

Programación Orientada a Objetos

Tema 2: Fundamentos de la programación orientada a objetos

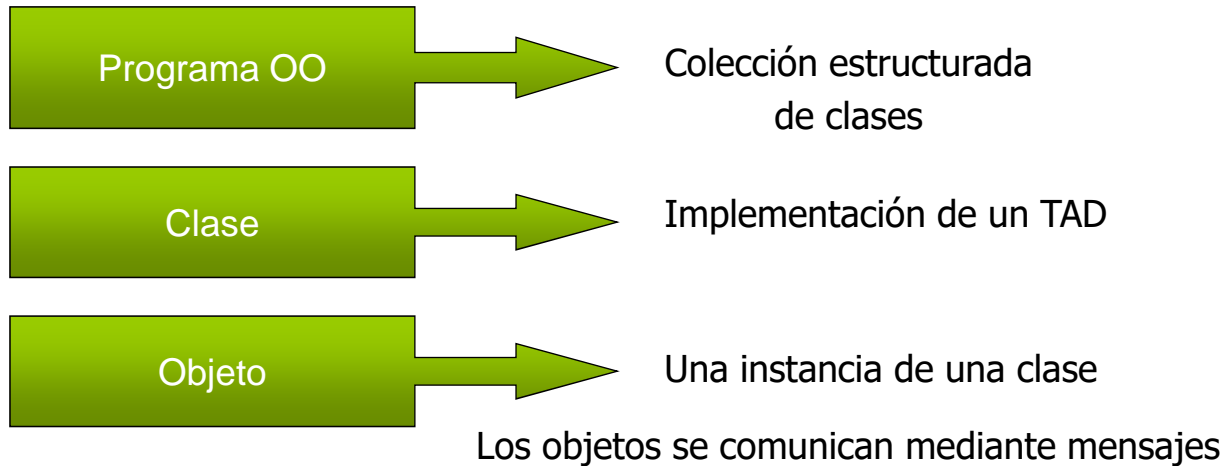
Tema 2-3: Conceptos básicos de OO

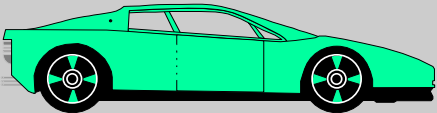
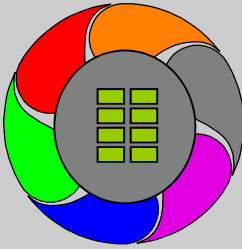
- Tema 2-3: Conceptos básicos de OO
- 1. Introducción
- 2. Objetos
- 3. Clases
- 4. Instancias
- 5. Diseño estructurado vs. OO



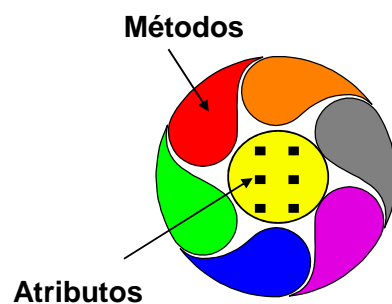
- **¿Qué es la POO?**

“Un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada una de las cuales representan una instancia de alguna clase, y cuyas clases son todas miembros de una jerarquía de clases unidas mediante relaciones de herencia”. Booch.



MUNDO REAL	SIMULACIÓN INFORMÁTICA
<p>Un elemento de un sistema físico se caracteriza por :</p> <p>Una IDENTIDAD, unas CARACTERÍSTICAS, un ESTADO, unos COMPORTAMIENTOS</p> 	<p>Un objeto se caracteriza por :</p> <p>Una IDENTIDAD, unos DATOS, unos PROCEDIMIENTOS</p> 

