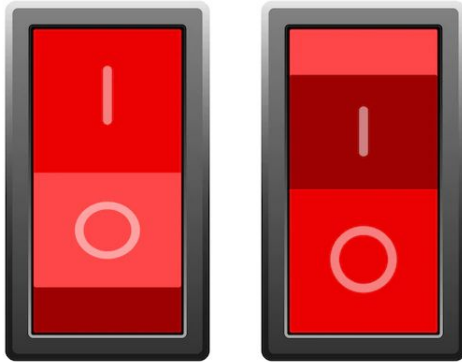


Identificador	Tipo de dato	Valor
nombre	Cadena de caracteres	“Hola Mundo”
edad	Número	13
confirmacion	Lógico	True

# Representación

*Lógica*

- 1 → Cierto
- 0 → Falso



Ejemplo

en Scratch: Un objeto con dos disfraces y una variable.

## Tipo de dato

# Representación

## Números

- Usando 1 Byte de información.
- 1 Byte = 8 bits
- Codificación de números enteros

¿Cómo se representa 4?

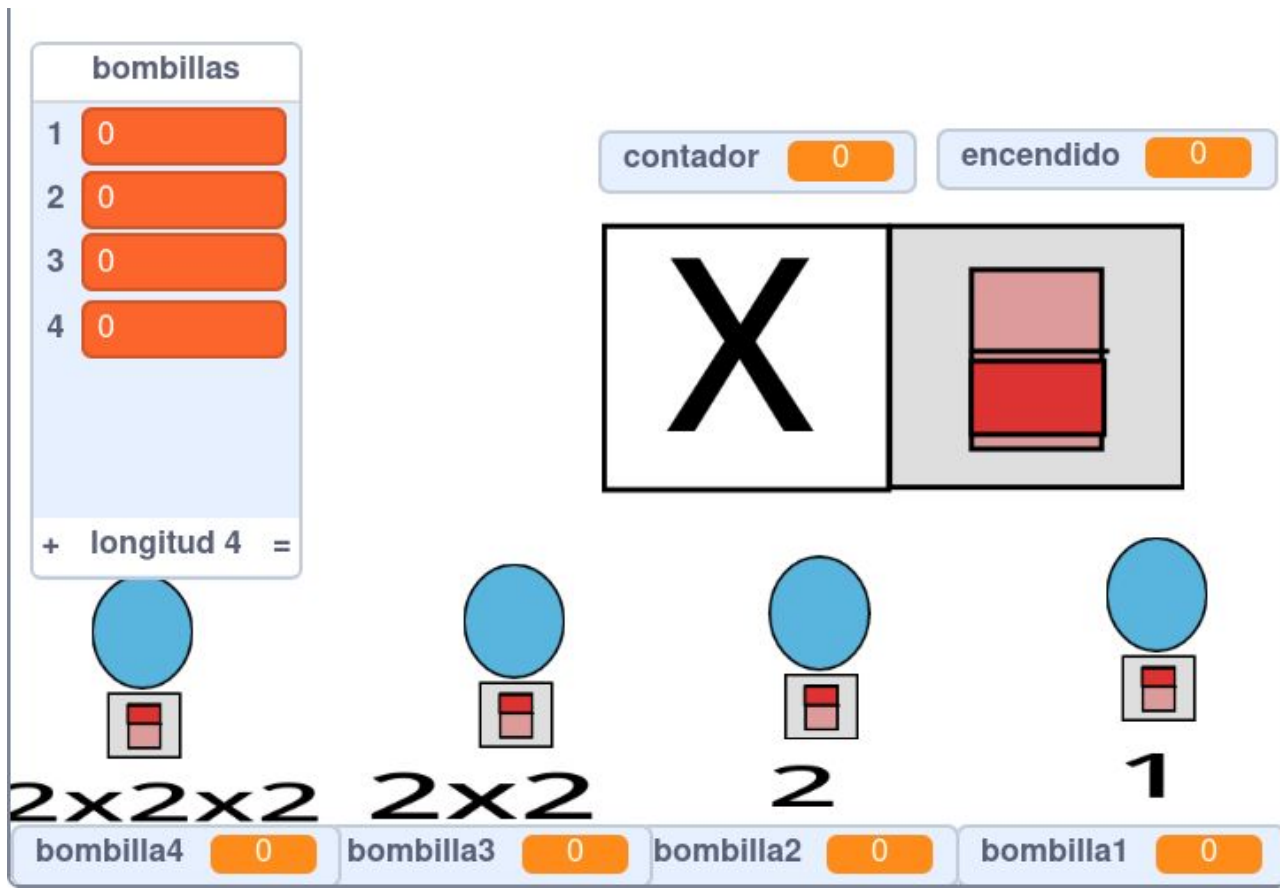
¿Cómo se representa -4?

