

ESTRUCTURA DE DATOS

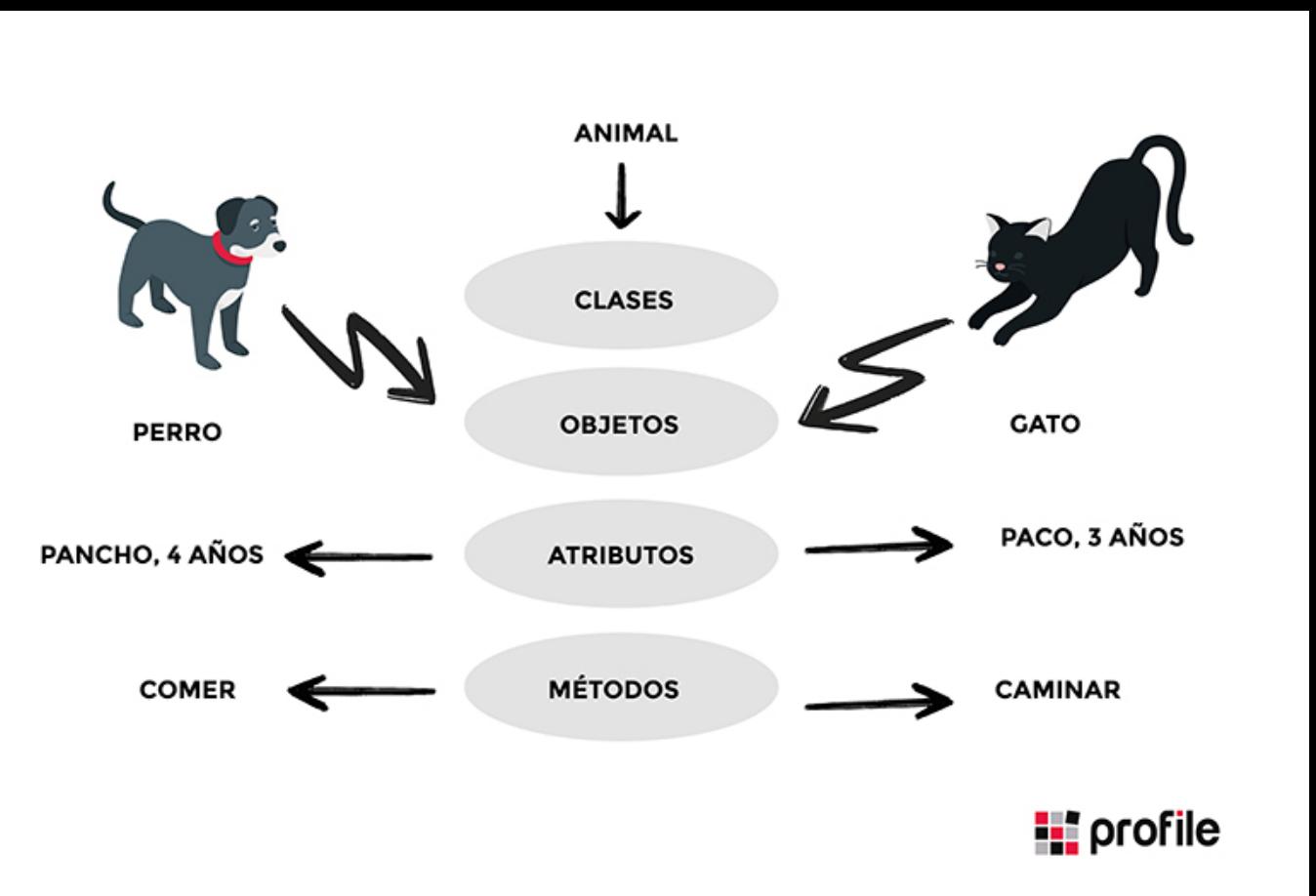
POO VARIABLES, ARRAYS, CLASES Y PACKAGES

2022 - ROMARIO TOLA QUISPE

The Java logo consists of the word "java" in a bold, red, sans-serif font. A small trademark symbol (TM) is located at the top right of the letter "a". Above the text, there is a stylized graphic element: a blue flame-like shape at the top, which sits atop a series of five nested, blue, swirling arcs.

