

ALGORITMIA

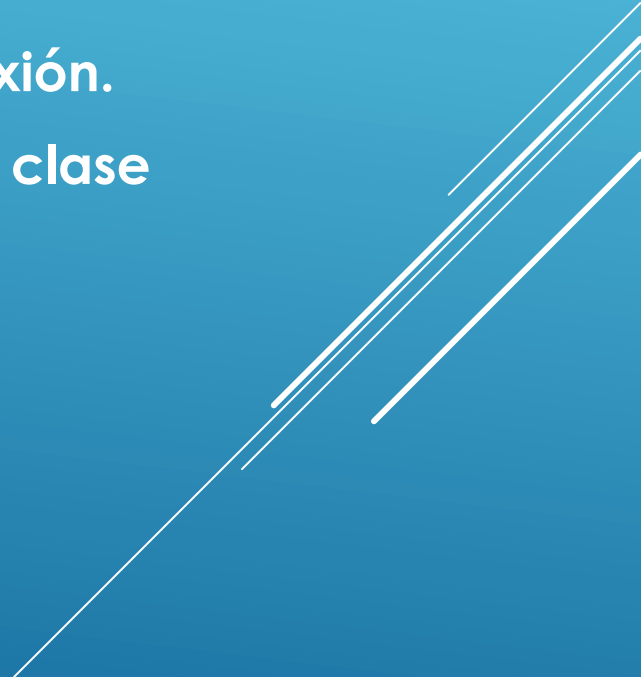
PSEUDOCÓDIGO Y DIAGRAMAS
DE FLUJO

Juan Carlos Molina Lozano
Docente

OBJETIVOS DE LA CLASE

1. Comprender que es Algoritmo.
2. Explicar el Enfoque Sistémico en la solución de problemas computacionales.
3. Definir y analizar entradas y salidas en un problema algorítmico.
4. Describir las partes, instrucciones, elementos y datos de un programa.
5. Comprender qué es el pseudocódigo y para qué sirve.
6. Reconocer características, ventajas y desventajas del pseudocódigo.
7. Conocer el diagrama de flujo y su importancia.
8. Identificar diferencias entre pseudocódigo y diagramas de flujo.
9. Desarrollar ejercicios prácticos para reforzar la comprensión.

METODOLOGÍA

1. Exposición con ejemplos prácticos durante la sección.
 2. Aprendizaje Activo: actividad en grupos con ejercicios.
 3. Discusión Guiada: preguntas estratégicas para fomentar la reflexión.
 4. Componente Práctico: resolución de problemas algorítmicos en clase
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

CONTENIDO

1. Definición de Algoritmo.
 2. Enfoque Sistémico.
 3. Entradas y Salidas de un Problema Algorítmico.
 4. Partes, Instrucciones, Elementos y Datos de un Programa.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

DEFINICIÓN DE ALGORITMO?

Es una secuencia de pasos con los cuales se puede dar solución a un problema determinado

1. Escribir la dirección de correo de destino
2. Escribir el asunto
3. Si el correo tiene un archivo, adjuntarlo
4. Escribir el mensaje
5. Darle clic en Enviar

1. Pon una sartén en la cocina a fuego medio
2. Añade un poco de aceite o mantequilla a la sartén mientras se calienta
3. Deja que la sartén se caliente hasta que la mantequilla se derrita
4. Rompe un huevo y viértalo en el sartén
5. Deja cocinar el huevo sin tocarlo por 3 minutos
6. Tapa la sartén
7. Cuando pasen los 3 minutos retira el sartén del fuego y listo!

1. Ver la cartelera
2. Seleccionar una película
3. Comprar las entradas
4. Entregar ticket en la entrada
5. Comprar palomitas y refresco
6. Dirigirse a la sala
7. Dirigirse a la silla indicada
8. Esperar que empiece la película
9. Mirar la película

ENTRADAS Y SALIDAS DE UN PROBLEMA ALGORÍTMICO

En un algoritmo, la entrada son los datos que recibe, y la salida es el resultado obtenido tras el procesamiento.

- Tipos de entrada:
 - Directa (usuario introduce datos manualmente).
 - Indirecta (datos preexistentes en archivos o bases de datos).
- Tipos de salida:
 - Visual (pantalla), impresa, sonora, almacenamiento en archivos.
- Ejemplo práctico: Algoritmo para calcular el área de un triángulo:
 - Entrada: Base y altura.
 - Proceso: Aplicar la fórmula $(\text{base} \times \text{altura}) / 2$.
 - Salida: Resultado en pantalla.

PARTES, INSTRUCCIONES, ELEMENTOS Y DATOS DE UN PROGRAMA.

Partes de un programa:

- Declaración de variables: Almacenan datos.
- Estructuras de control: Condiciones y bucles.
- Operaciones y funciones: Cálculos y reutilización de código.
- Entrada/salida: Comunicación con el usuario.

Tipos de instrucciones en un programa:

- Asignación: `x = 10`
- Entrada: `input("Ingrese un número: ")`
- Salida: `print("Hola mundo")`
- Condicionales: `if, else`
- Bucles: `for, while`

PARTES, INSTRUCCIONES, ELEMENTOS Y DATOS DE UN PROGRAMA.

Ejemplo práctico: Programa en Python que pide un número y dice si es par o impar.

```
Scanner scanner = new Scanner(System.in);  
System.out.print("Ingrese un número: ");  
int num = scanner.nextInt();  
if (num % 2 == 0) {  
    System.out.println("El número es par");  
} else {  
    System.out.println("El número es impar");  
}
```

ENFOQUE SISTÉMICO

Un sistema es un conjunto de elementos interrelacionados que trabajan juntos para lograr un objetivo.

