Teoría 3

Resolución de problemas

Resolución de Problemas y Algoritmos

Ingeniería en Computación
Ingeniería en Informática
Profesorado en Ciencias de la Computación

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.
  - Refinamiento sucesivo. Algoritmo que da solución a un problema.
  - Representación de algoritmos: pseudocódigo y diagrama de flujo
- ✔ **Etapa 3: Codificación en un lenguaje de programación (programa).**
- ✔ **Etapa 4: Verificar/revisar la solución**

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.
  - Refinamiento sucesivo. Algoritmo que da solución a un problema.
  - Representación de algoritmos: pseudocódigo y diagrama de flujo

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 aquellos en los que la solución se puede describir por pasos utilizando simplemente palabras. Por ejemplo: cebar mate, cambiar la llanta de un auto o buscar una palabra en el diccionario.

Problema computacional

son aquellos que para resolverlos involucran, generalmente, operaciones aritméticas, lógicas y relacionales. Por ejemplo: obtener la suma de dos números, encontrar el promedio de un conjunto de notas, encontrar el menor número en una secuencia, etc.

Problemas NO computacionales

- Hacer un licuado de frutillas.
- Llegar a la universidad.
- Grabar y subir un video en tiktok.
- Organizar una fiesta de cumpleaños.

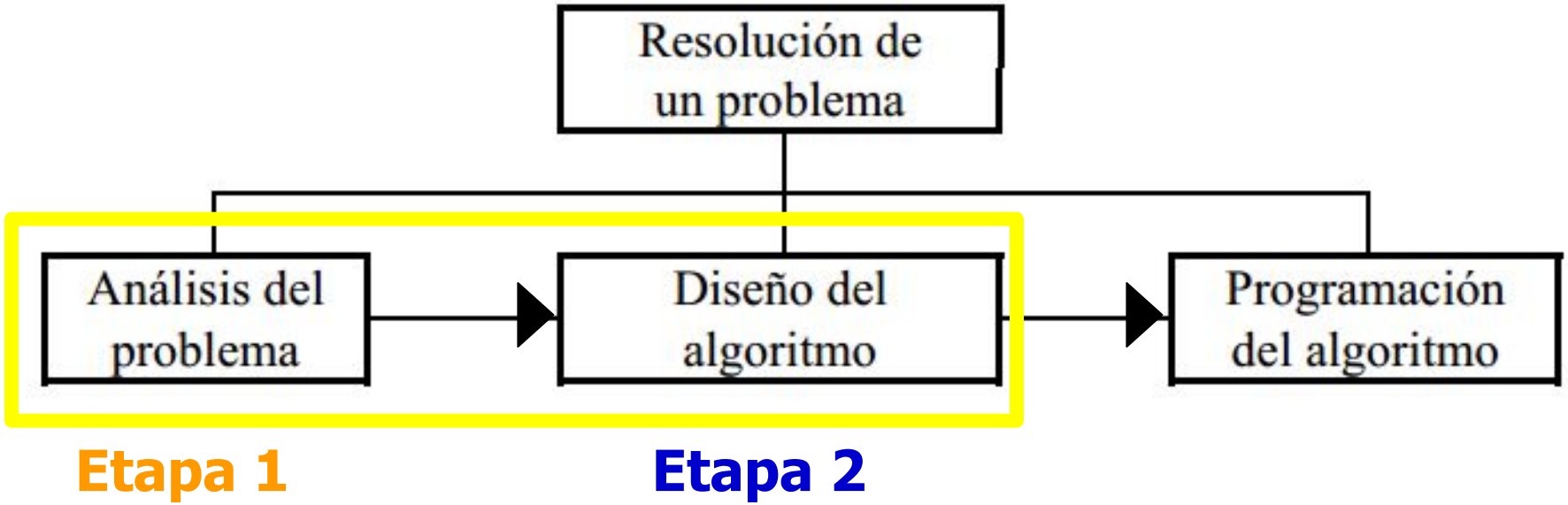
Problemas computacionales

- Sumar tres números cualquiera.
- Mostrar el nombre del estudiante con menos faltas a clase y mayor nota en el primer parcial.
- Calcular el promedio de todas las notas del año de un estudiante.
- Calcular la edad de una persona sabiendo el año de nacimiento.