¿A que se refiere cuando se habla de POO?

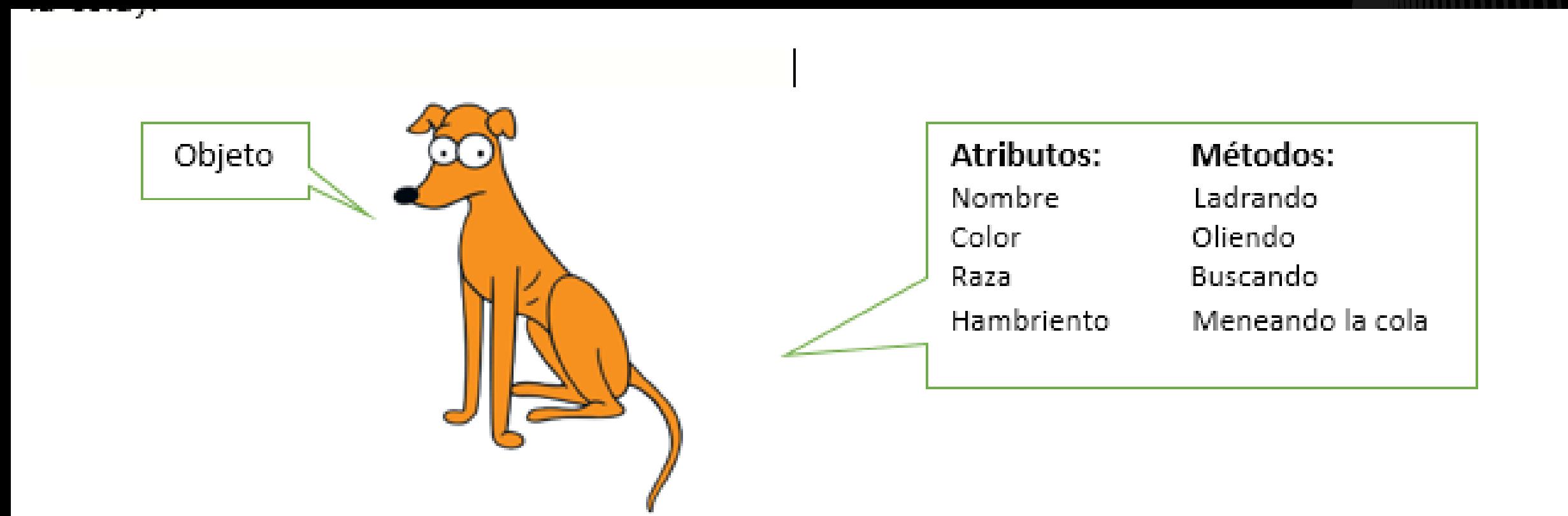
La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.



¿Cuáles son los 4 componentes que componen POO?