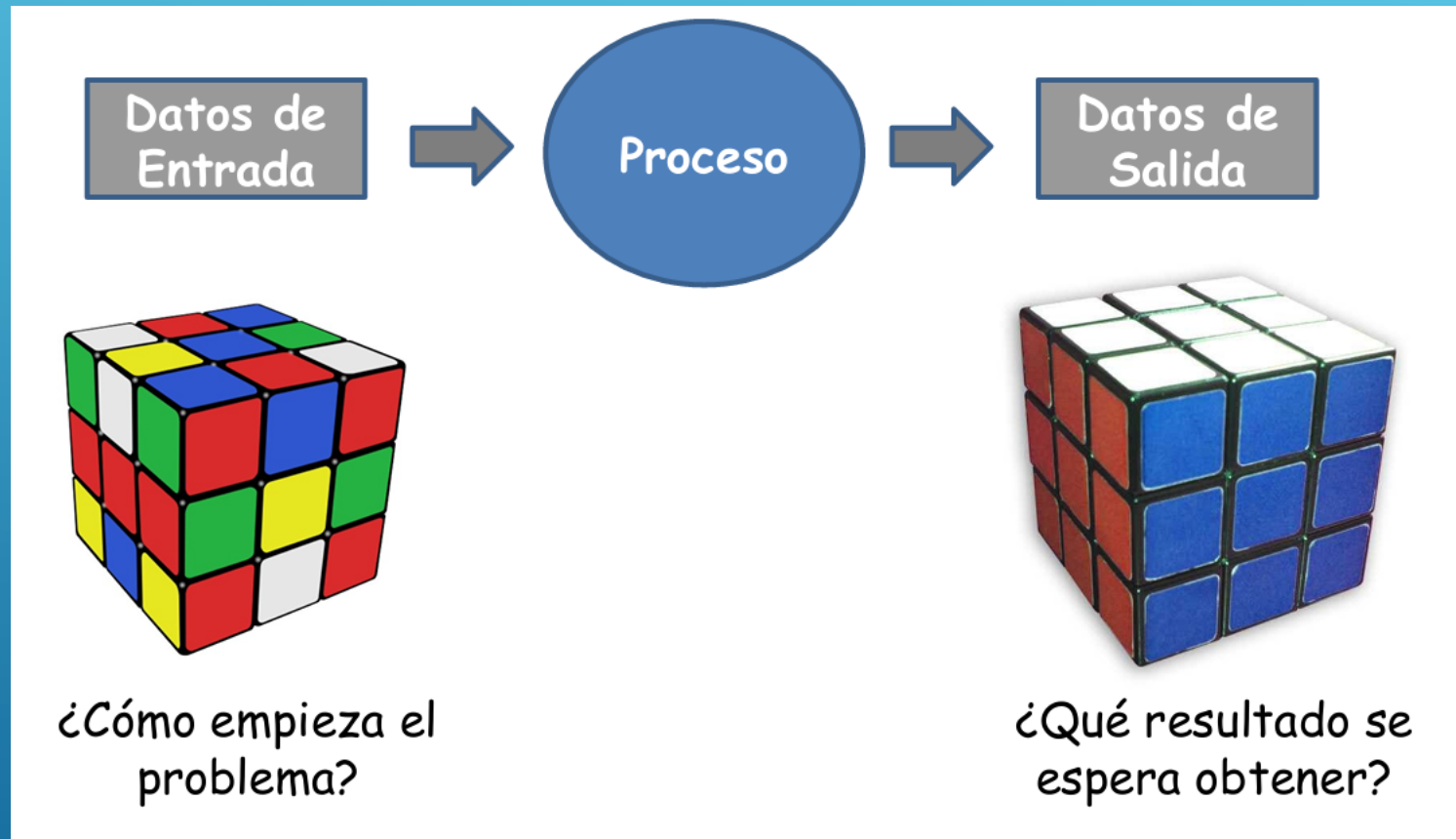
- Componentes clave: Entrada, procesamiento y salida.
- Relación con la computación: un programa informático es un sistema que toma datos de entrada, los procesa y genera una salida.

Ejemplo:

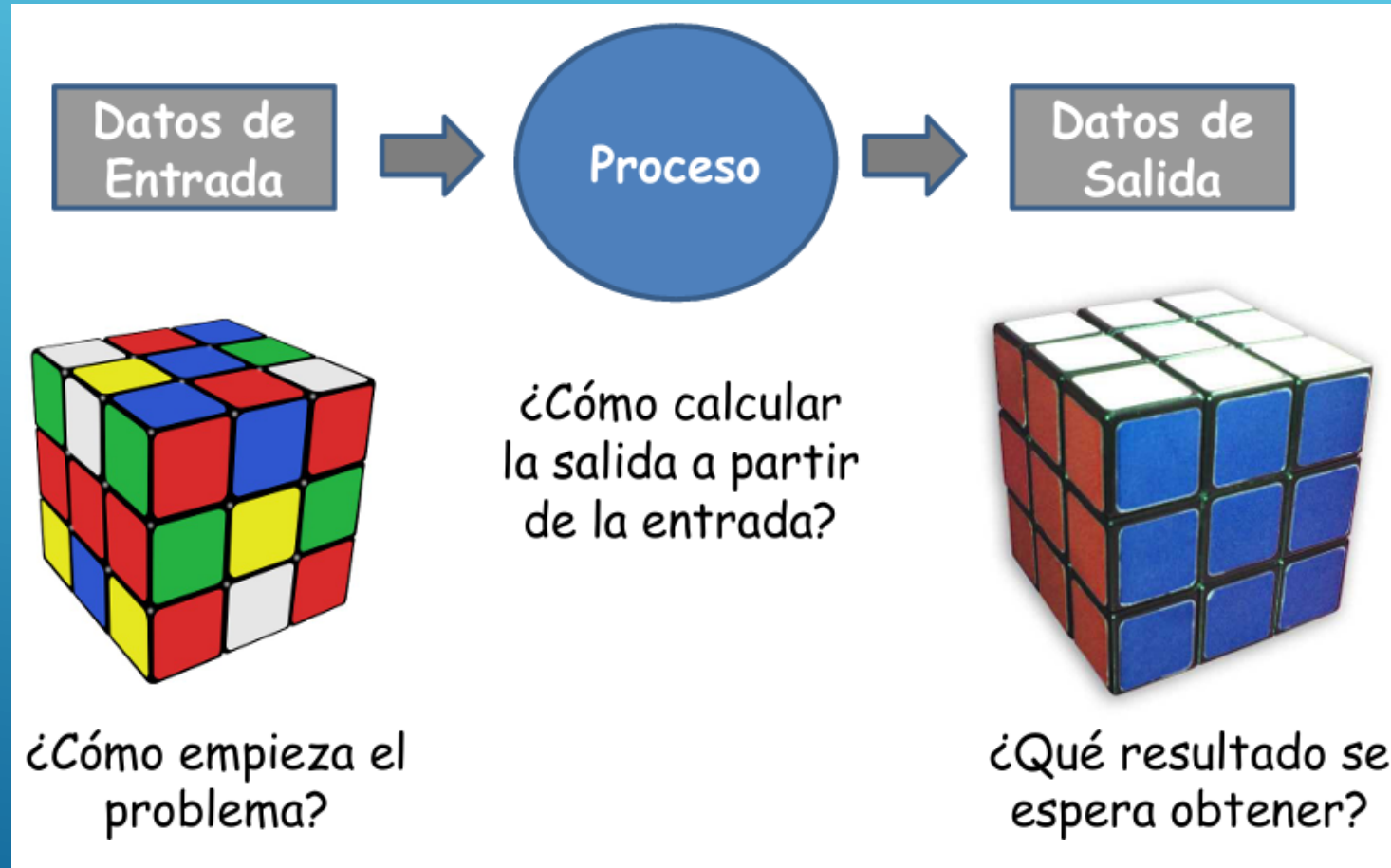
- Un cajero automático como sistema:
 - Entrada: Número de tarjeta, PIN.
 - Procesamiento: Validación de datos y consulta de saldo.
 - Salida: Dinero dispensado o mensaje de error.



