

Programación orientada a objetos

Sesión 1: Diseño de diagramas de clases UML

Mg. Evelyn Riveros



**Universidad
Tecnológica
del Perú**

Inicio

Silabo del curso



Revisamos el sílabo en la plataforma virtual de aprendizajes.

Logro de la sesión

Al finalizar la sesión el alumno analiza y realiza diagramas de clase mediante el uso del Lenguaje UML realizando ejercicios.

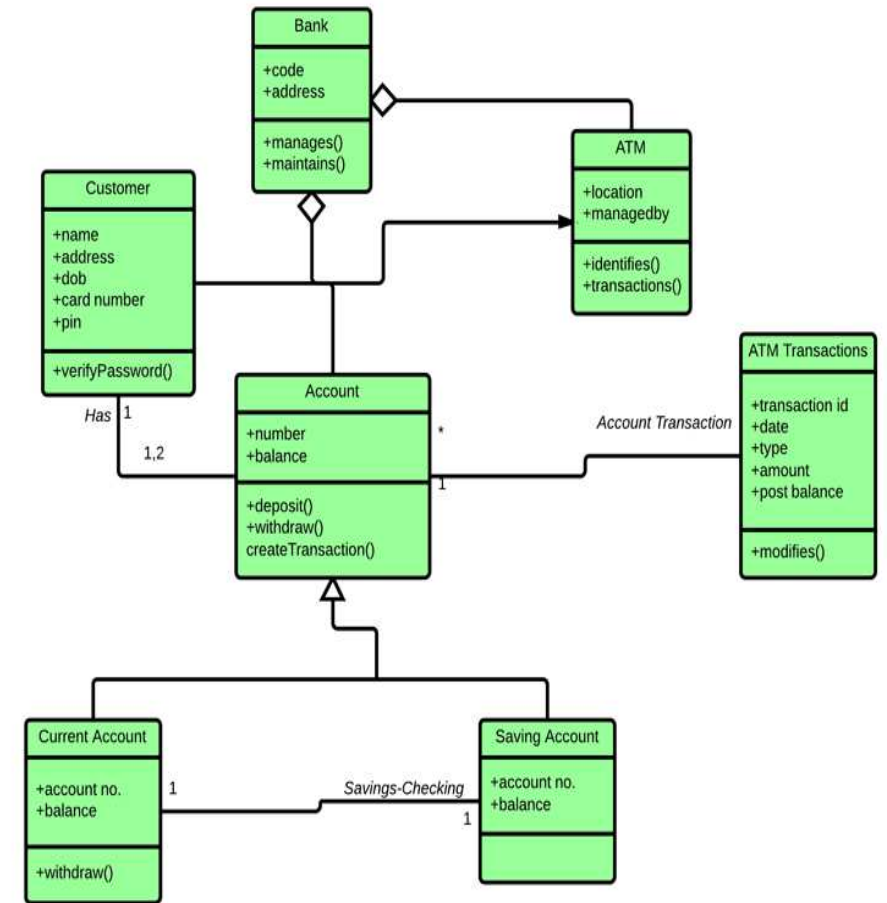
Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú

Utilidad

- Expresar visualmente cualesquier necesidades específicas de un sistema.
- Comprender mejor la visión general de los esquemas de una aplicación.



Modelo orientado a objetos

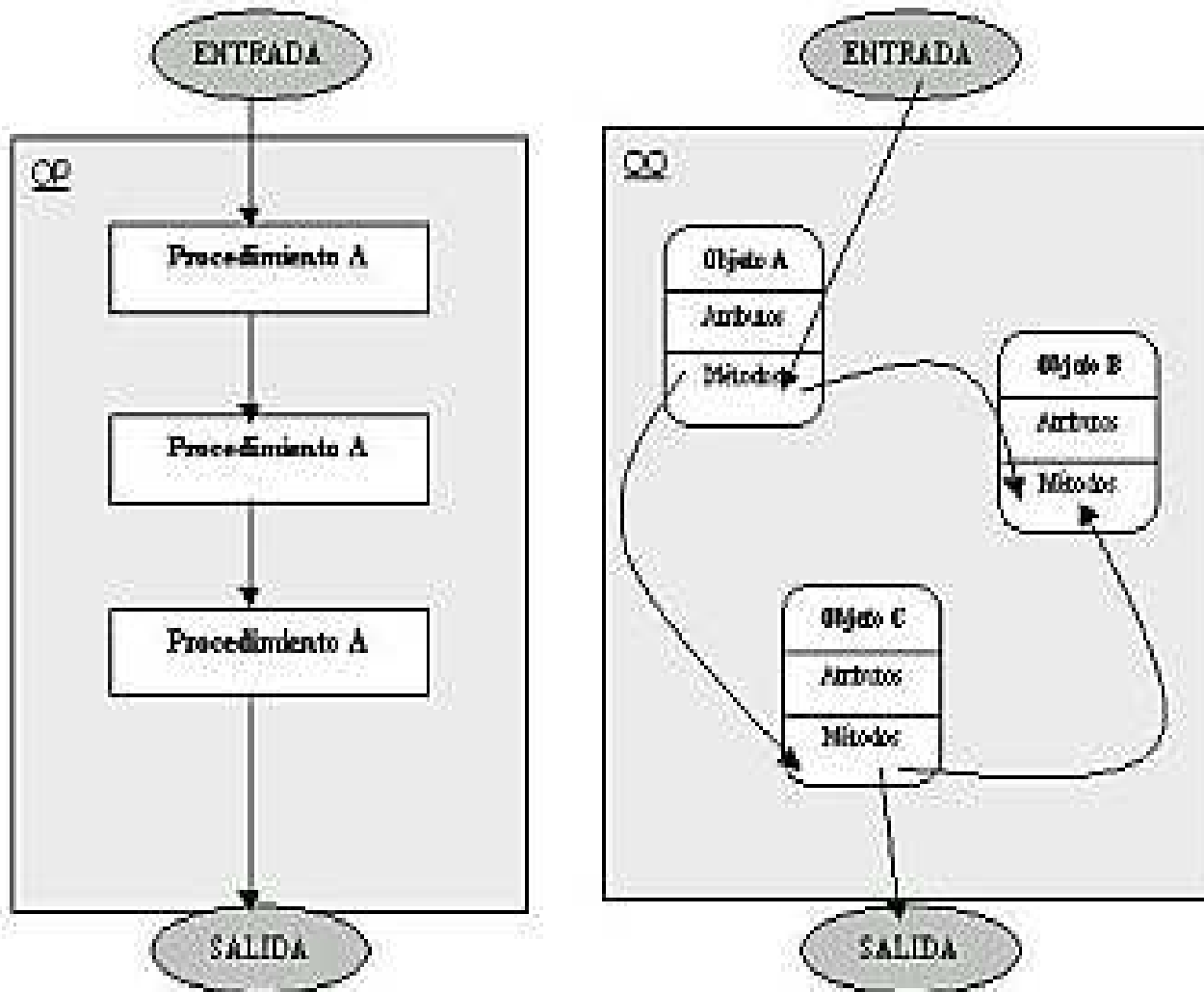
A diferencia de la orientación a procedimientos (OP), la orientación a objetos (OO) no concibe los procesos como una secuencia de procedimientos con su entrada y salida, sino que se basa en un conjunto de objetos interactuando:

Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú

Modelo orientado a objetos



```
class Perro {  
    private String nombre;  
  
    public void ladrar() {  
        System.out.println( "¡Ruff!, ¡Ruff!" );  
    }  
}  
  
public class PruebaPerro {  
    public static void main( String[] args ) {  
        Perro obj= new Perro();  
        obj.ladrar();  
    }  
}
```

UML principales diagramas

Diagrama de casos de uso

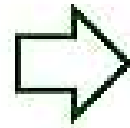
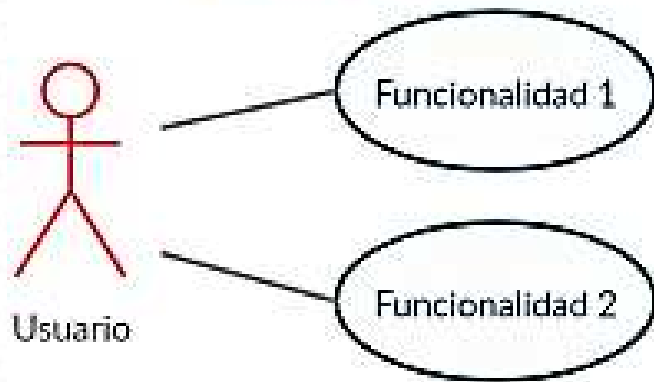