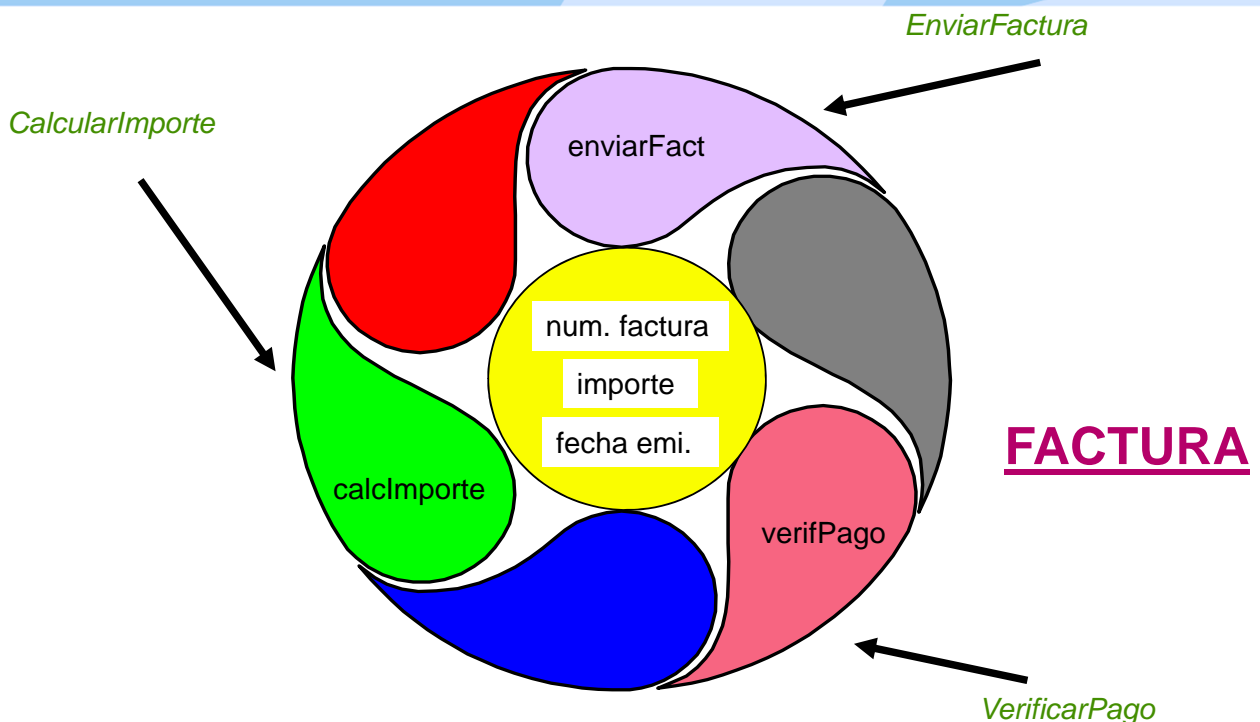
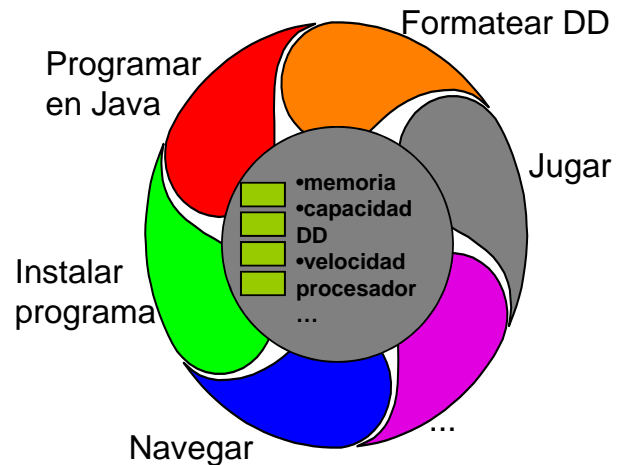
- Un “paquete” de software independiente formado por un conjunto de datos junto con los procedimientos que operan sobre estos datos.
- Un objeto puede ser real o abstracto.

