



Teoría 3

Resolución de problemas

Resolución de Problemas y Algoritmos

Ingeniería en Computación (TU y TFA)

Profesorado en Ciencias de la Computación (TU y TFA)



Teoría 3

Resolución de problemas



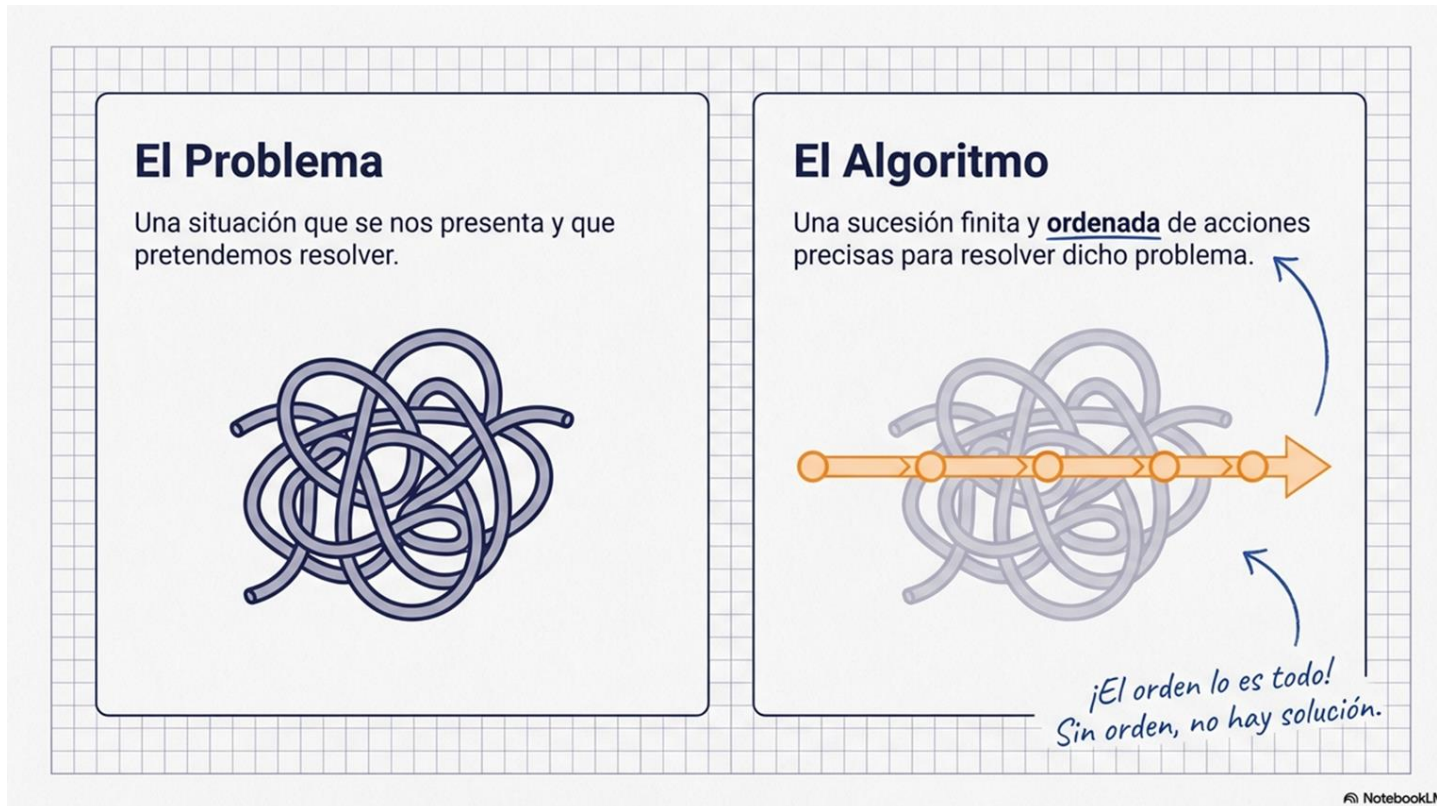
- ✓ Categorías de problemas: no computacionales y computacionales.
- ✓ **¿Cómo resolverlos?** Etapas en la resolución de problemas.
- ✓ **Etapa 1: Análisis del Problema.**
 - Entender el enunciado y comprender el problema.
 - Representación e interpretación de datos (entrada/proceso/salida).
- ✓ **Etapa 2: Diseño y desarrollo del algoritmo para resolver el problema.**
 - Descomposición de un problema en tareas.
 - Algoritmo que da solución a un problema: pseudocódigo y diagrama de flujo.
- ✓ **Etapa 3: Codificación en un lenguaje de programación (programa).**
- ✓ **Etapa 4: Verificar/revisar la solución.**

Problema

es una situación que se nos presenta y que mediante la aplicación de un **algoritmo** pretendemos resolver.

Algoritmo

es una sucesión **finita y ordenada** de acciones o pasos **precisos** que permiten resolver un problema.



