

Orientación a Objetos

Análisis Orientado a Objetos:

- Enfoque para entender y modelar sistemas.
- Se centra en identificar y definir los objetos, sus interacciones, sus características y responsabilidades.
- Permite flexibilidad y reutilización.

Características Claves:

- **Objetos:** entidades que combinan datos (atributos) y comportamientos (métodos) en una sola unidad.
- **Clases:** juntan objetos similares que comparten atributos y métodos.
- **Relaciones:** se analizan las relaciones entre objetos y clases, como la herencia y la asociación.
- **Encapsulamiento:** los datos y los métodos se agrupan dentro de los objetos.
- **Abstracción:** ocultar los detalles complejos, centrándose en lo que un objeto hace en lugar de cómo lo hace.
- **Responsabilidad:** cada objeto tiene responsabilidades específicas y es responsable de realizar ciertas tareas.
- **Modelo de Dominio:** representa los conceptos y las relaciones del mundo real en el sistema a desarrollar.
- **Polimorfismo:** capacidad de diferentes objetos para ser tratados de manera uniforme a través de una interfaz común.
- **Herencia:** mecanismo que permite a una clase heredar atributos y métodos de otra clase.
- **Asociación:** describe una relación entre clases que indica cómo los objetos de una clase se conectan con los objetos de otra clase.

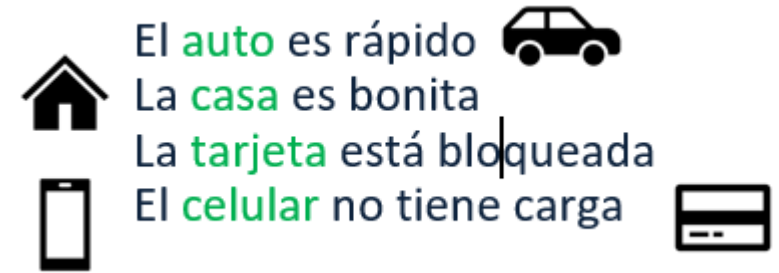
¿Qué es un Objeto?

El mundo está lleno de objetos:

- en la naturaleza
- en entidades hechas por el hombre
- y en los productos que usamos.

Pueden ser clasificados, descritos, organizados, combinados, creados y manipulados.

Objetos Descritos:



Informalmente, un objeto representa una entidad del mundo real

Entidades Físicas

(Ej.: Vehículo, Casa, Producto)

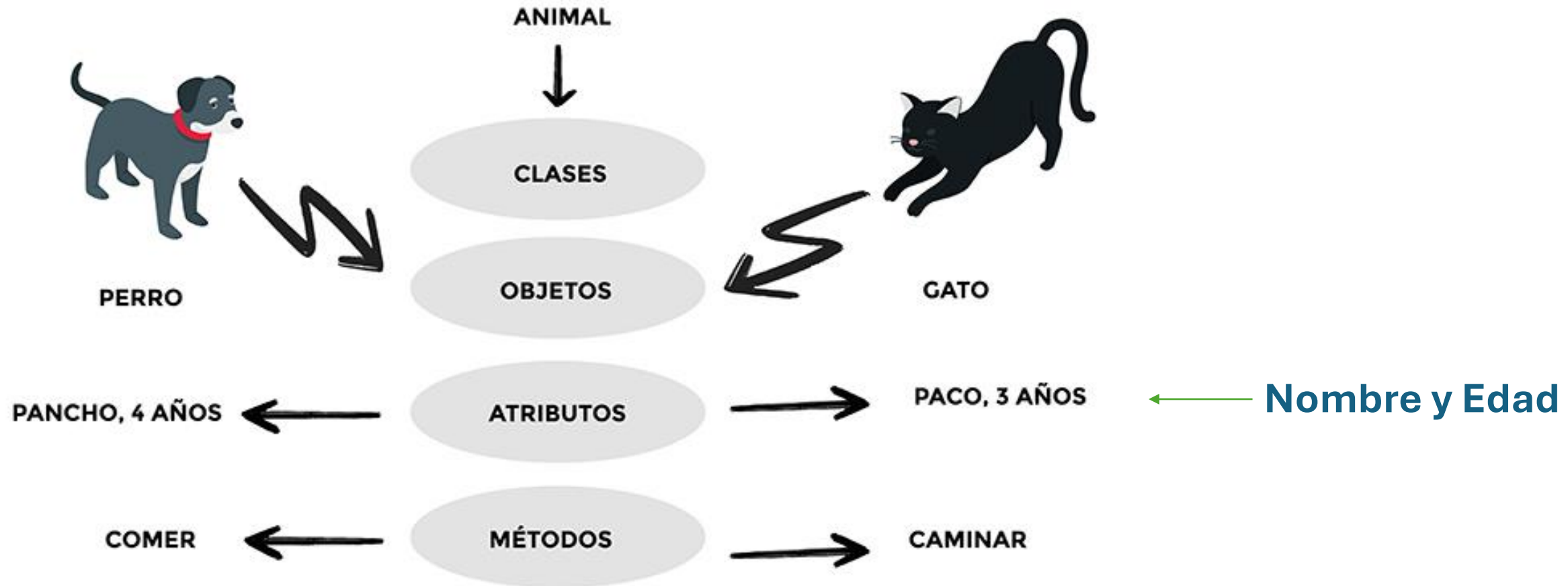
Entidades Conceptuales

(Ej.: Proceso Químico, Transacción Bancaria)

Entidades de Software

(Ej.: Lista Enlazada, Interfaz Gráfica)

Objetos:



Fuente: <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

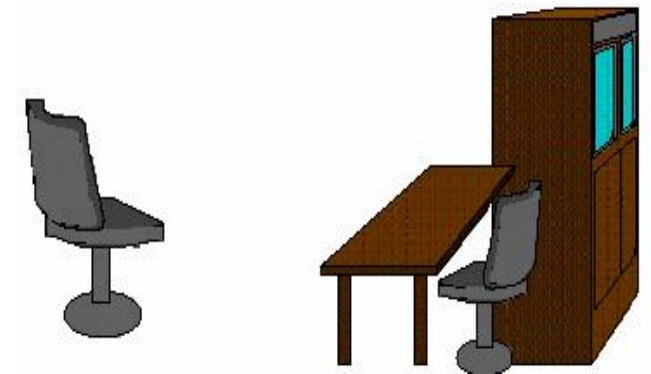
Ejemplos de OO:

Un objeto del mundo real: una silla.

