



UNIVERSIDAD  
Popular del cesar

# Ingeniería de Sistemas

Programación de Computadores II



## UNIDAD 1:

# INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS

### 1.1. Paradigmas de la programación

1.2. Características de la Programación Orientada a Objetos

1.3. Conceptos de Programación Orientada a Objetos

1.3.1. Abstracción

1.3.2. Encapsulamiento

1.3.3. Polimorfismo

1.3.4. Herencia

1.3.5. Modularidad

1.4. Clases y Objetos

1.5. Mensajes y Métodos

1.6 Nociones de lenguaje de programación orientado a objetos

1.6.1 Generalidades

1.6.2 Sintaxis básica y estructuras de control

1.6.3 Vectores y matrices



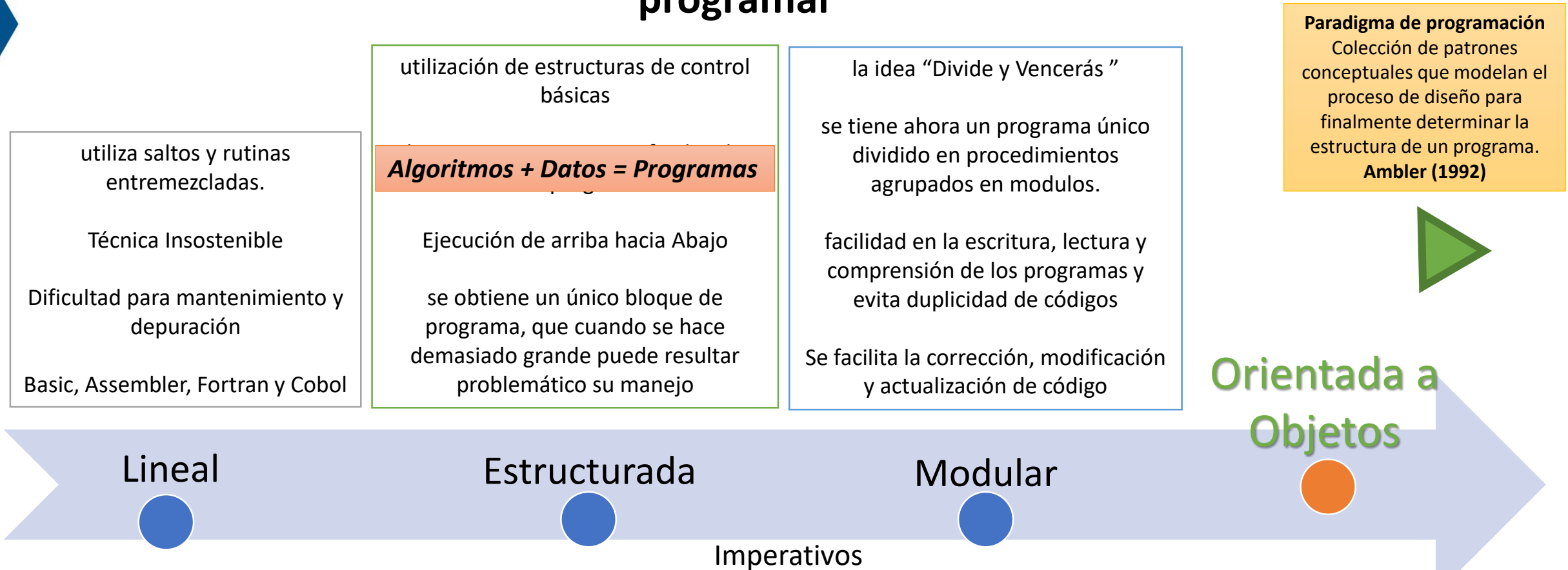
## UNIDAD 1:

### OBJETIVOS DE APRENDIZAJE

- Conocer la evolución de los paradigmas de programación, el propósito, y las características fundamentales de los principales paradigmas de programación.
- Comprender el concepto de Programación Orientada a Objetos, sus principales características, y su importancia dentro del contexto del desarrollo de software.
- Comprender los conceptos de abstracción, encapsulamiento, polimorfismo, herencia y modularidad, y su importancia dentro del paradigma de la orientación a objetos.
- Comprender los conceptos de Clases y Objetos, su relación e importancia dentro del paradigma orientado a objetos.
- Comprender los conceptos de Mensaje y Método, su relación e importancia dentro del paradigma orientado a objetos.



# Paradigmas de Programación: Evolución de las técnicas de programar



Toda la evolución de los lenguajes y técnicas de programación se ha desarrollado en base a una sola idea conductora: **hacer que la tarea de realizar programas para ordenadores sea cada vez lo más simple, flexible y portable posible.**



## Programación orientada a objetos: que es?, concepto y características.

La POO es un modelo de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. La idea central es simple: organizar los componentes del programa a imagen y semejanza de los objetos en el mundo real

**Fuente:** Ceballos, Sierra, Francisco Javier. Programación orientada a objetos con C++ (4a. ed.), RA-MA Edi.2007

- Define los programas en términos de "clases de objetos"
- Los elementos básicos son: objetos, mensajes, métodos y clases.
- Cambia el interrogante **¿Qué hace este programa?**, a otro interrogante **¿Qué objetos del mundo real debe modelar?**
- Ofrece mecanismos que facilitar la reutilización de código, es decir, facilita crear componentes de software reutilizables.
- Crear sistemas fiables que sean flexibles, mantenibles y capaces de evolucionar para cumplir las necesidades de cambio

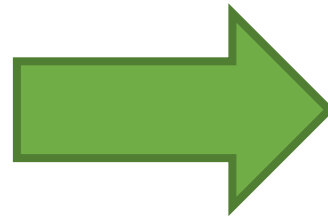
Técnica de programación mas utilizada en la actualidad.





## Programación orientada a objetos: elementos o propiedades del paradigma orientado a objetos?

POO



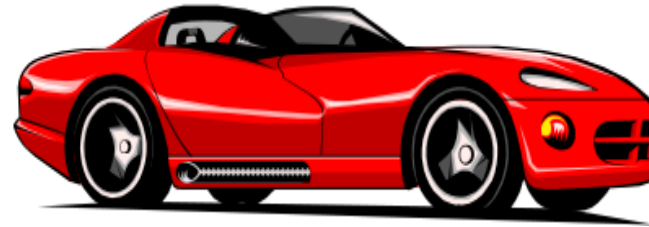
Los conceptos y herramientas orientados a objetos son tecnologías / mecanismos que permiten que los problemas del mundo real puedan ser expresados de modo fácil y natural en código.