Categorías de Problemas



Problema no computacional

son problemas que se resuelven mediante algoritmos que involucran acciones físicas o mentales sin requerir el uso de una computadora. La “ejecución” de la solución recae en un procesador humano que utiliza herramientas o elementos del entorno para llevar a cabo la tarea.

Problema computacional

Son problemas que se resuelven mediante algoritmos que luego serán ejecutados por computadoras o dispositivos digitales. Estos ejecutan una serie de acciones primitivas definidas en un lenguaje de programación, son aquellos que para resolverlos involucran, generalmente, operaciones aritméticas, lógicas y relacionales.

Categorías de Problemas



Problema no computacional

Por ejemplo: cebar mate, cambiar la llanta de un auto o buscar una palabra en el diccionario.

Problema computacional

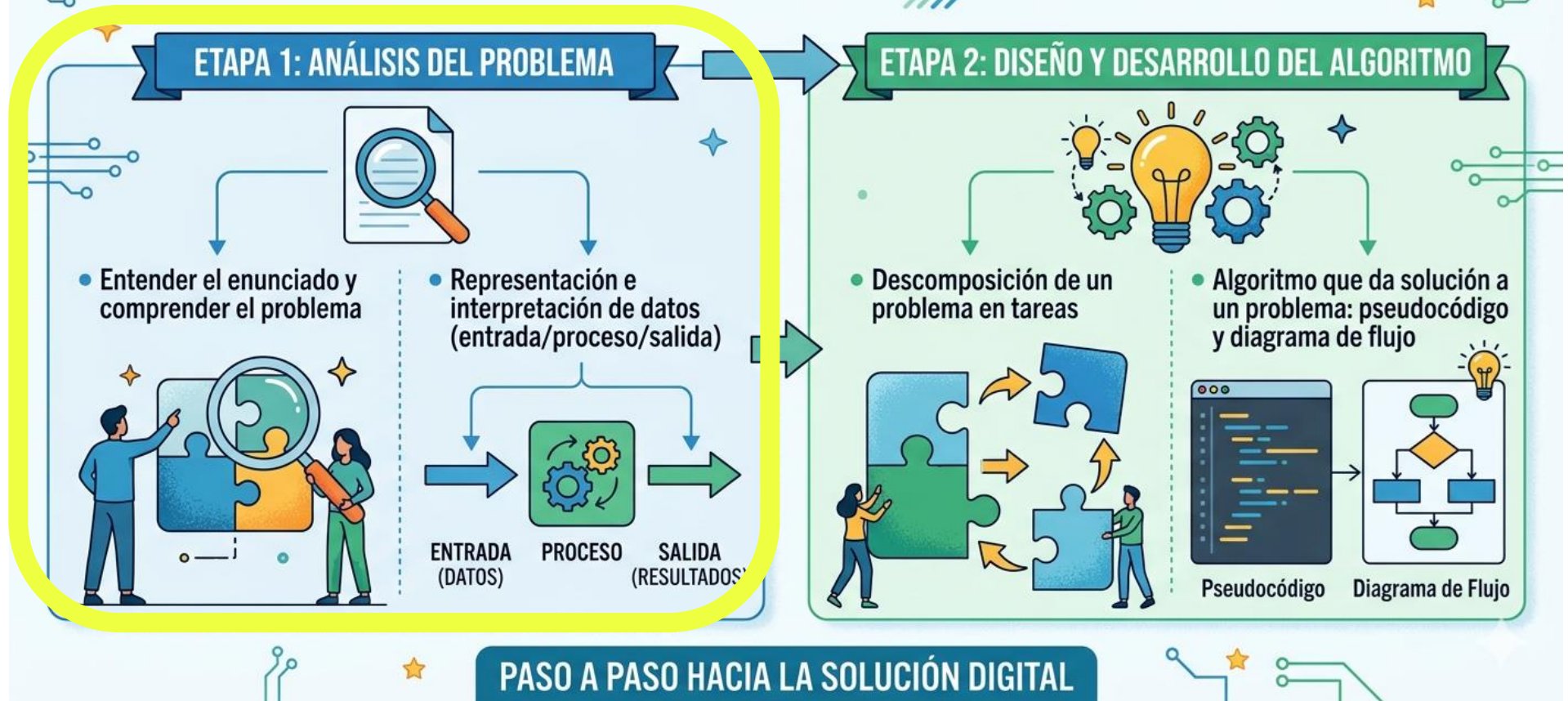
Por ejemplo: obtener la suma de dos números, encontrar el promedio de un conjunto de notas, encontrar el menor número en una secuencia, determinar si un número es par o impar, etc.



Problemas computacionales



¿CÓMO RESOLVERLOS? ETAPAS EN LA RESOLUCIÓN DE PROBLEMAS



Existen diferentes métodos, todos ellos tienen en común el trabajar en etapas.

Etapa 1: Análisis del Problema.



- Entender el enunciado y comprender el problema.
- Representación e interpretación de datos (entrada/proceso/salida).





Entender el enunciado y comprender el problema.