1000 0101	-5
1000 0011	-3
1000 0010	-2
1000 0001	-1
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3

## Tipo de dato

Representación ¿Cuántos números podemos representar con 4 bombillas?

*Números*



## Tipo de dato

# Representación de palabras codificando con ASCII

¿Cómo representarías en binario la palabra “**Hola**” usando la codificación ASCII?

Letra	Código Binario	Letra	Código Binario
A	01000001	Ñ	01101001
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	Q	01010001
E	01000101	R	01010010
F	01000110	S	01010011
G	01000111	T	01010100
H	01001000	U	01010101
I	01001001	V	01010110
J	01001010	W	01010111
K	01001011	X	01011000
L	01001100	Y	01011001
M	01001101	Z	01011010
N	01001110	(espacio)	00100000

Un grupo de 8 bits es un *byte*

Un bit es la unidad de datos más pequeña en una computadora. Es como un átomo. Un bit puede ser un 1 o un 0.

Nombre	Tamaño
Byte (B)	8 bits
Kilobyte (KB)	1024 bytes
Megabyte (MB)	1024 kilobytes
Gigabyte (GB)	1024 megabytes
Terabyte (TB)	1024 gigabytes

# Representación de palabras codificando con ASCII

¿Cómo representarías en binario la palabra “**Hola**” usando la codificación ASCII?

Letra	Código Binario	Letra	Código Binario
A	01000001	Ñ	01101001
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	Q	01010001
E	01000101	R	01010010
F	01000110	S	01010011
G	01000111	T	01010100
H	01001000	U	01010101
I	01001001	V	01010110
J	01001010	W	01010111
K	01001011	X	01011000
L	01001100	Y	01011001
M	01001101	Z	01011010
N	01001110	(espacio)	00100000



# Representación de palabras codificando con ASCII

¿Cómo representarías en binario la palabra “**HOLA**” usando la codificación ASCII?

H → 01001000

O → 01001111

L → 01001100

A → 01000001

Letra	Código Binario	Letra	Código Binario
A	01000001	Ñ	01101001
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	Q	01010001
E	01000101	R	01010010
F	01000110	S	01010011
G	01000111	T	01010100
H	01001000	U	01010101
I	01001001	V	01010110
J	01001010	W	01010111
K	01001011	X	01011000
L	01001100	Y	01011001
M	01001101	Z	01011010
N	01001110	(espacio)	00100000

# Representación de palabras codificando con ASCII

¿Cuántos **bits** tiene la palabra  
codificada “Hola”?

Letra	Código Binario	Letra	Código Binario
A	01000001	Ñ	01101001
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	Q	01010001
E	01000101	R	01010010
F	01000110	S	01010011
G	01000111	T	01010100
H	01001000	U	01010101
I	01001001	V	01010110
J	01001010	W	01010111
K	01001011	X	01011000
L	01001100	Y	01011001
M	01001101	Z	01011010
N	01001110	(espacio)	00100000

# Representación de palabras codificando con ASCII

¿Cuántos **bits** tiene la palabra  
codificada “Hola”?