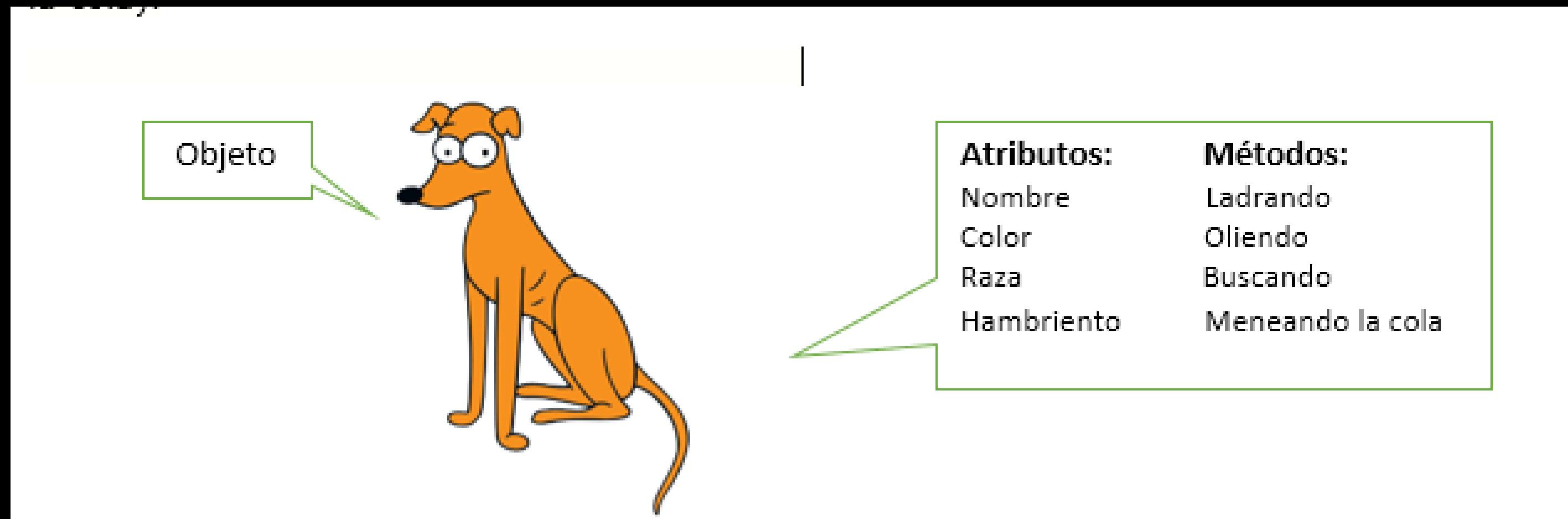
Clases: Las clases pueden ser definidas como un molde que contendrá todas las características y acciones con las cuales podemos construir N cantidad de objetos.

Objetos: Son aquellos que tienen propiedades y comportamientos, estos pueden ser físicos o conceptuales.



Métodos: Los métodos son las acciones que una clase puede realizar, siguiendo el mismo ejemplo anterior, estas podrían ser: caminar, comer, dormir, soñar, respirar, nadar, etc.

Propiedades: Las propiedades son las características de una clase, tomando como ejemplo la clase humanos, las propiedades podrían ser: nombre, el género, la altura, color de cabello, color de piel, etc.