Etapas en la Resolución de un Problema

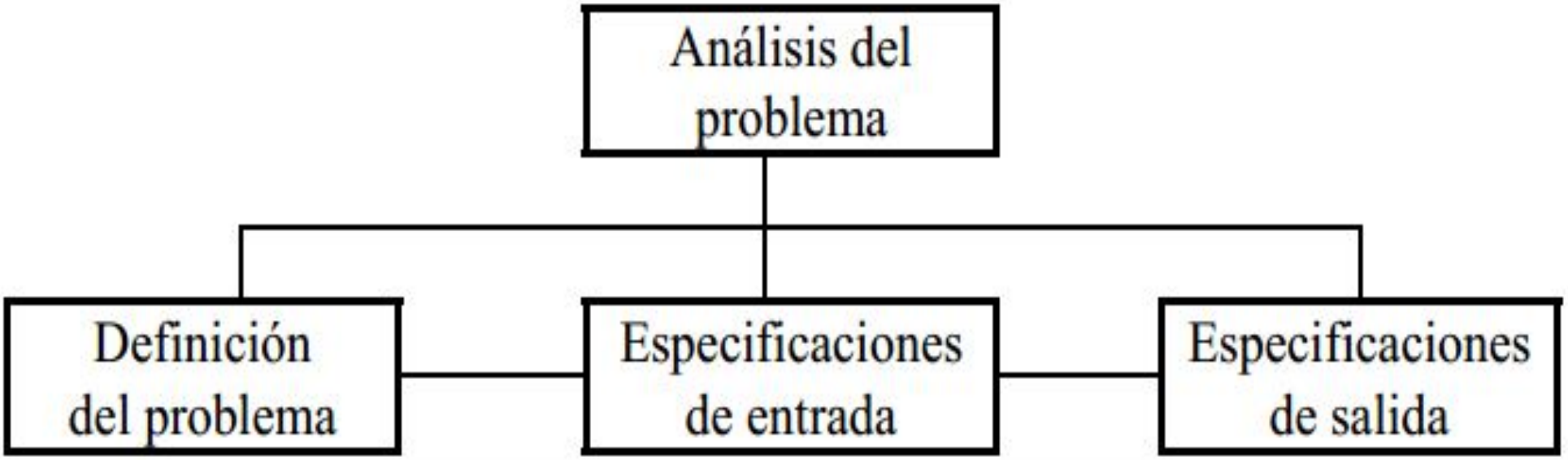
¿Cómo resolver un problema?

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



Etapa 1: Análisis del Problema.

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



Etapa 1: Análisis del Problema.

1. Entender el enunciado y comprender el problema.

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

Enunciado: En una vieja caja en el depósito de mi casa, encontré más de 200 cartas antiguas de mi abuelo. Las cartas eran de distintos tipos, tamaños y colores. La caja medía 10 pulgadas de largo, estaba muy sucia y arruinada. Necesito encontrar un algoritmo para ordenar todas las cartas.

?

Entender el enunciado y comprender el problema.

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

1. ¿Entiendo cada palabra/siglas del enunciado?
2. ¿Qué categoría de problema es? ¿Cuál es el objetivo?

   ¿computacional o no computacional?

1. ¿Está completo el enunciado? ¿Hay datos que podrían estar faltando para resolver el problema?
2. ¿Hay datos en el enunciado que no son importantes para resolver el problema o no están claros?
3. ¿Necesito “googlear” o tener otros conocimientos para resolver el problema planteado?

Ejemplos de problemas:

1. Determinar si una persona tiene baja estatura.
2. Pintar un cuadro de Picasso.
3. Calcular la suma de un conjunto de números.
4. Realizar ejercicio de RCP a una persona.
5. Ordenar descendente alfabéticamente un conjunto de palabras.

Actividad 1: Elegir 2 problemas y completar una tabla para cada uno:

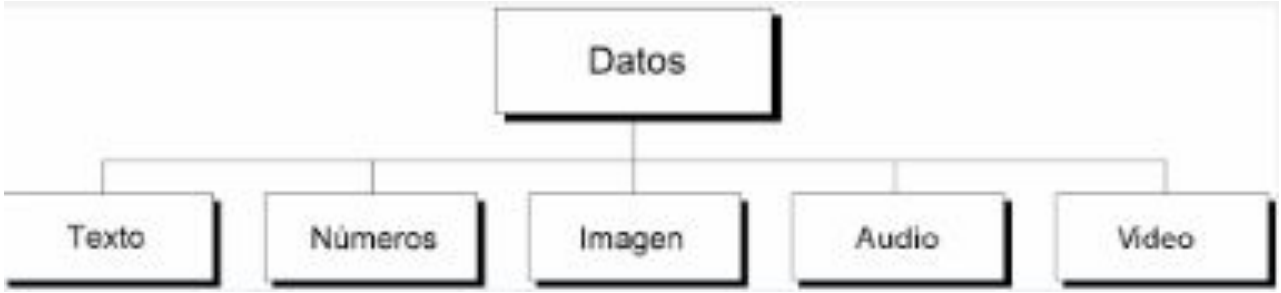
Problema número:	
¿Es computacional o no computacional?	
¿Qué palabras del enunciado no entiendo o no se su significado?	
¿Tiene todos los datos necesarios?Si su respuesta en No, ¿cuáles faltan?	
¿Sabe a qué tiene que llegar, un número, una palabra, una lista?	

Etapa 1: Análisis del Problema.

2. Representación e interpretación de datos.

Saber **representar e interpretar datos** es muy importante para resolver problemas.

Es fundamental desarrollar la capacidad para analizar diferentes **representaciones de datos** y proponer procesos capaces de obtener información relevante para la toma de decisiones.

