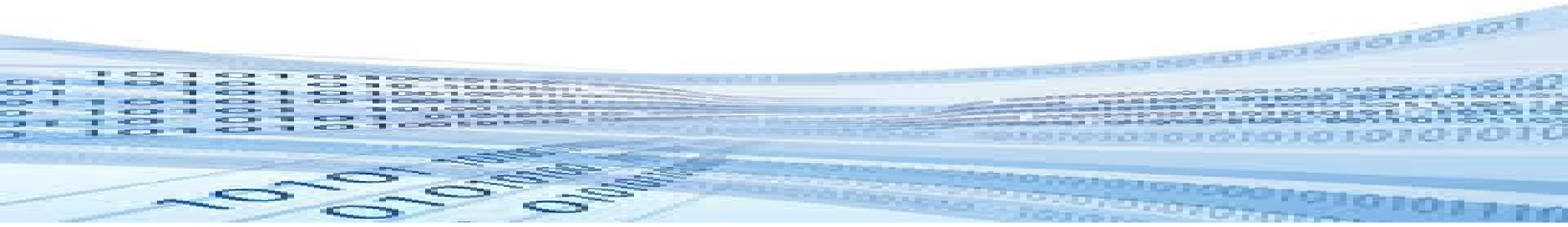


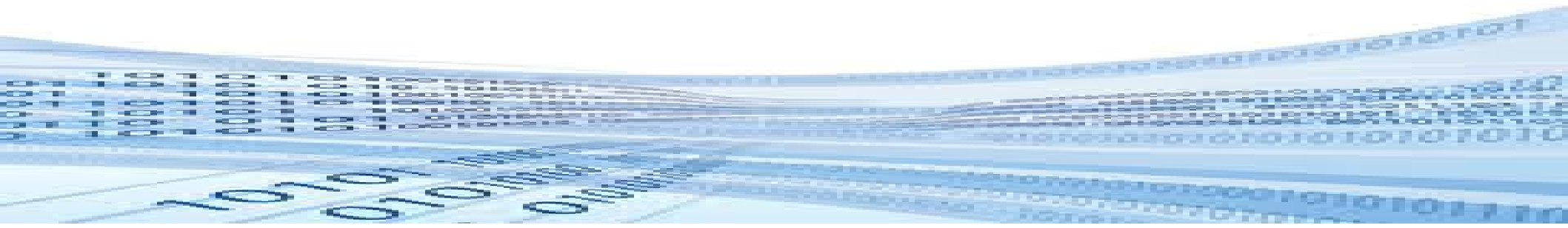
UD5.- Diseño de programas

Módulo: Programación
1.º DAM

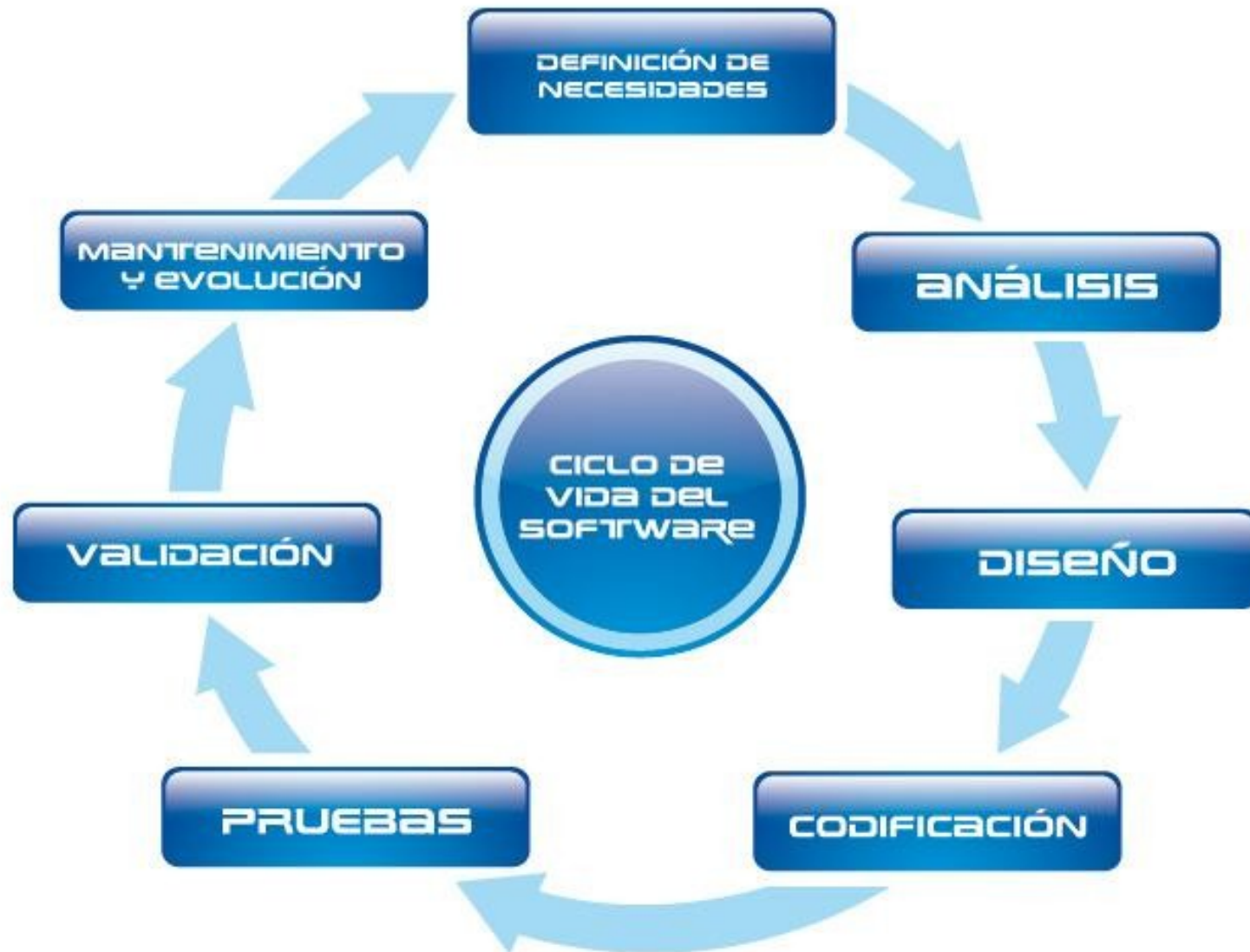


CONTENIDOS