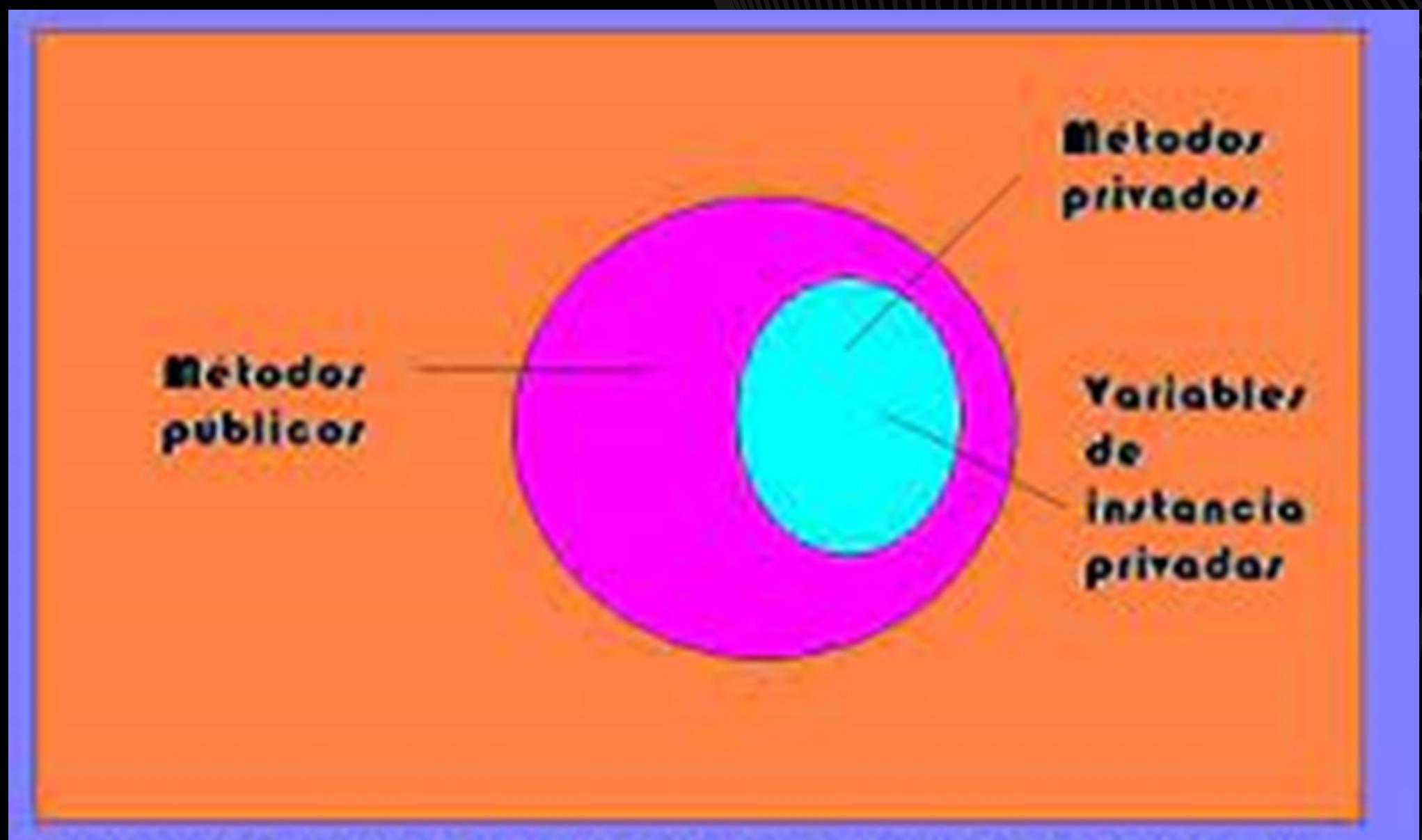
¿Cuáles son los pilares de POO?



¿Qué es Encapsulamiento?