Diagrama de secuencias

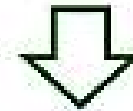
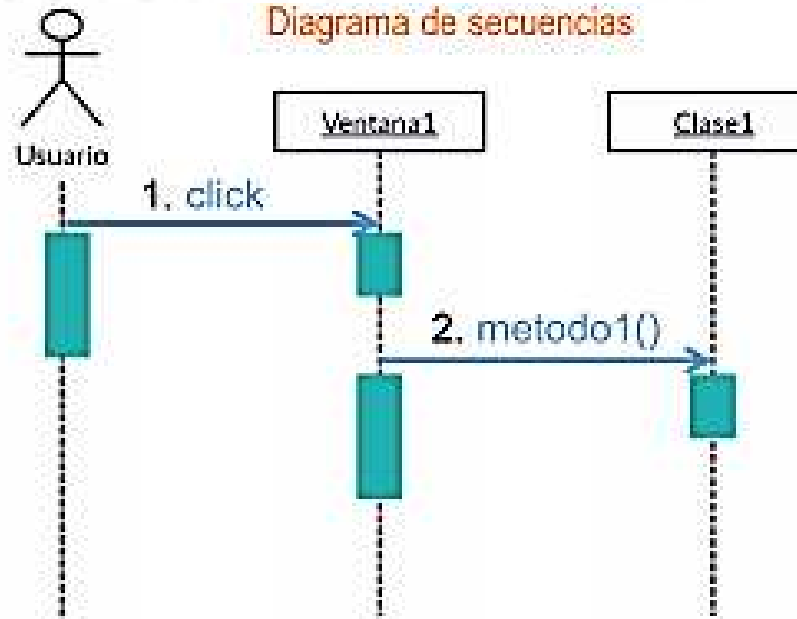
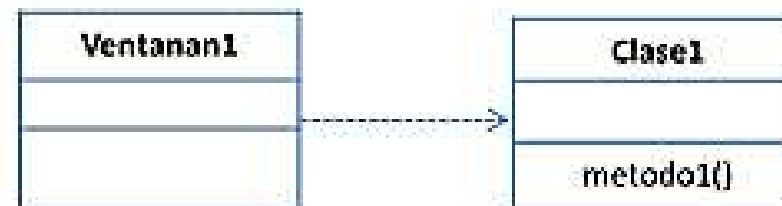


Diagrama de clases



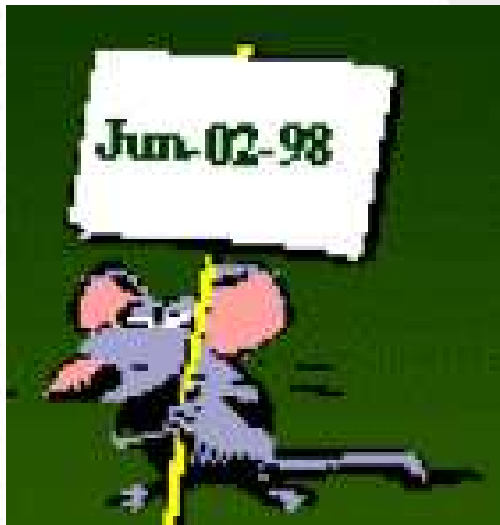
Clases y Objetos



- Una **clase** en Java es básicamente una plantilla que sirve para crear objetos.
- En Java, un **objeto** es una instancia de una clase.
- Un **programa Java Orientado a Objetos** está formado por un conjunto de clases. A partir de esas clases se crearán objetos que interactuarán entre ellos enviándose mensajes para resolver un problema.

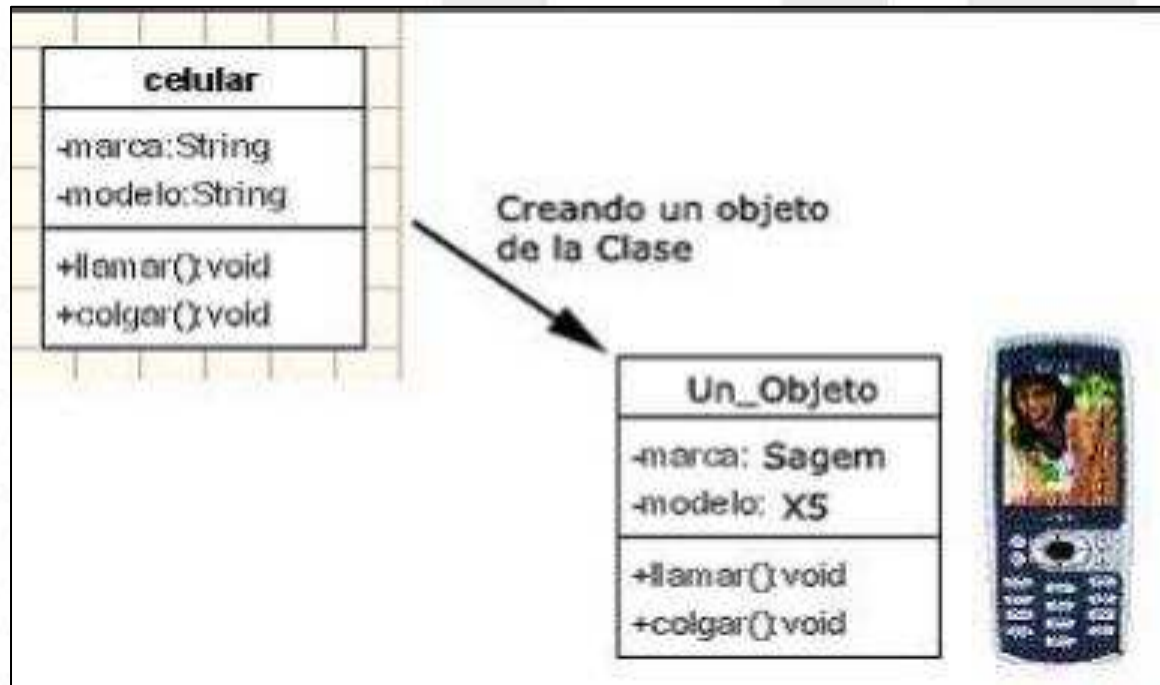
¿QUE ES UN OBJETO?

Un objeto podría ser real o abstracto , por ejemplo un bebe , una factura, una medida, una fecha ,etc.

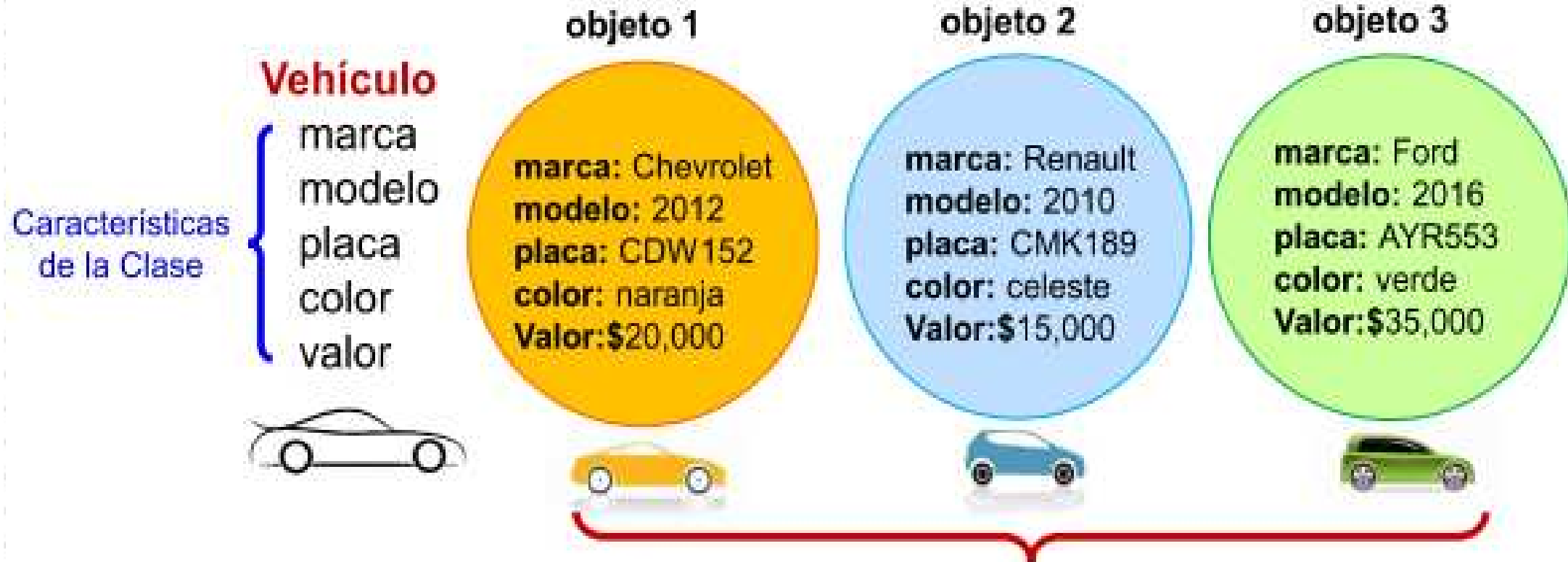


¿QUÉ ES UNA CLASE?

Conjunto de objetos que poseen características similares, es decir objetos del mismo tipo.