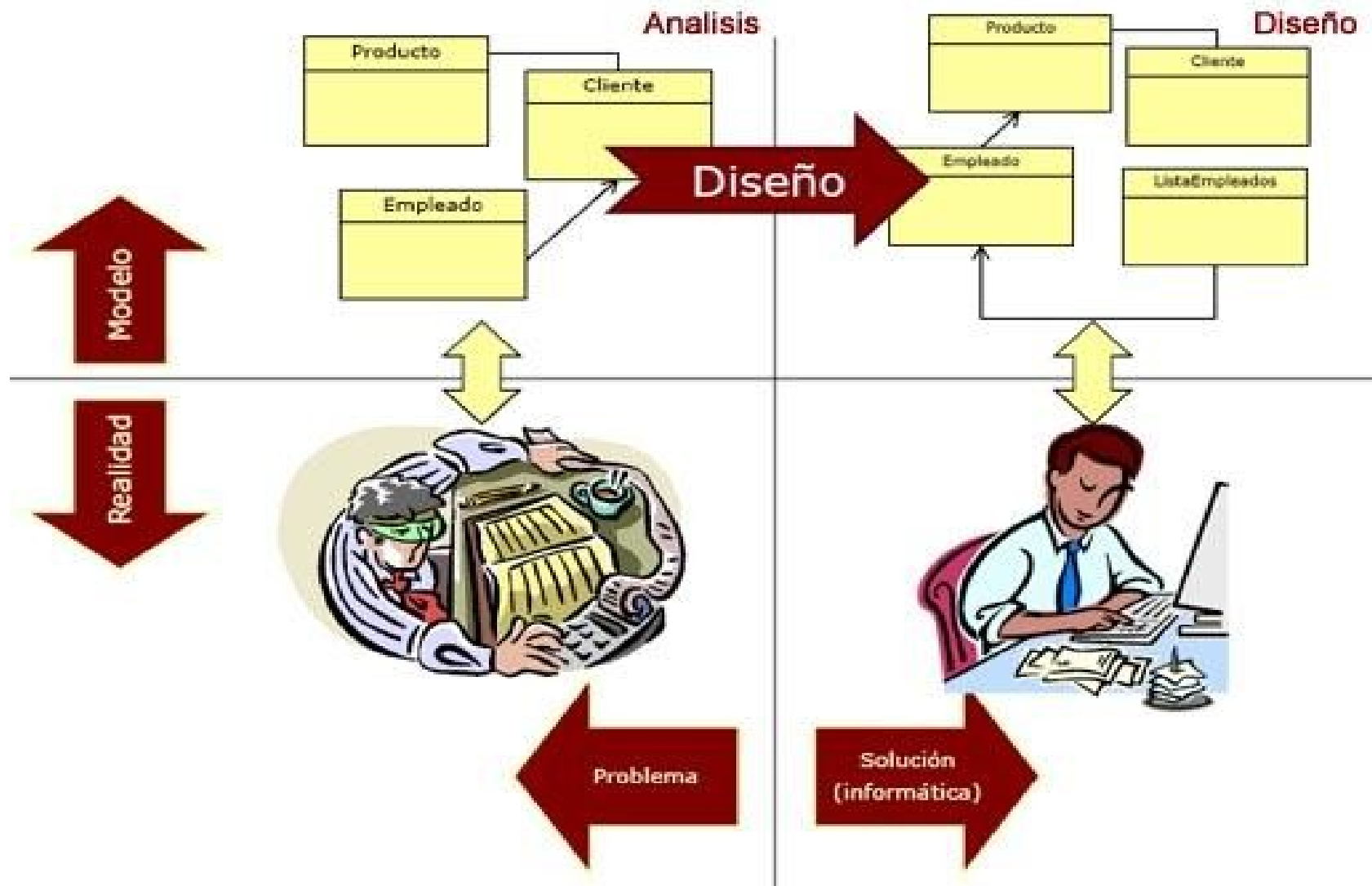
- Etapas de desarrollo del *software*
- Análisis y diseño OO
 - UML (Unified Modelling Language)
- Diseño de algoritmos
 - Diagramas de flujo
 - Pseudocódigo



