

# **Principios de la Tecnología de Objetos**

---

## **Conceptos básicos de la Orientación a Objetos**

## **Copyright**

-  Copyright (c) 2004  
José M. Ordax
-  Este documento puede ser distribuido solo bajo los términos y condiciones de la Licencia de Documentación de javaHispano v1.0 o posterior.
-  La última versión se encuentra en  
<http://www.javahispano.org/licencias/>

## Definición de objeto



Un objeto es:



Una cosa tangible y/o visible.



Algo que puede comprenderse intelectualmente.



Una entidad software.



## Características de un objeto



Un objeto tiene:



Identidad: un identificador único.



Estado: un conjunto de propiedades (atributos).



Comportamiento: un conjunto de operaciones (métodos).



Los términos objeto e instancia son intercambiables.

## Otras definiciones



Un objeto se caracteriza por un número de **operaciones** y un **estado** que recuerda el efecto de estas operaciones.

Ivar Jacobson



Un objeto tiene un **estado**, **comportamiento** e **identidad**; la estructura y comportamiento de objetos similares se definen en sus clases comunes.

Grady Booch

## Otras definiciones



Un objeto es una entidad que tiene un **estado** (cuya representación está oculta) y un conjunto definido de **operaciones** que operan sobre ese estado.

Ian Sommerville

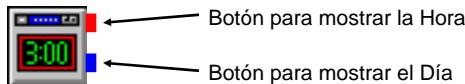


Un objeto es una identidad con unos límites bien definidos que encapsulan **estado** y **comportamiento**. El estado se representa por atributos y relaciones, el comportamiento es representado por operaciones y métodos.

Object Managemente Group

## Ejemplo práctico

Modelo físico:



Modelo informático.

unReloj	
Atributos	Métodos
hora (horas, min, seg)	getHora
dia (dia, mes, año)	getDia
modelo	incrementarHora
numSerie	incrementarDia
	limpiarPantalla

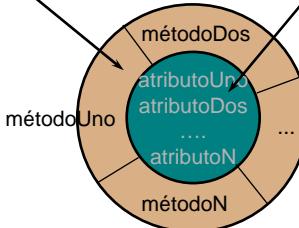
## Estructura de un objeto

Métodos

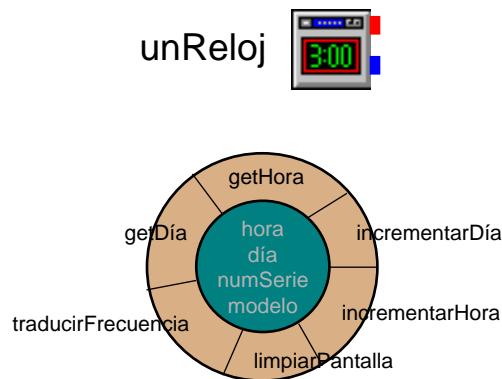
Operaciones permitidas.  
Pueden estar o no ocultas para el usuario.

Atributos

Estructura encapsulada de los datos.



# Ejemplo práctico



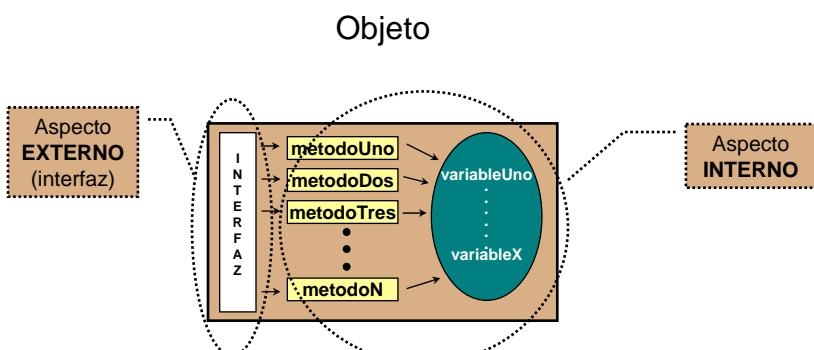
## Definición de atributo

- Es una característica fundamental de cada objeto de una clase.
- Una clase puede definir un cierto número de atributos estáticos.
- Todos los atributos tienen algún valor. Este valor puede ser una cantidad, una relación con otro objeto, etc...