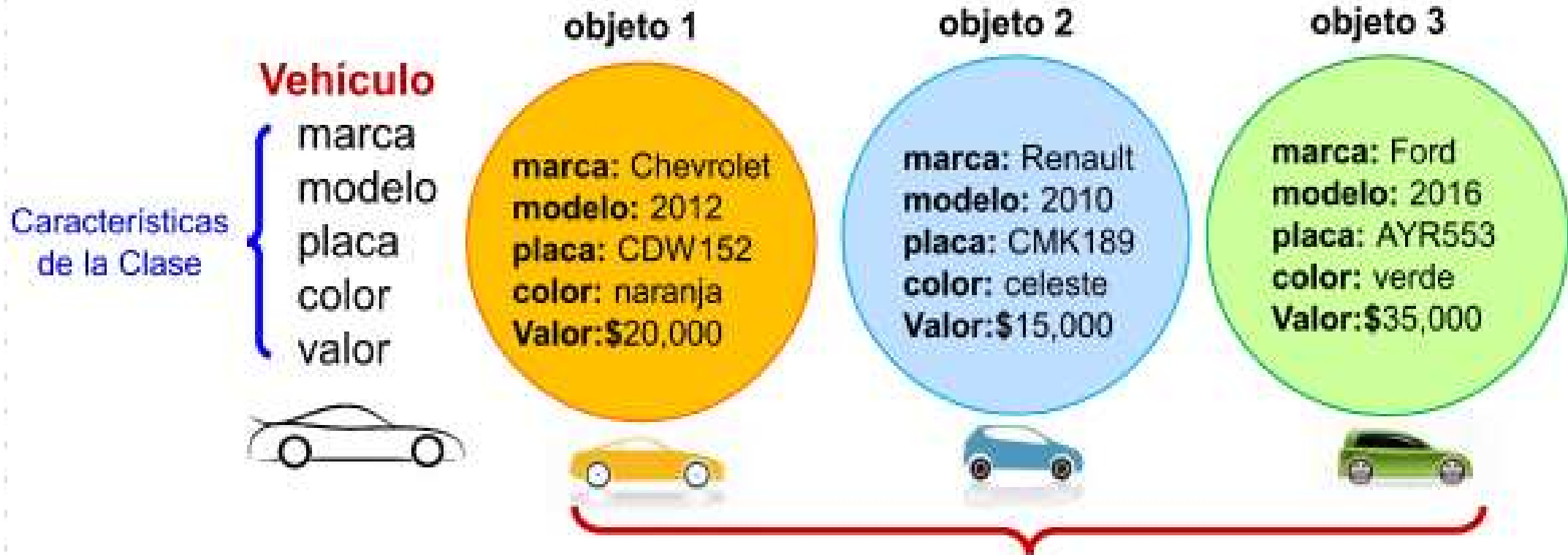


Ejemplos de Clases y Objetos



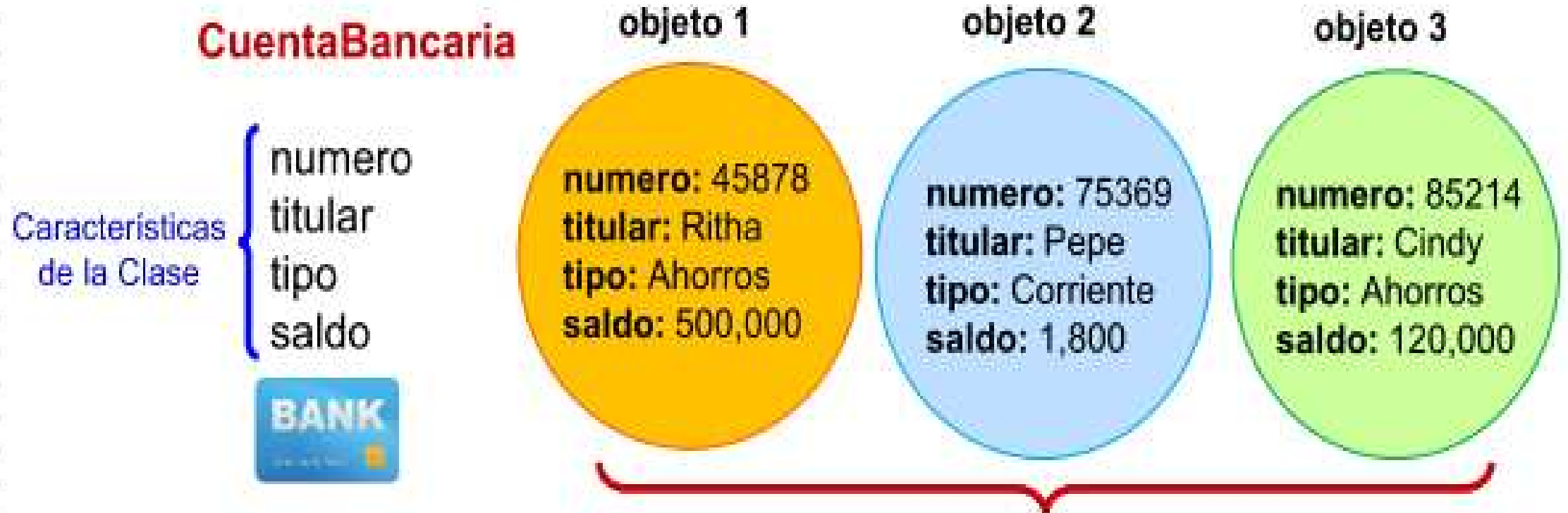
Cada objeto tiene valores propios para cada una de las características de la clase.

Ejemplos de Clases y Objetos



Los objetos que existen en el mundo real
pueden ser **conceptuales** o **físicos**.

Ejemplos de Clases y Objetos

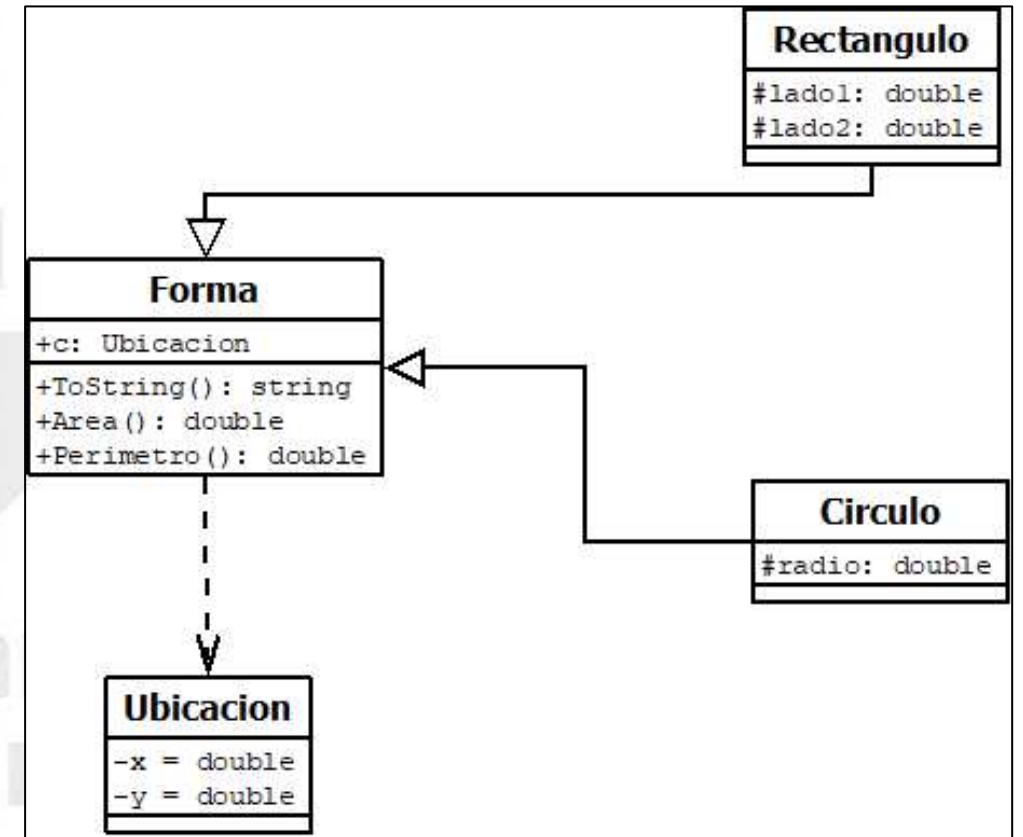


Los **objetos conceptuales** existen en el mundo real pero no son tangibles.

Diagramas de clase

Es una herramienta para comunicar el diseño de un programa que se creó para orientar objetos y que permite modelar relaciones entre diferentes entidades.

En UML, una clase se representa con un rectángulo que posee tres divisiones, nombre de la clase, atributos que tiene y mensajes que entiende.



Diagramas de clase



https://www.youtube.com/watch?v=M0S3AEPP_bw

Diagrama de clase y su representación

El diagrama de clase es la descripción de un conjunto de objetos que comparten los mismos atributos, operaciones y relaciones.

Se representa mediante un rectángulo con hasta 3 compartimientos.