Un problema se define, generalmente, mediante un **enunciado** escrito en lenguaje natural. Por ejemplo:

Enunciado: Se ha escaneado una vieja caja que contiene cartas antiguas, generando 215 archivos de imagen.

Las imágenes están guardadas en una carpeta llamada `DEPÓSITO_CASA`, la cual ocupa 4.5 GB de espacio en el disco duro marca Kingston.

Los archivos tienen distintos formatos (JPG, PNG y PDF) y algunos nombres de archivo están escritos en mayúsculas.

El objetivo es desarrollar un algoritmo que permita ordenar todos los archivos de la carpeta.

Para **ayudarnos a entender correctamente el enunciado** nos podemos plantear preguntas, como por ejemplo:

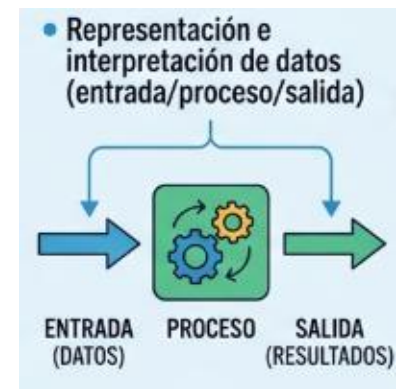
1. ¿Entiendo cada palabra o siglas del enunciado?
2. ¿Qué categoría de problema es? ¿computacional o no computacional?
3. ¿Cuál es el objetivo? ¿la salida esperada?
4. ¿Está completo el enunciado? ¿Hay datos que podrían estar faltando para resolver el problema?
5. ¿Hay datos en el enunciado que no son importantes para resolver el problema o no están claros?
6. ¿Necesito consultar o tener otros conocimientos para resolver el problema planteado?

Enunciado: Se ha escaneado una vieja caja con 215 archivos de imagen que contienen cartas antiguas. Las imágenes están guardadas en una carpeta llamada DEPÓSITO_CASA, la cual ocupa 4.5 GB de espacio en el disco duro marca Kingston. Los archivos tienen distintos formatos (JPG, PNG y PDF) y algunos nombres de archivo están escritos en mayúsculas.

El objetivo es desarrollar un algoritmo que permita ordenar todos los archivos de la carpeta.

1. ¿Entiendo cada palabra o siglas del enunciado?
2. ¿Qué categoría de problema es? ¿computacional o no computacional?
3. ¿Cuál es el objetivo? ¿cuál es la salida esperada?
4. ¿Está completo el enunciado? ¿Hay datos que podrían estar faltando para resolver el problema?
5. ¿Hay datos en el enunciado que no son importantes para resolver el problema o no están claros?
6. ¿Necesito consultar o tener otros conocimientos para resolver el problema planteado?

Representación e interpretación de datos.



Para resolver problemas es muy importante saber **representar e interpretar datos**.

Aprender a identificar los tipos de **datos de entradas** a utilizar, nos permite crear **procesos** claros, que nos ayuda a obtener **resultados** precisos.

DATOS NUMÉRICOS

- Enteros (e.g., 5, -12)
- Flotantes (e.g., 3.14, -0.01)

DATOS DE TEXTO (ALFANUMÉRICOS)

- Cadenas (e.g., 'Hola', 'Ecuación')
- Caracteres (e.g., 'a', '@')

DATOS LÓGICOS (BOOLEANOS)