*4 caracteres,  
cada carácter 8 bits.*

*En total  $8 \times 4 = 32$  bits*

Letra	Código Binario	Letra	Código Binario
A	01000001	Ñ	01101001
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	Q	01010001
E	01000101	R	01010010
F	01000110	S	01010011
G	01000111	T	01010100
H	01001000	U	01010101
I	01001001	V	01010110
J	01001010	W	01010111
K	01001011	X	01011000
L	01001100	Y	01011001
M	01001101	Z	01011010
N	01001110	(espacio)	00100000

# Representación de palabras codificando con ASCII

¿Cuántos **byte** tiene la palabra codificada “Hola”? 4 bytes.

*Como tiene  
8 x 4 = 32 bits y  
1 Byte son 8 bits.  
4 bytes.*

Letra	Código Binario	Letra	Código Binario
A	01000001	Ñ	01101001
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	Q	01010001
E	01000101	R	01010010
F	01000110	S	01010011
G	01000111	T	01010100
H	01001000	U	01010101
I	01001001	V	01010110
J	01001010	W	01010111
K	01001011	X	01011000
L	01001100	Y	01011001
M	01001101	Z	01011010
N	01001110	(espacio)	00100000

# Algoritmo <Implementación>

El lenguaje algorítmico es aquel por medio del cual se realiza un análisis previo del problema a resolver y además permite encontrar un método para resolverlo.

Al conjunto de todas las operaciones a realizar así como el orden en el que deben efectuar, se le denomina algoritmo.

- **Preciso:** Un algoritmo debe describir claramente el orden de realización de cada uno de sus pasos.
- **Definido:** Un algoritmo es definido cuando al ejecutarse con datos iguales en distintas ocasiones, produce resultados iguales en todas las ocasiones.
- **Finito:** Los algoritmos deben estar constituidos por un número finito de pasos.
- **Independiente del Lenguaje de Programación:** Los algoritmos deben ser diseñados de forma independiente al lenguaje de programación en el que se implemente. Es decir, su implementación debe ser posible en cualquier lenguaje de programación.

# PROGRAMA






## Lenguaje informático



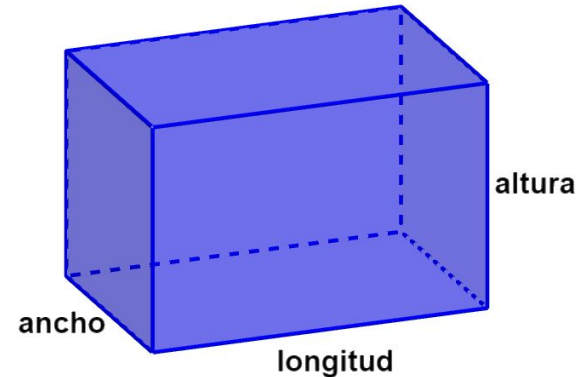
El lenguaje informático es aquel por medio del cual dicho algoritmo se codifica un sistema comprensible por el ordenador o computador.

Se denomina lenguaje de alto nivel aquel que es más cercano a la comprensión humana y lenguaje de bajo nivel a aquellos que son más comprensibles por la máquina



Símbolo	Significado/Función
	Inicio/Fin/Conector
	Proceso
	Entrada/Salida
	Decisión
	Flujo de ejecución