- Nombre de la clase
- Lista de atributos
- Lista de métodos

Diagrama de clase y su representación

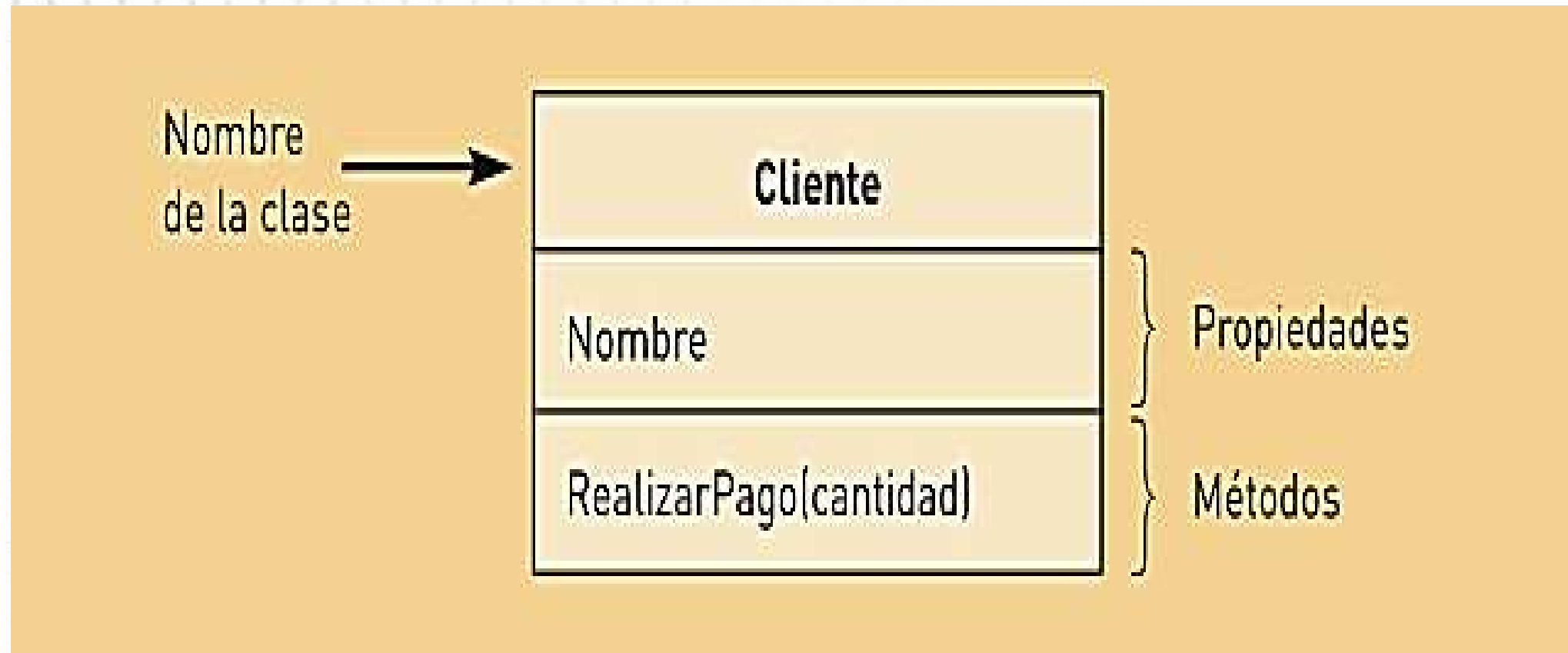


Diagrama de clase y su representación

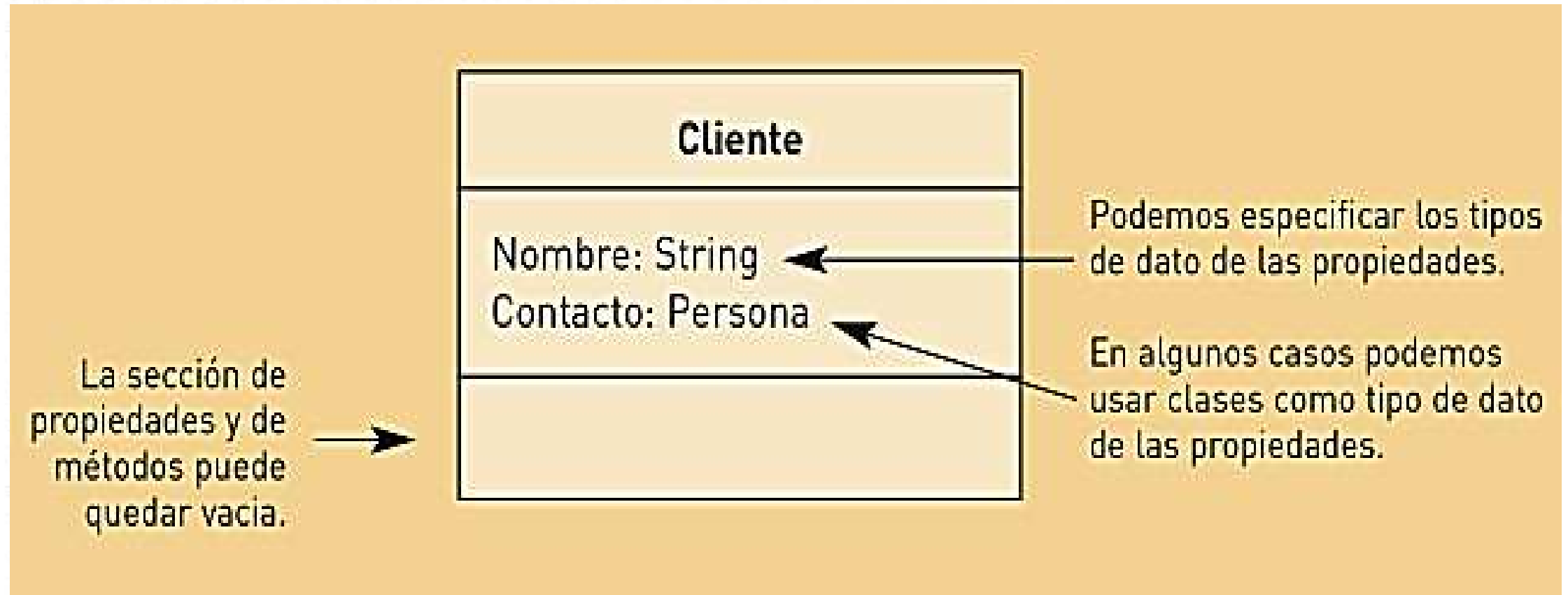


Diagrama de clase: Atributos

- Atributo: Son valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una persona, sus propiedades serían: nombre, edad, sexo, etc.
- Tipo: Puede llegar a depender del lenguaje de programación a utilizar (public, protected, private).
- Valor inicial: valor que poseerá el atributo al crear un objeto.

Diagrama de clase: Atributos

- **Multiplicidad:** Determinar si un atributo debe estar o no, y si posee un único valor o una lista de valores.
 - 1-> El atributo debe tener un único valor.
 - 0..1-> El atributo puede o no tener un valor.
 - 0..* -> El atributo puede tener varios valores o ninguno.
 - 1..*-> El atributo puede tener varios valores, pero debe tener al menos uno
 - *-> El atributo puede tener varios valores.
 - M..N -> El atributo puede tener entre M y N valores.

Diagrama de clase: Atributos

- Visibilidad
- El encapsulamiento presenta las siguientes ventajas básicas:
 - Se protegen los datos de accesos indebidos
 - Favorece la modularidad y el mantenimiento
- Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos.

Diagrama de clase: Atributos

Niveles de encapsulamiento

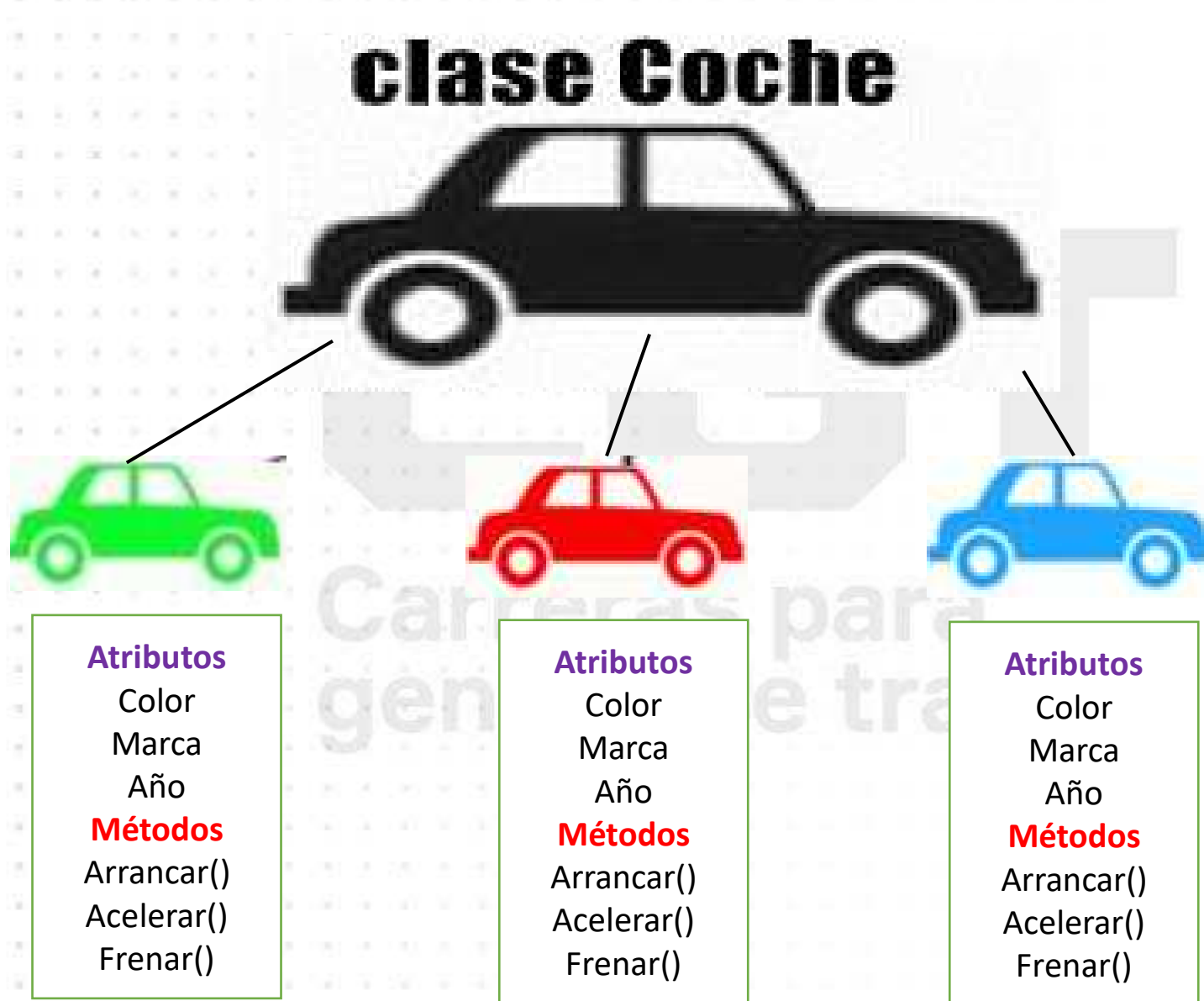
Todas las clases exhiben un nivel de modificador de acceso variable en función de su visibilidad. Los marcadores de visibilidad limitan a las personas que pueden acceder a la información en una clase en particular. A continuación, se presentan los marcadores de visibilidad y su información de acceso:

Marcador	Visibilidad	Descripción
+	Público	Todas las clases pueden ver la información.
-	Privado	La información está oculta a todas las clases fuera de la partición.
#	Protegido	La clase secundaria puede acceder a la información heredada de la clase principal.