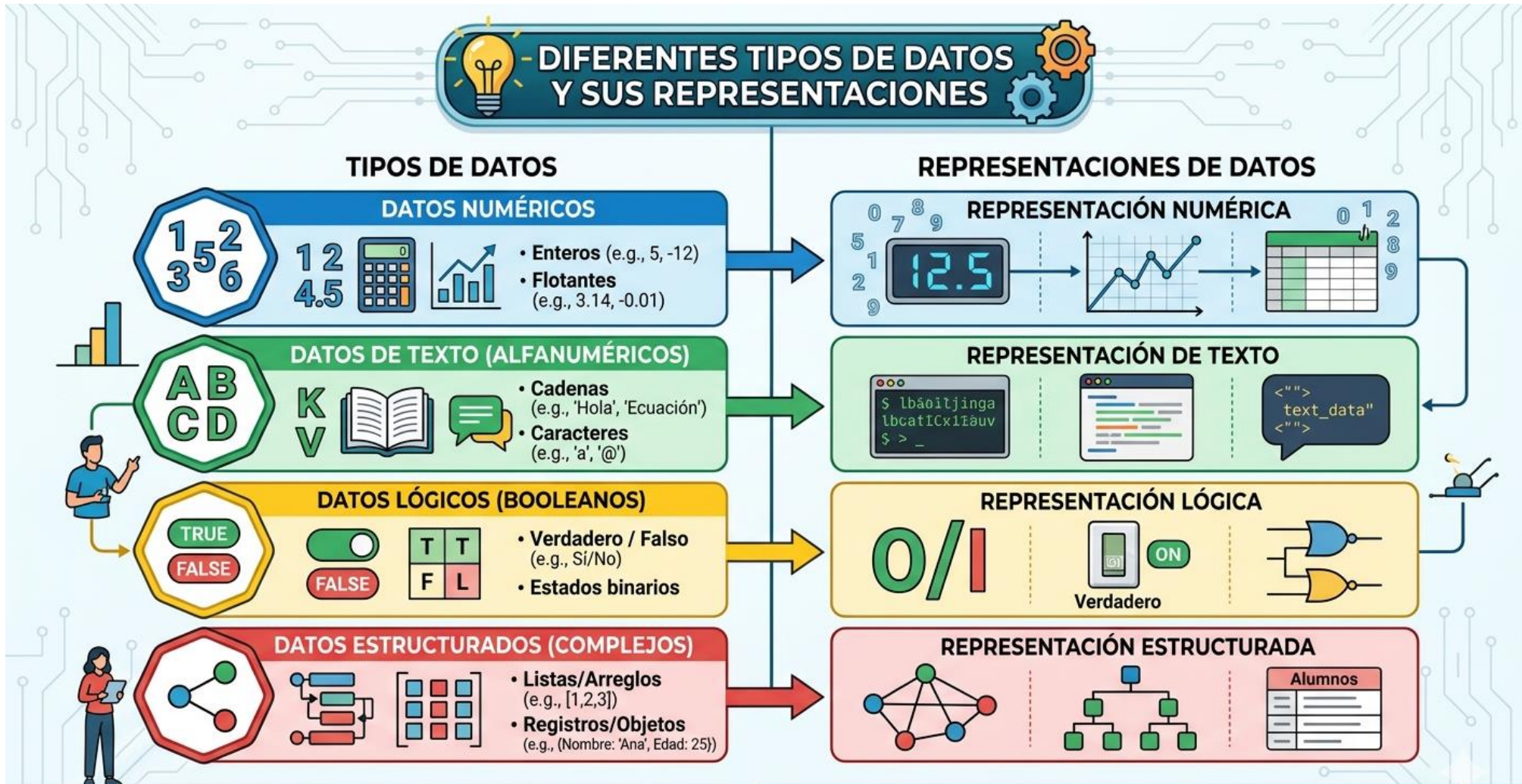
T	T
F	L

- Verdadero / Falso (e.g., Si/No)
- Estados binarios

DATOS ESTRUCTURADOS (COMPLEJOS)

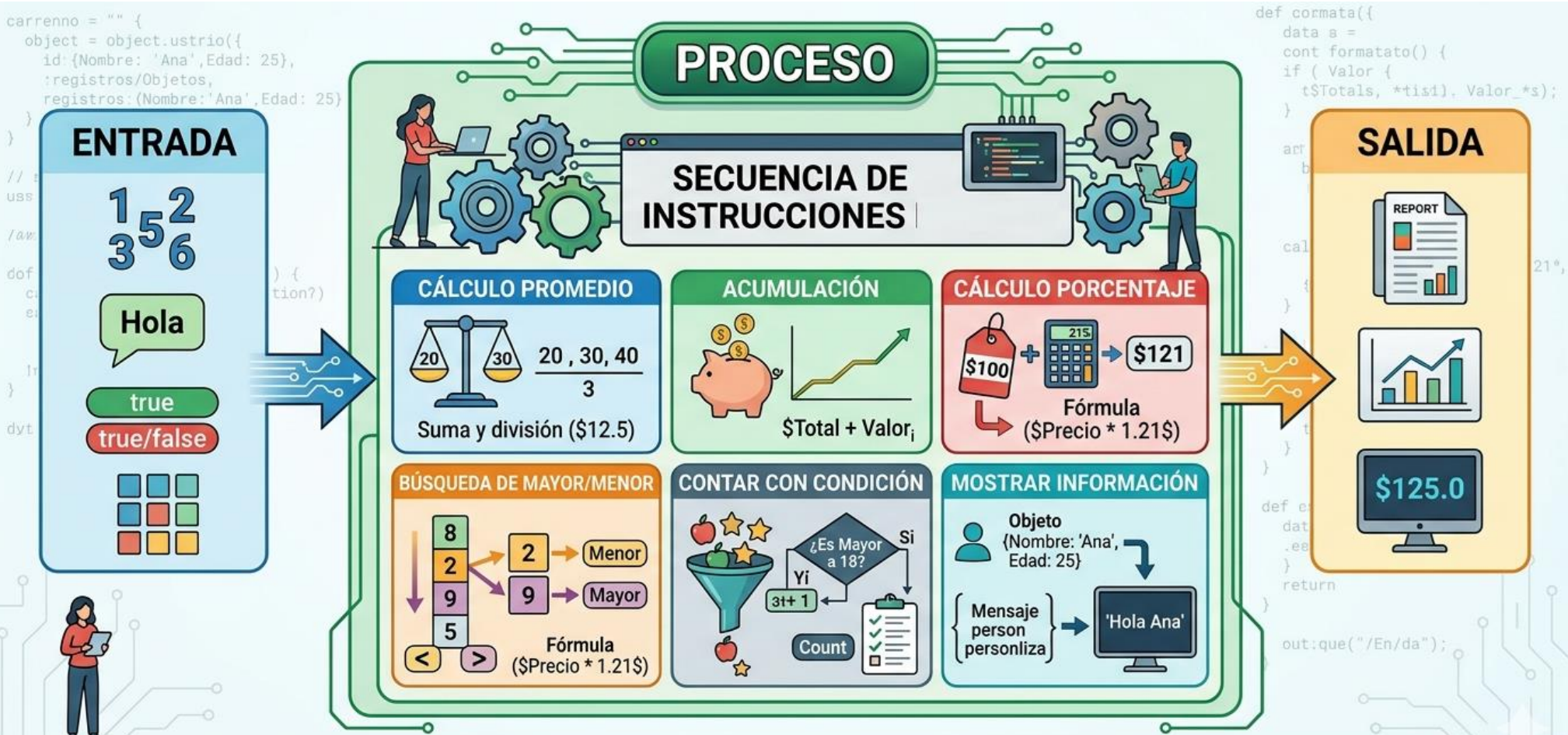
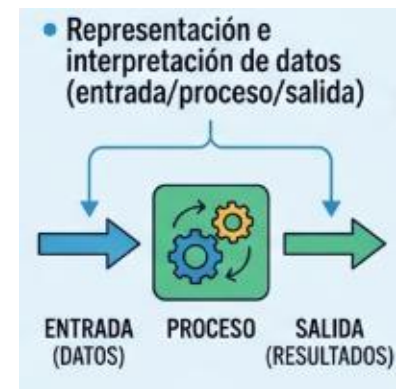
- Listas/Arreglos (e.g., [1,2,3])
- Registros/Objetos (e.g., (Nombre: 'Ana', Edad: 25))