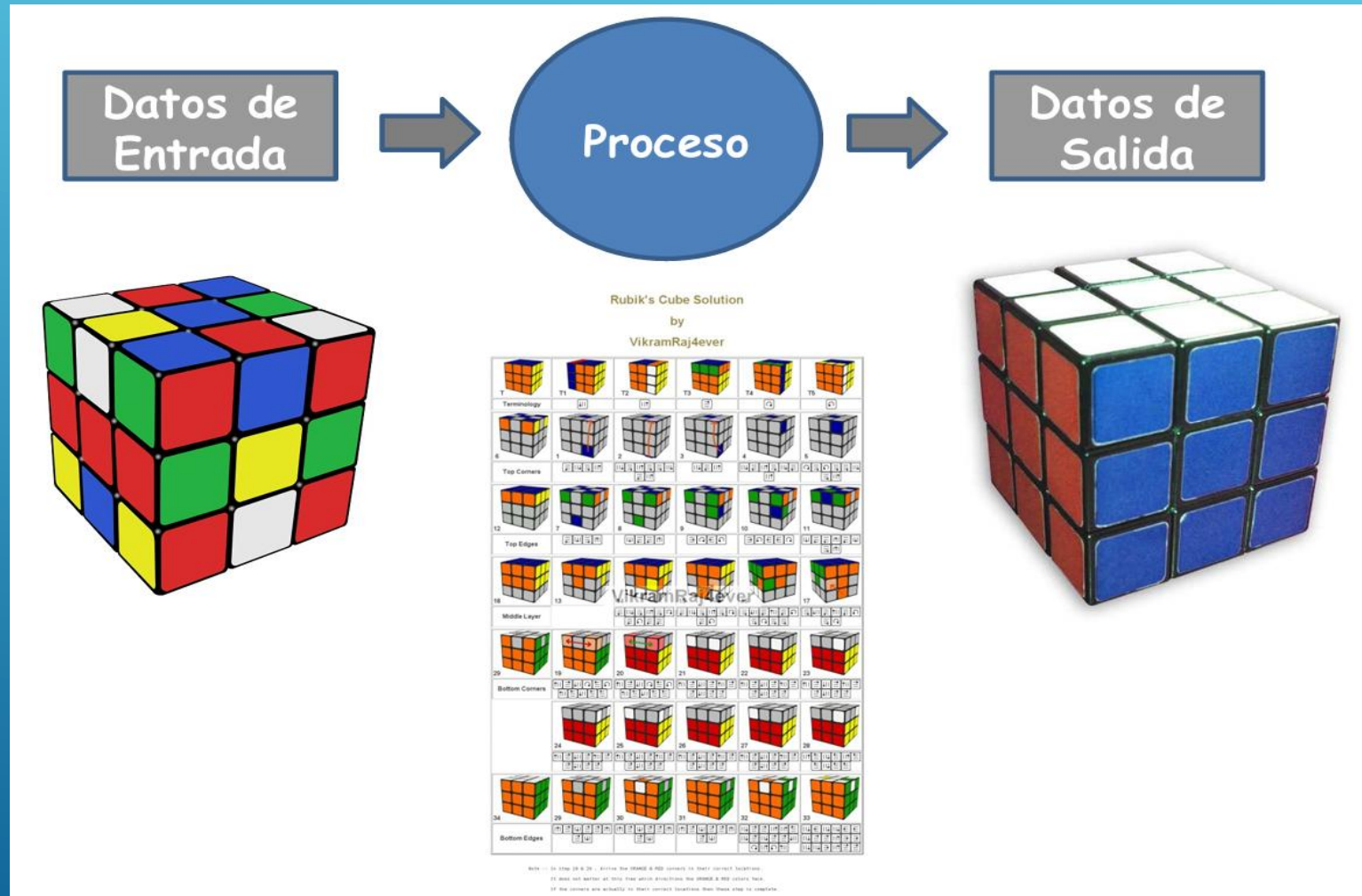
ENFOQUE SISTÉMICO



ENFOQUE SISTÉMICO



ENFOQUE SISTÉMICO



METODOLOGÍA PARA RESOLVER PROBLEMAS

1. Analizar el problema.
 2. Diseñar el algoritmo y escribirlo en pseudocódigo.
 3. Realizar una prueba de escritorio para el algoritmo.
 4. Codificar el algoritmo usando el Lenguaje de programación escogido (Python, Java, C#, Javascript, etc).
 5. Probar el algoritmo implementado.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

QUÉ ES EL PSEUDOCÓDIGO?

Forma de escribir algoritmos usando lenguaje natural estructurado. No depende de un lenguaje de programación específico.

Ejemplo:

Inicio

Leer número

Si número > 0 entonces

 Escribir 'Es positivo'

Sino

 Escribir 'Es negativo'

Fin



CARACTERÍSTICAS DEL PSEUDOCÓDIGO

1. Claridad en la descripción de los pasos.
 2. No depende de un lenguaje de programación específico.
 3. Fácil de entender para personas sin conocimientos técnicos.
 4. Permite estructurar el pensamiento lógico antes de programar.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

VENTAJAS Y DESVENTAJAS DEL PSEUDOCÓDIGO

Ventajas:

- Sencillo de aprender y escribir.
- Facilita la traducción a cualquier lenguaje de programación.
- Buena herramienta de enseñanza.

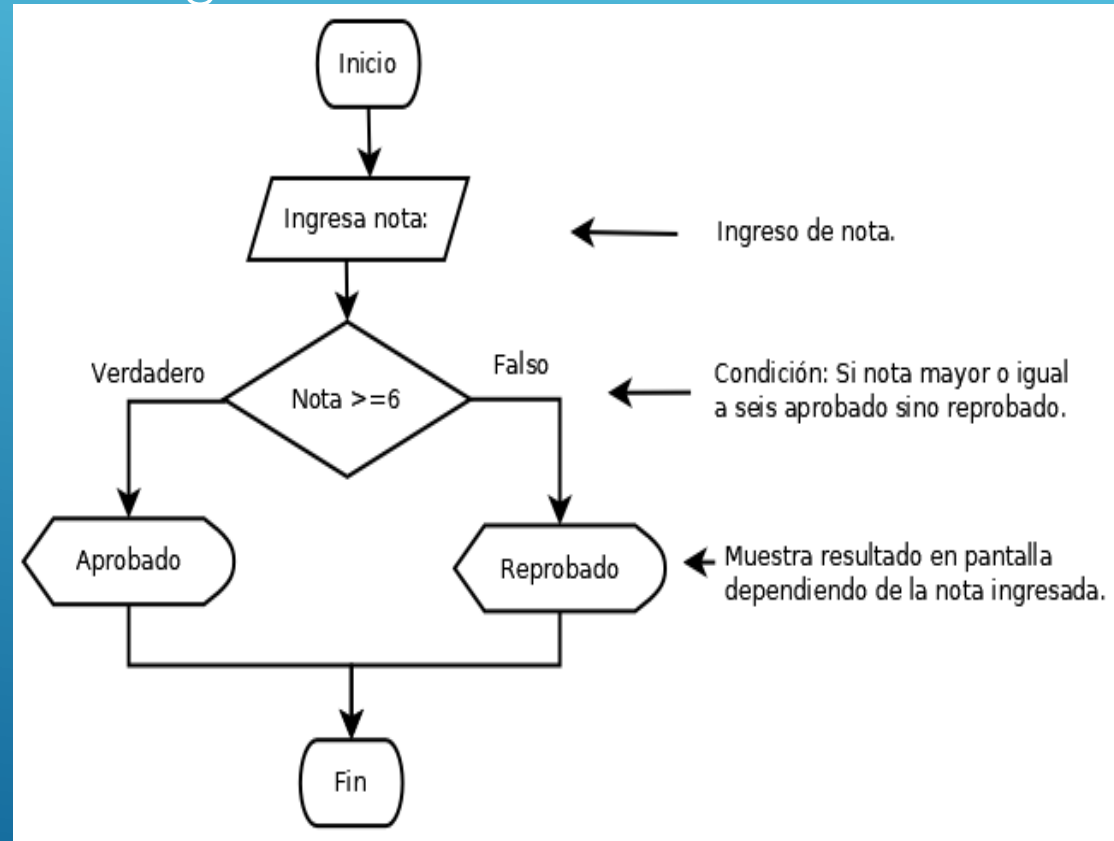
Desventajas:

- No existe un estándar único.
- Puede prestarse a ambigüedades.
- No se puede ejecutar directamente en un computador.