**Ejercicio:** Crea un diagrama de flujo que represente el proceso de cálculo del volumen de un prisma rectangular.



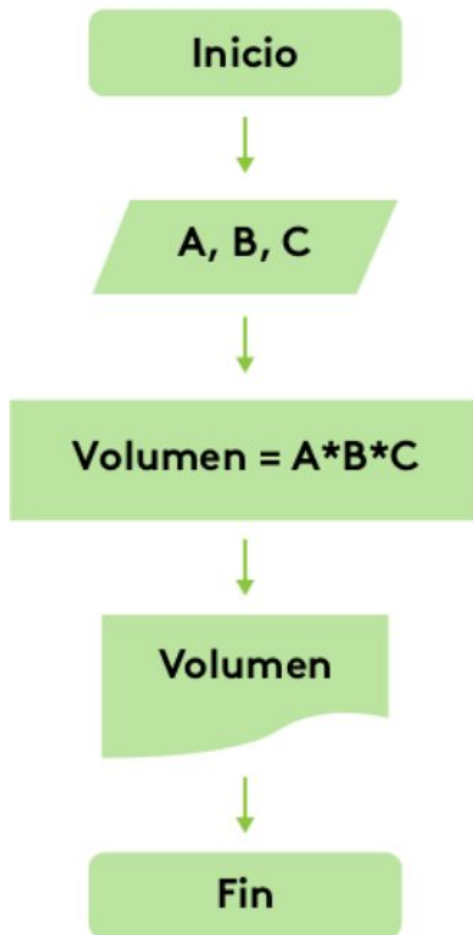
Inicio

Leer las medidas A, B  
y C.

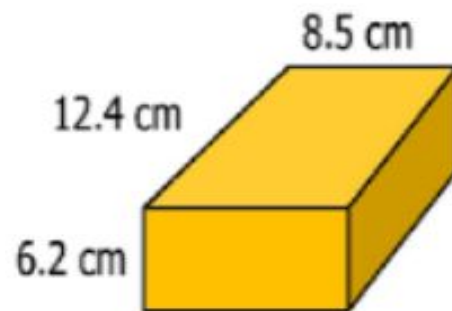
Calcular el volumen.

Escribir el volumen  
obtenido.

Fin.



# Algoritmo



Volumen = ?



# **Fases del desarrollo de una aplicación**

1. **Análisis:** Determinar qué debe hacer la aplicación.
2. **Diseño:** Planificar cómo se va a desarrollar.
3. **Codificación:** Escribir el programa.
4. **Pruebas:** Asegurarse de que funcione correctamente.
5. **Documentación:** Registrar cómo usarlo y cómo fue creado.
6. **Explotación:** Implementar para usuarios finales.
7. **Mantenimiento:** Hacer mejoras y correcciones a lo largo del tiempo.



## Problema

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

## Análisis (del Problema)

Para resolver este problema, se necesita desarrollar un programa que realice los siguientes pasos:

1. **Entrada de Datos:** El programa debe solicitar al usuario que introduzca dos valores: la base y la altura de un rectángulo.
2. **Validación de Datos:** Debería haber una validación para asegurar que los valores introducidos son números positivos.
3. **Cálculo del Área:** Una vez que se han recibido y validado estos dos valores, el programa debe calcular el área del rectángulo utilizando la fórmula:  
$$\text{Área} = \text{base} \times \text{altura}$$
4. **Salida de Datos:** Finalmente, el programa debe mostrar el resultado del cálculo al usuario.

# Diseño (del Problema)

## 1. Especificación de Entradas

- Dos números reales positivos: base y altura del rectángulo.

## 2. Especificación de Salidas

- Un número real positivo: el área del rectángulo.

## 3. Especificación de Proceso

- Solicitar al usuario la base del rectángulo.
- Validar que la base sea un número real positivo, si no, solicitar de nuevo.
- Solicitar al usuario la altura del rectángulo.
- Validar que la altura sea un número real positivo, si no, solicitar de nuevo.
- Calcular el área del rectángulo multiplicando base por altura.
- Mostrar el resultado al usuario.