# Programación orientada a objetos: que es la Abstracción?



Definir una abstracción significa describir una entidad del mundo real (objeto), no importa lo compleja que pueda ser, seleccionando las características y comportamientos esenciales y a continuación utilizar esta descripción en un programa.



## TAD / Clase

### Carro

**String** marca  
**int** modelo  
**int** color  
**int** velMax  
**int** matricula

Encender ( )  
Frenar ( )  
Acelerar ( )



# Programación orientada a objetos: Ejercicio sobre abstracción?



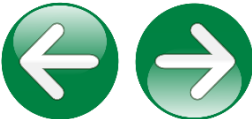
Televisor
ATRIBUTOS
COMPORTAMIENTOS

TAD



Reloj
ATRIBUTOS
COMPORTAMIENTOS

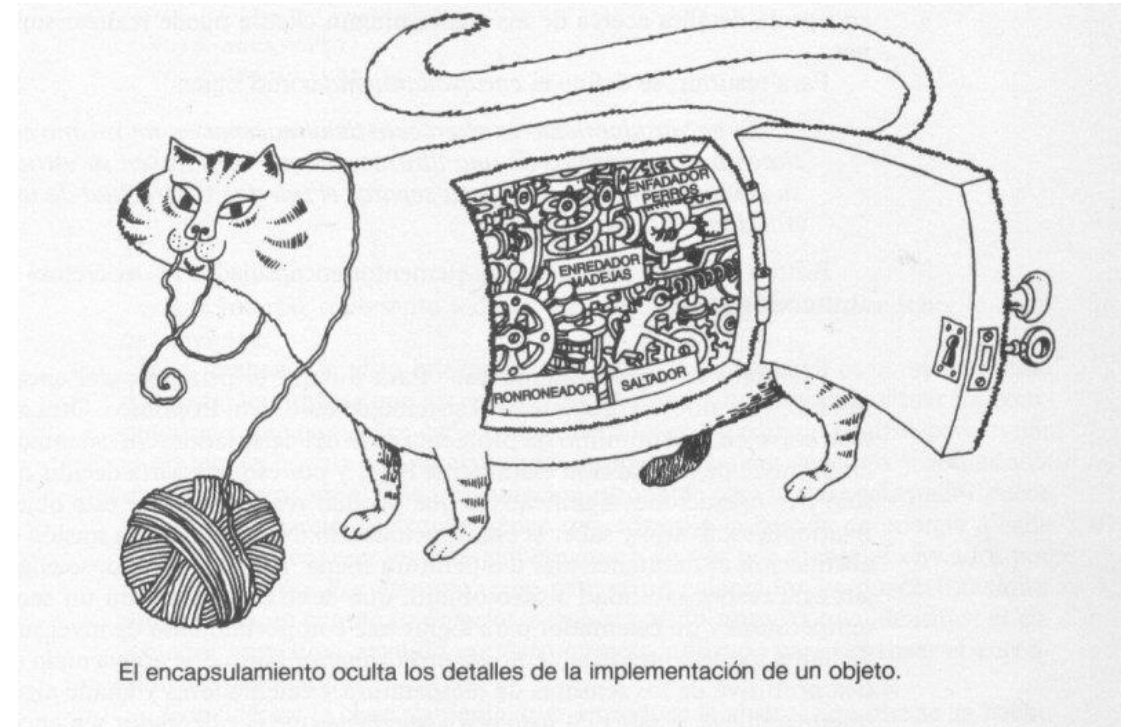
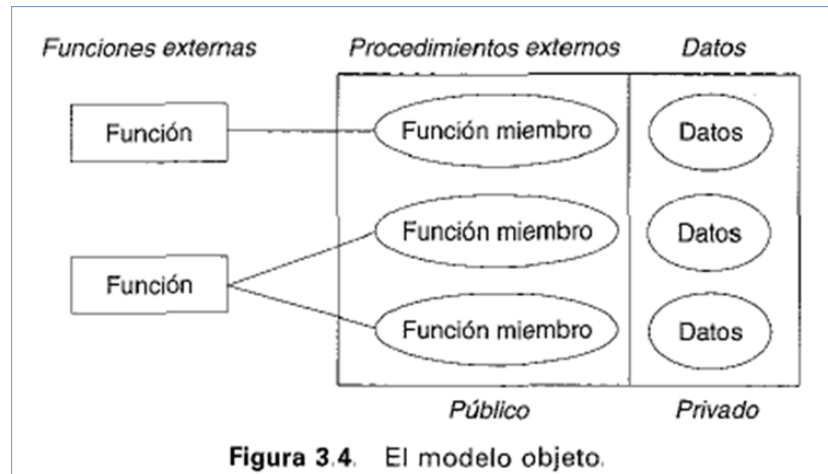
TAD





## Programación orientada a objetos: que es el encapsulamiento?

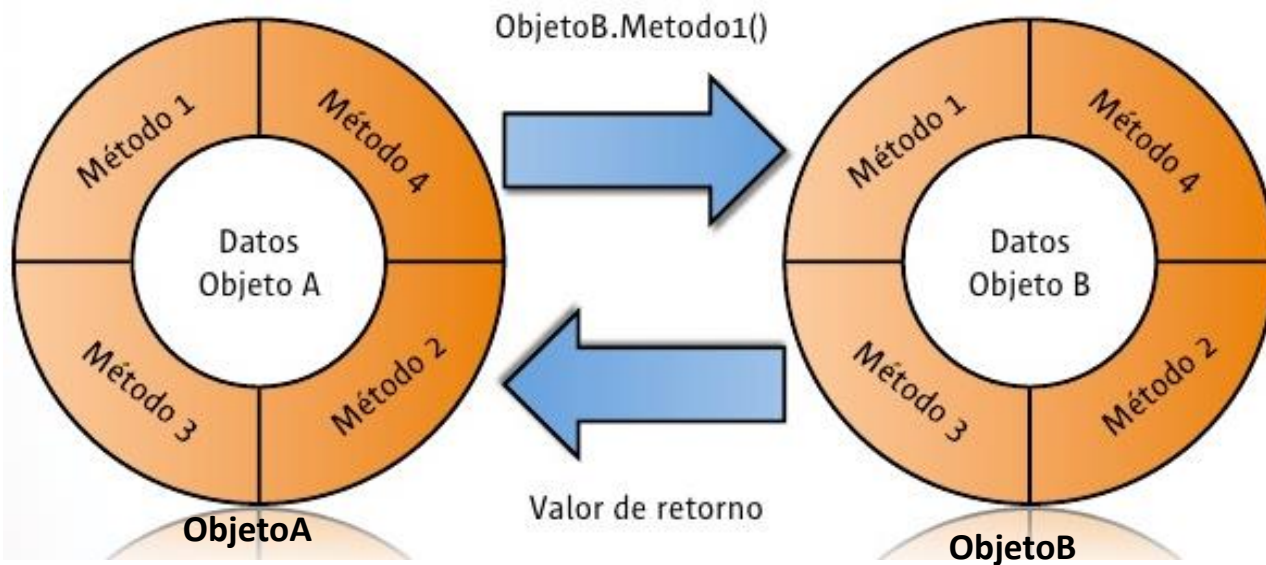
El **encapsulamiento** es la propiedad que permite asegurar que el contenido de información de un objeto está oculta para los otros objetos: **El objeto A no conoce los datos de un objeto B, y viceversa.**