Etapas de desarrollo del software

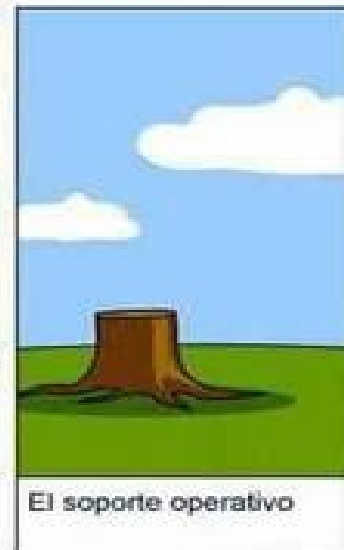
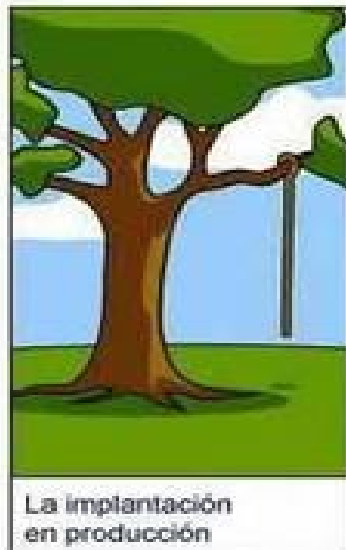
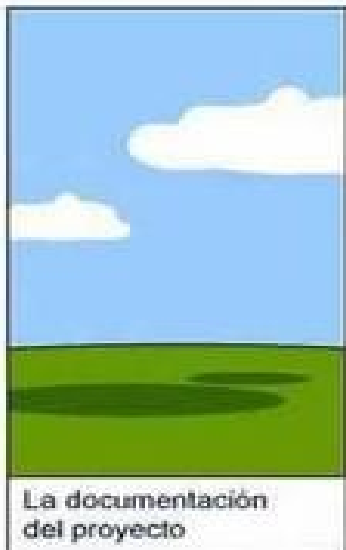
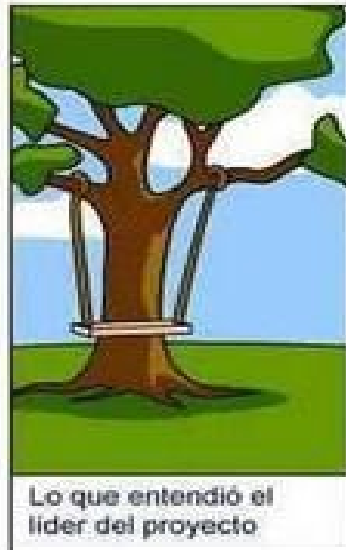


Análisis y diseño del software



¿Para qué sirve el análisis y el diseño?

Para EVITAR situaciones como esta...



Herramientas para el análisis y el diseño

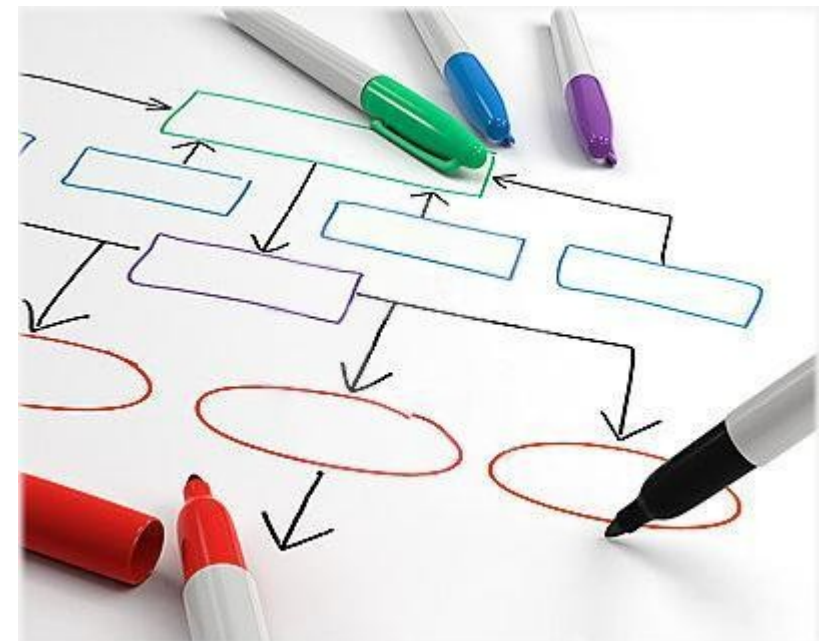
UML (*Unified Modelling Language*)

- **Diagrama de clases:** representa las clases
- **Diagrama de casos de uso:** representa los actores



Diseño de algoritmos:

- **Organigrama-Diagrama de flujo:** esquema que representa como funciona el algoritmo.
- **Pseudocodi:** código ordenado de soluciones.



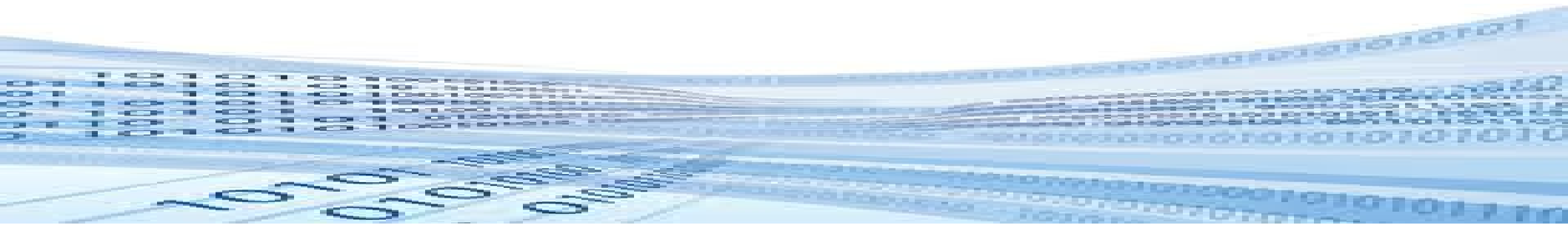
UML

- Lenguaje de modelado para describir procesos
- Ideal para modelar la Orientación a Objetos (OO)
- Tipo de diagramas **UML**:
 - Diagrama de clases
 - Diagrama de casos de uso
 - Diagrama de estados
 - Diagrama de actividades
 - Diagrama de secuencia
 - Diagrama de colaboraciones