## 4. Diseño de la Solución

Inicio

Escribir "Introduzca la base del rectángulo:"

Leer base

Mientras base  $\leq 0$

Escribir "Por favor, introduzca un número positivo para la base:"

Leer base

Escribir "Introduzca la altura del rectángulo:"

Leer altura

Mientras altura  $\leq 0$

Escribir "Por favor, introduzca un número positivo para la altura:"

Leer altura

$\text{área} \leftarrow \text{base} \times \text{altura}$

Escribir "El área del rectángulo es:", área  
Fin

# Codificación CON:

## C

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

[enlace a código fuente](#)

**C** calculoarea.c

```
1  #include <stdio.h>
2  int main() {
3      double base, altura, area;
4
5      printf("Introduzca la base del rectángulo:\n");
6      scanf("%lf", &base);
7      while(base <= 0) {
8          printf("Por favor, introduzca un número positivo para la base:\n");
9          scanf("%lf", &base);
10     }
11     printf("Introduzca la altura del rectángulo:\n");
12     scanf("%lf", &altura);
13     while(altura <= 0) {
14         printf("Por favor, introduzca un número positivo para la altura:\n");
15         scanf("%lf", &altura);
16     }
17     area = base * altura;
18     printf("El área del rectángulo es: %lf\n", area);
19     return 0;
20 }
```

**Código objeto:**

gcc -c calculoarea.c

**Programa:**

gcc calculoarea.c -o programa\_area

# Codificación CON: Python

[enlace a código fuente](#)

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

calculoarea.py

```
1 print("Introduzca la base del rectángulo:")
2 base = float(input())
3 while base <= 0:
4     print("Por favor, introduzca un número positivo para la base:")
5     base = float(input())
6
7 print("Introduzca la altura del rectángulo:")
8 altura = float(input())
9 while altura <= 0:
10     print("Por favor, introduzca un número positivo para la altura:")
11     altura = float(input())
12
13 area = base * altura
14 print("El área del rectángulo es:", area)
```

# Codificación CON: Java

[enlace a  
código fuente](#)

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

```
AreaRectangulo.java X
1 import java.util.Scanner;
2
3 public class AreaRectangulo {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         double base, altura, area;
7
8         System.out.println("Introduzca la base del rectángulo:");
9         base = scanner.nextDouble();
10        while(base <= 0) {
11            System.out.println("Por favor, introduzca un número positivo para la base:");
12            base = scanner.nextDouble();
13        }
14
15        System.out.println("Introduzca la altura del rectángulo:");
16        altura = scanner.nextDouble();
17        while(altura <= 0) {
18            System.out.println("Por favor, introduzca un número positivo para la altura:");
19            altura = scanner.nextDouble();
20        }
21
22        area = base * altura;
23        System.out.println("El área del rectángulo es: " + area);
24    }
25 }
```

# Codificación CON: Javascript

[enlace a  
código fuente](#)

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

```
<script>
  function calcularArea() {
    let base = document.getElementById('base').value;
    let altura = document.getElementById('altura').value;

    base = parseFloat(base);
    altura = parseFloat(altura);

    if (base <= 0 || altura <= 0) {
      document.getElementById('resultado').textContent = 'Por favor, introduzca números positivos para la base y la altura';
      return;
    }

    const area = base * altura;
    document.getElementById('resultado').textContent = 'El área del rectángulo es: ' + area;
  }
</script>
```



# Codificación CON:

## PHP

[enlace a  
código fuente](#)

Desarrollar un programa que solicite al usuario las dimensiones de un rectángulo (base y altura) y calcule su área.

calculoarea.php

```
1  <?php
2  $base = $altura = $area = $error = null;
3
4  if ($_SERVER["REQUEST_METHOD"] == "POST") {
5      $base = isset($_POST['base']) ? (float)$_POST['base'] : null;
6      $altura = isset($_POST['altura']) ? (float)$_POST['altura'] : null;
7
8      if ($base <= 0 || $altura <= 0) {
9          $error = "Por favor, introduzca números positivos para la base y la altura.";
10     } else {
11         $area = $base * $altura;
12     }
13 }
14 ?>
15
16 <!DOCTYPE html>
17 <html lang="es">
18 <head>
19     <meta charset="UTF-8">
20     <title>Calculadora de Área de Rectángulo</title>
21 </head>
22 <body>
23     <h2>Calculadora de área de rectángulo</h2>
24
25     <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
26         <label for="base">Base:</label>
```