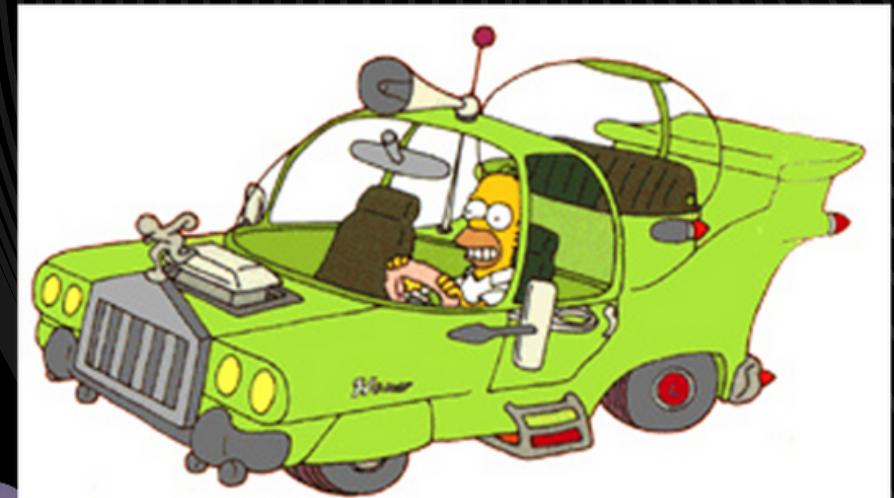
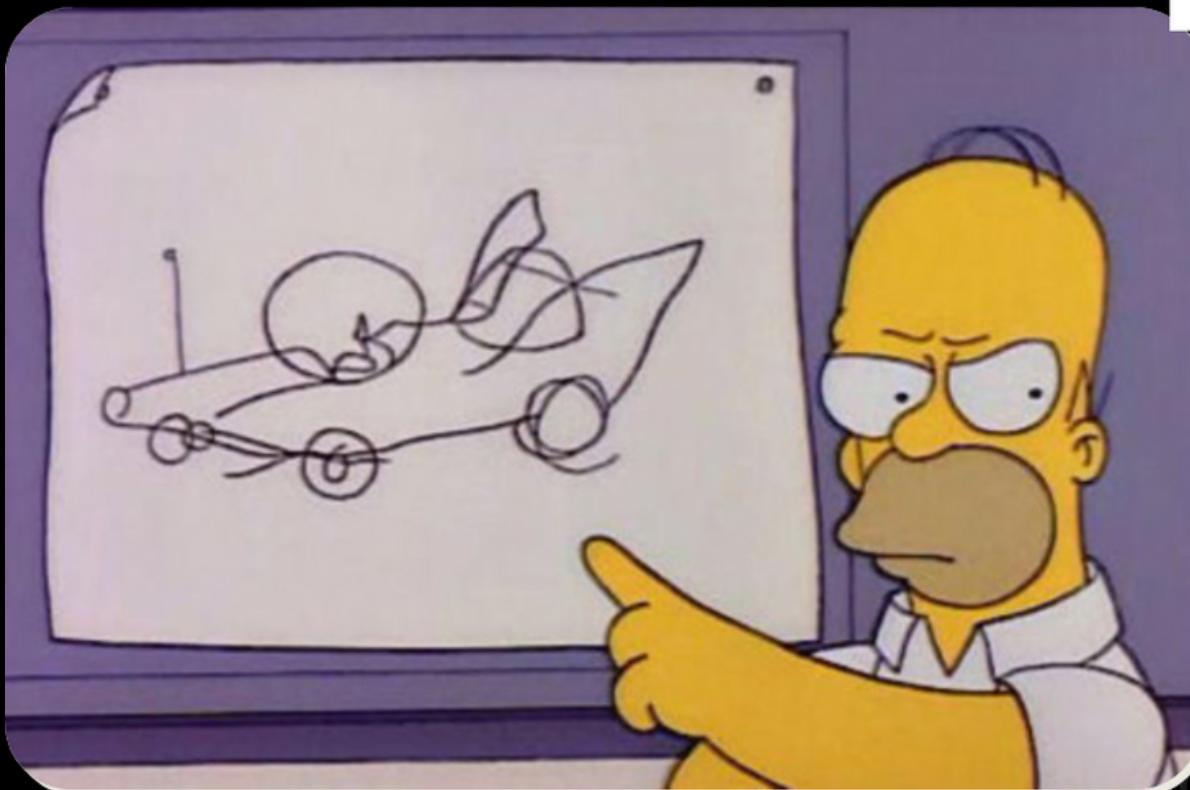
Lo puedes utilizar cuando deseas que ciertos métodos o propiedades sean inviolables o inalterables.

Un ejemplo del encapsulamiento podría ser una cuenta de banco, donde el usuario no puede simplemente aumentar su balance de dinero, si no que debe depender de unos métodos previamente validados para aumentar dicho balance (depósitos, transferencias, etc).



¿Qué es Abstracción?