- ✍ La silla es un **miembro (instancia)** de una **clase** mucho más grande de objetos que llamaremos mobiliario.

Un conjunto de **atributos** genéricos pueden asociarse con cada objeto, en la clase mobiliario.

- Por ejemplo, todo mueble tiene un costo, dimensiones, peso, localización y color, entre otros muchos posibles atributos.
- Estos atributos también son aplicables a una mesa o silla, un sofá o un armario.



Como silla es **miembro de la clase** mobiliario, silla **hereda** todos los atributos definidos para la clase.

Orientación a Objetos:

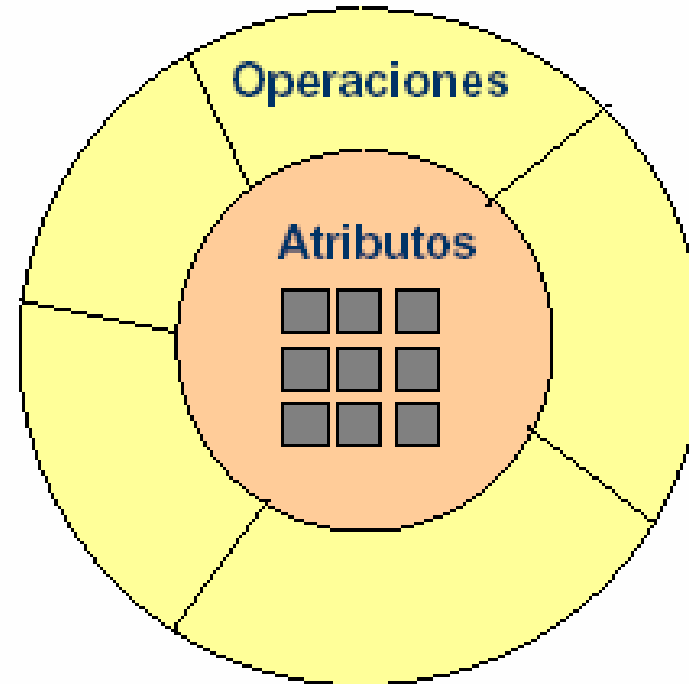
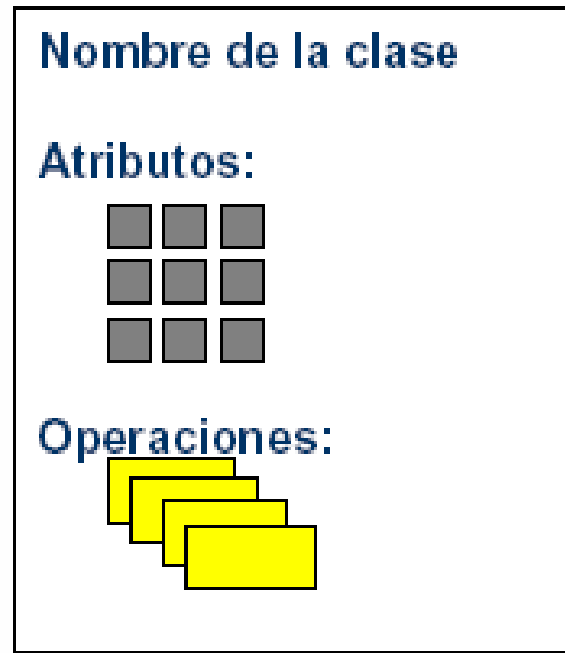
Una vez definida la clase, los atributos pueden reutilizarse al crear nuevas **instancias** de la clase

- Por ejemplo, supongamos que debemos definir un nuevo objeto llamado sillesa (mezcla entre silla y mesa)



Principios de OO.:

- ✍ Las abstracciones de datos (atributos) que describen la clase están encerradas por una “muralla” de abstracciones procedimentales capaces de manipular los datos de alguna manera (**Encapsulamiento**).



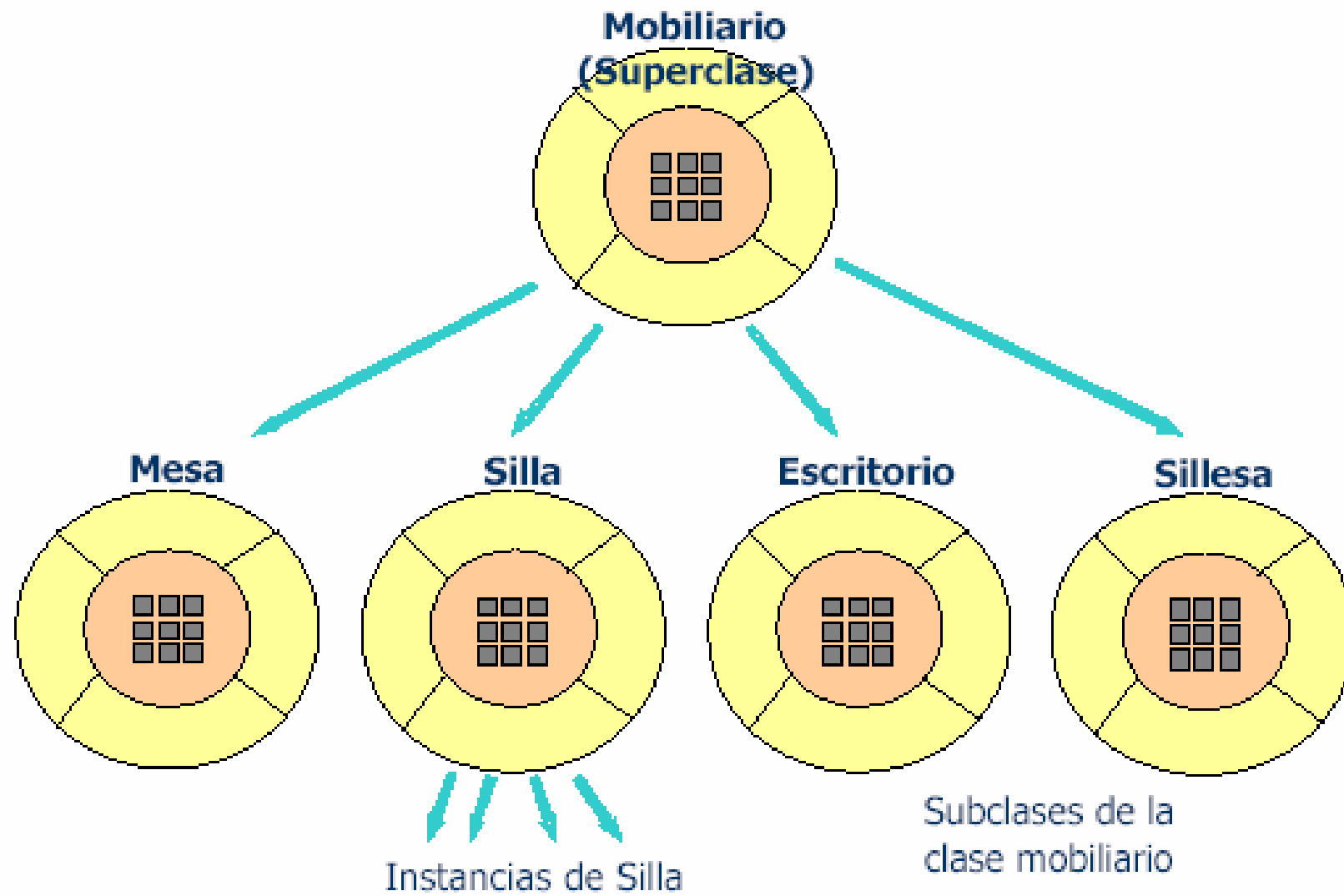
Principios de OO.:

- ✍ Esto posibilita el **ocultamiento** de información y reduce el impacto asociado a los cambios.
- ✍ Como estos métodos manipulan un número limitado de atributos (alta **cohesión**) y como la comunicación sólo ocurre a través de los métodos que encierra la "muralla", la clase tiende a un bajo **acoplamiento** con otros elementos del sistema.

Jerarquía:

- ✍ **Clase:** colección de objetos similares, los cuales heredan atributos y operaciones disponibles para la manipulación de éstos.
- ✍ **Superclase:** colección de clases.
- ✍ **Subclase:** instancia de una clase.
- ✍ Estas definiciones implican la existencia de una **jerarquía** de clases, en la cual los atributos y operaciones de la superclase son heredados por subclases que pueden añadir atributos “privados” y métodos.

Jerarquía:



Clases y Objetos:

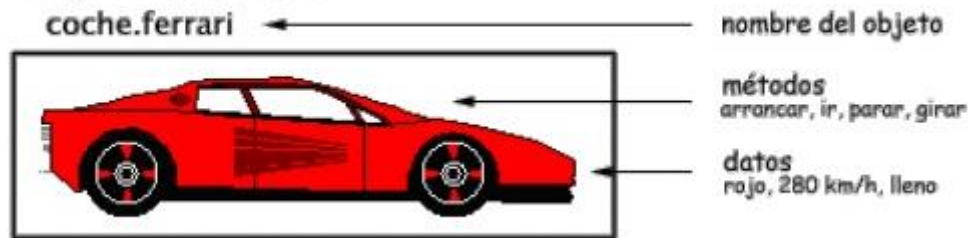
Clase:
Coche



Fuente:

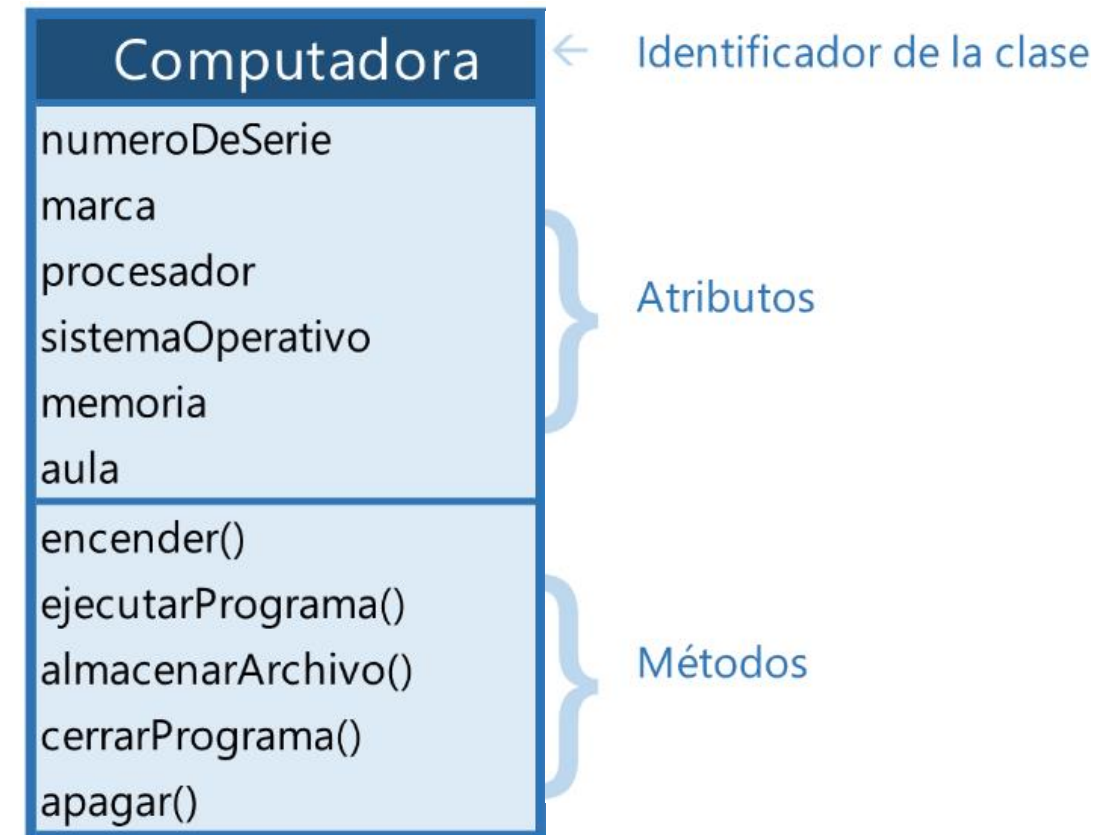
<https://conogasi.org/articulos/programacion-orientada-a-objetos/>