## Definición de método

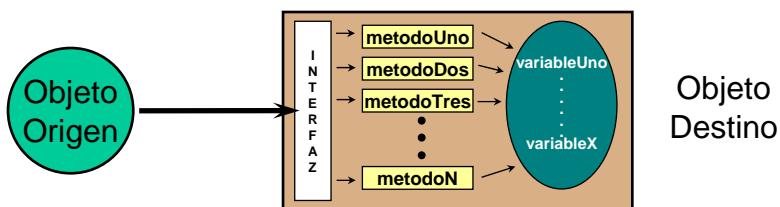
- Es una acción que se realiza sobre un objeto para consultar o modificar su estado.
- Tipos de operaciones:
  - Modificador (setter): altera el estado de un objeto.
  - Selector (getter): accede al estado de un objeto sin alterarlo.
  - Iterador: permite acceder a todas las partes de un objeto.
  - Constructor: crea un objeto e inicializa su estado.
  - Destructor: limpia el estado de un objeto y lo destruye.
  - Propósito general: la lógica del programa.

## El aspecto de los objetos



## Interfaz

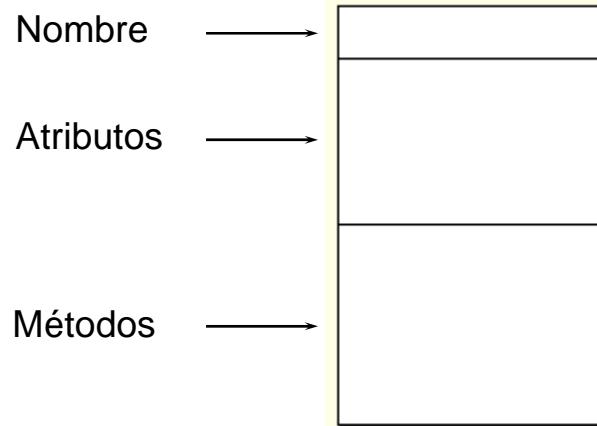
- Es el aspecto externo del objeto. La parte visible y accesible para el resto de objetos.
- También se le define como el protocolo de comunicación de un objeto.
- Puede estar formado por uno o varios métodos. No todos los métodos de un objeto tienen porque formar parte del interfaz.



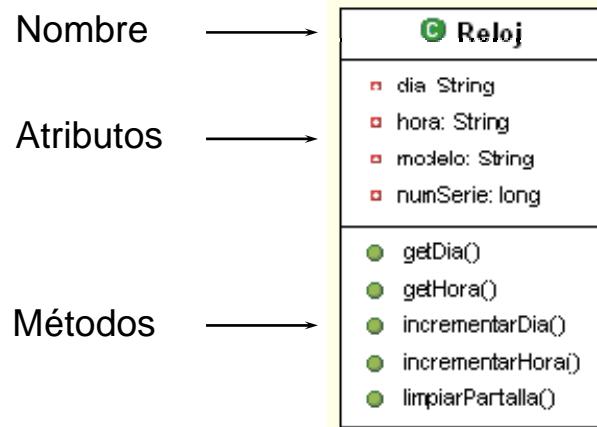
## Definición de clase

- Una clase es la representación de la estructura y comportamiento de un objeto
- Es un patrón para la definición de atributos y métodos para un tipo particular de objetos.
- Todos los objetos de una clase dada son idénticos en estructura y comportamiento pero son únicos (aunque tengan los mismos valores en sus atributos).
- Instancia es el término utilizado para referirse a un objeto que pertenece a una clase concreta.