Un problema puede ser de distintos tipos como así también sus **datos** pueden tomar distintas representaciones.



Debemos identificar tres elementos claves:

- **Entrada:** Los datos iniciales disponibles.
- **Proceso:** El tratamiento o cálculos que haremos con esos datos.
- **Salida:** La información final solicitada, el resultado.



Etapa 1: Análisis del Problema.



- Entender el enunciado y comprender el problema.
- Representación e interpretación de datos (**E**ntrada/**P**roceso/**S**alida).



Problemas computacionales



¿CÓMO RESOLVERLOS? ETAPAS EN LA RESOLUCIÓN DE PROBLEMAS

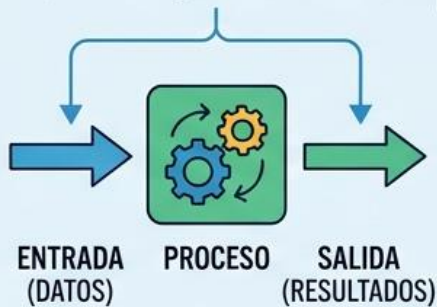
ETAPA 1: ANÁLISIS DEL PROBLEMA



- Entender el enunciado y comprender el problema



- Representación e interpretación de datos (entrada/proceso/salida)



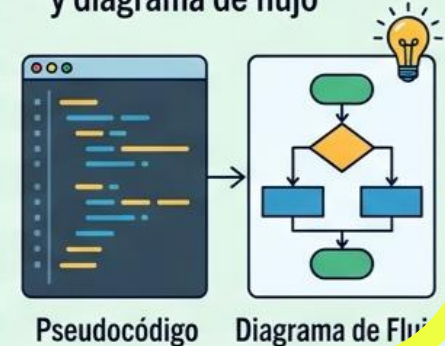
ETAPA 2: DISEÑO Y DESARROLLO DEL ALGORITMO



- Descomposición de un problema en tareas



- Algoritmo que da solución a un problema: pseudocódigo y diagrama de flujo



PASO A PASO HACIA LA SOLUCIÓN DIGITAL

Etapa 2: Diseño y desarrollo del algoritmo



- Descomposición de un problema en tareas.
- Algoritmo que da solución a un problema: pseudocódigo y diagrama de flujo.



• Descomposición de un problema en tareas



Descomposición de un problema en tareas.

La **estrategia de descomponer** un problema es dividirlo en partes más pequeñas y manejables que sean más simples de resolver.

$$\neg (P \rightarrow \neg (Q \wedge R)) \vee (P \rightarrow \neg S) \wedge (((P \wedge (\neg Q)) \wedge R) \vee S)$$



Enunciado: Determinar el presupuesto total para organizar una fiesta temática de "Rock argentino". El algoritmo debe calcular el costo total sumando costo del catering, decoración y música. Tener en cuenta que el costo de catering es por persona, pero la decoración y música es un valor fijo.

Versión 1 (Estrategia de descomponer el problema)

T1. Definir los datos para catering por persona, decoración y música.

T2. Ingresar los datos para costo de catering por persona, decoración y música.

T3. Sumar los costos de catering, decoración y música.

T4. Informar el resultado calculado en T3.