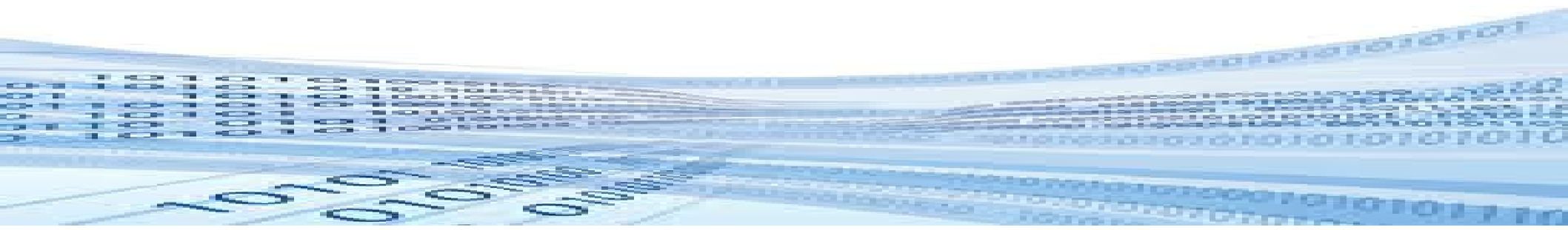
Pasos para desarrollar una CLASE

1. Averiguar datos sobre la clase y hacer una lista con sus variables de instancia y métodos
2. Dibujar el diagrama de clases – UML
3. Escribir el algoritmo de sus métodos
4. Escribir el código que hará el test de los métodos (llamamientos desde main())
5. Implementar la clase (escribir su código en JAVA)
6. Probar los métodos ejecutando el main()
7. Volver a rediseñar si fuera necesario



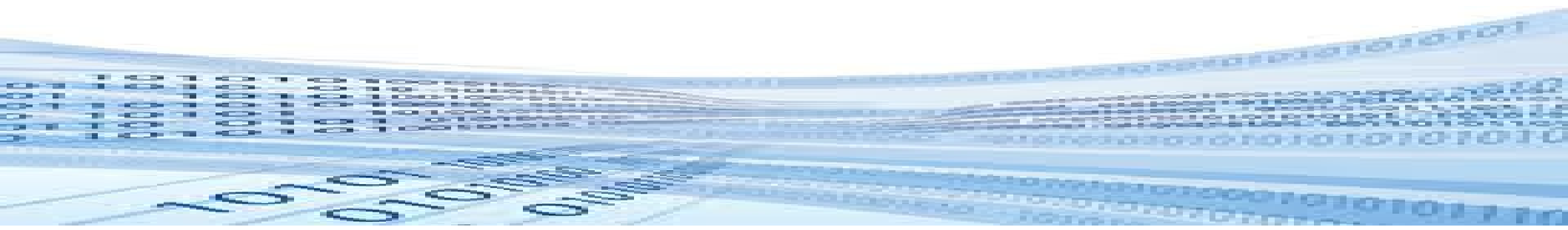
Diseñar un PROGRAMA en Java

1. Entender el enunciado del problema
2. Diseñar las clases necesarias, si es el caso
3. Diseñar el algoritmo de funcionamiento del programa principal
 1. Pseudocodigo
 2. Diagramas de flujo
4. Codificar los métodos de la clase y del programa en Java



¿Qué es un ALGORITMO?

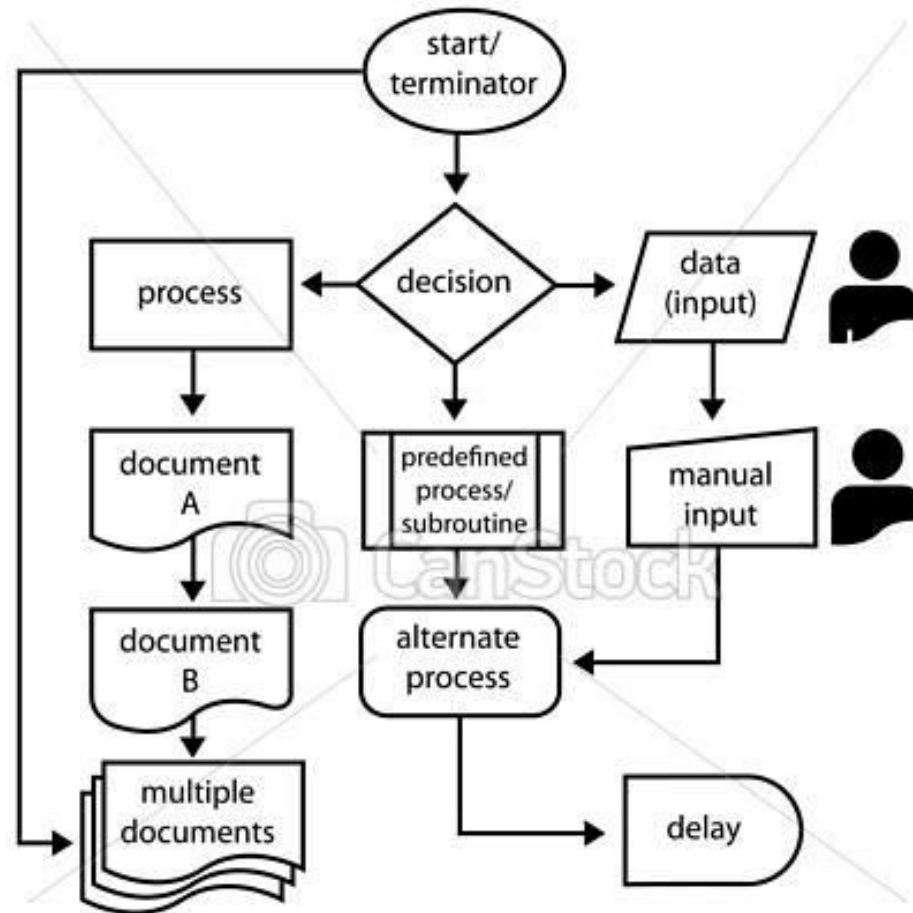
- Conjunto de acciones o secuencia de operaciones que, ejecutadas en un orden determinado, resuelven un problema.
- **Características:**
 - Tiene que ser concreto, muy definido y finito.
 - Independiente del lenguaje de programación.
- Qué es **implementar** o programar: adaptar el algoritmo al lenguaje de programación.