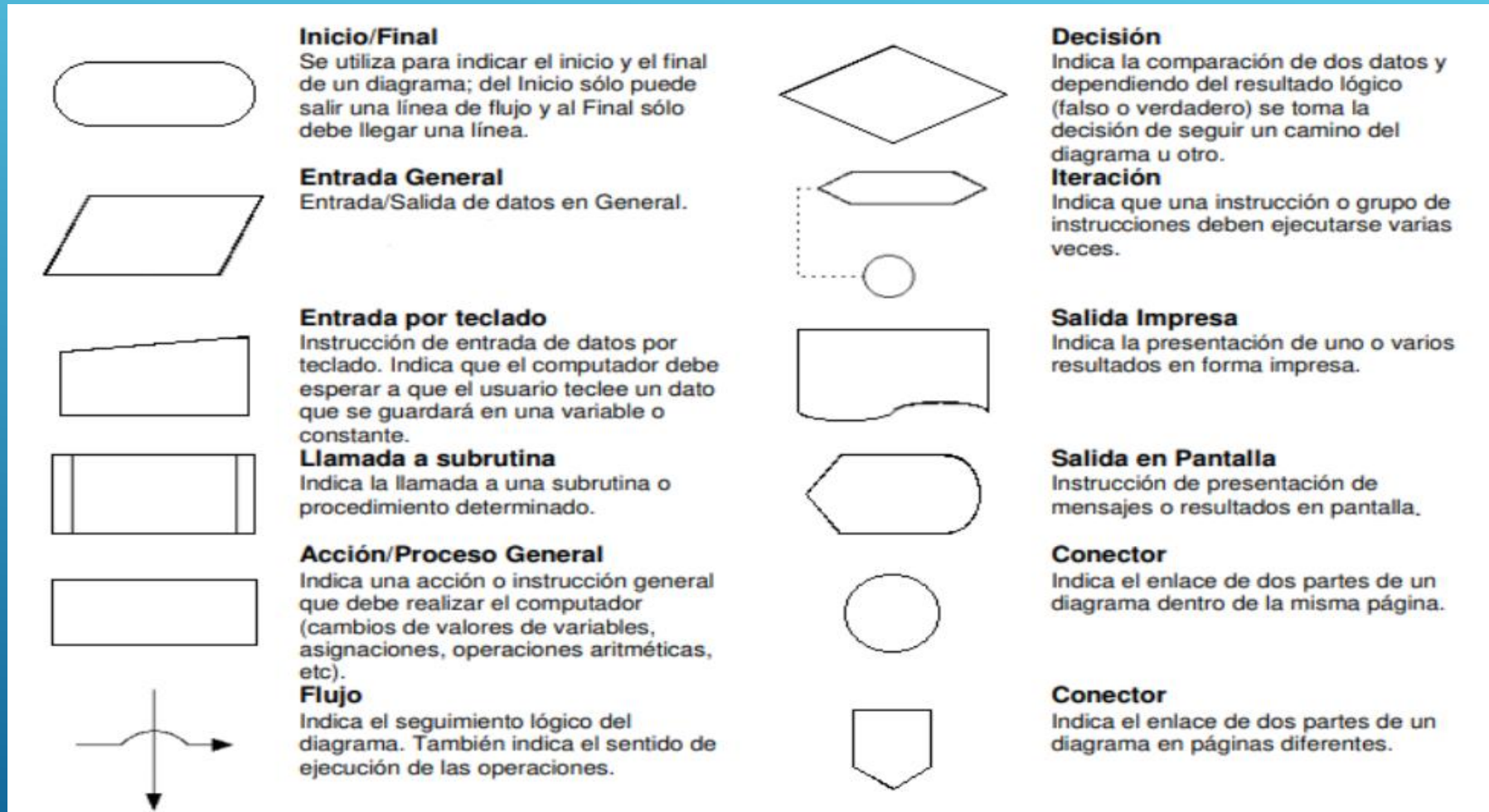
QUÉ ES UN DIAGRAMA DE FLUJO?

Representación gráfica de los pasos de un algoritmo. Se utilizan símbolos estandarizados:

- Óvalo → Inicio / Fin
- Rectángulo → Proceso
- Paralelogramo → Entrada / Salida
- Rombo → Decisión



SIMBOLOGÍA DE LOS DIAGRAMAS DE FLUJO



CARACTERÍSTICAS DEL DIAGRAMA DE FLUJO

1. Claridad visual y estructurada.
 2. Uso de símbolos universales.
 3. Ayuda a entender procesos complejos de forma simple.
 4. Útil para documentar y explicar algoritmos.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

VENTAJAS Y DESVENTAJAS DEL DIAGRAMA DE FLUJO

Ventajas:

- Comunicación visual clara y universal.
- Facilita la depuración de procesos.
- Útil para explicar a personas no técnicas.

Desventajas:

- Puede ser extenso para algoritmos grandes.
 - Difícil de modificar cuando cambia el algoritmo.
 - Requiere conocer la simbología.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

COMPARACIÓN PSEUDOCÓDIGO VS DIAGRAMA DE FLUJO

Pseudocódigo: Texto estructurado.

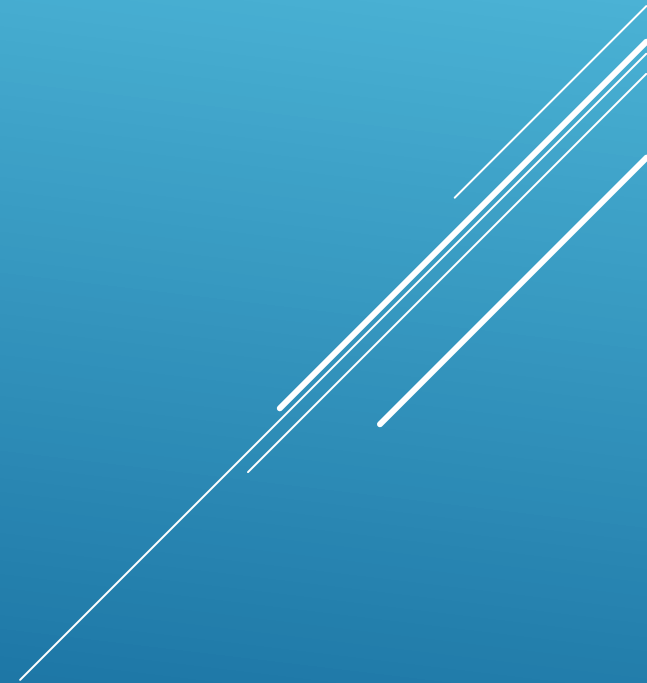
Diagrama de flujo: Gráfico con símbolos.