Diagrama de clase: Métodos

- Un método (operación) es la implementación de un servicio que puede ser requerido a cualquier objeto de la clase para que muestre un comportamiento. Una operación es una abstracción de algo que se puede hacer a un objeto y que es compartido por todos los objetos de la clase.
- Para cada método debe especificarse:
 - **Tipo devuelto:** Puede llegar a depender del lenguaje de programación a utilizar.
 - **Parámetros:** Especificación del tipo de datos y la información que determina el funcionamiento de la clase.
 - **Visibilidad:** Está relacionada con el encapsulamiento (-, #, +).

Ejemplo de clase



Atributos

Color
Marca
Año

Métodos

Arrancar()
Acelerar()
Frenar()

Diagrama de clase: Relaciones

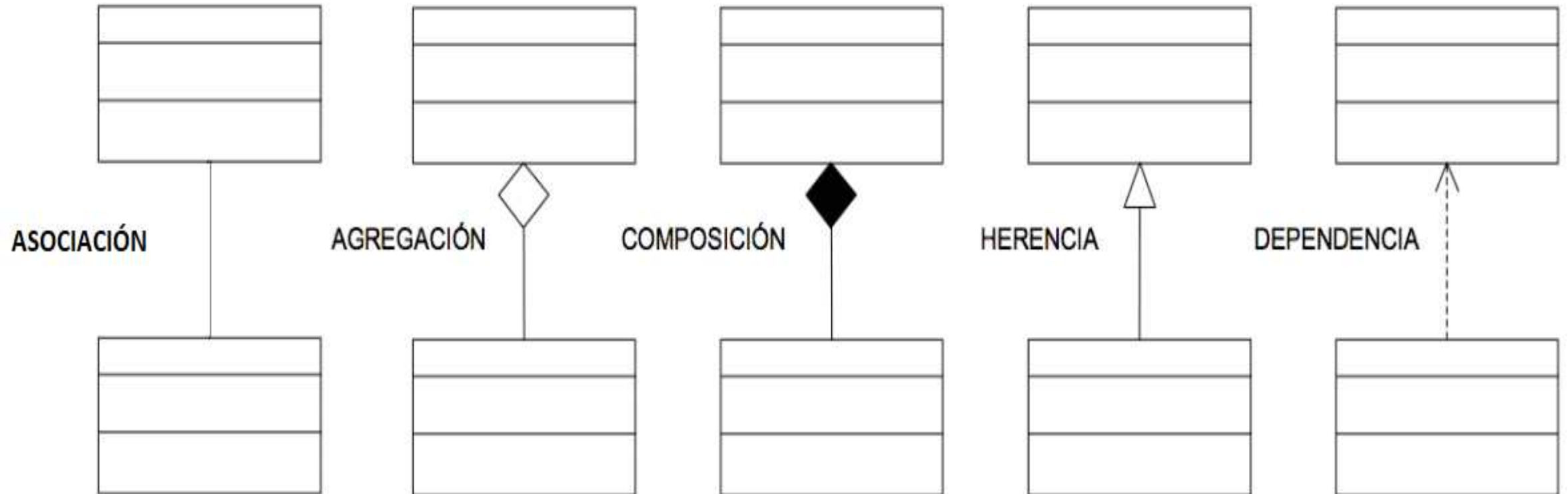
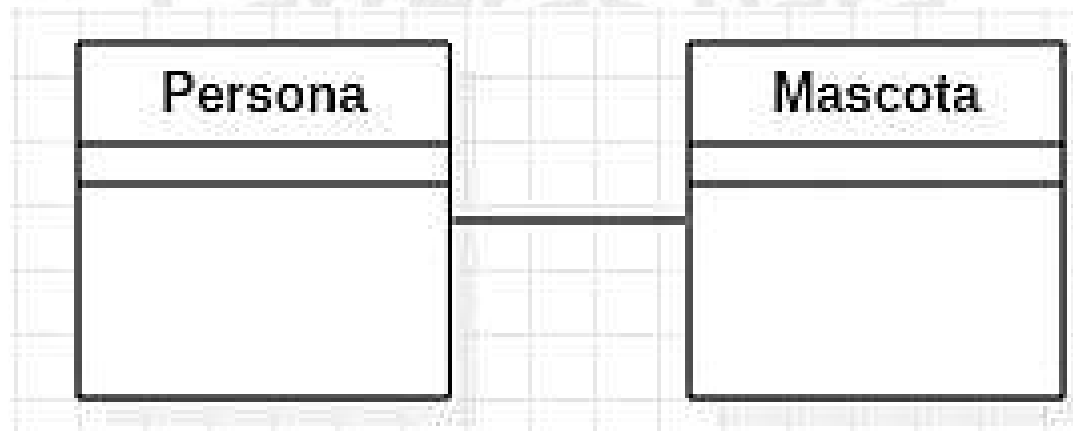


Diagrama de clase: Relaciones

Asociación

Este tipo de relación es el más común y se utiliza para representar dependencia semántica. Se representa con una simple línea continua que une las clases que están incluidas en la asociación.

Un ejemplo de asociación podría ser: «Una mascota pertenece a una persona».



Ejemplo de asociación

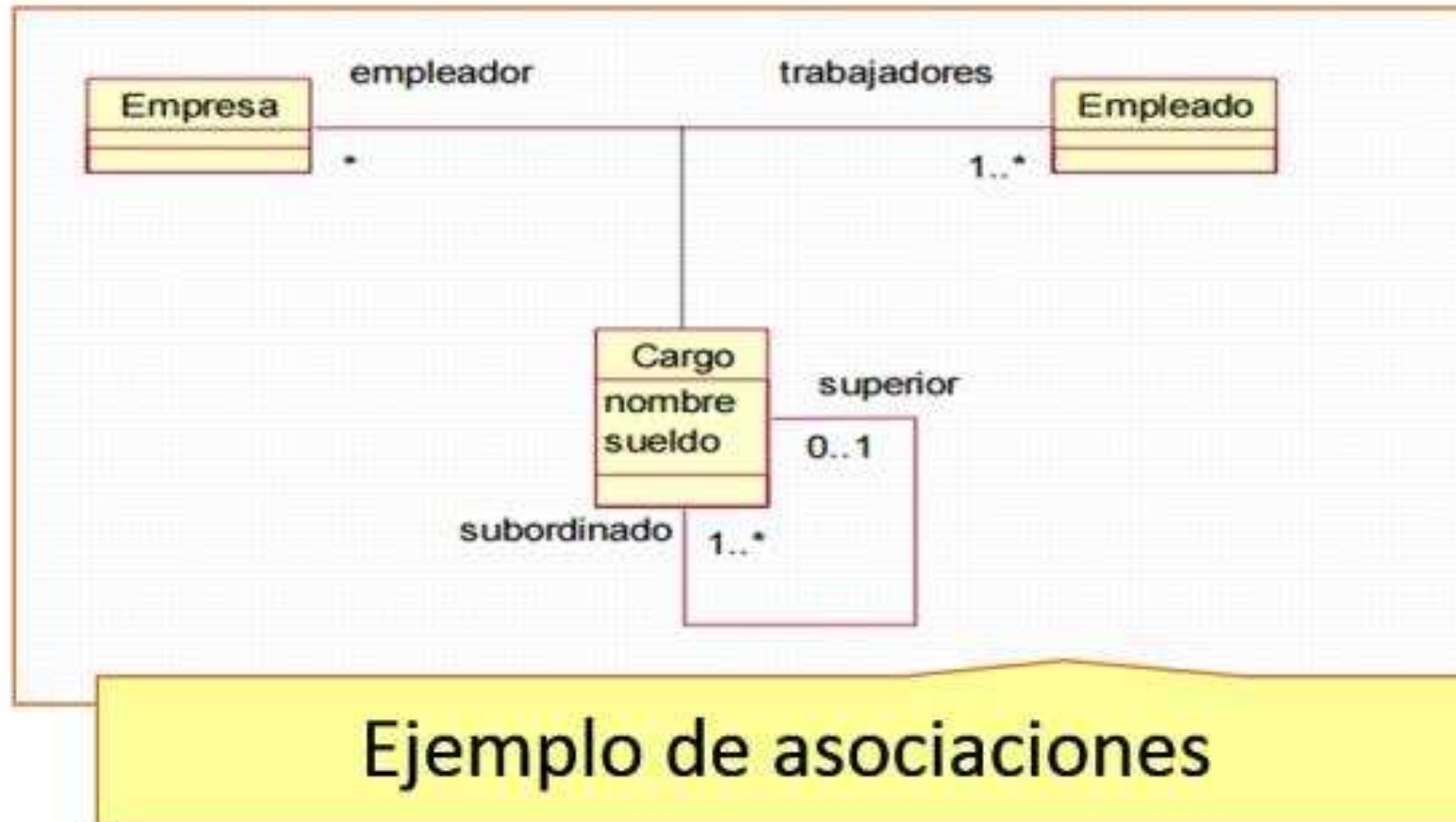


Diagrama de clase: Relaciones

Agregación

Es una representación jerárquica que indica a un objeto y las partes que componen ese objeto. Es decir, representa relaciones en las que un objeto es parte de otro, pero aun así debe tener existencia en sí mismo.