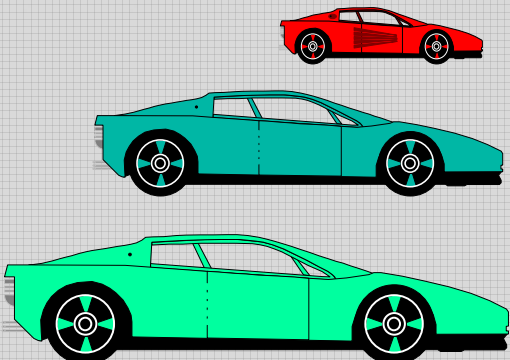
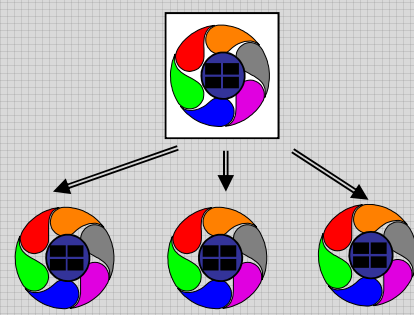


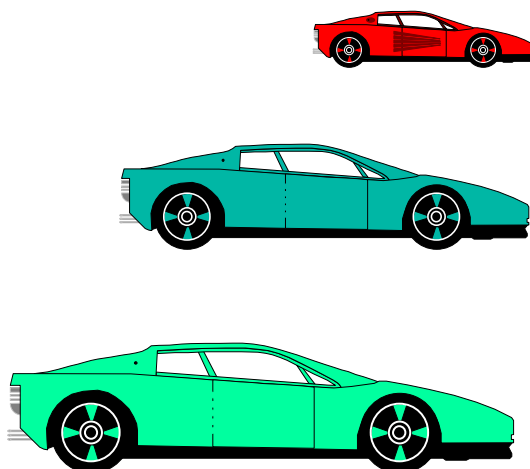
Estructura de un objeto

- Objeto = Módulo de software.
- Cada objeto constituye un universo cerrado bien definido.
- Todo lo que un objeto “sabe” se expresa en sus atributos.
- Todo lo que un objeto “puede hacer” se expresa por sus operaciones (métodos).

Ejemplo: Ordenador



MUNDO REAL	SIMULACIÓN INFORMÁTICA
<p>Los elementos de un sistema físico pueden ser agrupados en CATEGORIAS :</p> 	<p>Los objetos que muestran la misma estructura y comportamiento se agrupan en CLASES</p> 



Cualquier tipo de
coche puede
agruparse dentro
de la Clase
AUTOMOVIL



- Automóvil

Atributos	Operaciones
Matrícula	poner en marcha
marca	parar
modelo	acelerar
estado	frenar
potencia	pintar
color	encenderLuces
luces	apagarLuces
velocidad	
peso	...

Esta lista de atributos y de operaciones asegura el hecho de que el conjunto de instancias de la clase Automóvil dispondrá de estos atributos y de los valores asociados, así como de la facultad de ejecutar las operaciones citadas.



- Un Automóvil

Atributos
Matrícula: 5671BCD
marca: Ford
modelo: Focus
estado: en marcha
potencia: 130 c.
color: rojo
luces: encendidas
Velocidad: 60 km/h
peso: 1500 kg

- Se especifican sus atributos.
- Este coche tiene ahora una “existencia informática”.
- Está en condiciones de participar en las secuencias de interacciones.



Definición

“Implementación total o parcial de un tipo abstracto de dato (TAD)”

- Una clase es un tipo definido que determina la estructura de datos y las operaciones asociadas a ese tipo.
- Una clase se puede ver como una plantilla que describe objetos que van a tener la misma estructura y el mismo comportamiento.
- Un programa OO es una colección estructurada de clases.



- **Doble naturaleza de las clases:**

Una clase es un **módulo** y un **tipo de dato**:

- **Módulo** (concepto sintáctico)
 - Mecanismo para organizar el software (sintáctico)
 - Encapsula componentes software
- **Tipo** (concepto semántico)
 - Mecanismo de definición de nuevos tipos de datos: describe una estructura de datos (objetos) para representar valores de un dominio y las operaciones aplicables.

“Los servicios proporcionados por una clase, vista como un módulo, son precisamente las operaciones disponibles sobre las instancias de la clase, vista como un tipo”.



Componentes de una clase:

- **Atributos**

- Determinan una estructura de almacenamiento para cada objeto de la clase.