Pseudocódigo: Muy fácil de escribir.

Diagrama de flujo: Requiere práctica.

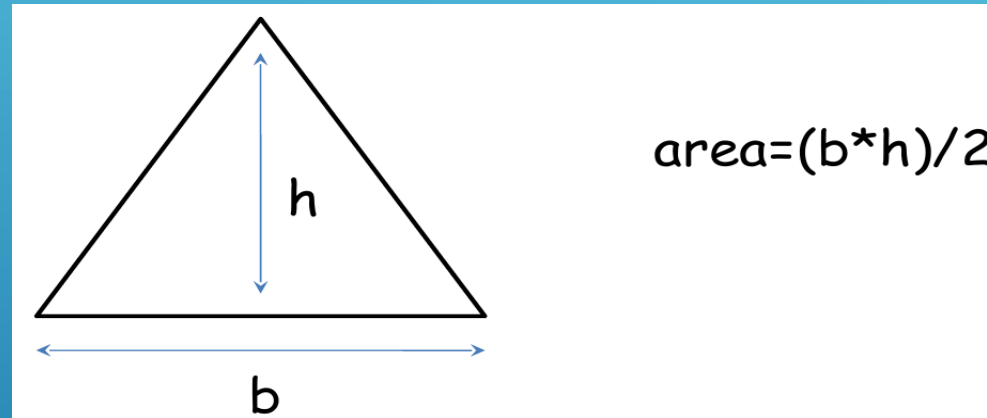
Pseudocódigo: No estándar universal.

Diagrama de flujo: Estándar universal.



METODOLOGÍA PARA RESOLVER PROBLEMAS

Problema: Desarrollar un programa que permita calcular el área de un triángulo dados su base y su altura.



Digite la base: 3

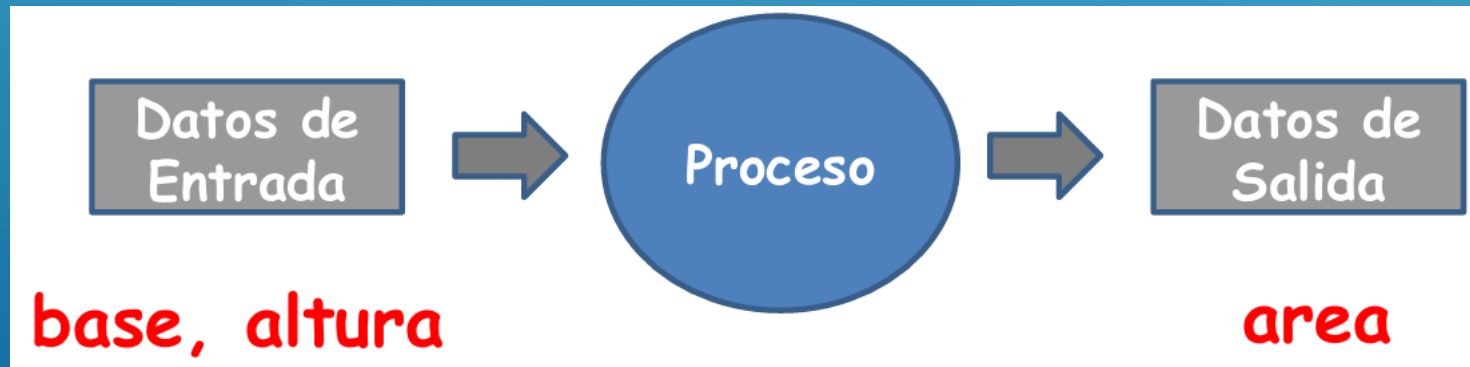
Digite la altura: 5



El area del triangulo es: 7.5

METODOLOGÍA PARA RESOLVER PROBLEMAS

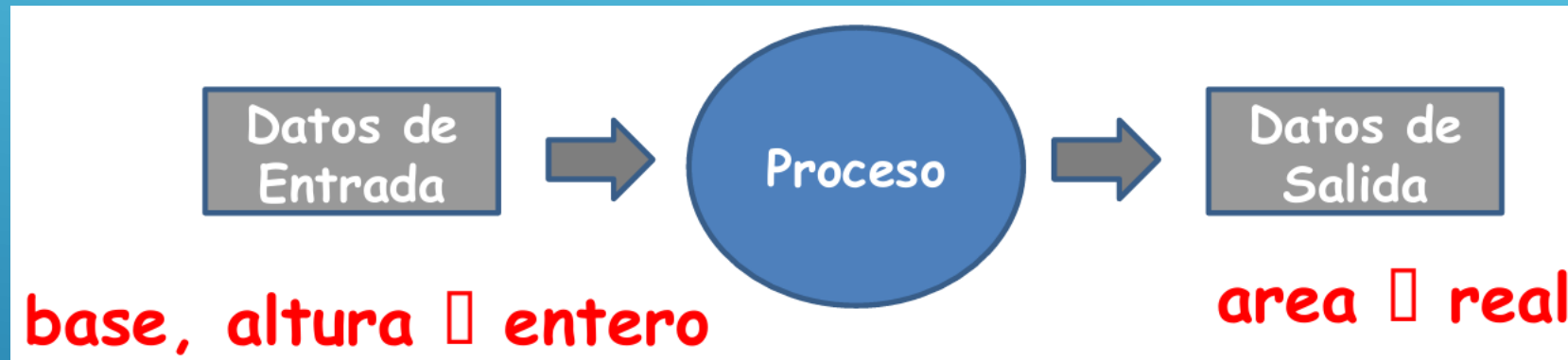
1. **Analizar el problema:** Consiste en identificar los datos de entrada y de salida.



- Las entradas y las salidas se conocen como variables
- Las variables pueden ser de 3 tipos: Número real, Número entero, Texto, etc.

METODOLOGÍA PARA RESOLVER PROBLEMAS

1. **Analizar el problema:** Consiste en identificar los datos de entrada y de salida.



Las entradas:

Base, Altura: Int

La Salida

Area: real

METODOLOGÍA PARA RESOLVER PROBLEMAS

2. Diseñar el algoritmo y escribirlo en pseudocódigo.

El pseudocódigo es una descripción detallada y legible de lo que debe hacer un programa. Está escrito en un estilo formal y legible que utiliza una sintaxis y un formato natural para que los desarrolladores y otras personas involucradas en el proceso de desarrollo puedan entenderlo fácilmente.