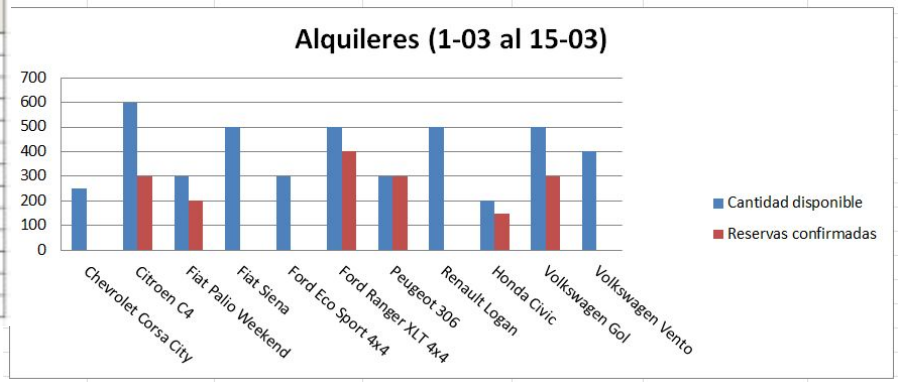


Representación e interpretación de datos

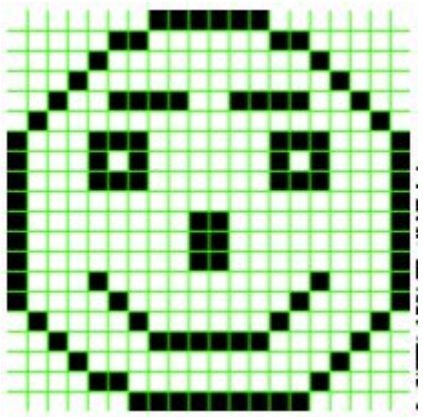
Un problema puede ser de distintos tipos como así también sus **datos** pueden tomar distintas representaciones.

Total de Clases: 22

Estudiante	Asistencias	Inasistencias	% de Asistencia	Nota 1	Nota 2	Nota 3	Presentó certificado?	Promedio	tiene acceso a una recuperación por trabajo?
Estudiante 1	15	7	68,18%	6,00	5,00	7,00	si	6,00	Si tiene acceso
Estudiante 2	20	2	90,91%	7,00	8,00	7,00	si	7,33	Si tiene acceso
Estudiante 3	18	4	81,82%	6,00	0,00	7,00	no	4,33	No tiene acceso
Estudiante 4	20	2	90,91%	7,00	8,00	7,00	no	7,33	No tiene acceso
Estudiante 5	20	2	90,91%	8,00	8,00	7,00	no	7,67	No tiene acceso
Estudiante 6	20	2	90,91%	8,00	8,00	5,00	no	7,00	No tiene acceso
Estudiante 7	20	2	90,91%	8,00	8,00	6,00	si	7,33	Si tiene acceso
Estudiante 8	20	2	90,91%	9,00	0,00	4,00	si	4,33	Si tiene acceso
Estudiante 9	20	2	90,91%	9,00	9,00	7,00	no	8,33	No tiene acceso
Estudiante 10	10	12	45,45%	9,00	9,00	8,00	no	8,67	No tiene acceso
Estudiante 11	6	16	27,27%	5,00	10,00	5,00	no	6,67	No tiene acceso
Estudiante 12	9	13	40,91%	7,00	0,00	9,00	no	5,33	No tiene acceso
Estudiante 13	18	4	81,82%	4,00	10,00	8,00	si	7,33	Si tiene acceso



Licuadora
duraznos
leche
azúcar



Entrada/proceso/Salida

Entrada

Datos de entrada con los que se dispone

Proceso

Proceso o tratamiento a realizarse con los datos.

Salida

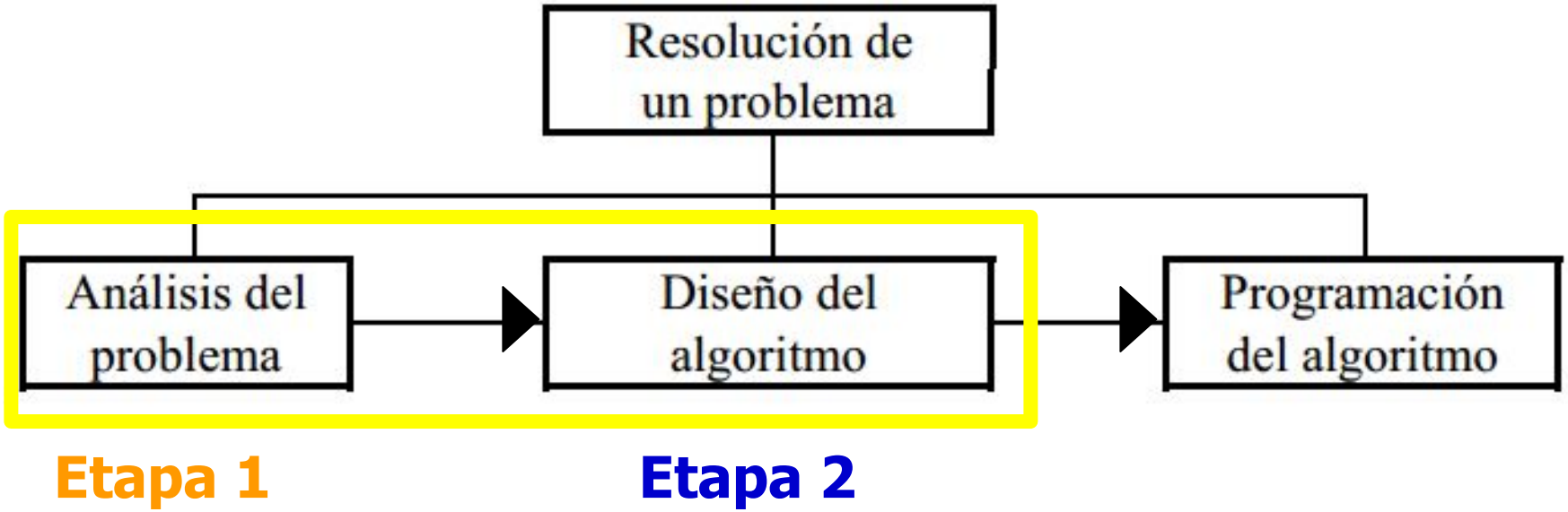
Información deseada de salida.