Diseño de algoritmos

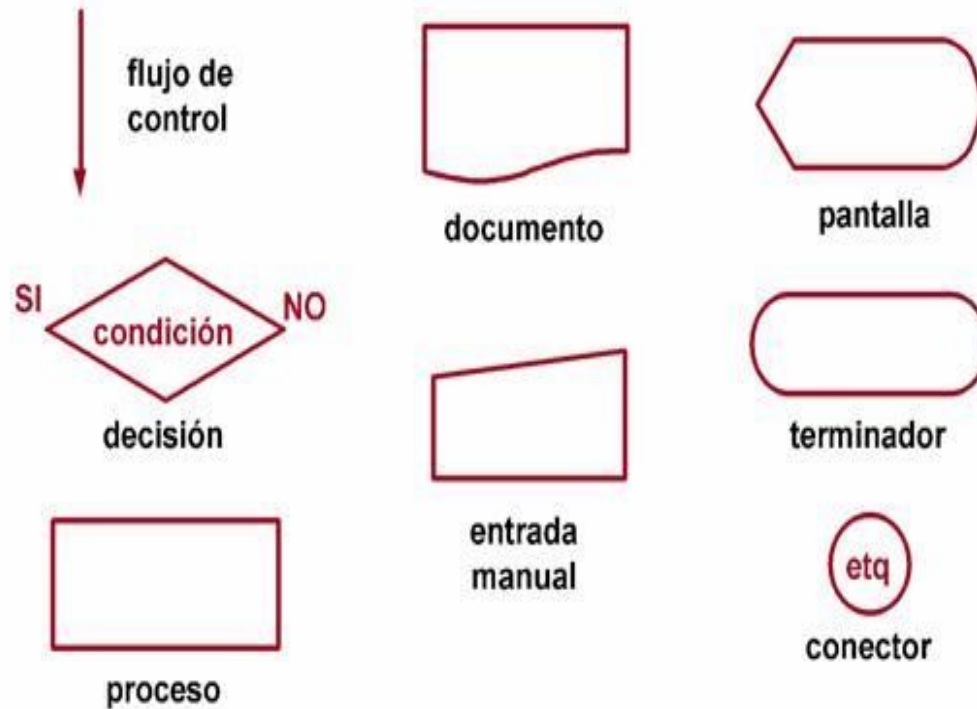
- **Pseudocodigo**
 - Lenguaje para expresar el flujo de ejecución de un programa.
- **Diagrama de flujo**
 - Esquema para representar como funciona un algoritmo.
 - Secuencia de pasos de la ejecución de un algoritmo.
- Son dos herramientas o modelos para representar el mismo: el **funcionamiento de un algoritmo**.

Elementos de un diagrama de flujo (I)



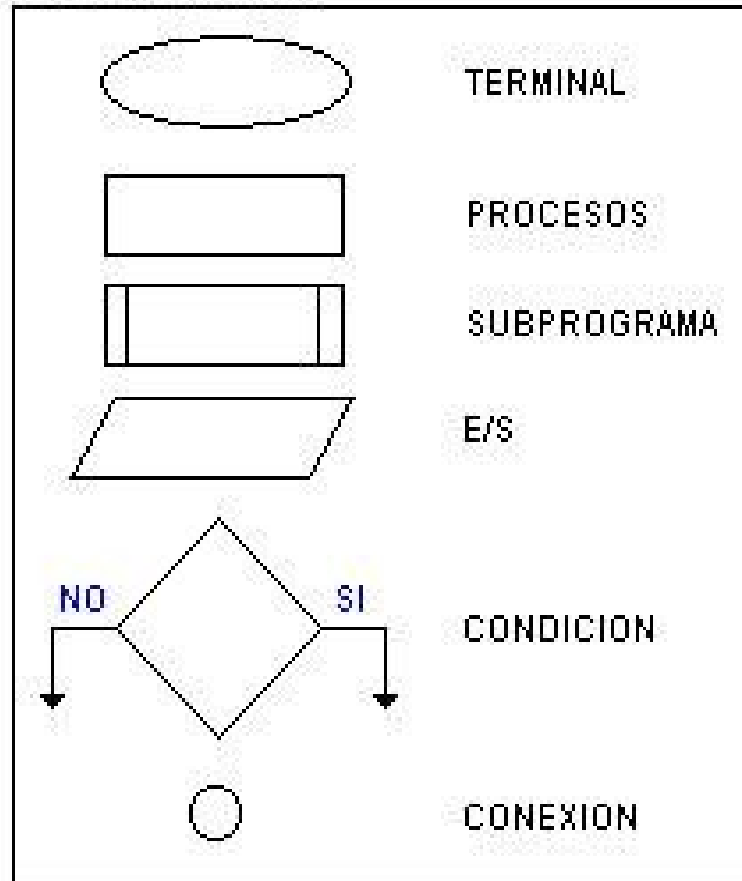
© Can Stock Photo - csp4239513

Elementos de un diagrama de flujo (II)