Ocultación de  
Información



## Programación orientada a objetos: que es el encapsulamiento?



En la practica la **encapsulación** da lugar a que las clases se dividan en dos partes:

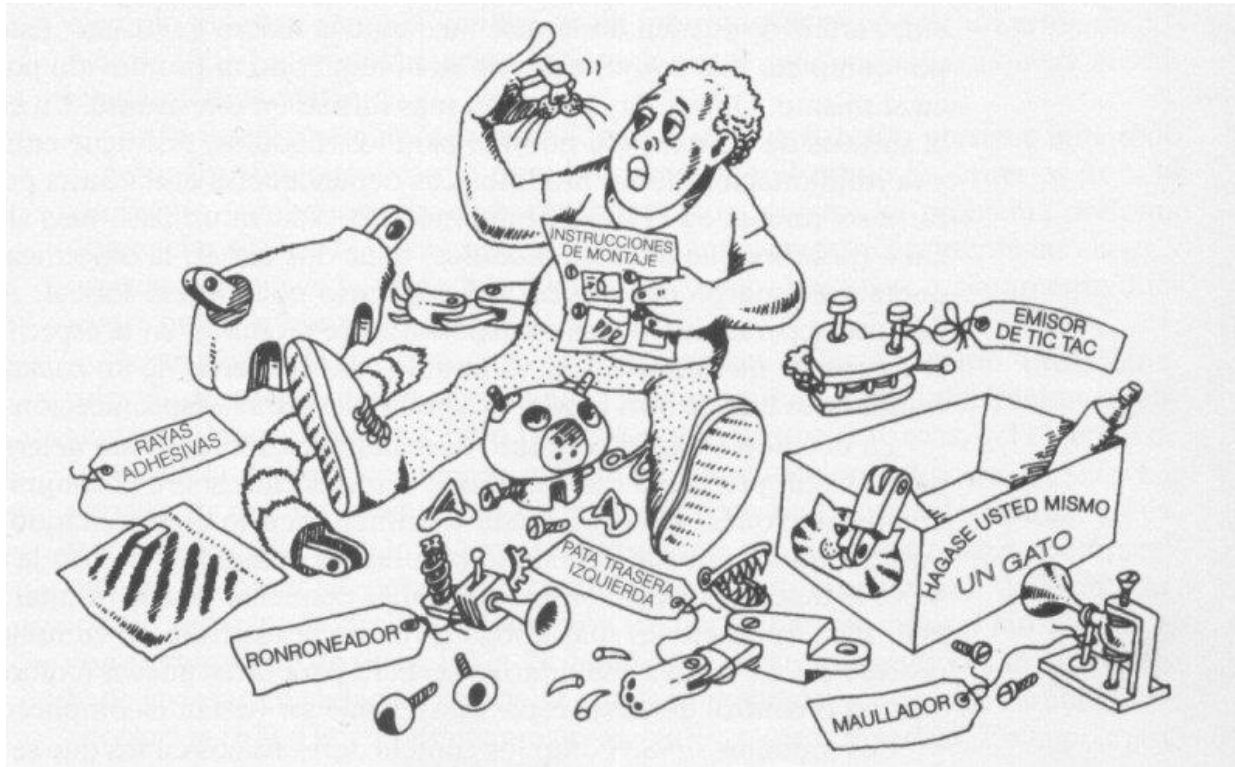
**Interface:** captura la visión externa de una clase, abarcando la abstracción del comportamiento común a los ejemplos de esa clase.

**Implementación:** comprende la representación de la abstracción, así como los mecanismos que conducen al comportamiento deseado.

Con la **encapsulación**, las clases son vistas como cajas negras, es decir que se conoce lo que hace la clase pero no la forma en que lo hace.



## Programación orientada a objetos: la modularidad?



La modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

**Débilmente Acoplados y  
Altamente cohesivos**



## Programación orientada a objetos: la modularidad?

El **Acoplamiento** es la medida del grado de interdependencia entre módulos.



El objetivo es minimizar el acoplamiento entre módulos

Un módulo **cohesivo** ejecuta una tarea sencilla de un procedimiento de software



Un módulo cohesivo sólo hace (idealmente) una cosa

El **bajo acoplamiento** y la **alta cohesividad** de los módulos permite garantizar que se puedan ensamblar fácilmente para construir una aplicación completa.