Pseudocódigo

Inicio

Fin

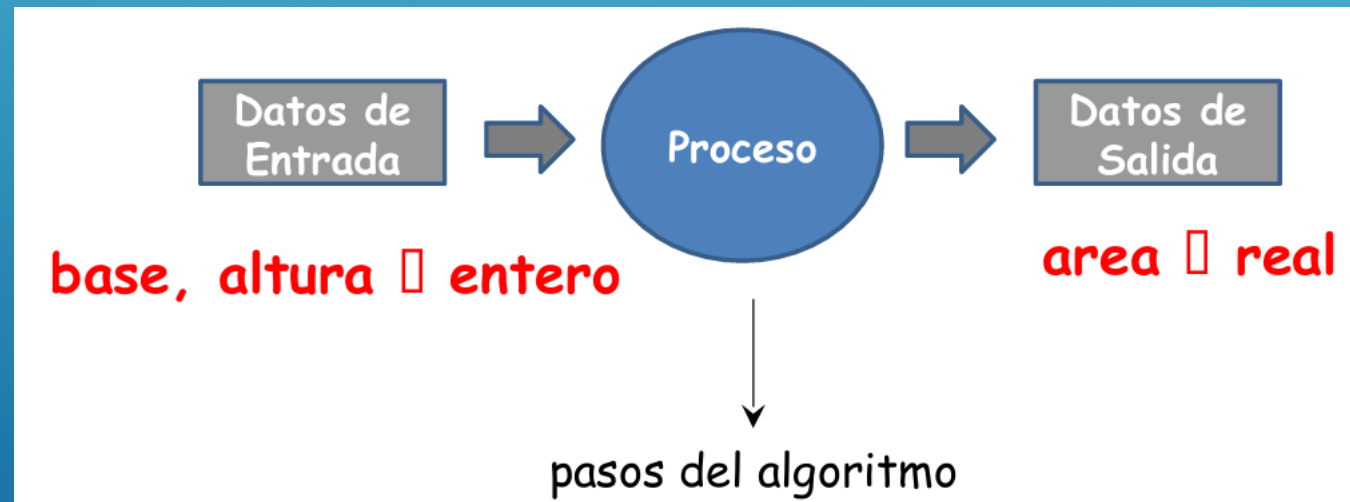
METODOLOGÍA PARA RESOLVER PROBLEMAS

2. Diseñar el algoritmo y escribirlo en pseudocódigo.

Inicio

base, altura: real

area: real



Fin

METODOLOGÍA PARA RESOLVER PROBLEMAS

2. Diseñar el algoritmo y escribirlo en pseudocódigo.

Inicio

base, altura: real
area: real

Pseudocódigo

Inicio

base, altura $\leftarrow$ entero

area $\leftarrow$ real

} pasos del algoritmo

Fin

Fin

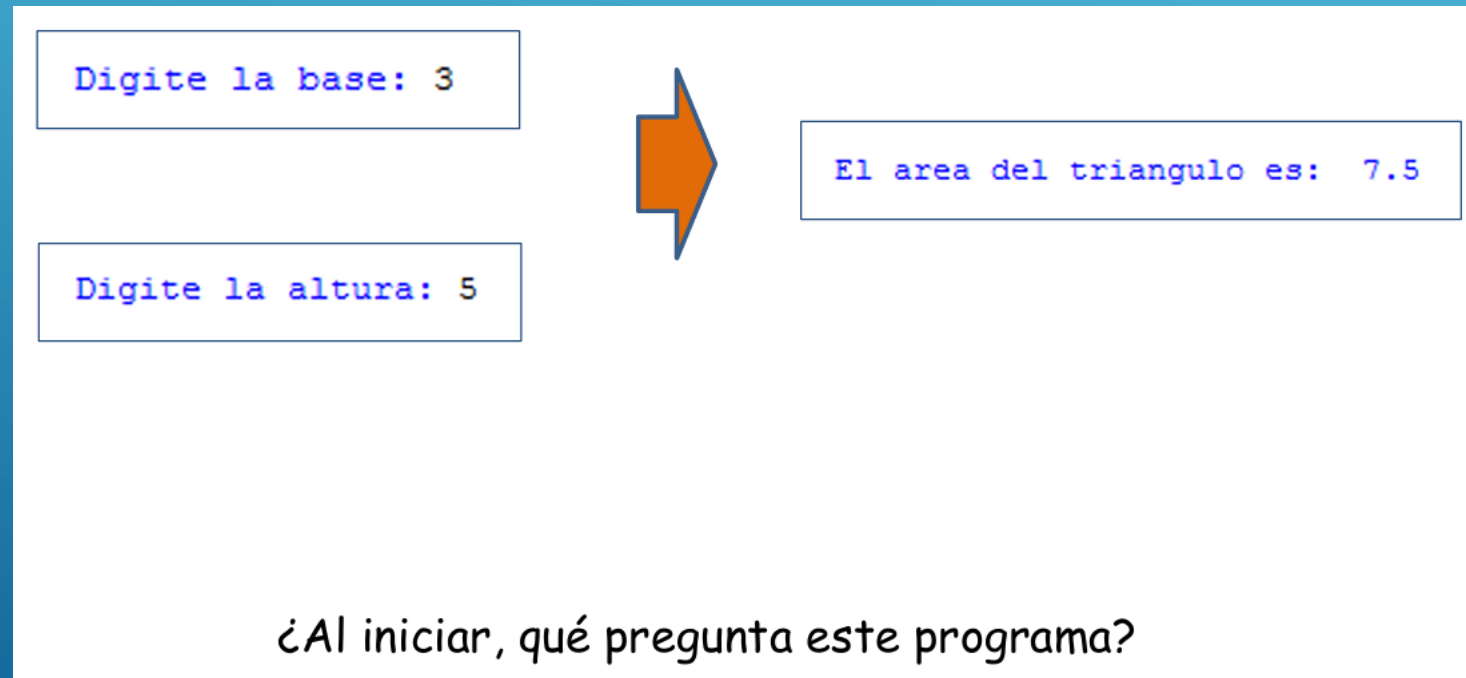
METODOLOGÍA PARA RESOLVER PROBLEMAS

2. Diseñar el algoritmo y escribirlo en pseudocódigo.

Inicio

base, altura: real

area: real



Fin

METODOLOGÍA PARA RESOLVER PROBLEMAS

2. Diseñar el algoritmo y escribirlo en pseudocódigo.

Inicio

base, altura: real

area: real

base = Leer("Digite el valor de la base del Triangulo: ")

altura = Leer("Digite el valor de la altura del Triangulo: ")

area = (base * altura)/2

Imprimir ("El área del triangulo es: ", área)

Fin

El area del triangulo es: 7.5

METODOLOGÍA PARA RESOLVER PROBLEMAS

3. Realizar una prueba de escritorio para el algoritmo

Inicio

base, altura: real

area: real

base = Leer("Digite el valor de la base del Triangulo: ")

altura = Leer("Digite el valor de la altura del Triangulo: ")