**Enumerar las tareas
y escribir en infinitivo**

**¿Faltan datos?
¿Son importantes?**

Enunciado: Determinar el presupuesto total para organizar una fiesta temática de "Rock argentino" para 100 personas. El algoritmo debe calcular el costo total sumando catering, decoración y música. Tener en cuenta que el costo de catering es por persona, pero la decoración y música es un valor fijo.

Versión 1 (Estrategia de descomponer el problema)

T1. Definir los datos para catering por persona, decoración y música.

T2. Ingresar los datos para catering por persona, decoración y música.

T3. Calcular el costo de catering para 100 personas.

T4. Sumar los costos de catering, decoración y música.

T5. Informar el resultado calculado en T4.

Enunciado: Determinar el presupuesto total para organizar una fiesta temática de "Rock argentino". El algoritmo debe calcular el costo de catering, decoración y música. Tener en cuenta que el costo de decoración y música es un valor fijo.

¿Faltan datos?

¿Son importantes?

LOS PIDO (Definir e ingresar)

Versión 1 (Estrategia de descomponer el problema)

T1. Definir los datos para costo de catering por persona, decoración, música **y cantidad de personas.**

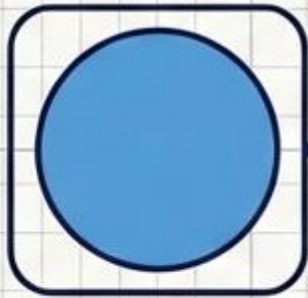
T2. Ingresar los datos para costo de catering por persona, decoración, música **y cantidad de personas.**

T3. Calcular el costo de catering para **la cantidad de personas ingresadas.**

T4. Sumar los costos de catering, decoración y música.

T5. Informar el resultado calculado en T4.

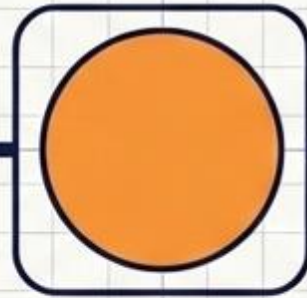
Descomposición de un problema en tareas.



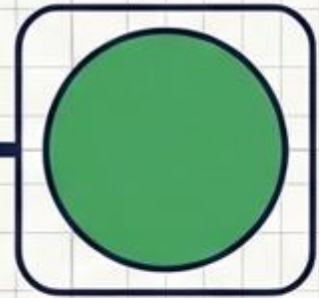
Definir los datos de entrada necesarios.



Ingresar la información al sistema.



Procesamiento. Cálculos simples y compuestos. ¡Aquí ocurre la magia!



Informar el resultado. Un algoritmo mudo no es útil.

Separar siempre los cálculos en pasos más comprensibles.



Versión 1 (Estrategia de descomponer el problema)

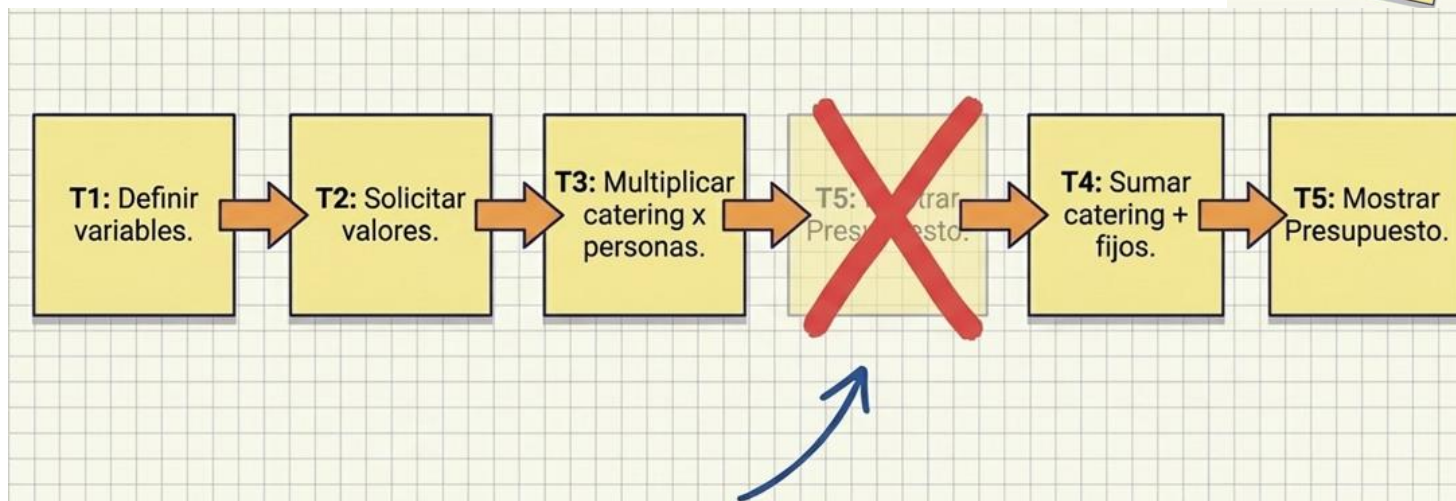
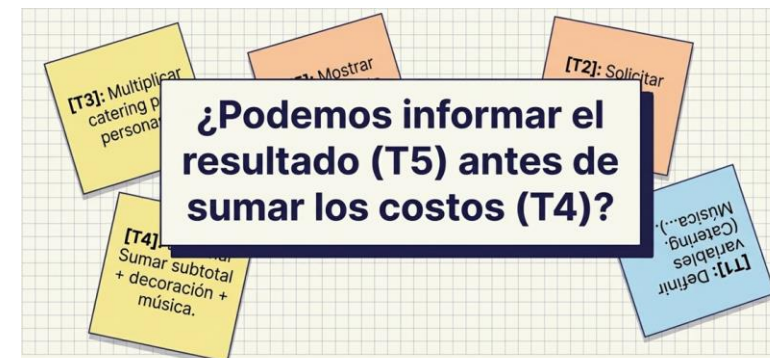
T1. Definir los datos para costo de catering por persona, decoración, música **y cantidad de personas.**