Se representa con una línea que tiene un rombo en la parte de la clase que es una agregación de la otra clase (es decir, en la clase que contiene las otras).

Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú

Diagrama de clase: Relaciones

Un ejemplo de esta relación podría ser: «Las mesas están formadas por tablas de madera y tornillos o, dicho de otra manera, los tornillos y las tablas forman parte de una mesa». Como ves, el tornillo podría formar parte de más objetos, por lo que interesa especialmente su abstracción en otra clase.

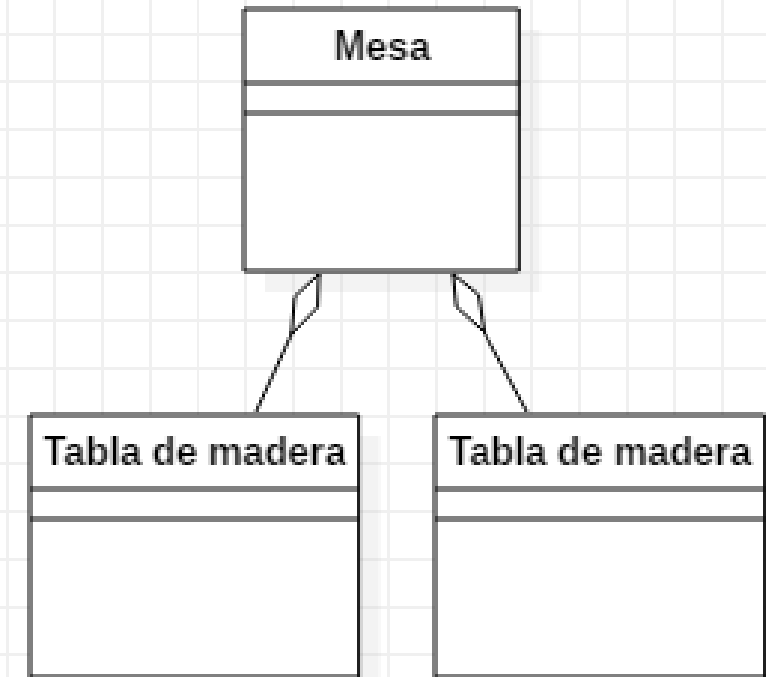


Diagrama de clase: Relaciones

Composición

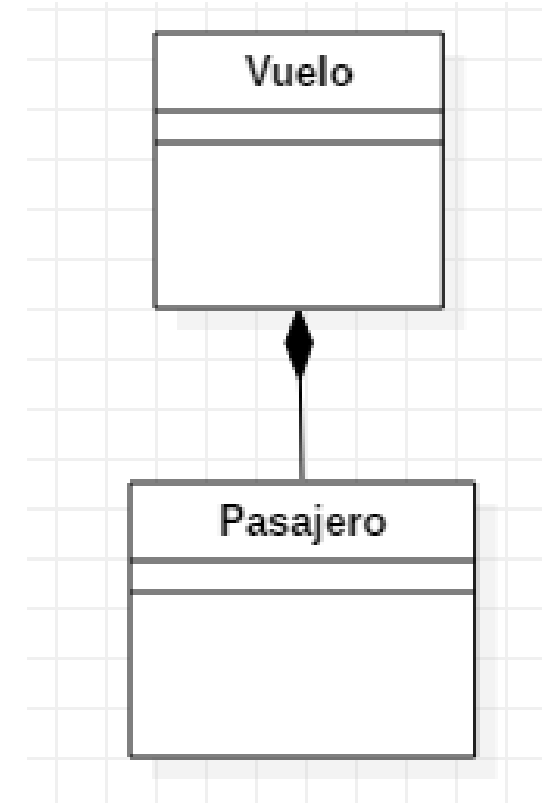
La composición es similar a la agregación, representa una relación jerárquica entre un objeto y las partes que lo componen, pero de una forma más fuerte.

En este caso, los elementos que forman parte no tienen sentido de existencia cuando el primero no existe. Es decir, cuando el elemento que contiene los otros desaparece, deben desaparecer todos ya que no tienen sentido por sí mismos sino que dependen del elemento que componen. Además, suelen tener los mismos tiempo de vida. Los componentes no se comparten entre varios elementos, esta es otra de las diferencias con la agregación.

Diagrama de clase: Relaciones

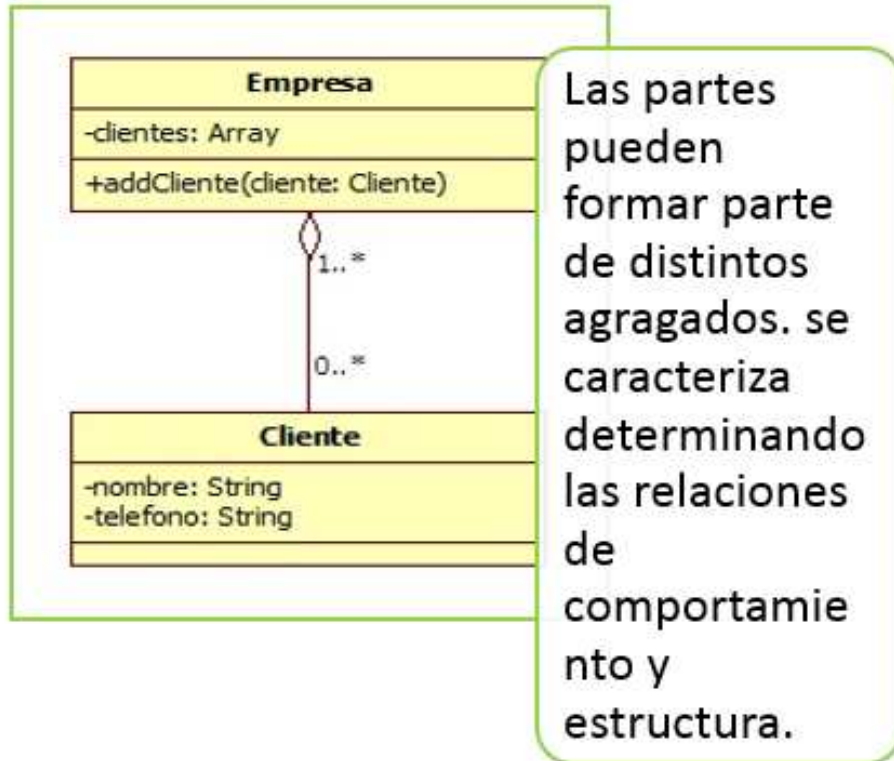
Se representa con una línea continua con un rombo relleno en la clase que es compuesta.

Un ejemplo de esta relación sería: «Un vuelo de una compañía aérea está compuesto por pasajeros, que es lo mismo que decir que un pasajero está asignado a un vuelo»



Ejemplo de agregación y composición

Agregación



Composición

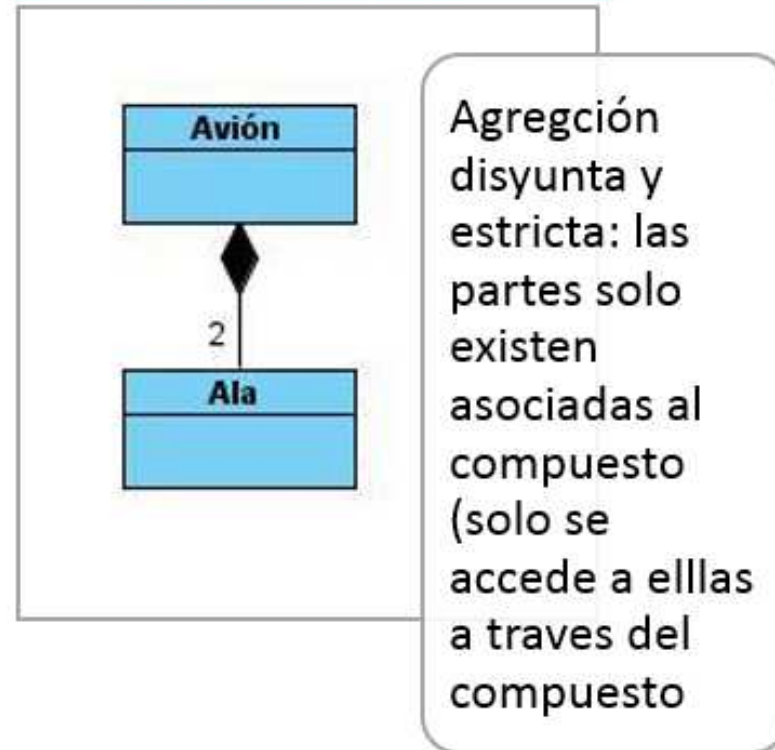
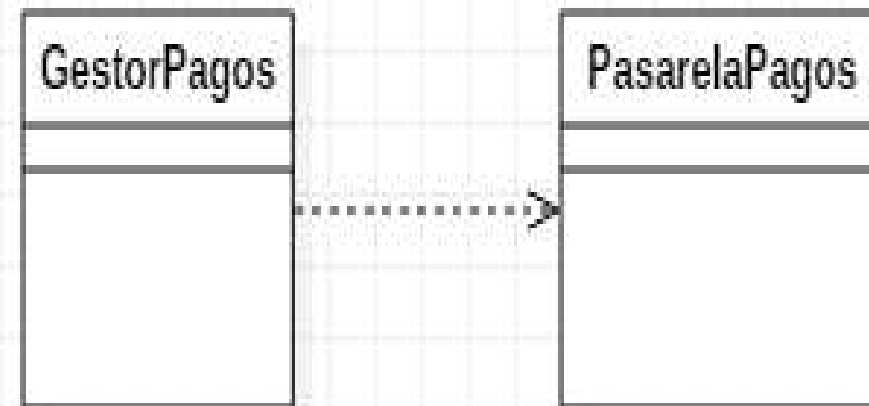