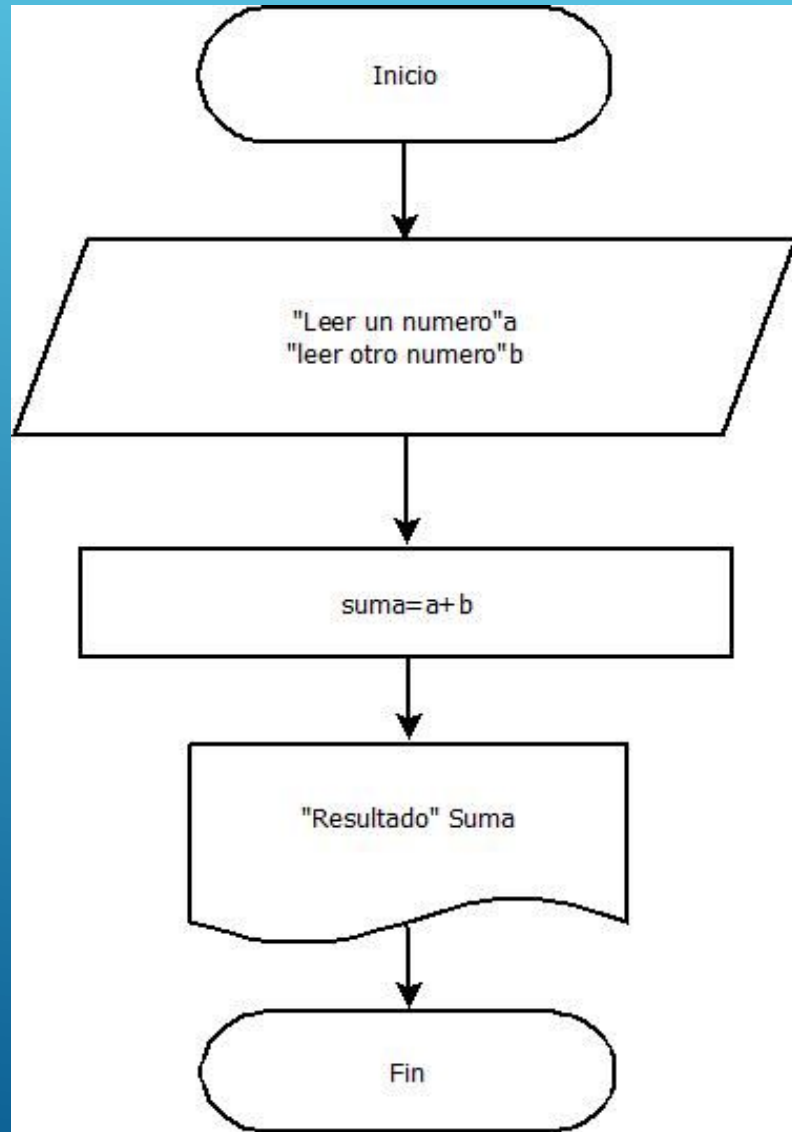
area = (base * altura)/2

Imprimir ("El área del triangulo es: ", área)

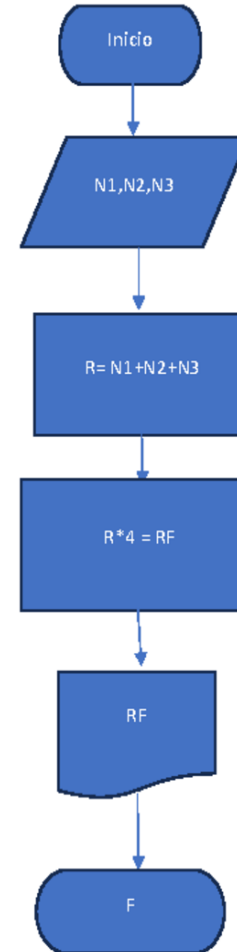
Fin

base	altura	area
1	2	1.0
4	5	10.0
2	3	3.0

EJEMPLOS DE DIAGRAMAS DE FLUJO



Crear un diagrama de flujo que sume 3 números y luego multiplíquelo por 4, imprima el resultado.



Campos	Variables
Numero 1	N1
Numero 2	N2
Numero 3	N3
Resultados	R
Resultado Final	RF

EJEMPLOS DE DIAGRAMAS DE FLUJO

a) Hacer el Algoritmo para sumar 3 números leídos por teclado, dividirlos entre 2 e imprimir el resultado.