♦ **Objeto:** *Ferrari*



Fuente:

<https://portalacademico.cch.unam.mx/cibernetica1/algoritmos-y-codificacion/caracteristicas-POO>



Atributos:

- ✍ Los **atributos** están asociados a clases y objetos, describiéndolos de alguna manera.
 - Esta relación implica que un atributo puede tomar un valor definido por un **dominio** enumerado.
 - ✍ Una clase **auto** tiene atributo *color*. El dominio de valores de *color* es {blanco, negro, azul, rojo, amarillo, verde}.
 - En situaciones más complejas, el dominio puede ser un conjunto de clases.
 - ✍ La clase **auto** también tiene un atributo *motor* que abarca los siguientes valores de dominio: {valor 32 válvulas opción de lujo, valor 24 válvulas opción deportiva, valor 15 opción económica}.
 - ✍ Estas características se representan como asociaciones, no atributos.

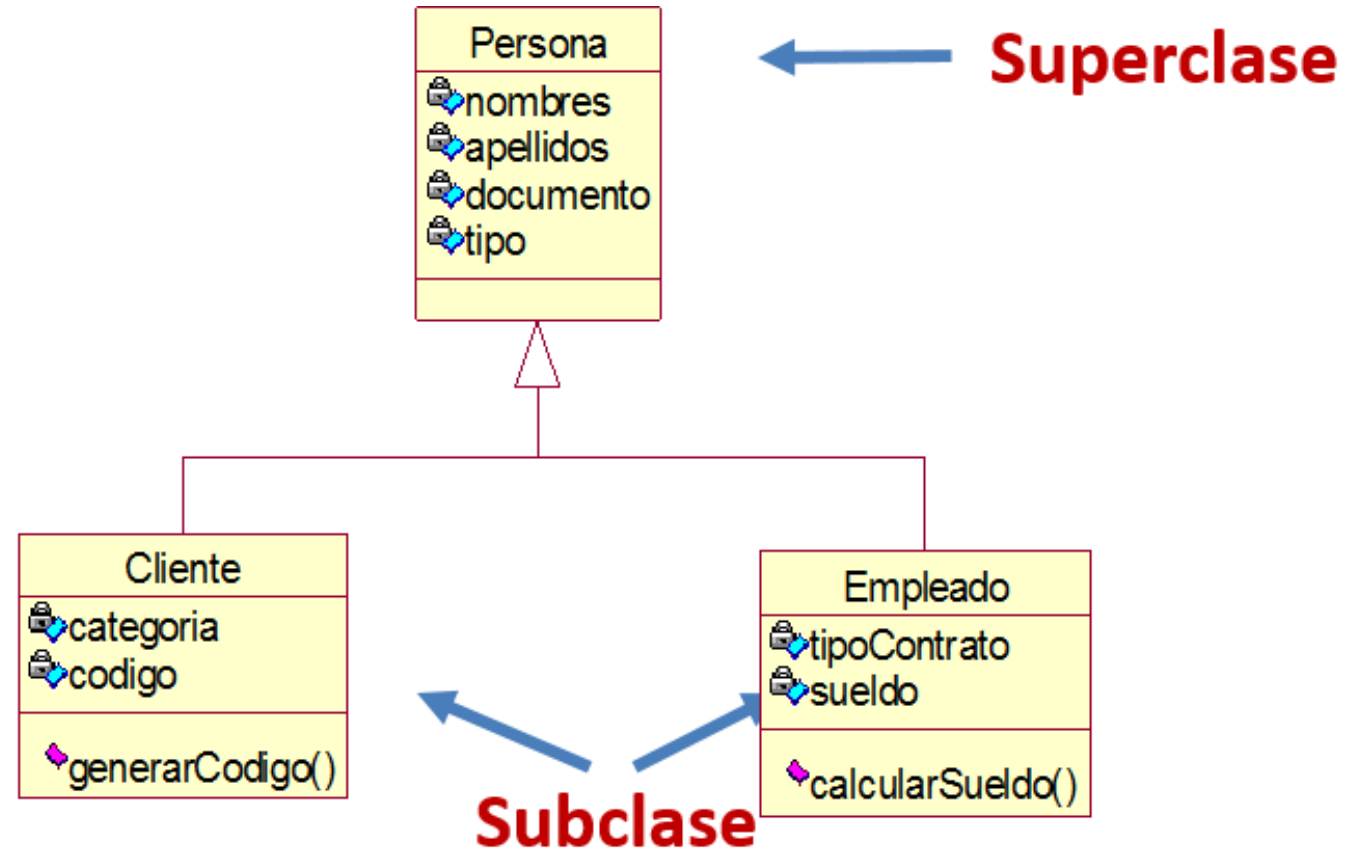
Herencia

Clase Base (Superclass):

- Es la clase que proporciona atributos y métodos que serán heredados por otras clases.
- **Vehículo** podría ser una superclase que define atributos comunes como: marca, modelo y métodos como mover().