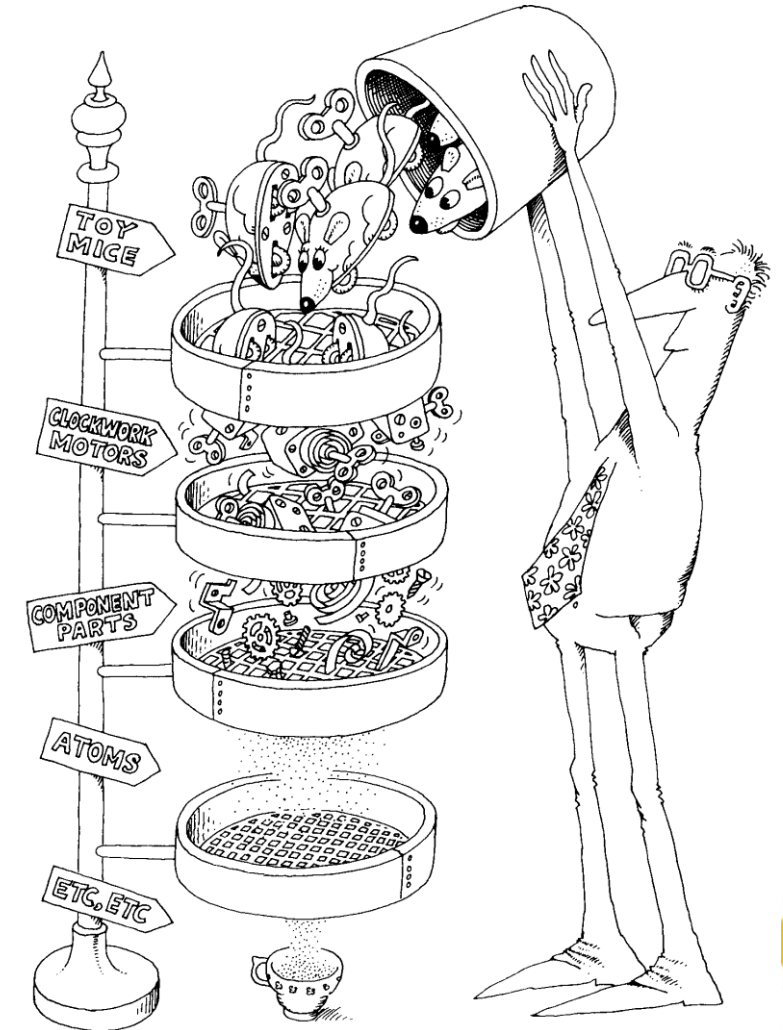




## Programación orientada a objetos: jerarquía?

La jerarquía es una propiedad que permite una ordenación de las abstracciones. Las mas importantes son:

- Estructura de clases (**herencia**)
- Estructura de objetos (**asociaciones**)