todos los números del 1 al 14 → **sumar** → **1**

2 → **3** →

Actividad 2: copiar y completar los valores para **1, 2 y 3.**

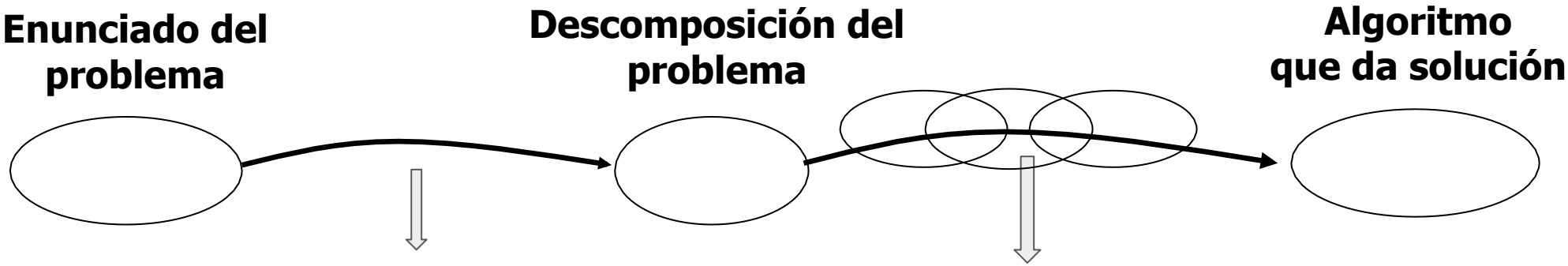
Etapas en la Resolución de un Problema



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

Etapa 2: Diseño y desarrollo del algoritmo

1. Descomposición de un problema en tareas.
2. Refinamiento sucesivo. Algoritmo que da solución a un problema.
3. Pseudocódigo y diagrama de flujo.



Etapa 2: Diseño y desarrollo del algoritmo

1. Descomposición de un problema en tareas.

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

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

Etapa 2: Diseño y desarrollo del algoritmo

Enunciado: Debo planificar una fiesta para 100 personas, compañeros y profesores de la facultad, la temática debe ser "Rock argentino", me indicaron que debe hacerse antes del mes de mayo, que cumple años la UNSL.

Versión 1 (Estrategia de Descomponer el problema)

T1. Elegir fecha y lugar del evento.

T2. Realiz**ar** invitaciones y enviarlas.

T3. Organizar el catering.

T4. Elegir ropa y calzado.

T5. Comprar adornos y souvenirs.

T6. Encargar torta y mesa dulce.

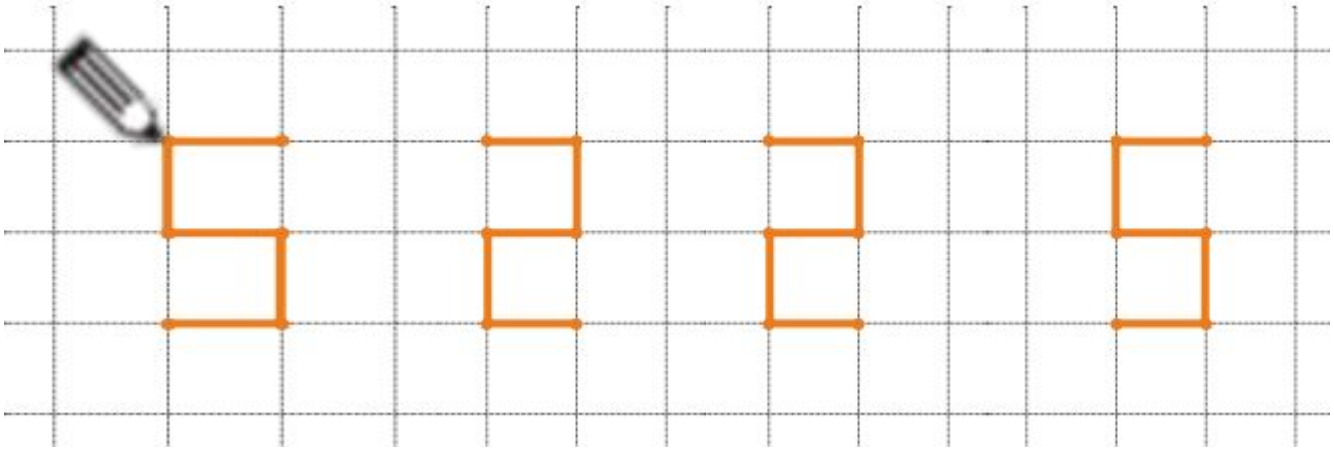
T7. Elegir música.