- **Operaciones (Métodos)**

- Operaciones aplicables a los objetos.
- Único modo de acceder a los atributos.

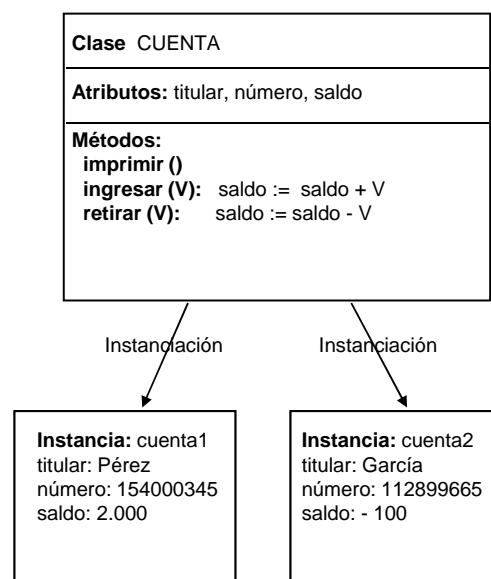
Ejemplo: Al modelar un banco, encontramos objetos “*cuenta*”.
Todos los objetos “*cuenta*” tienen propiedades comunes:

- atributos: número, saldo, titular, ...
- operaciones: ingresar, retirar ...

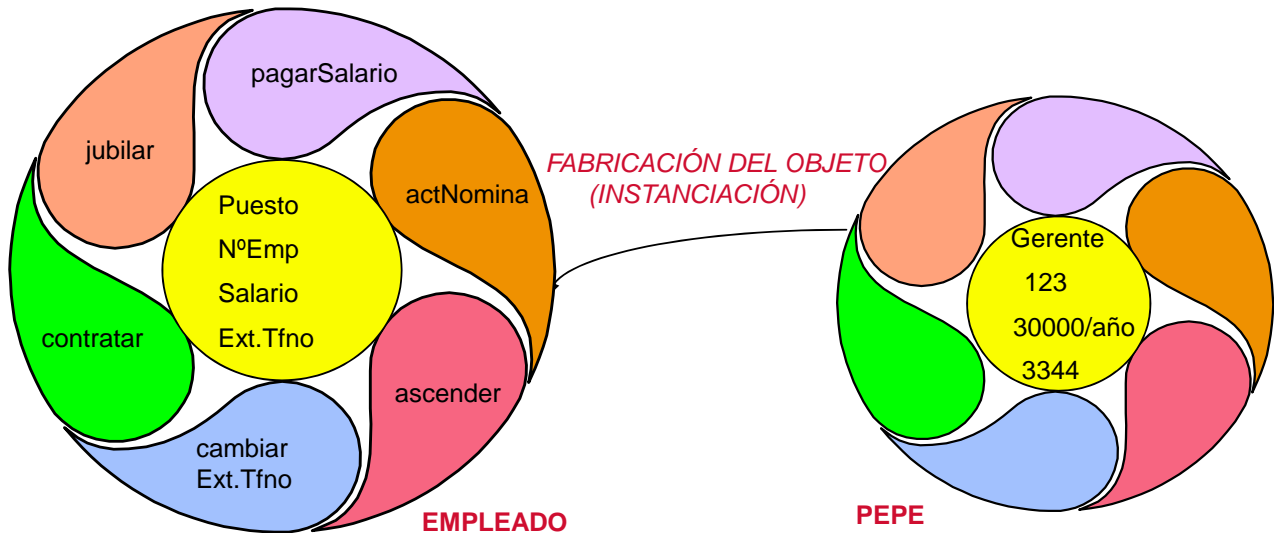
Definimos una clase CUENTA.



- Una clase es un generador de objetos (las instancias de la clase).
- Una instancia es una estructura constituida por los atributos descritos para la clase.
- La creación de un objeto a partir de una clase se llama instanciación.
- Muchas instancias de una misma clase pueden existir simultáneamente en memoria.



La clase Empleado describe a todos los empleados, a partir de ella podemos dirigirnos a un empleado concreto dando valores a sus datos (atributos).