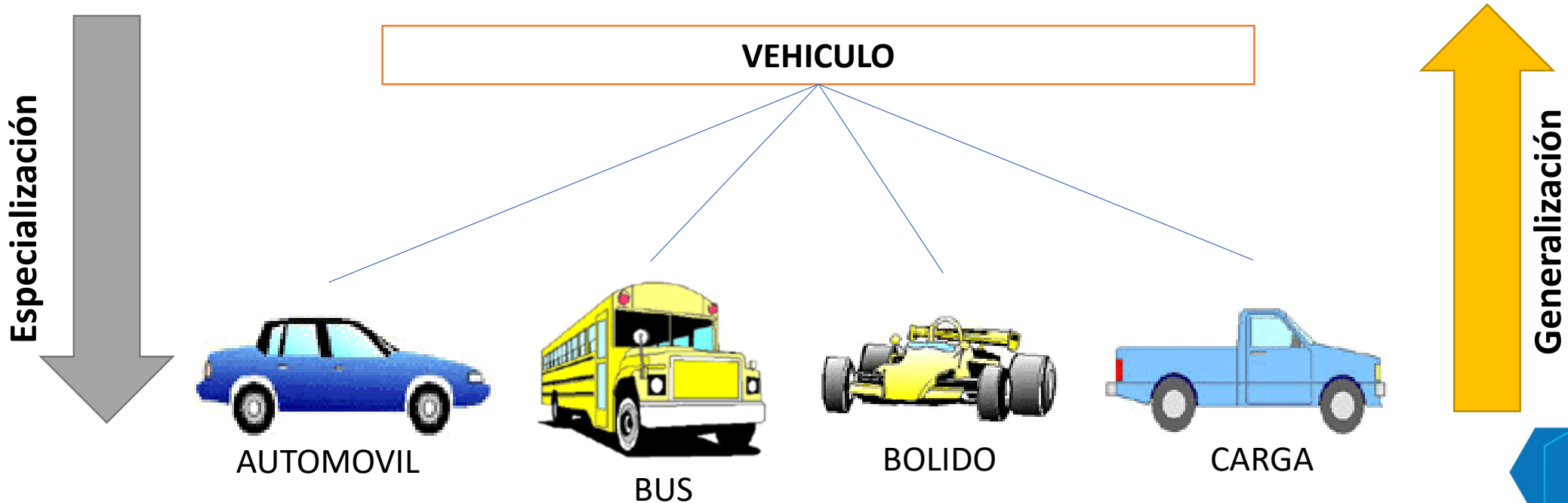




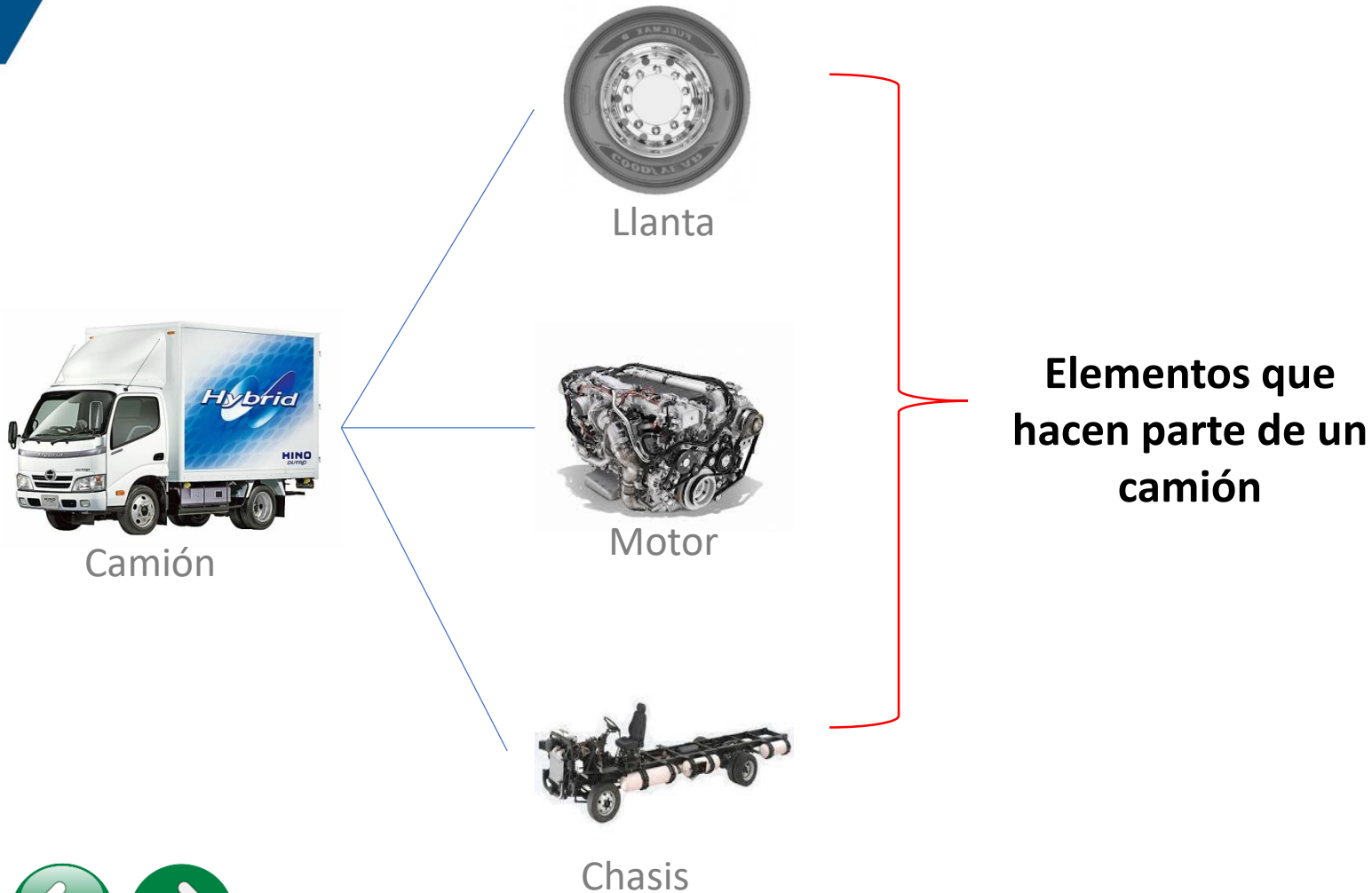
## Programación orientada a objetos: jerarquía?

Las jerarquías de **generalización/especialización** se conocen como HERENCIA la cual define una relación entre clases, en donde una clase comparte la estructura o comportamiento definido en una o más clases.

**“Es un tipo de”**



## Programación orientada a objetos: jerarquía?



La **asociación** es el concepto que permite el agrupamiento físico entre objetos.

**"Hace parte de"**



# Programación orientada a objetos: ejercicio de jerarquía?

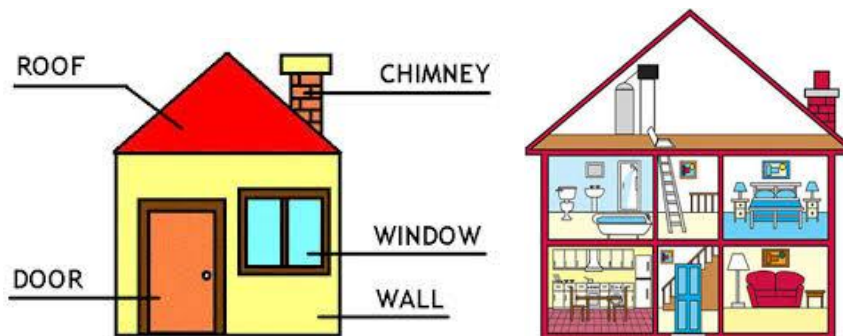


**Herencia**

Son un tipo de



**Vivienda**



**Asociacion**

Hacen parte de



# Programación orientada a objetos: ejercicio de jerarquía?



bicicleta



Dirección



Automóvil



Frenos



Lámpara



Llanta



Avión

**Vehículo**



**Agregación**

**Herencia**

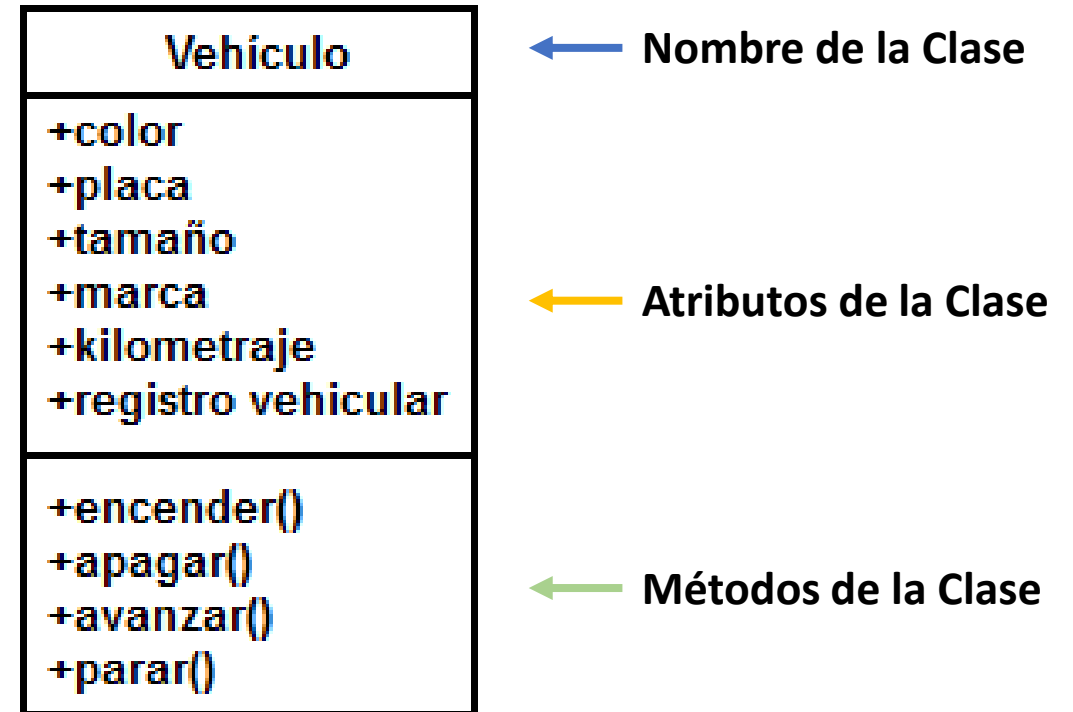


# Programación orientada a objetos: clases y objetos ?

## QUE ES UNA CLASE?

Una clase es la descripción de un conjunto de objetos; consta de métodos y datos que resumen características comunes de un conjunto de objetos.