Enumerar las tareas y escribir en infinitivo

¿Faltan datos al descomponer? ¿Son importantes?

Actividad 3: Descomponer el problema planteado en el siguiente enunciado.

Enunciado: Se tiene una cuadrícula y se necesita dibujar los números resaltados en color naranja según se muestra en la imagen.



Versión 1 (Estrategia de Descomponer el problema)

T1. Dibujar el número 5 y ubicarse para dibujar próximo dibujo.

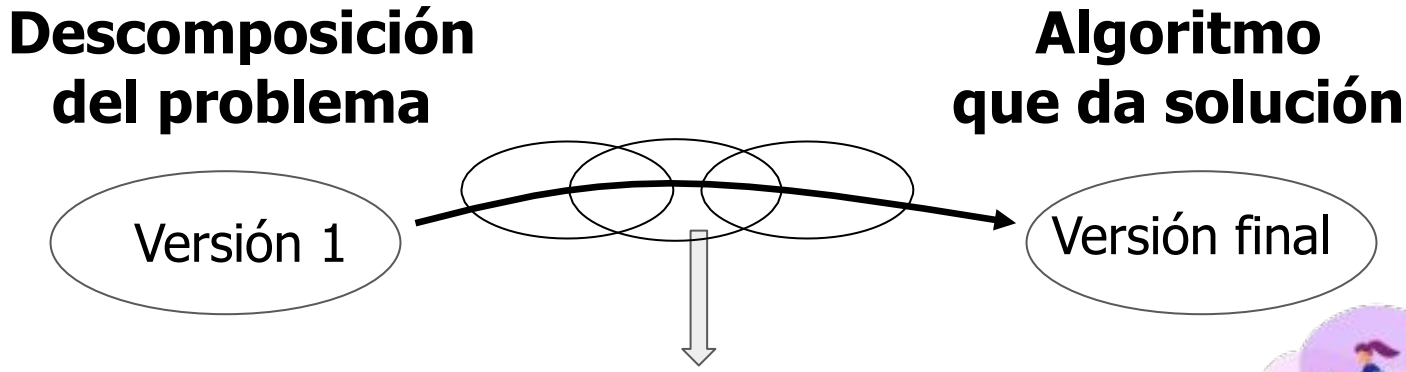
...

Etapa 2: Diseño y desarrollo del algoritmo

2. **Refinamiento sucesivo. Algoritmo que da solución a un problema.**

¿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.



Problema: es una situación a resolver.

Enunciado: descripción detallada de un problema a resolver.

Procesador: Toda entidad capaz de comprender las acciones primitivas de un algoritmo y ejecutar el trabajo indicado por el mismo.

Ambiente: El conjunto de todos los recursos necesarios para la ejecución de un algoritmo.

Algoritmo: conjunto de **acciones (pasos)** que debe realizar el procesador para resolver el problema planteado.

Ejemplo:

Problema: realizar un cálculo aritmético.

Enunciado: Encontrar la solución al siguiente cálculo:

(5+10+8) - 4 * 10 : 2.

Procesador: una persona que tenga los conocimientos para realizar las operaciones binarias +, - , * y :.

Ambiente: números enteros y operadores aritméticos.

Algoritmo:

5+10=15	Sumar 5+10+8
15+8=23	Restar 4
4*10=40	Multiplicar por 10
40:2=20	Dividir por 2
23-20=3	

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.

Tiene que tener

- **Entrada**: Todo algoritmo tiene **un comienzo**. Se considera como entrada el conjunto de **datos requerido** para resolver un problema dado.
- **Salida:** Todo algoritmo tiene **un fin**. La salida es un conjunto de resultados que se obtienen al aplicar el algoritmo al conjunto de datos de entrada.

Ejemplo:

Problema: realizar un cálculo aritmético.

Enunciado: Encontrar la solución al siguiente cálculo:

(5+10+8) - 4 * 10 : 2.

Procesador: una persona que tenga los conocimientos para realizar operaciones con +, - , * y :.

Ambiente: números enteros y operadores aritméticos.

Algoritmo:

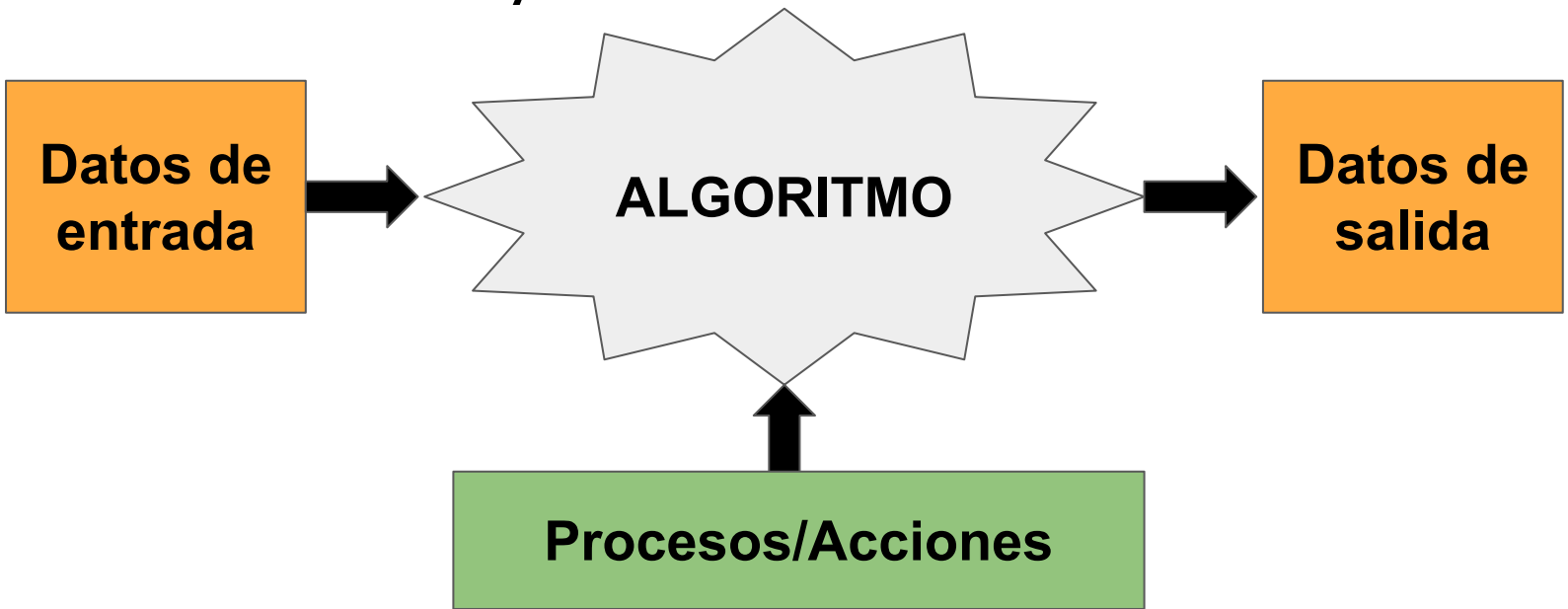
5+10=15	~~Sumar 5+10+8~~
15+8=23	~~Restar 4~~
4*10=40	~~Multiplicar por 10~~
40:2=20	~~Dividir por 2~~
23-20=3	

¿Qué se considera ACCIÓN?

Es un **evento** que modifica al **ambiente** establecido.

Una acción **no primitiva** debe ser descompuesta en acciones primitivas, para que pueda ser ejecutada por un procesador.

Una acción es **primitiva** si dicha acción pueda ser comprendida por el procesador sin mayor información adicional.

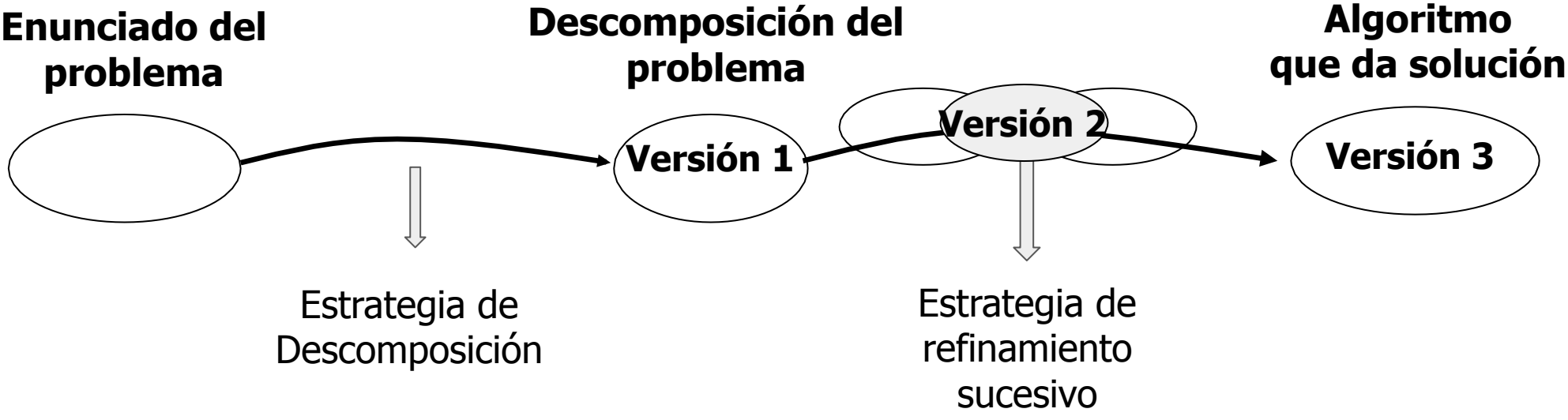


Para definir un algoritmo en la materia se deberá aplicar la estrategia o técnica de **refinamiento sucesivo**:

Versión 1: aplicar estrategia de descomposición del problema en tareas más simples.

Versión 2: desagregar las tareas de la versión 1 en acciones más simples.