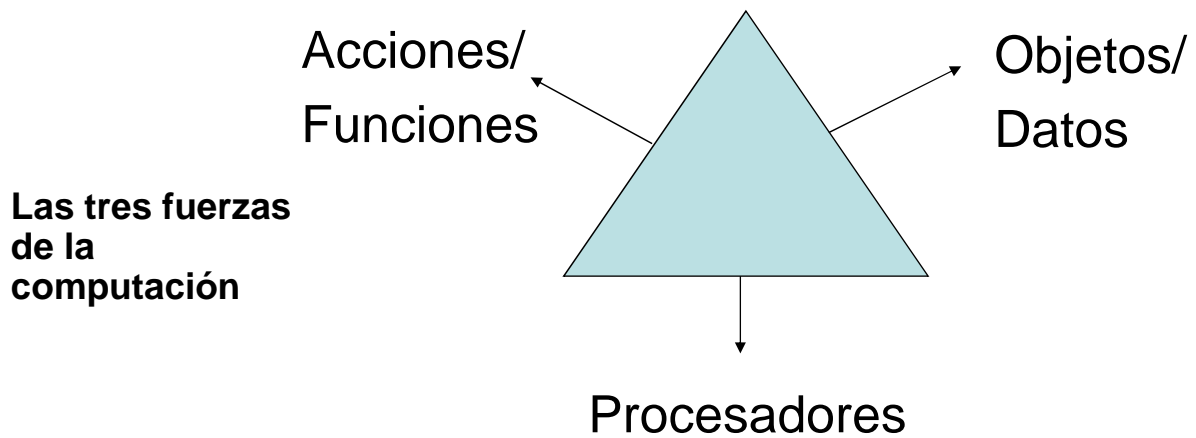
Definición

**“Es una instancia de una clase,
creada en tiempo de ejecución”**

- Es una estructura de datos formada por tantos **atributos** como tiene la clase.
- El **estado** de un objeto viene dado por el valor de los atributos. El estado suele cambiar con el paso del **tiempo**.
- Los **métodos** permiten consultar y modificar el estado del objeto.
- Durante la ejecución de un programa OO se crearán un conjunto de objetos.

Diseño estructurado vs. OO

- ¿Qué criterio usamos para encontrar los módulos?

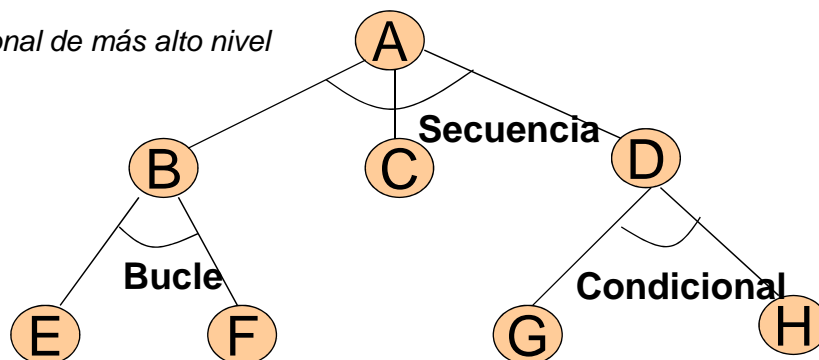


- A) Unidades de descomposición funcional → Enfoque tradicional
- B) Basándose en los principales tipos de datos → Enfoque OO

Diseño estructurado vs. OO

- **Descomposición funcional:**
- Respuesta tradicional a la cuestión de modularización.
- ¿Responde a los requisitos de modularidad?

Abstracción funcional de más alto nivel



Refinamientos sucesivos



Diseño estructurado vs. OO

- **Inconvenientes de la descomposición funcional:**
- Se basa en propiedades poco estables que dificulta la **extensibilidad**:
- ⊗ **Función principal: “Cima del sistema”**
 - El “programa principal” es una propiedad volátil.
 - Sistemas reales no tienen “cima”.
 - Mejor la visión de un “conjunto de servicios”.
- ⊗ **Centrado en la interfaz externa**
 - Primera pregunta: **¿Que hará el sistema?**
 - La arquitectura del software debe basarse en propiedades más profundas.
- ⊗ **Ordenación temporal prematura**



Diseño estructurado vs. OO

- **Inconvenientes de la descomposición funcional:**
- No promueve la **reutilización**:
- ⊗ Se desarrollan elementos software para satisfacer necesidades específicas de otro elemento del nivel superior.
- ⊗ “Cultura del proyecto actual”.
- Las estructuras de datos son descuidadas.
 - Funciones y datos deben jugar un papel complementario.
- Cuando un sistema evoluciona los datos son más estables que los procesos.