Una clase es una plantilla utilizada para la creación de objetos. Define los atributos y métodos que contendrán los objetos que se creen a partir de ella.



Representación grafica de una clase mediante UML





# Programación orientada a objetos: clases y objetos ?

## QUE ES UN OBJETO?

Un objeto es un ejemplar creado de una CLASE. Cada vez que se construye un objeto de una clase, se crea una instancia de esa clase.

Vehículo
+color +placa +tamaño +marca +kilometraje +registro vehicular
+encender() +apagar() +avanzar() +parar()

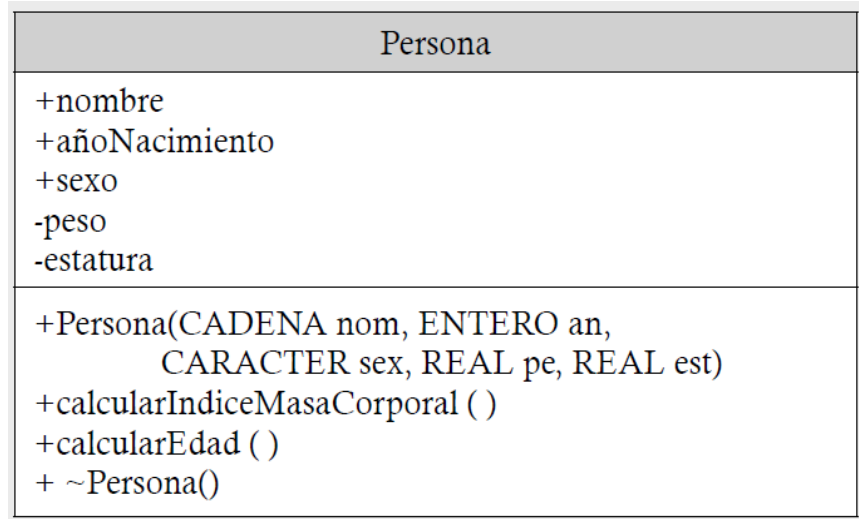
Clase



Un objeto es cada instancia de una clase



# Programación orientada a objetos: clases y objetos ?



**Clase**



Objeto1: Persona

nombre=SAMUEL  
 añoNacimiento=2010  
 sexo=H  
 peso=13.7  
 estatura=90.5

Persona(CADENA nom, ENTERO an,  
                   CHARACTER sex, REAL pe, REAL est)  
 calcularIndiceMasaCorporal( )  
 calcularEdad( )  
 ~Persona()

Objeto 1



Objeto2: Persona

nombre=MARÍA ISABEL  
 añoNacimiento=2002  
 sexo=M  
 peso=45  
 estatura=140

Persona(CADENA nom, ENTERO an,  
                   CHARACTER sex, REAL pe, REAL est)  
 calcularIndiceMasaCorporal( )  
 calcularEdad( )  
 ~Persona()

Objeto 2

**Diagrama de objetos en UML**



# Programación orientada a objetos: ejercicio clases y objetos ?

Asignatura
código: <b>String</b> nombre: <b>String</b> noCréditos: <b>int</b> habilitable: <b>boolean</b>
obtenerCodigo( ) obtenerNombre( )



ObjA: \_\_\_\_ ?

longRadio: 15,5  
color: "rojo"

obtenerArea()  
obtenerLongitud()

ObjB: \_\_\_\_ ?

longRadio: 20  
color: "Azul"

obtenerArea()  
obtenerLongitud()



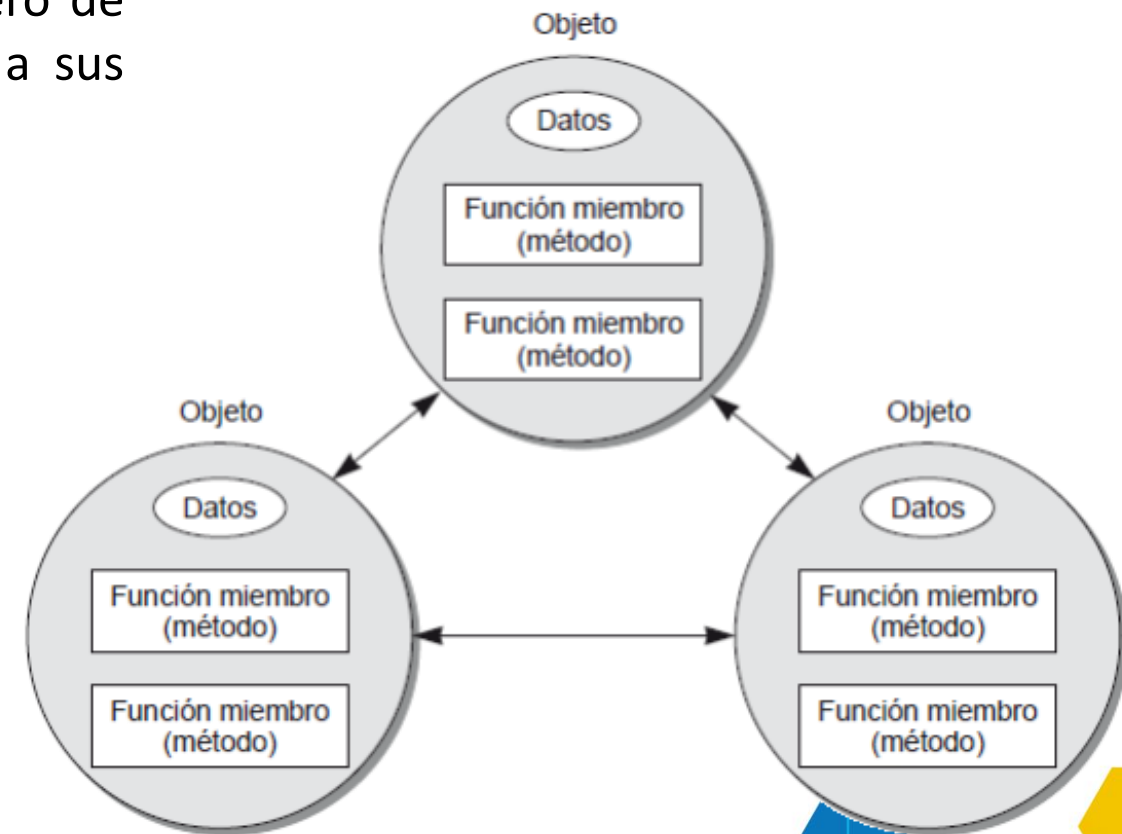
## Programación orientada a objetos: métodos y mensajes?