Diagrama de clase: Relaciones

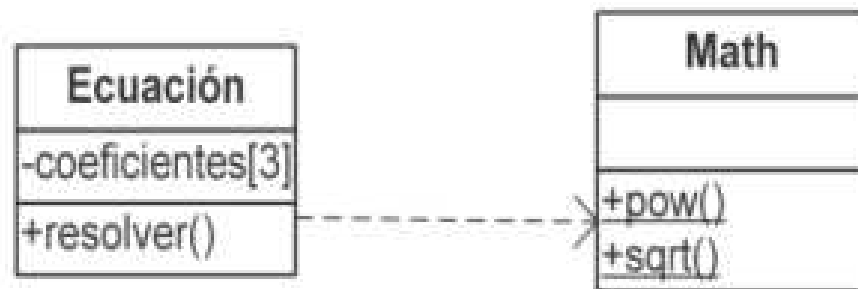
Dependencia

Se utiliza este tipo de relación para representar que una clase requiere de otra para ofrecer sus funcionalidades. Es muy sencilla y se representa con una flecha discontinua que va desde la clase que necesita la utilidad de la otra flecha hasta esta misma.

Un ejemplo de esta relación podría ser la siguiente:



Ejemplo de Dependencia



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Resolución de una ecuación de segundo grado.

Para resolver una ecuación de segundo grado hemos de recurrir a la función *sqrt* de la clase *Math* para calcular la raíz cuadrada.

Diagrama de clase: Relaciones

Herencia

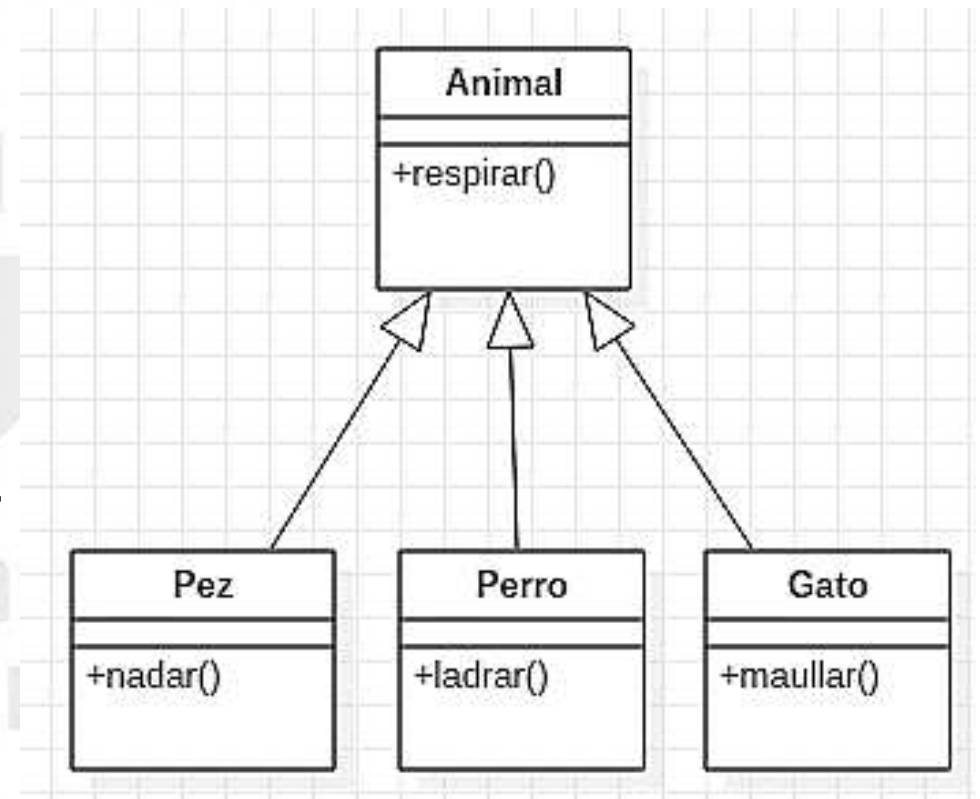
Otra relación muy común en el diagrama de clases es la herencia. Este tipo de relaciones permiten que una clase (clase hija o subclase) reciba los atributos y métodos de otra clase (clase padre o superclase). Estos atributos y métodos recibidos se suman a los que la clase tiene por sí misma. Se utiliza en relaciones «es un».

Un ejemplo de esta relación podría ser la siguiente: Un pez, un perro y un gato son animales.

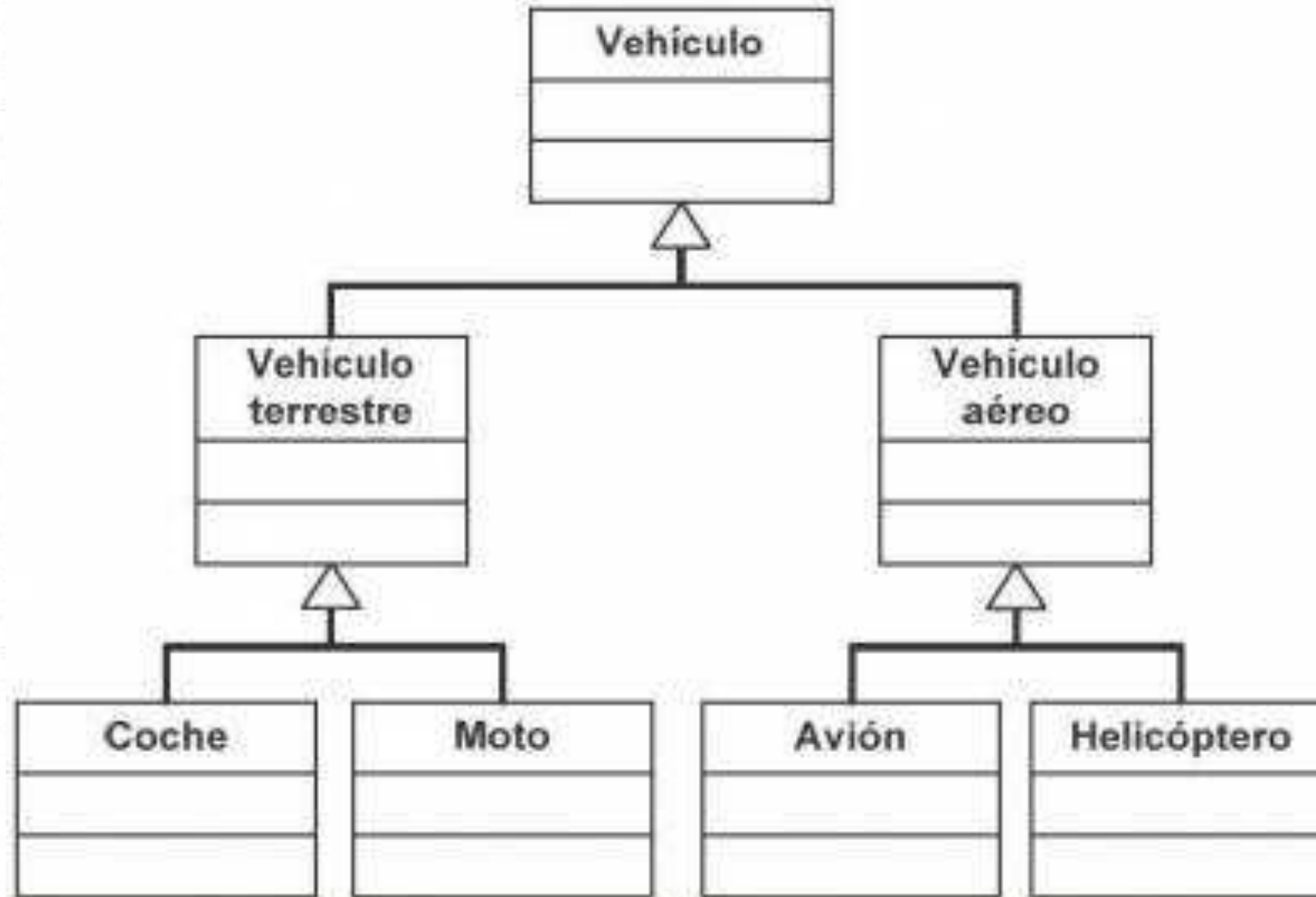
Diagrama de clase: Relaciones

En este ejemplo, las tres clases (Pez, Perro, Gato) podrán utilizar la función respirar, ya que lo heredan de la clase animal, pero solamente la clase Pez podrá nadar, la clase Perro ladrar y la clase Gato maullar.