Algoritmo: Suma

Var

Entero: A,B,C,P

Real: S

Inicio

Leer A

Leer B

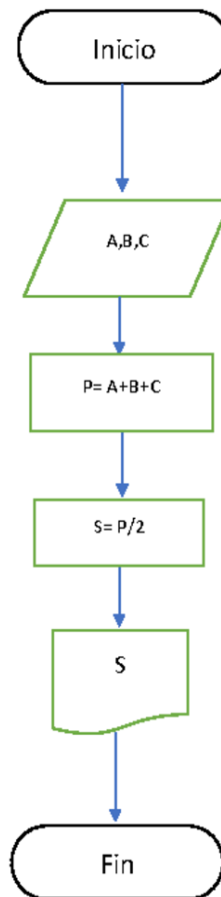
Leer C

$P = A + B + C$

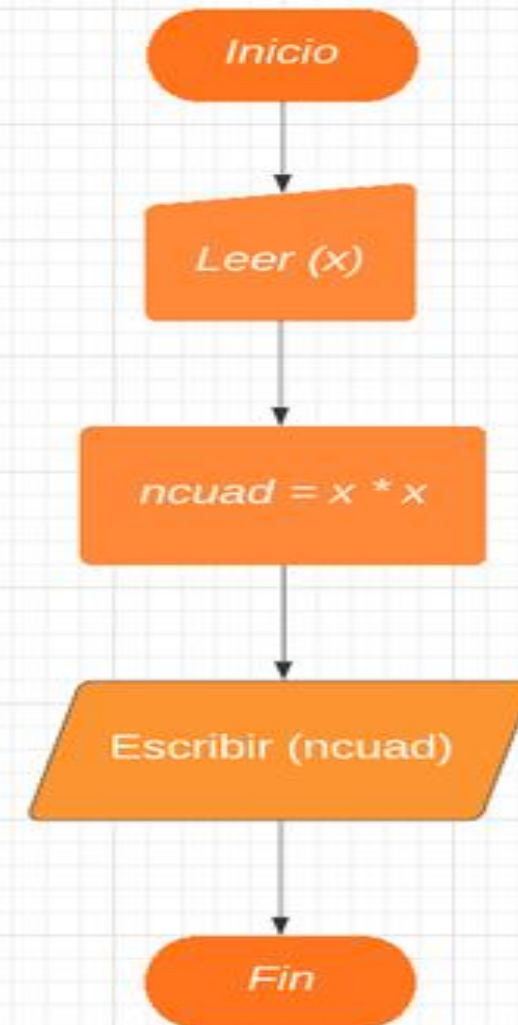
$S = P / 2$

Imprimir: S

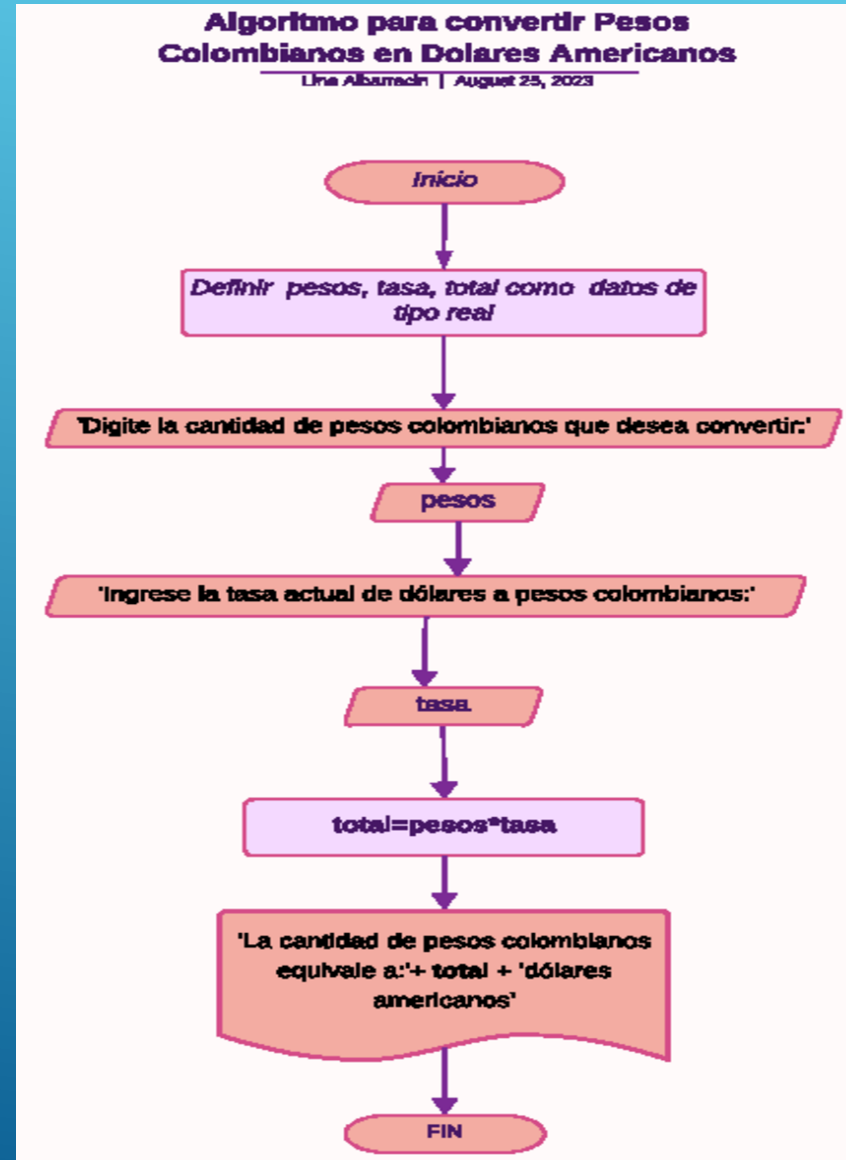
Fin



Leer un número y calcular su cuadrado



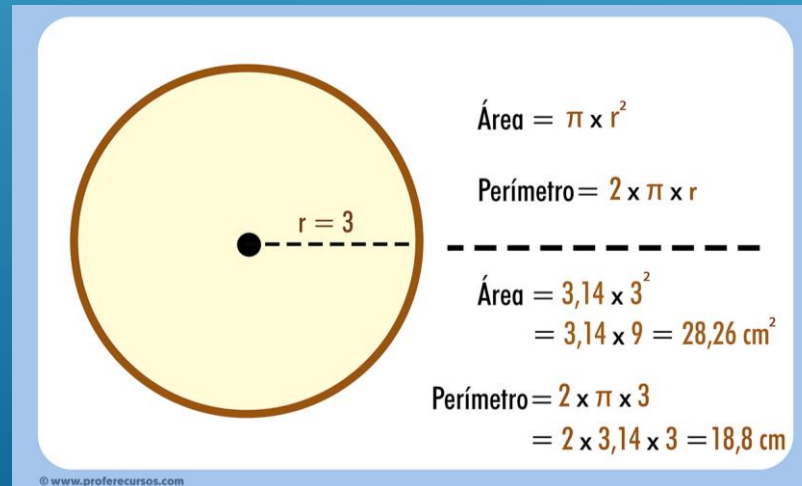
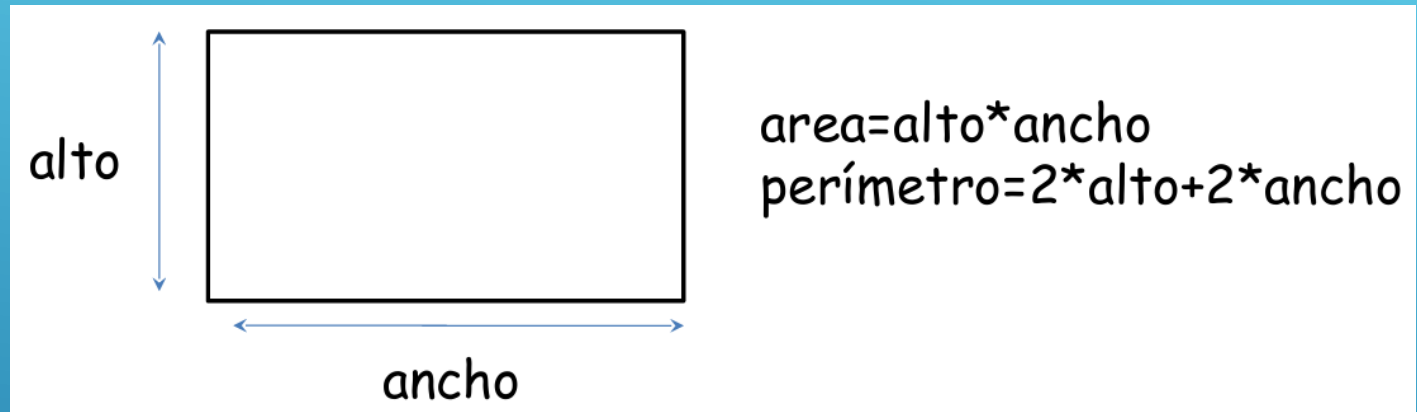
EJEMPLOS DE DIAGRAMAS DE FLUJO



EJERCICIOS

1. Desarrollar un programa que permita calcular el valor de la función $f(x,y) = 3x^2 - 2xy + y^2$
2. Desarrollar un programa que permita calcular el área y el perímetro de un rectángulo de dimensiones alto y ancho.
3. Desarrollar un programa que permita calcular el IVA (16%) dado el valor de una venta.
4. Desarrollar un programa que permita calcular el área y el perímetro de un círculo de radio.
5. La nota final del curso de Algoritmia se calcula a partir de la nota del primer parcial, segundo parcial y proyecto, los pesos son 30%, 30% y 40% respectivamente.

RECOMENDACIONES



RESUMEN

1. Algoritmo.
 2. Enfoque Sistémico.
 3. Entradas y Salidas de un Problema Algorítmico.
 4. Partes, Instrucciones, Elementos y Datos de un Programa
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.