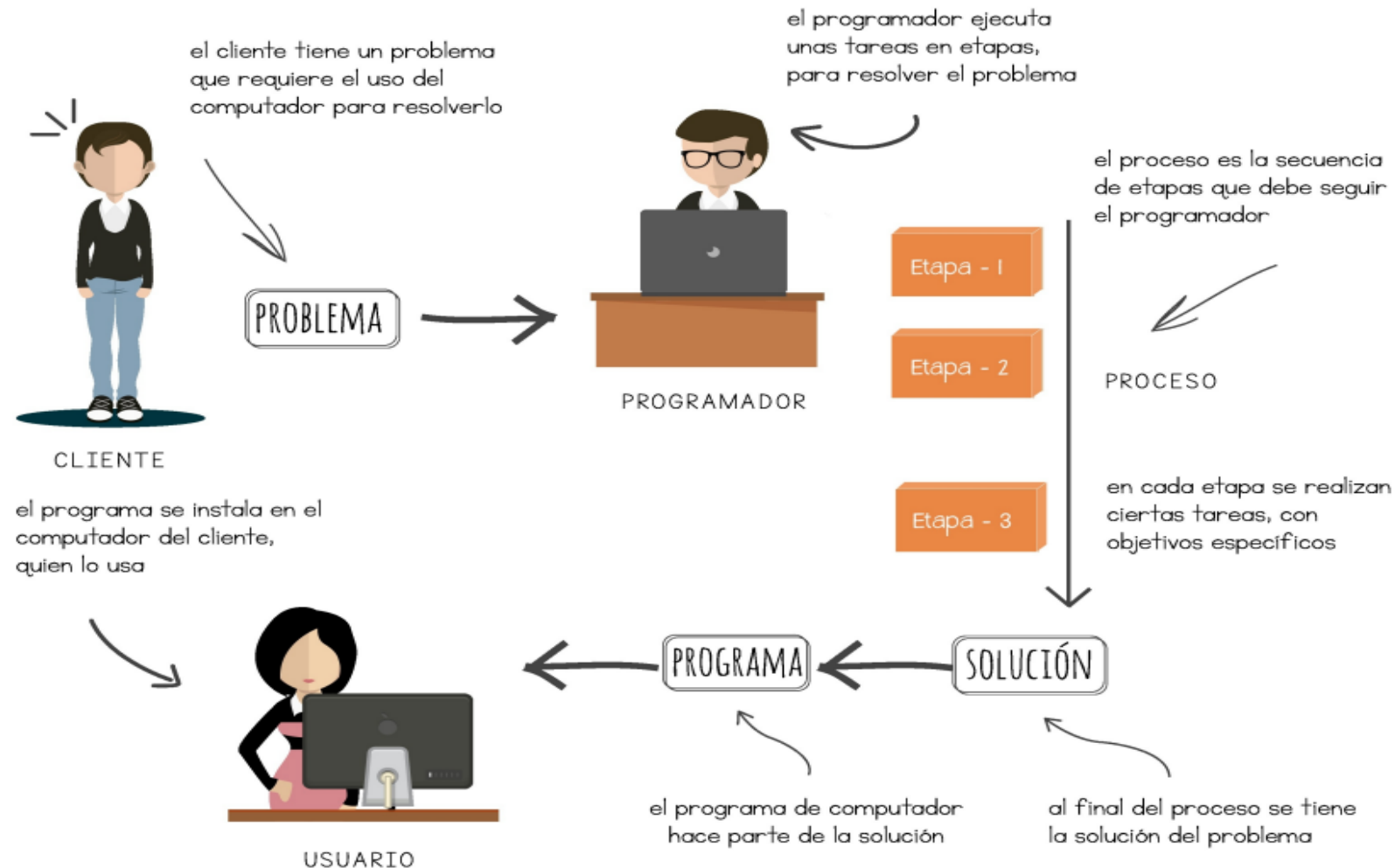
Un programa orientado a objetos consiste en un número de objetos que se comunican unos con otros invocando a sus métodos.

Un **método** es el procedimiento o función que se invoca para actuar sobre un objeto.

Un **mensaje** es la indicación de la acción que debe hacer un objeto, es decir, la ejecución de uno de sus métodos.