La clase Animal podría plantearse ser definida abstracta, aunque no es necesario.



Ejemplo de Herencia



Sintaxis de una clase en Java

También denominadas
atributos o campos
(fields)

```
public class Persona {
```

```
    private String nombre;  
    private int edad;
```

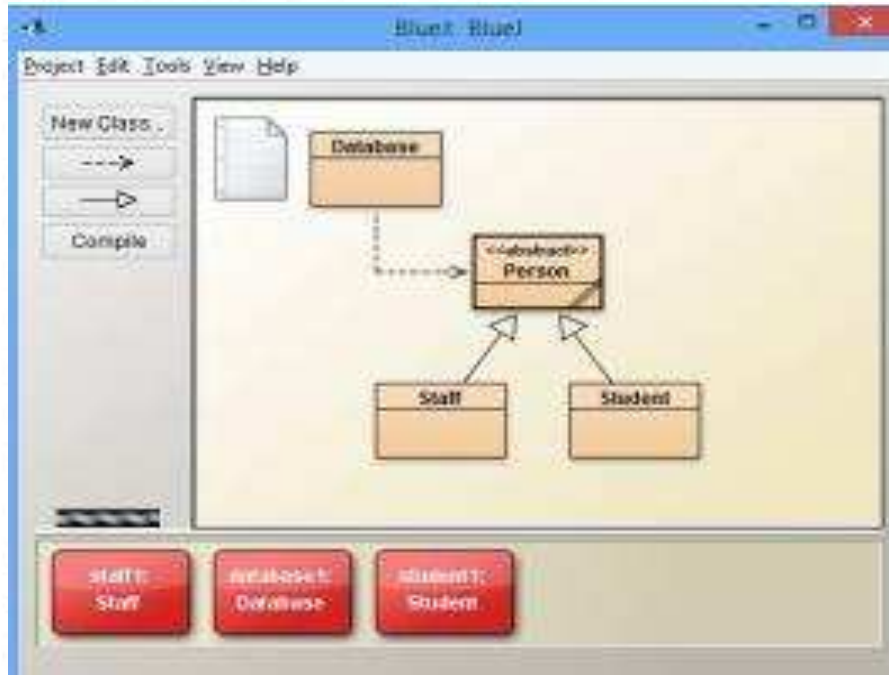
Métodos de la
clase

```
    public void inicializar() {...}  
    public void imprimir() {...}  
    public void esMayorEdad() {...}  
    public static void main(String[ ] ar) {...}
```

```
}
```

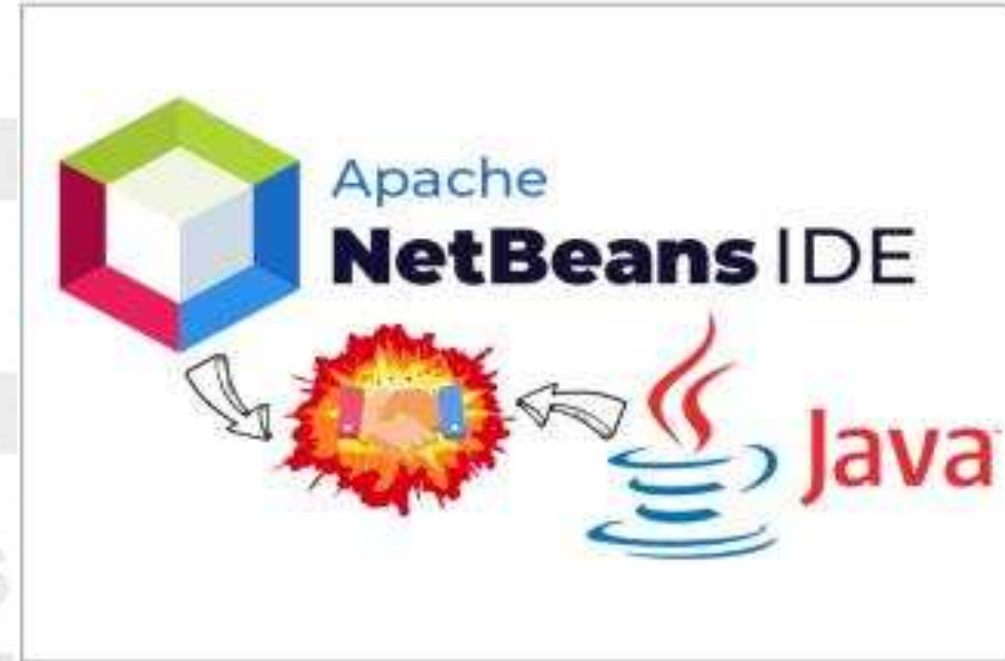
Este método sirve para
definir qué ocurre
cuando se crea un
objeto.

Software que utilizaremos en este curso



Descarga:

<https://www.bluej.org/versions.html>



Descarga:

<https://netbeans.apache.org/download/archive/index.html>

Practica 1: Clases y Objetos

Relaciona clase con sus objetos:

**ANIMALES
DOMESTICOS**

TRABAJADORES

GATO

PERRO

PROFESOR

SECRETARIA

DIRECTOR

CONEJO



Universidad
Tecnológica
del Perú

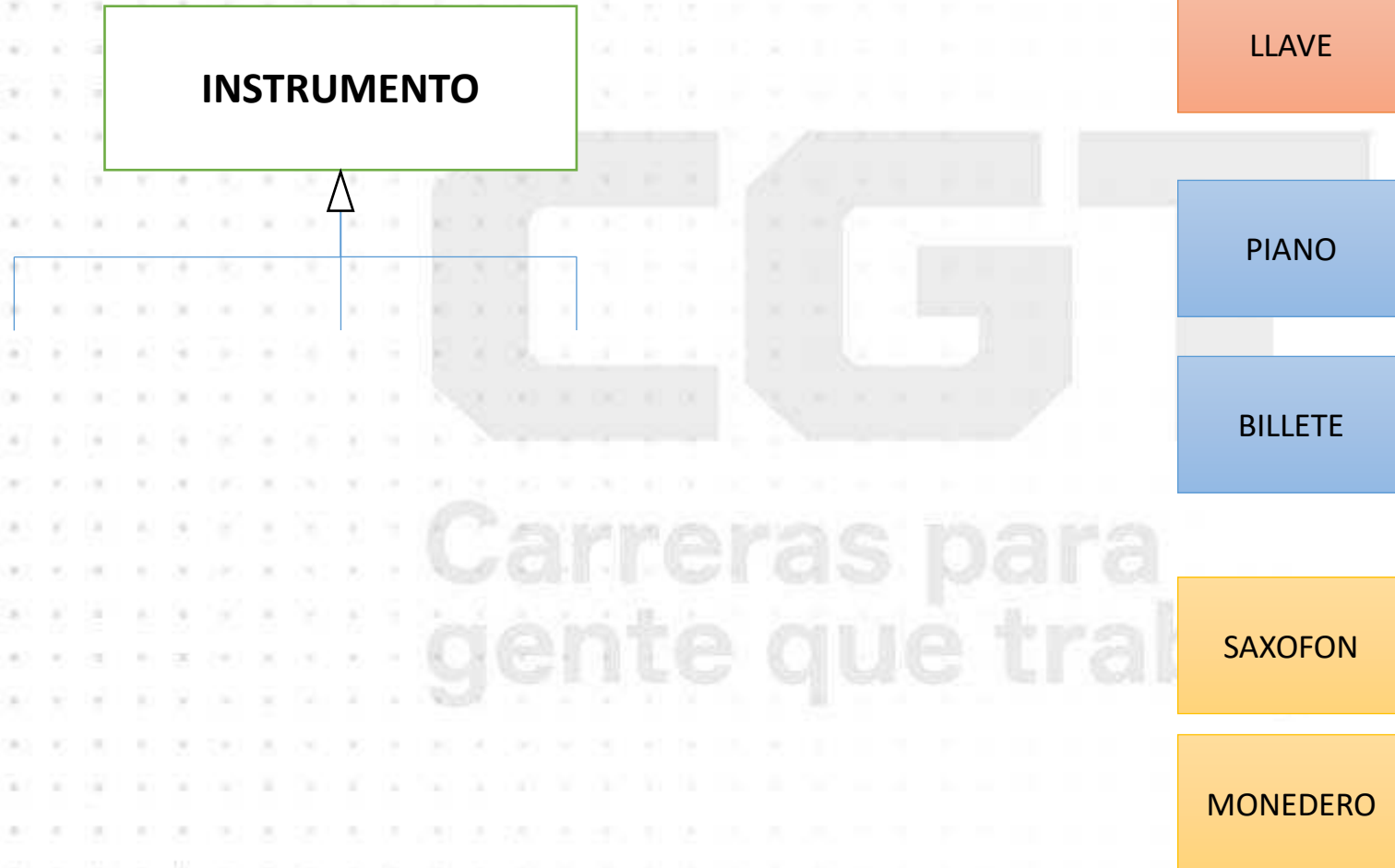
Practica 2: Clases

Relaciona clase con sus objetos:



Practica 3: Herencia

Relaciona :



Tarea

- Plantea 5 ejemplos de clases y objetos
- Plantea 5 ejemplos de asociación, agregación, dependencia, composición, herencia

Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú

CIERRE

¿Qué hemos aprendido hoy?

CGT

Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú

Prueba

Ahora se responderá las siguientes preguntas:

<https://forms.gle/AAM9jXyM8HAooxoQ6>

Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú

CGT

Carreras para
gente que trabaja



Universidad
Tecnológica
del Perú