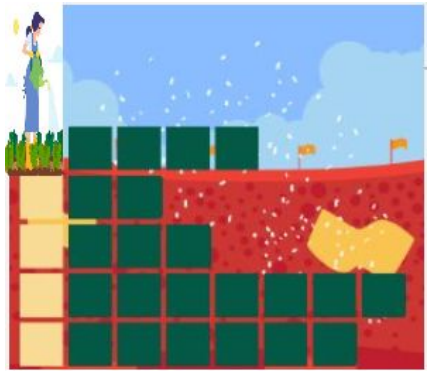
Versión 3: versión final, algoritmo que da solución al problema teniendo en cuenta las acciones primitivas dadas y los estados (inicial y final si se han dado).



Veamos un ejemplo:

Enunciado: La jardinera planta al final de cada camino un árbol.

Procesador: una persona que comprende órdenes.	**Acciones primitivas** Mover a la derecha. Mover a la izquierda. Pasar siguiente fila. Plantar el árbol. Volver al estado inicial.
Estado inicial: ubicación de la persona.	



Estado inicial

Estado final

Enunciado: La jardinera planta al final de cada camino un árbol.

Versión 1	Versión 2
T1: Recorrer fila 1 y plantar el árbol. **T2: Recorrer fila 2 y plantar el árbol.** **T3: Recorrer fila 3 y plantar el árbol.** **T4: Recorrer fila 4 y plantar el árbol.** **T5: Recorrer fila 5 y plantar el árbol.** **T6: Volver al inicio de la primera fila.**	T1.1. Mover 5 celdas a la derecha. T1.2: Plantar el árbol. T1.3: Volver al comienzo e ir a la siguiente fila T2.1. Mover 3 celdas a la derecha. T2.2: Plantar el árbol. T2.3: Volver al comienzo e ir a la siguiente fila.

Enunciado: La jardinera planta al final de cada camino un árbol.

Versión 1	Versión 2
T1: Recorrer fila 1 plantar el árbol. **T2: Recorrer fila 2 y plantar el árbol.** **T3: Recorrer fila 3 y plantar el árbol.** **T4: Recorrer fila 4 y plantar el árbol.** **T5: Recorrer fila 5 y plantar el árbol.** **T6: Volver al inicio de la primera fila.**	T1.1. Mover 5 celdas a la derecha. T1.2: Plantar el árbol. T1.3: Volver al comienzo e ir a la siguiente fila T2.1. Mover 3 celdas a la derecha. T2.2: Plantar el árbol. T2.3: Volver al comienzo e ir a la siguiente fila.

Actividad 4: Escribir la versión 2 de la tarea T3, enumerando correctamente. Ver como se escribieron versión 2 de las tareas T1 y T2.

Enunciado: La jardinera planta al final de cada camino un árbol.

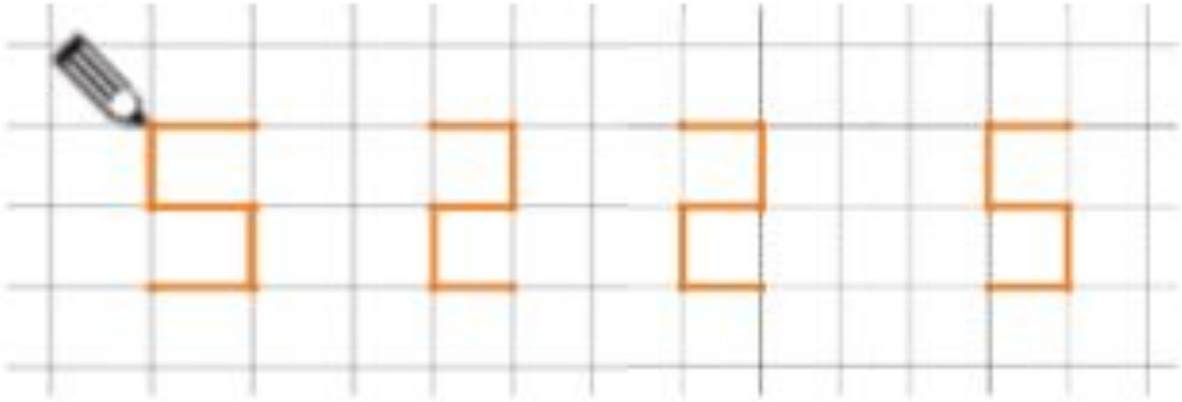
Versión 2	Versión final (Algoritmo con primitivas)
T1.1. Mover 5 celdas a la derecha. T1.2: Plantar el árbol. T1.3: Volver al comienzo e ir a la siguiente fila T2.1. Mover 3 celdas a la derecha. T2.2: Plantar el árbol. T2.3: Volver al comienzo e ir a la siguiente fila.	Mover a la derecha Mover a la derecha Mover a la derecha Mover a la derecha Mover a la derecha Plantar el árbol Mover a la izquierda Mover a la izquierda Mover a la izquierda Mover a la izquierda Mover a la izquierda Pasar siguiente fila ¿Y si hubiera una acción? **REPETIR** <número> **[** <acción primitiva> **]**

Enunciado: La jardinera planta al final de cada camino un árbol.