Diseño estructurado vs. OO

➤ **Ventajas de la descomposición funcional:**

- ☺ Disciplina de pensamiento lógica y bien organizada.
- ☺ Técnica simple, fácil de aplicar.
- ☺ Útil para pequeños programas y algoritmos individuales.
- ☺ Buena para documentar diseños (describir algoritmos).
- ☺ Promueve el desarrollo ordenado de sistemas.
- ☺ Adecuada para dominar la complejidad.



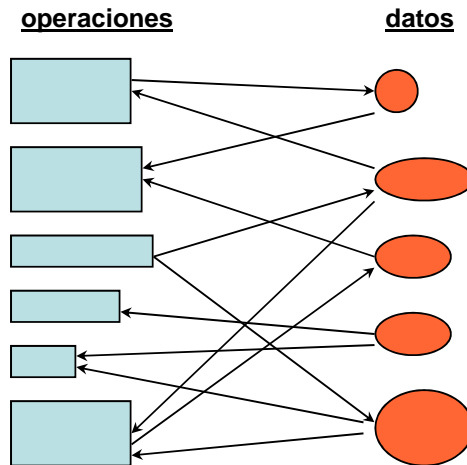
Diseño estructurado vs. OO

➤ **Descomposición modular basada en los datos:**

- Los datos son más estables que las funciones lo que favorece la extensibilidad.
- Los tipos de datos equipados con las operaciones asociadas proporcionan unidades estables para la reutilización.
- ¿Qué significa Orientación a Objetos?
 - El software se organiza como una colección de objetos que contienen tanto estructura como comportamiento.
- ¿Qué es el desarrollo Orientado a Objetos?
 - Una nueva forma de pensar acerca del software basándose en abstracciones que existen en el mundo real.

Diseño estructurado vs. OO

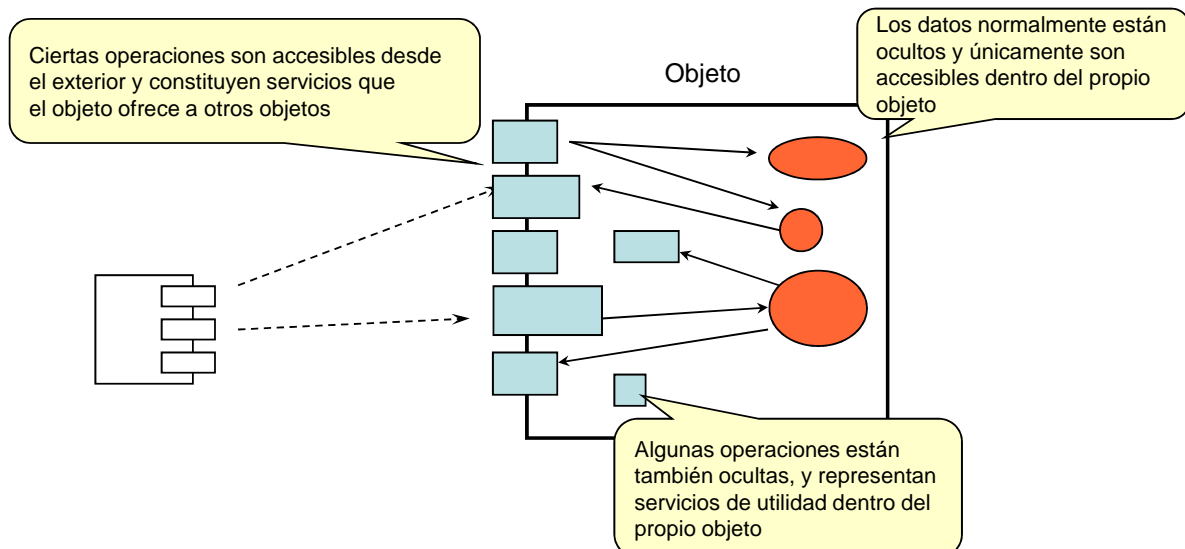
En la programación tradicional tenemos por un lado los **datos** y por otro las **operaciones sobre esos datos**, pero son entidades independientes.