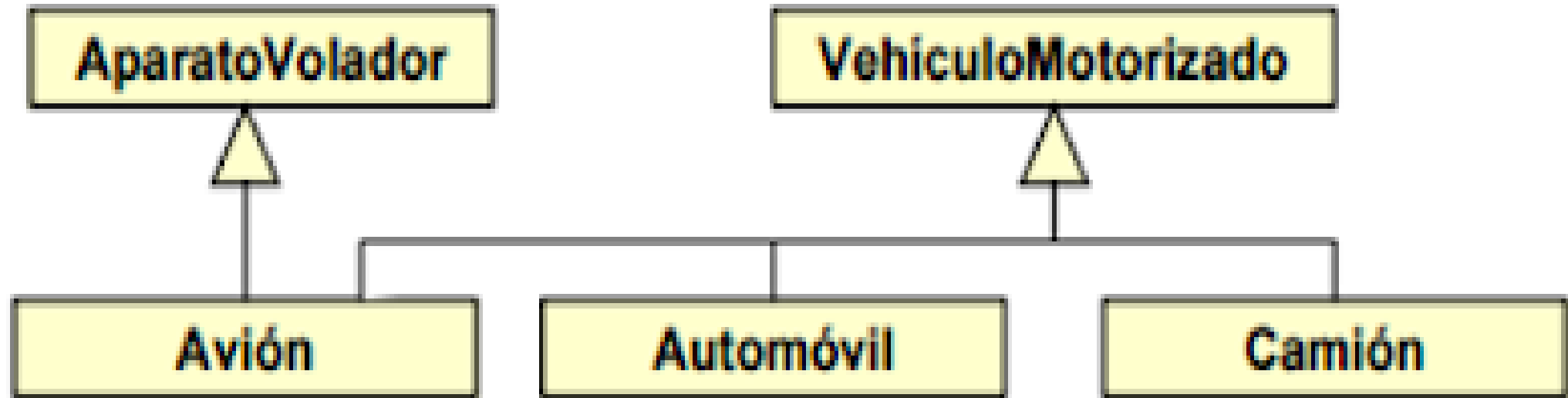
Clase Derivada (Subclase):

- Es la clase que hereda atributos y métodos de la clase base.
- Puede agregar nuevos atributos y métodos o modificar los existentes.
- Una **clase Coche** que hereda de la **clase Vehículo** puede tener atributos adicionales como: cantidad de puertas y métodos específicos.



Fuente: <https://www.incanatoit.com/2017/02/herencia-y-polimorfismo-poo-curso-de.html>

Herencia Múltiple:

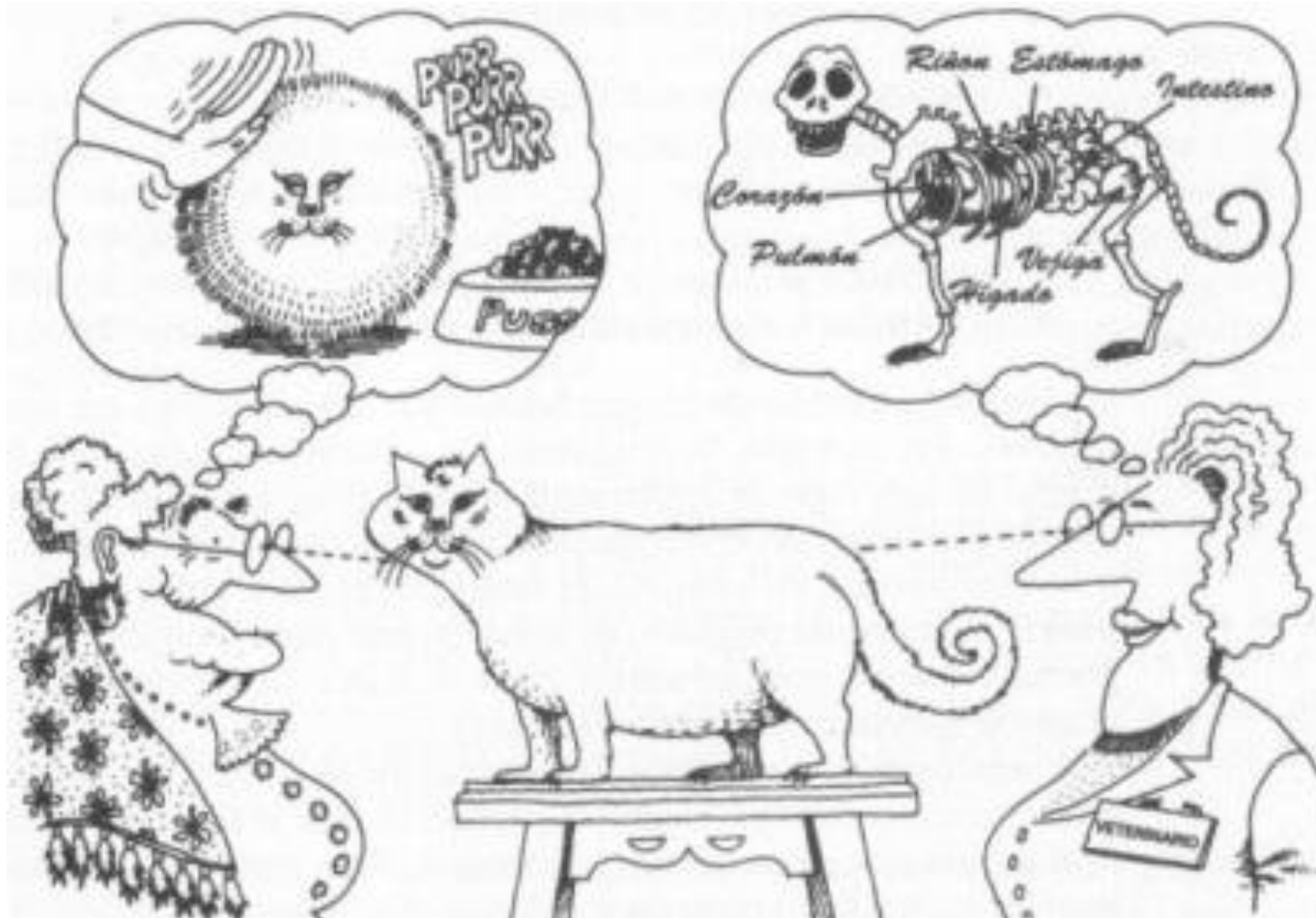


Fuente: <https://paomendozais.blogspot.com/2015/07/relaciones-entre-clases.html>

Operaciones, Métodos o Servicios

- ✍ Un objeto **encapsula** datos y algoritmos que procesan estos datos.
- ✍ Cada operación proporciona uno de los **comportamientos** del objeto.
- ✍ La operación se ejecuta en la medida en que se reciba un estímulo - **mensaje**.