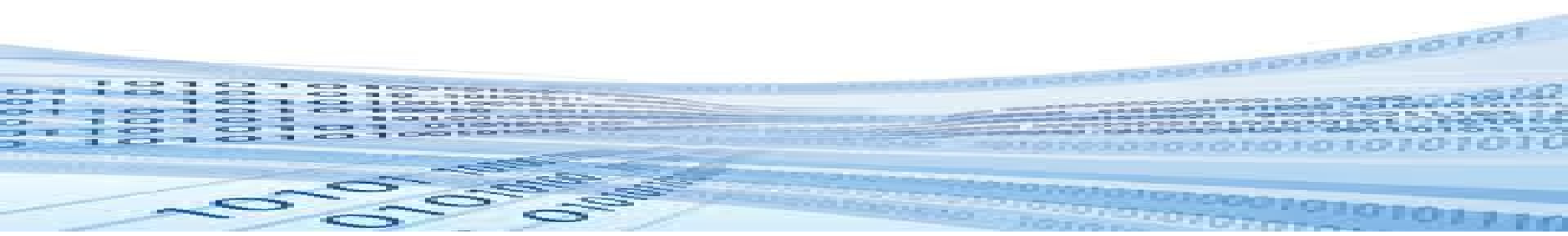
Elementos de un diagrama de flujo (III)