T2. Ingresar los datos para costo de catering por persona, decoración, música **y cantidad de personas.**

T3. Calcular el costo de catering para **la cantidad de personas ingresadas.**

T4. Sumar los costos de catering, decoración y música.

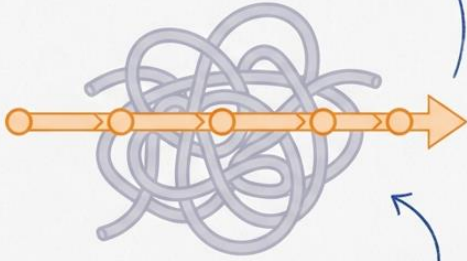
T5. Informar el resultado calculado en T4.



Respuesta: ¡Imposible! Un algoritmo es una sucesión estricta y ordenada.

El Algoritmo

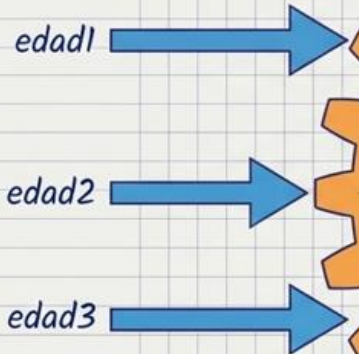
Una sucesión finita y **ordenada** de acciones precisas para resolver dicho problema.



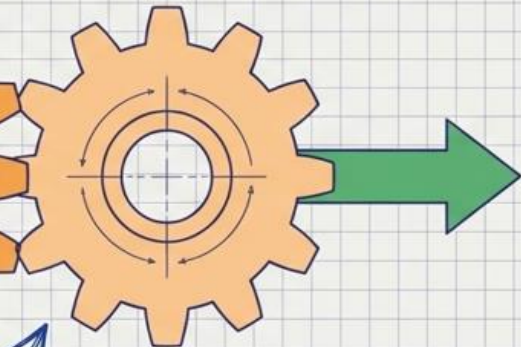
*¡El orden lo es todo!
Sin orden, no hay solución.*

¡El orden lo es todo!
Sin orden, no hay solución.

1. Acumulación (Suma)



2. División (/ 3)



El engranaje 2 no puede girar si el engranaje 1 no termina su trabajo.

Promedio

¿Qué es un Algoritmo?

Un algoritmo es una **secuencia finita y ordenada de acciones primitivas** que pueden ser ejecutadas por un **procesador** y que lleva a la solución de un problema planteado en un **enunciado**, valiéndose de un **conjunto de recursos necesarios** para su ejecución.



¿Qué es un Algoritmo?

Un algoritmo es una **secuencia finita y ordenada de acciones primitivas** que pueden ser ejecutadas por un **procesador** y que lleva a la solución de un problema planteado en un **enunciado**, valiéndose de un **conjunto de recursos necesarios** para su ejecución.



Características de un algoritmo

- 1. Finito:** un algoritmo debe terminar después de ejecutar un número finito de pasos.
- 2. Preciso:** cada paso en un algoritmo debe estar definido con precisión, esto es, la acción a seguir no debe ser ambigua, sino rigurosamente especificada. Considerando el ambiente en el que se trabaja, cada **acción primitiva** debe significar, una sola tarea, bien determinada y sin lugar a dudas.
- 3. Efectivo:** Un algoritmo debe llevar a la solución del problema.

Para que sea un algoritmo debe ser:



Finito

Debe terminar en algún momento. No hay bucles infinitos.



Preciso

Cada paso rigurosamente definido, cero ambigüedades.



Efectivo

Debe lograr resolver el problema planteado inicialmente.

Si falla una de estas tres, no es un algoritmo válido.

Problema: realizar un cálculo aritmético.

Enunciado: Escribir un algoritmo para calcular y mostrar el promedio de 3 números cualesquiera.