Abstracción: Centrarse en las características esenciales de algún objeto (perspectiva del observador).

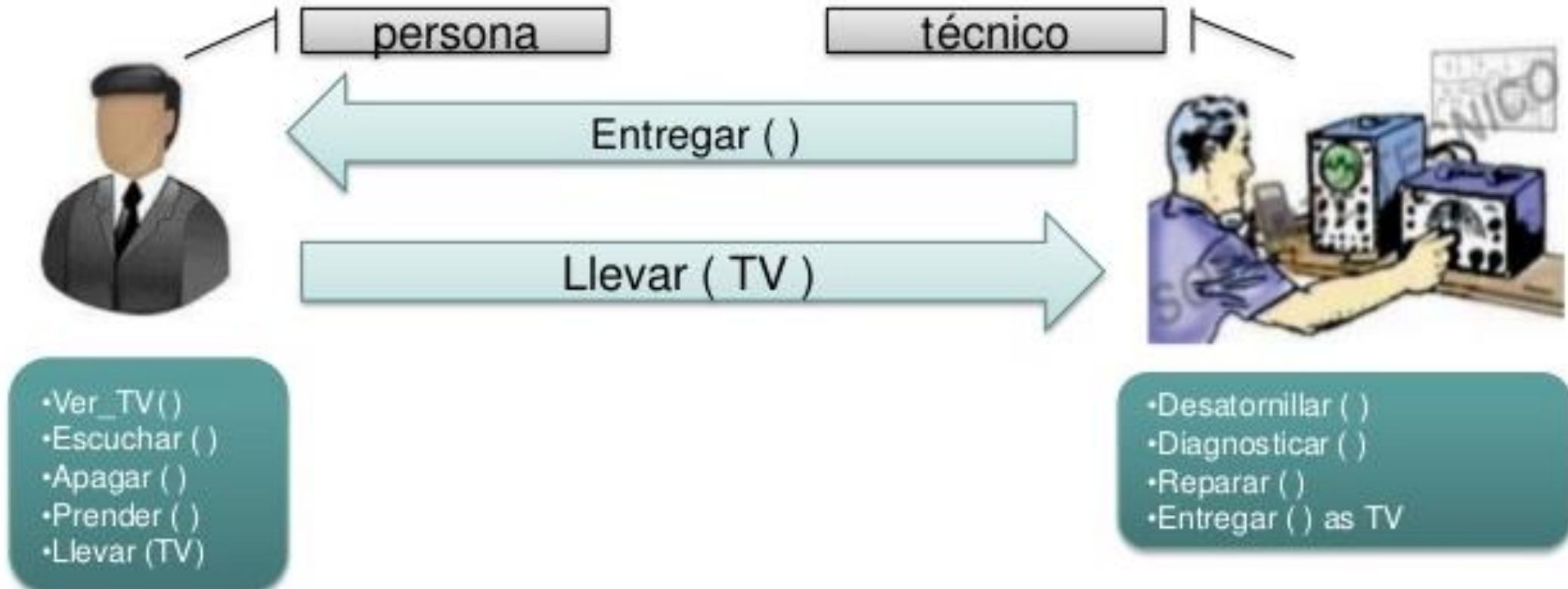


Fuente: https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoría/concepts.html

Encapsulamiento:

Ocultar la información, dejando ver sólo lo público.

Ejemplo: Llevar una TV. en mal estado a un centro técnico para repararla.



Ejercicio:

- ✍ Describa a una bicicleta como clase definiendo dos operaciones y al menos tres atributos.
- ✍ Escriba el código java para la clase bicicleta sin especificar los métodos.



Bicicleta



vel_actual : int

cambio_actual : int



cambiarVelocidad()

frenar()

```
Public class Bicicleta {
```

```
    private int vel_actual;
```

```
    private int cambio actual;
```

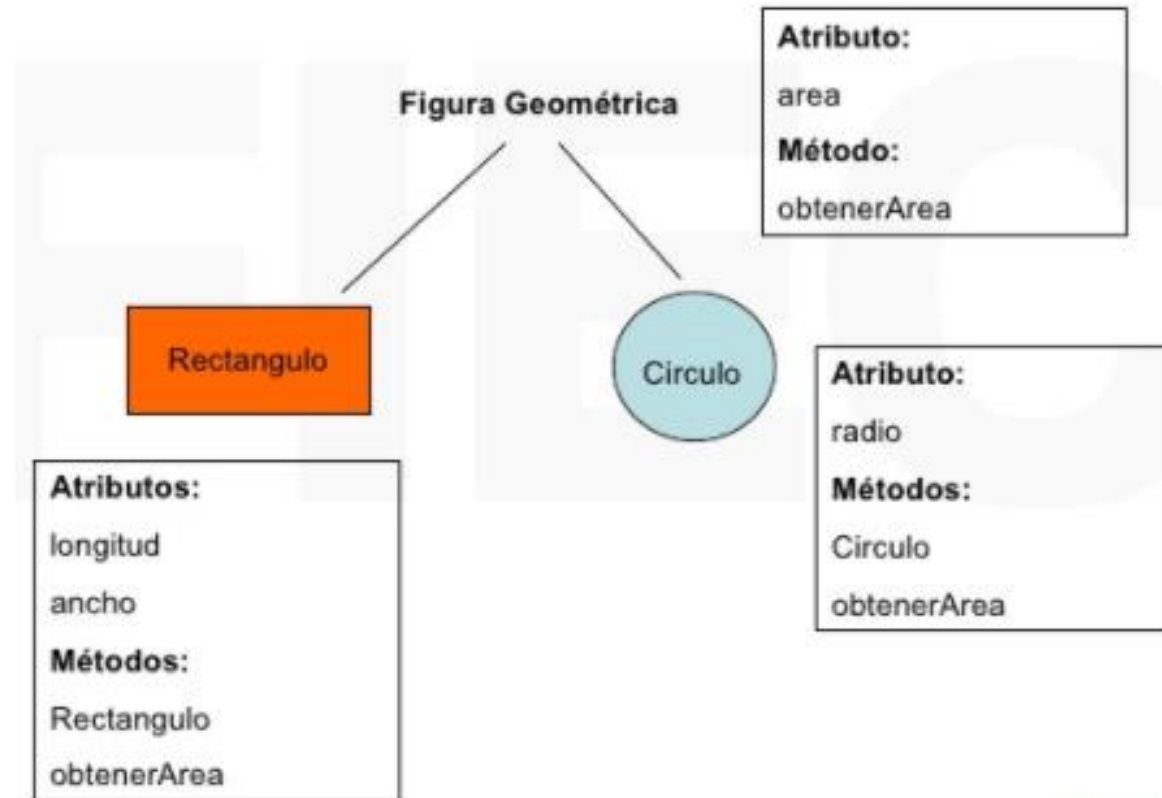
```
    public int cambiarVelocidad (int vel_nueva){}
```

```
    public int frenar(){}
```

```
}
```