## Estructura de una clase



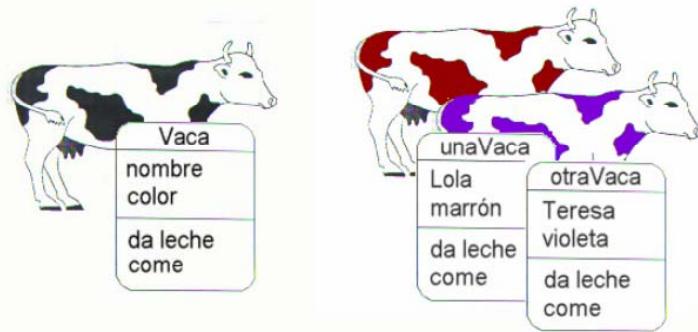
## Ejemplo práctico



## Clases vs. Objetos

- Clase:
  - Un patrón para la definición del estado y el comportamiento de un tipo particular de objetos.
  - Todos los objetos de una clase dada son idénticos en estructura y comportamiento, pero tienen identidad única.
- Objeto (instancia):
  - Pertenece a una clase en particular.
  - Los objetos son creados y destruidos en tiempo de ejecución. Residen en el espacio de memoria.

## Ejemplo práctico



## Clasificación

- La clasificación es el medio por el que ordenamos el conocimiento:
- Es fundamentalmente un problema de búsqueda de similitudes.
- Al clasificar buscamos grupos de cosas que tengan una misma estructura o exhiban un comportamiento común.
- Clasificación y desarrollo O.O.:
- Clasificación significa que los objetos con la misma estructura de datos y con el mismo comportamiento se agrupan para formar una clase.
- Esta es una de las tareas fundamentales en el análisis y diseño O.O.

## Ejercicio

- ¿Quién dice qué?: Clase, Objeto, Atributo, Método
- El valor de mis atributos puede ser distinto al de los de mi semejante:\_\_\_\_\_.
- Yo me comporto como una plantilla:\_\_\_\_\_.
- A mi me gusta hacer cosas:\_\_\_\_\_.
- Yo puedo tener muchos métodos:\_\_\_\_\_.
- Yo represento el estado:\_\_\_\_\_.
- Yo represento el comportamiento:\_\_\_\_\_.
- Yo estoy en los objetos:\_\_\_\_\_.

## Ejercicio (solución)

- ¿Quién dice qué?: Clase, Objeto, Atributo, Método
- El valor de mis atributos puede ser distinto al de los de mi semejante: **Objeto**.
- Yo me comporto como una plantilla: **Clase**.
- A mi me gusta hacer cosas: **Objeto, método**.
- Yo puedo tener muchos métodos: **Clase, objeto**.
- Yo represento el estado: **Atributo**.
- Yo represento el comportamiento: **Método**.
- Yo estoy en los objetos: **Atributo, método**.

## Ejercicio

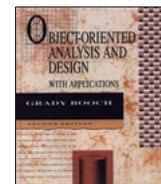
- ¿Quién dice qué?: Clase, Objeto, Atributo, Método
- Yo vivo en memoria:\_\_\_\_\_.
- Yo soy usado para crear instancias:\_\_\_\_\_.
- Mi estado puede cambiar:\_\_\_\_\_.
- Yo declaro métodos:\_\_\_\_\_.
- Yo puedo cambiar en ejecución:\_\_\_\_\_.

# Ejercicio

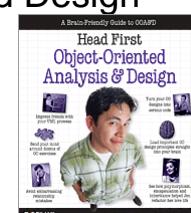
- ¿Quién dice qué?: Clase, Objeto, Atributo, Método
- Yo vivo en memoria: **Objeto**.
- Yo soy usado para crear instancias: **Clase**.
- Mi estado puede cambiar: **Objeto**.
- Yo declaro métodos: **Clase**.
- Yo puedo cambiar en ejecución: **Objeto, atributo**.

# Bibliografía

- Object-Oriented Analysis and Design  
Grady Booch.  
Addison-Wesley.



- Head First Object-Oriented Analysis and Design  
Brett McLaughlin, Gary Pollice, David West  
O'Reilly



- Design patterns  
Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides  
Addison-Wesley.