ORGANIGRAMAS

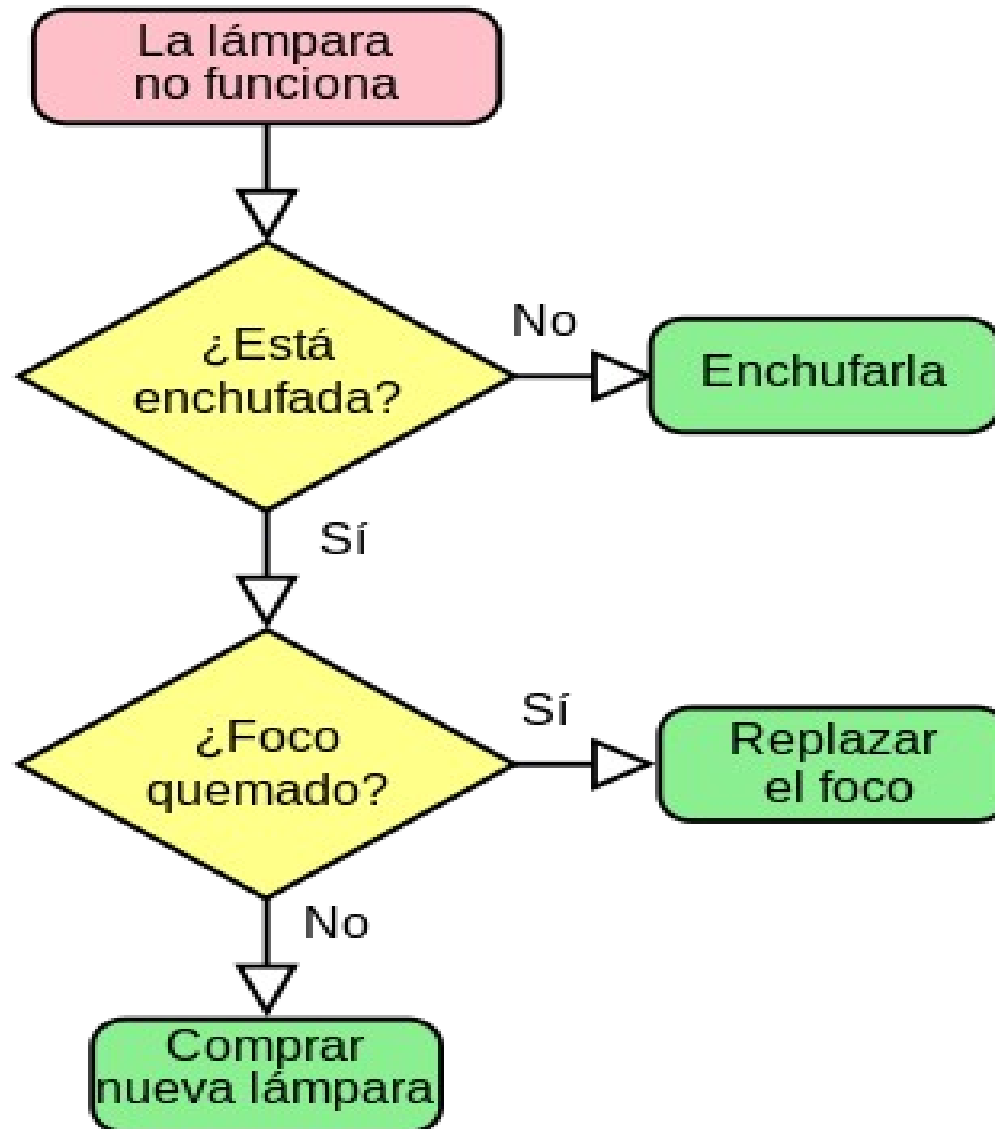


Diseño de algoritmos

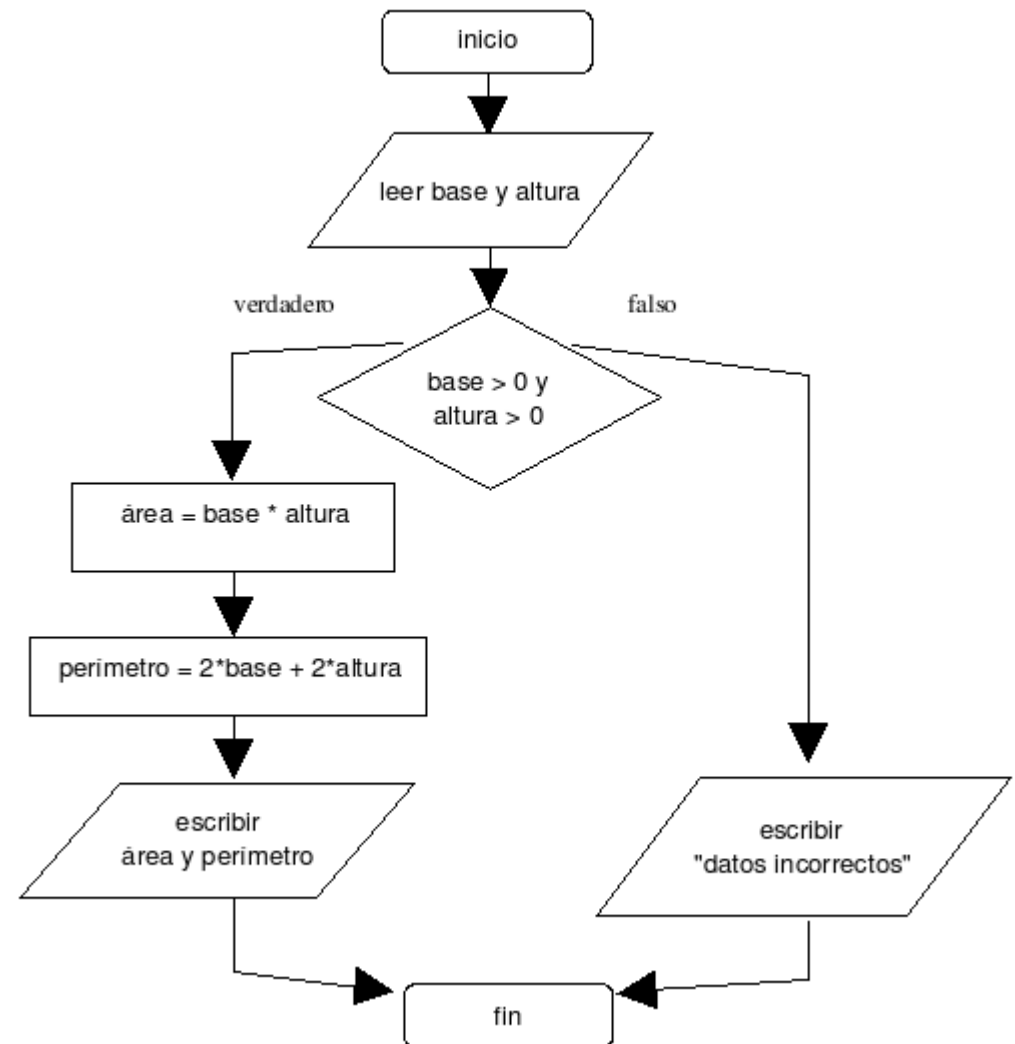
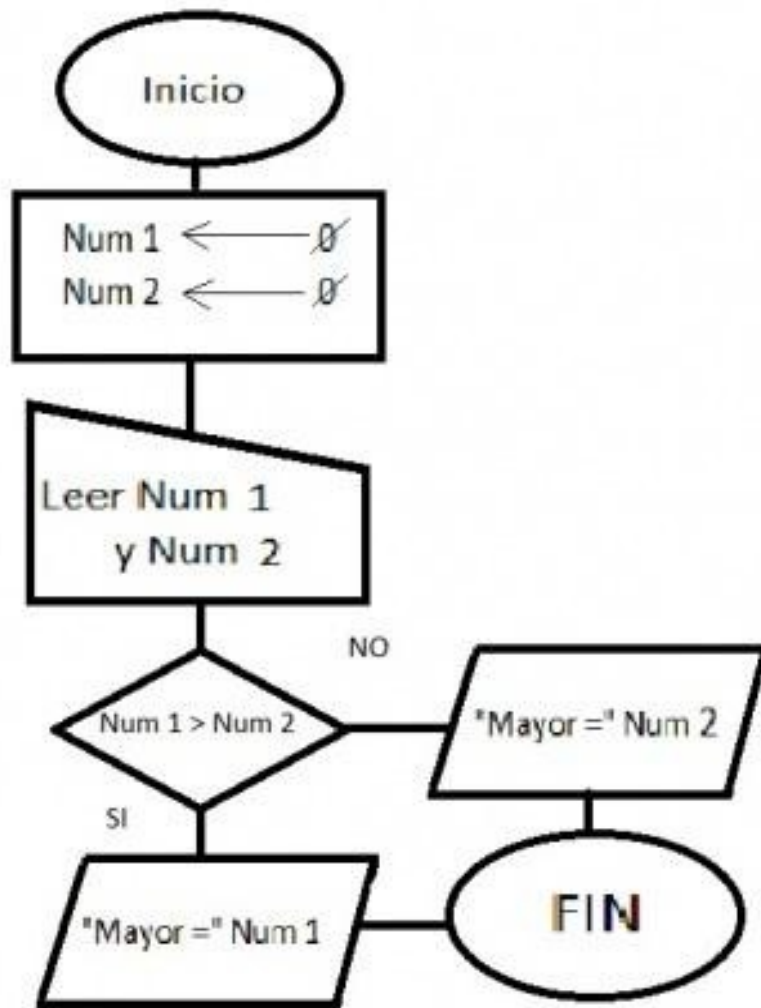
- Los **algoritmos**, normalmente, tienen los siguientes **elementos**:
 - Inicio
 - Entrada de datos
 - Proceso
 - Salida de datos
 - Fin