Algoritmo:

Inicio

Definir num1, num2, num3, prom de tipo reales

Escribir "Ingrese el primer número:"

Leer num1

Leer num2

Escribir "Ingrese el segundo número:"

$\text{promedio} = \text{num1} + \text{num2} + \text{num3} / 3$

Escribir "Ingrese el tercer número:"

Leer num3

Escribir "El promedio es:", prom

Fin

**¿Resuelve el enunciado?
Llega a la salida esperada
esperada**



¿Cómo saber si un algoritmo resuelve el enunciado? Es decir, llega a la salida esperada.

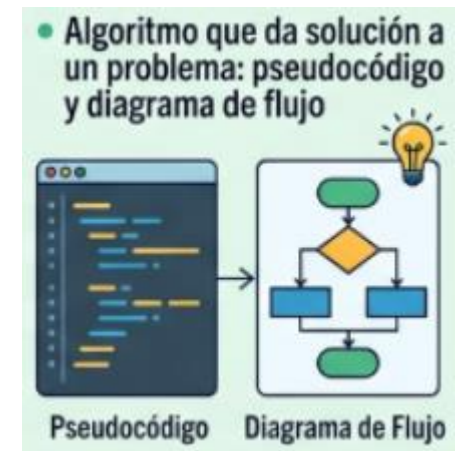
La **ejecución de un algoritmo** es el proceso de seguir paso a paso todas las instrucciones definidas en el mismo, transformando los datos de entrada en los resultados deseados. Es decir, es la puesta en práctica del algoritmo para resolver un problema de forma sistemática y ordenada.

Al hacerlo, podemos encontrar y corregir errores, como usar datos incorrectos o seguir un orden equivocado.

En resumen, la **ejecución de un algoritmo** permite detectar y corregir errores, lo que es fundamental para garantizar la precisión en cualquier tarea o proceso automatizado.

Etapa 2: Diseño y desarrollo del algoritmo para resolver el problema.

1. Descomposición de un problema en tareas.
2. Algoritmo que da solución a un problema: **pseudocódigo y diagrama de flujo.**



Etapa 2: Pseudocódigo y diagrama de flujo



CONTINUARÁ...en la Teoría 4

Pseudocódigo

Representación textual detallada. Parecida al lenguaje de la máquina.

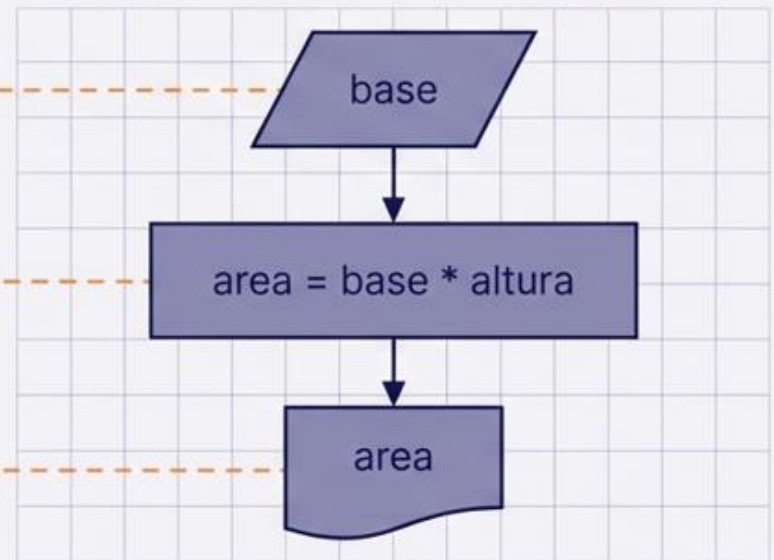
LEER base

calcular $\text{area} = \text{base} * \text{altura}$

MOSTRAR area

Diagrama de Flujo

Representación gráfica. Permite visualizar rápidamente cómo fluye la ejecución.



Etapa 2: Pseudocódigo y diagrama de flujo



CONTINUARÁ...en la Teoría 4

PSEUDOCÓDIGO es usado para la **representación no gráfica o textual** de algoritmos de la forma más detallada posible y a su vez lo más parecida posible a lenguaje que posteriormente se utilizará para la codificación del mismo.

DIAGRAMA DE FLUJO es utilizado para **representar gráficamente** algoritmos, facilita la visualización del flujo de ejecución del mismo.

Teoría 3

Resolución de problemas



- ✓ Categorías de problemas: no computacionales y computacionales.
- ✓ **¿Cómo resolverlos?** Etapas en la resolución de problemas.
- ✓ **Etapa 1: Análisis del Problema.**
 - Entender el enunciado y comprender el problema.
 - Representación e interpretación de datos (entrada/proceso/salida).
- ✓ **Etapa 2: Diseño y desarrollo del algoritmo para resolver el problema.**
 - Descomposición de un problema en tareas.
 - Algoritmo que da solución a un problema: **pseudocódigo y diagrama de flujo.**



Teoría 3

Resolución de problemas

¡Ya podemos comenzar con el
Práctico 3!