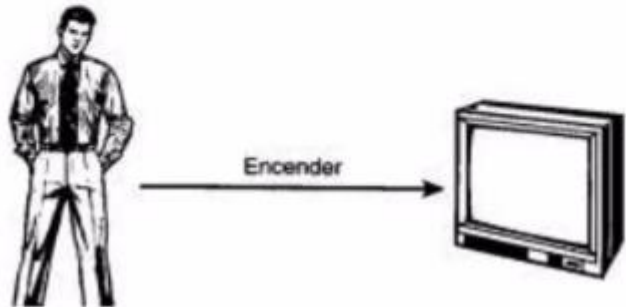
Polimorfismo:

- ✍ **Polimorfismo** permite que un número de operaciones diferentes tengan el mismo nombre, haciendo que cada objeto sea más independiente.



Asociación

Por ejemplo, cuando enciende su televisor, en términos de orientación a objetos, usted se asocia con su televisor.
La asociación "encendido" es en una sola dirección (una vía), esto es, usted enciende la televisión.

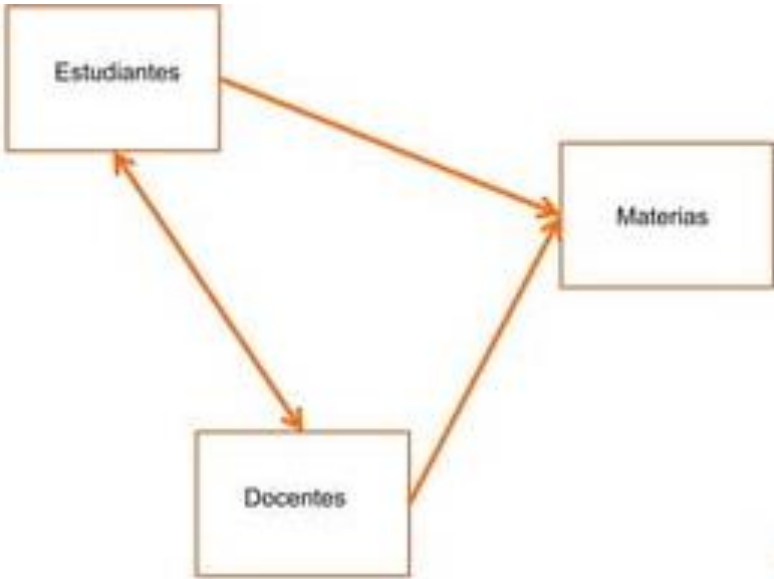


Fuente: <https://es.slideshare.net/slideshow/programacion-orientada-a-objetos-45285363/45285363>

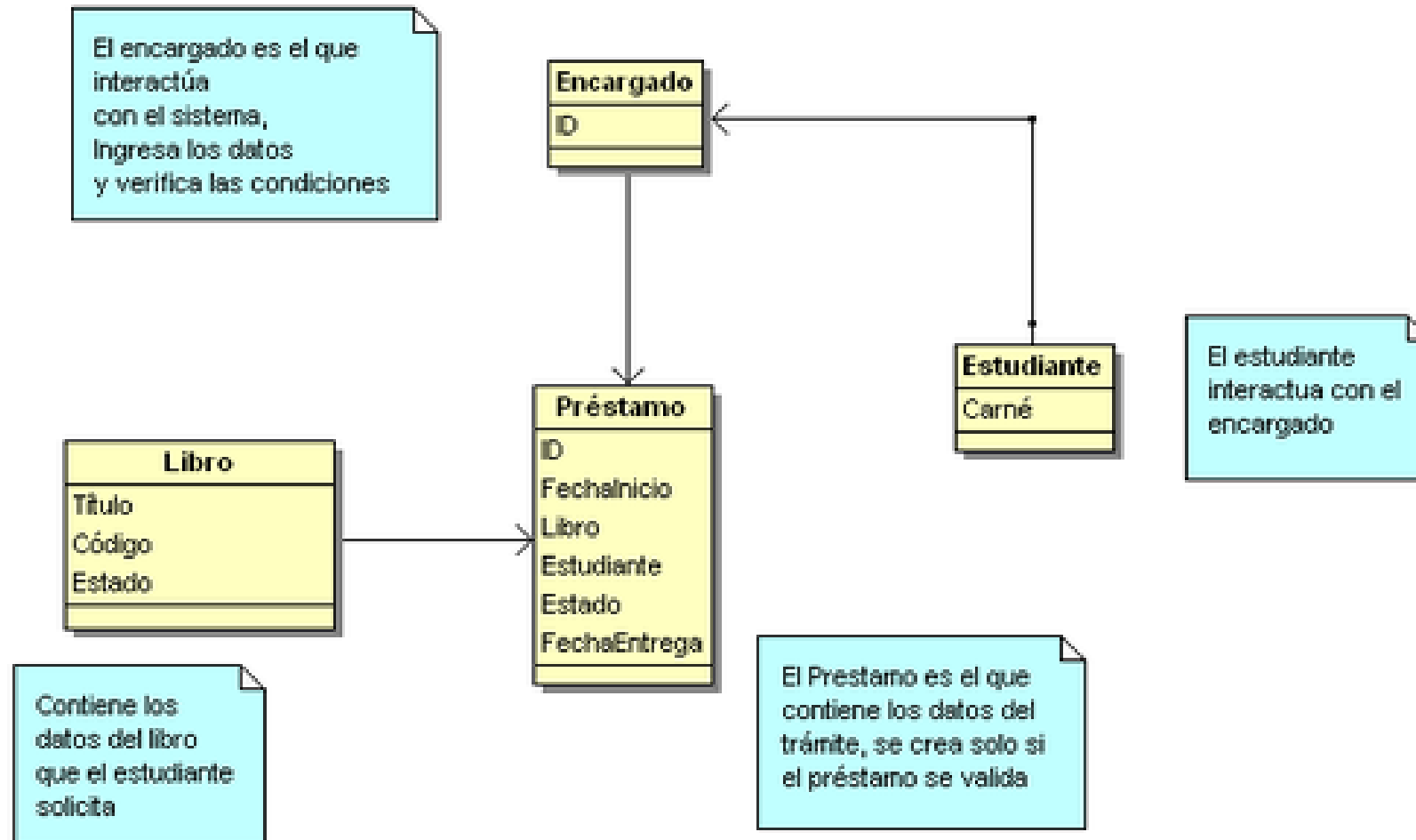
Fuente: <https://es.slideshare.net/slideshow/relaciones-entre-objetos-asociaciones/44328755>