25

Diseño estructurado vs. OO

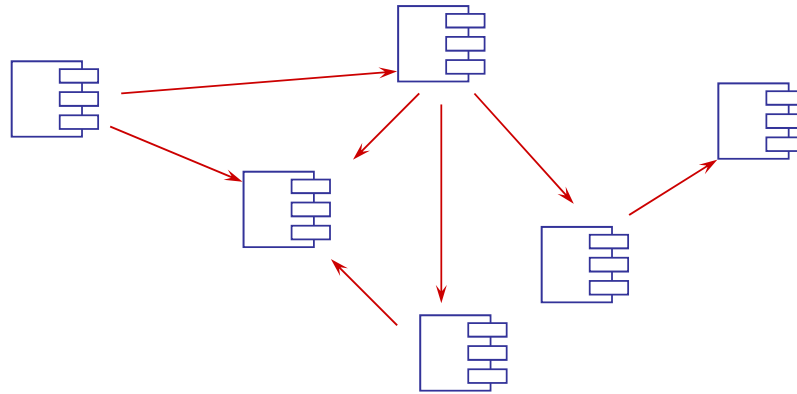
En la programación OO agrupamos (encapsulamos) **conjuntos de datos** relacionados entre sí y **operaciones sobre esos datos** en entidades que llamamos objetos.



26

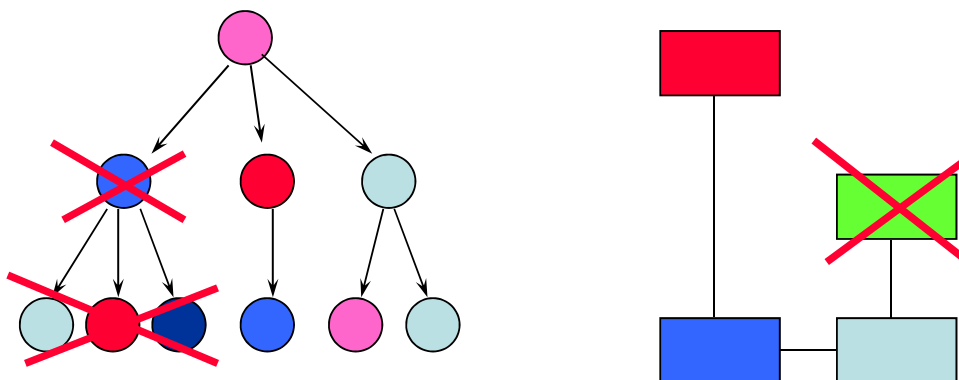
Diseño estructurado vs. OO

- Con la orientación a objetos, construimos pequeños **modelos software** de la realidad y simulamos ésta.
- Un sistema O.O. es un **conjunto de objetos** que interactúan entre sí enviándose **mensajes** mediante los cuáles se solicitan servicios unos a otros.



27

Diseño estructurado vs. OO



Las abstracciones funcionales son más volátiles que las de datos.

Esa es una de las ventajas de la OO.

28



Diseño estructurado vs. OO

- **Desarrollo de software orientado a objetos :**

Definición

- ♦ Método de desarrollo de software que basa la arquitectura del sistema en módulos deducidos de los tipos de objetos que se manipulan (en lugar de basarse en la función o funciones a las que el sistema está destinado a asegurar).
- ♦ Hay que centrar la atención no sobre lo que **HACE** el sistema, sino principalmente sobre lo que **ES** el sistema, en términos de datos, de componentes, en término de manejo de entidades, de reacción a las solicitudes.