Ejemplo de diagrama de flujo



Qué hacen los siguientes algoritmos?



PSeInt – Herramienta para aprender lógica de programación

The screenshot displays the PSeInt software interface. The main window is divided into three sections: a code editor on the left, a central toolbar, and a flowchart area on the right.

Code Editor: The code is written in a pseudo-code style for calculating the average of a list of numbers. It includes comments in Spanish and uses standard programming constructs like loops and variables.

```
1 // Calcula el promedio de una lista de N datos
2 Proceso Promedio
3
4   Escribir "Ingrese la cantidad de datos:"
5   Leer n
6
7   i<-0
8   Repetir
9     i<-i+1
10    Escribir "Ingrese el dato ",i, ":"
11    Leer dato
12    acum<-acum+dato
13  hasta que i=n
14
15  prom<-acum/n
16  Escribir "El promedio es: ",prom
17
18 FinProceso
```

Execution Window (Ejecucion): This window shows the runtime status. It indicates that the execution started successfully but was interrupted due to an error. The error message is: "Lin 12 (inst 1): ERROR 215: Variable no inicializada (ACUM)".

Flowchart: The flowchart on the right illustrates the logic of the code. It starts with an initialization step $i \leftarrow 0$, followed by a loop that increments i and prompts the user to enter a data point. The data point is stored in $DATO$, and it is added to the accumulator $ACUM$. A decision diamond checks if $i = N$. If not (F), it loops back to the input step. If yes (V), it calculates the average $PROM \leftarrow ACUM / N$ and displays the result before reaching the **FinProceso** terminal.

Error Message (Ayuda Rapida): The bottom section provides a detailed explanation of the error: "ERROR 215: Variable no inicializada (ACUM)". It states: "Está intentando utilizar el valor de la variable señalada entre paréntesis para mostrar en pantalla o como parte de una expresión, pero esta variable no tiene ningún valor aún."

v20121013

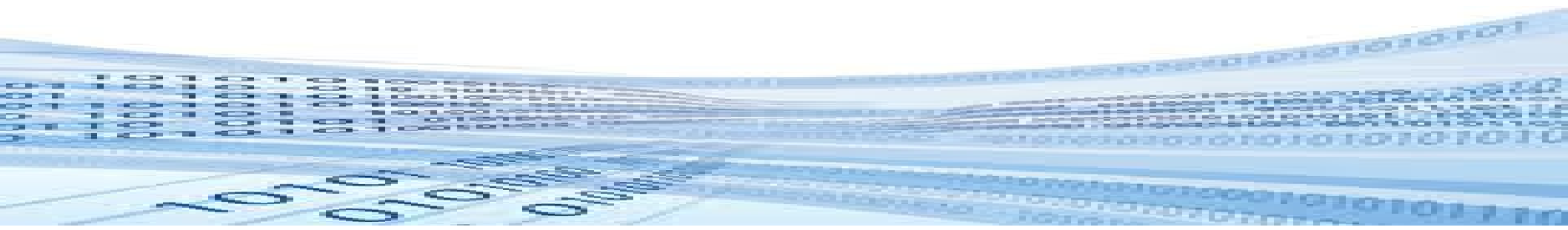
Ejemplos de pseudocodigo

```
PROGRAMA: MAYOR_AB
MÓDULO: PRINCIPAL
  INICIO
  DATOS:
    A, B NUMÉRICOS
  ALGORITMO:
    LEER A, B
    SI (A > B) . >
      ESCRIBIR "VERDADERO"
    FINSI
  FIN
```

```
PROGRAMA: NUMEROS_50
MÓDULO: PPAL
  INICIO
  DATOS:
    NUM NUMÉRICO
  ALGORITMO:
    ESCRIBIR "Numeros de 1 a 50:"
    NUM=1
    MIENTRAS (NUM <= 50) HACER
      ESCRIBIR NUM
      NUM=NUM+1
    FINMIENTRAS
  FIN
```

Prácticas con PSeInt

- Probar los algoritmos de la diapositiva 17 con PSeInt.
- Actividades guiadas
- Actividades1-7



Actividades guiadas (I)

- Un n.º perfecto es un entero positivo que es igual a la suma de todos los números enteros positivos (él excluido) que son divisores suyos.
 - Ejemplo: 6 es un número perfecto porque $6=1+2+3$
- Diseña e implementa un algoritmo para decidir si un número n (introducido por teclado) es o no perfecto.

Actividades guiadas (II)

- Diseñar e implementar un programa para calcular ecuaciones de 2.º grado.