??

Unidireccional



Modelo de Dominio:



Fuente: <https://diagramasumlerickolmososati102.weebly.com/modelado-del-dominio.html>

Ejemplo:

En la **construcción de un sistema de información** para el control hospitalario se definieron los siguientes **elementos**:

Hospital, con los datos nombre, dirección y teléfono.

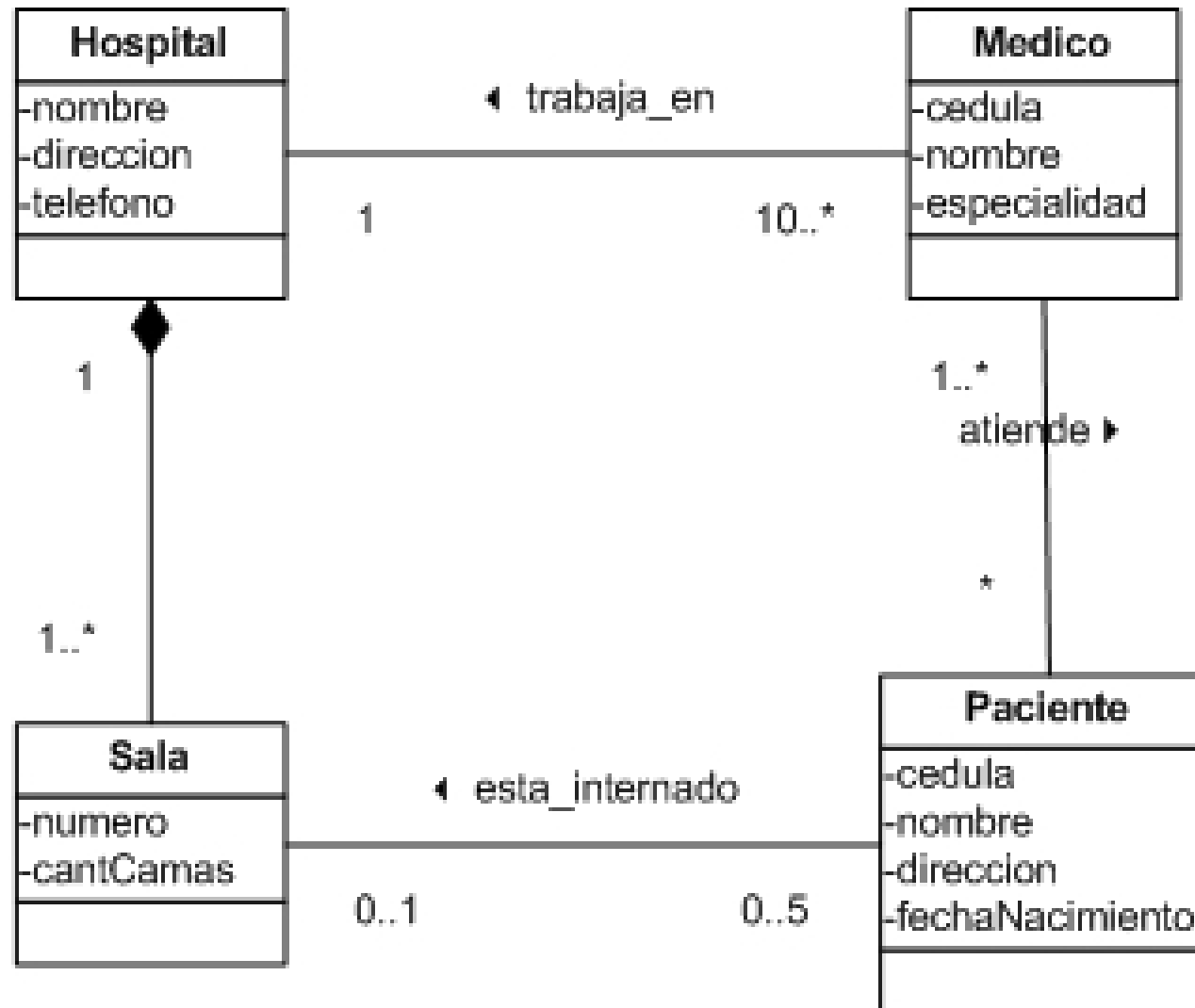
Sala, con los datos número y cantidad de camas.

Médico, con los datos cédula de identidad, nombre y especialidad.

Paciente, con los datos cédula de identidad, nombre, dirección y fecha de nacimiento.

- Cada hospital tiene varias salas. Todas y cada una de ellas pertenecen a un hospital (y solo a uno).
- Cada médico trabaja en un único hospital. Todo hospital tiene al menos 10 médicos.
- Un paciente puede estar internado; si lo está, estará en una sala (y sólo en una).
- La capacidad máxima de camas que puede tener una sala es de cinco pacientes
- Cada paciente puede ser atendido por más de un médico (pero por lo menos por uno), y a su vez cada médico puede atender varios pacientes.

Modelo Conceptual:



Gracias por su Atención ii