Versión 2	Versión final (Algoritmo con primitivas)
T1.1. Mover 5 celdas a la derecha. T1.2: Plantar el árbol. T1.3: Volver al comienzo e ir a la siguiente fila T2.1. Mover 3 celdas a la derecha. T2.2: Plantar el árbol. T2.3: Volver al comienzo e ir a la siguiente fila.	Mover a la derecha Mover a la derecha Mover a la derecha Mover a la derecha Mover a la derecha Plantar el árbol Mover a la izquierda Mover a la izquierda Mover a la izquierda Mover a la izquierda Mover a la izquierda Pasar siguiente fila ¿Y si hubiera una acción? **REPETIR** <número> **[** <acción primitiva> **]**

Actividad 5: Reescribir las dos partes del algoritmo (dentro del recuadro) reemplazando con la nueva acción REPETIR.

Actividad 6: Realizar Versión 2 a partir de la versión 1 realizada en la Actividad 3.



Versión 1 (Estrategia de Descomponer el problema)

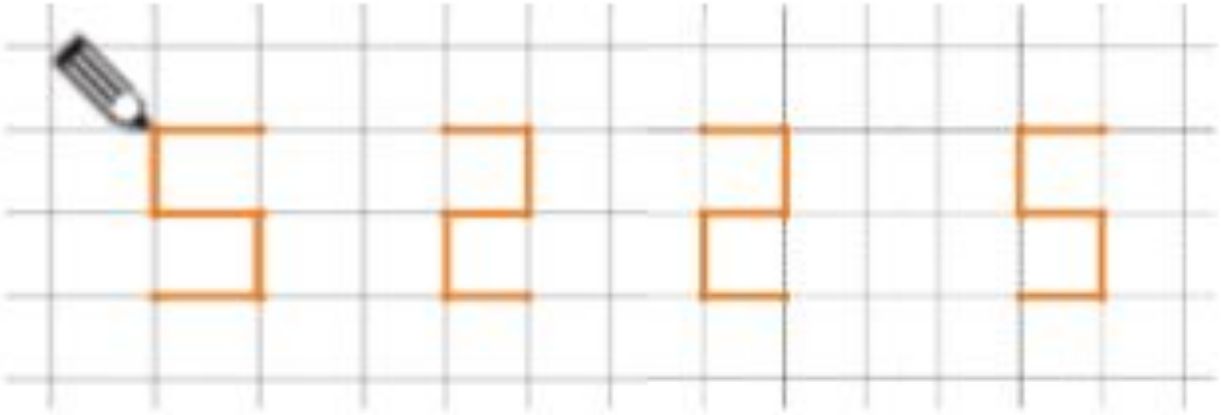
T1. Dibujar el número 5 y ubicarse para dibujar próximo dibujo.

...

Actividad 6: Realizar Versión 3 con las acciones primitivas, a partir de la versión 2 realizada.

Acciones primitivas	Qué significa?	
↑	moverse un cuadrado hacia arriba trazando una línea	
↓	moverse un cuadrado hacia abajo trazando una línea	
→	moverse un cuadrado hacia la derecha trazando una línea	
←	moverse un cuadrado hacia la izquierda trazando una línea	
□	Si la acción primitiva está dentro de este cuadro se desplaza pero no se traza línea.	

Acciones primitivas



Etapa 2: Diseño y desarrollo del algoritmo

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

1. Descomposición de un problema en tareas.
2. Refinamiento sucesivo. Algoritmo que da solución a un problema.
3. **Pseudocódigo y diagrama de flujo.**

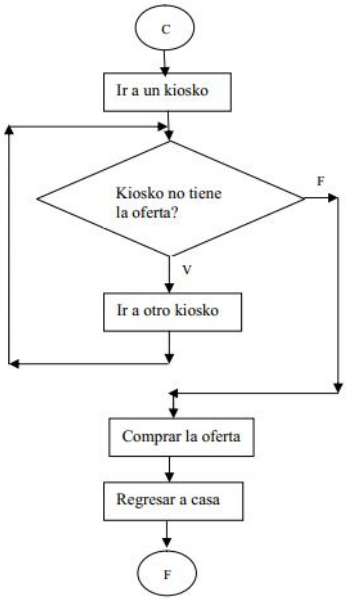
Etapa 2: Pseudocódigo y diagrama de flujo

CONTINUARÁ...en la Teoría 4

Representación NO GRÁFICA de un algoritmo

```
INICIO
    Entero: num, absnum ;
    Leer num;
    Si num<0
        absnum= num * (-1);
    Sino
        absnum=num;
    Fin_si
    Imprimir "El valor absoluto es: " absnum;
FIN
```

Representación GRÁFICA de un algoritmo



Etapa 2: Pseudocódigo y diagrama de flujo

CONTINUARÁ…en la Teoría 4

Pseudocódigo es usado para la **representación no gráfica** de algoritmos de la forma más detallada posible y a su vez lo más parecida posible a lenguaje que posteriormente su 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.
  - Refinamiento sucesivo. Algoritmo que da solución a un problema.
  - Representación de algoritmos: pseudocódigo y diagrama de flujo.

Teoría 3

Resolución de problemas

¡Ya podemos comenzar con el Práctico 3!