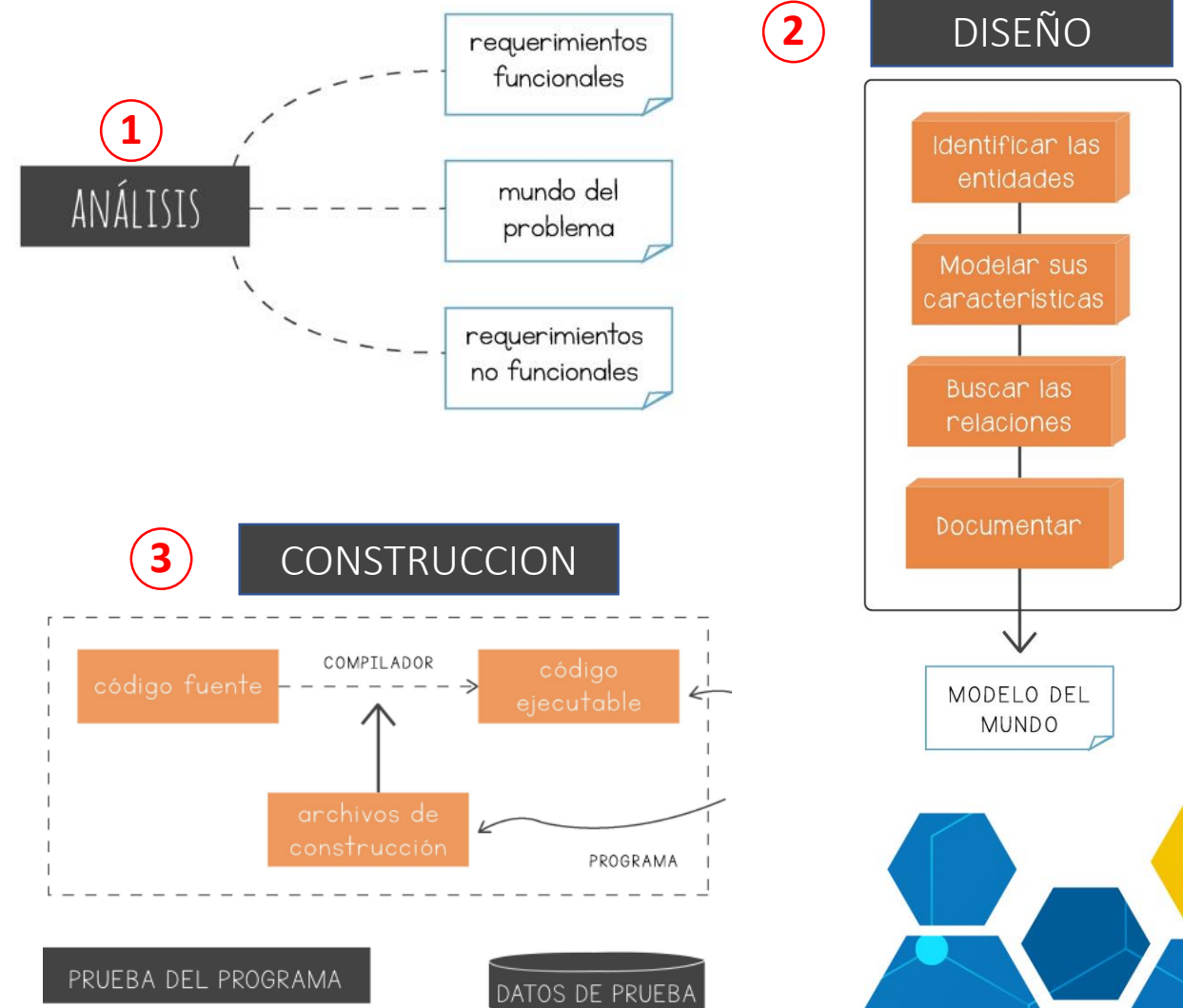
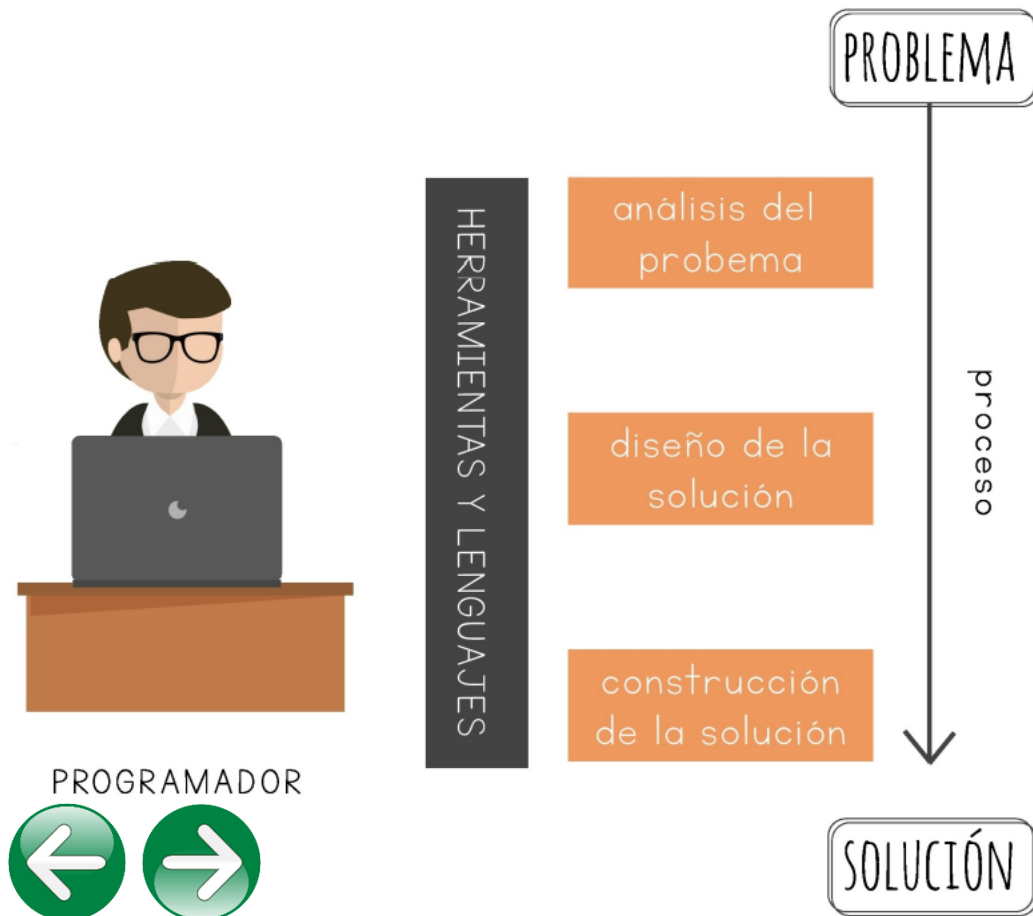
# Programación orientada a objetos: etapas para la solución de problemas en el enfoque orientado a objetos?





# Programación orientada a objetos: etapas para la solución de problemas en el enfoque orientado a objetos?

Fig. 1.3 Principales etapas del proceso de solución de problemas



# Programación orientada a objetos: etapas para la solución de problemas en el enfoque orientado a objetos?

## Construcción de la solución

Desarrollo y  
codificación  
de algoritmos

Generación  
de código  
ejecutable

Pruebas y  
refinamiento

## Análisis y diseño orientado a objetos

Identificar entidades

Modelar sus  
características

Definir relaciones

Documentar

Crear modelo del mundo  
del problema

## Análisis del problema

Identificación de  
requerimientos  
funcionales

Entendimiento del  
mundo del problema

Identificación de  
requerimiento no  
funcionales



## Recomendaciones



**coursera**

Introducción a la programación orientada a objetos en Java

**Universidad de los Andes**

Introducción a UML

**Universidad de los Andes**



**ORACLE®**  
Academy



## Bibliografía de Consulta

# coursera

Introducción a la programación orientada a objetos en Java

**Universidad de los Andes**

Introducción a UML

**Universidad de los Andes**

**ORACLE®**  
Academy





**UNIVERSIDAD**  
Popular del cesar