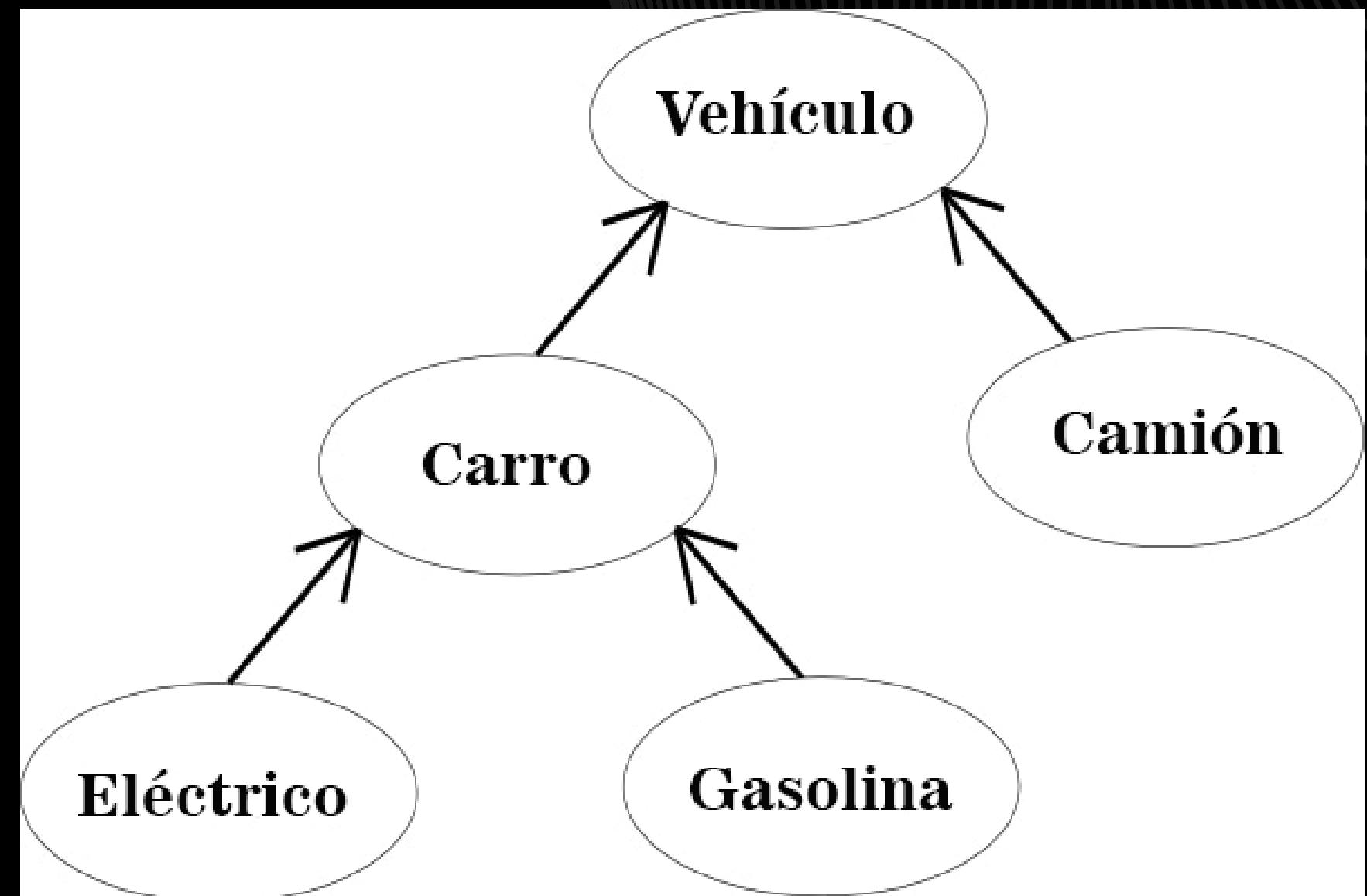
LEs cuando separamos los datos de un objeto para luego generar un molde (una clase).



Énfasis
dado a
que

¿Qué es Herencia?

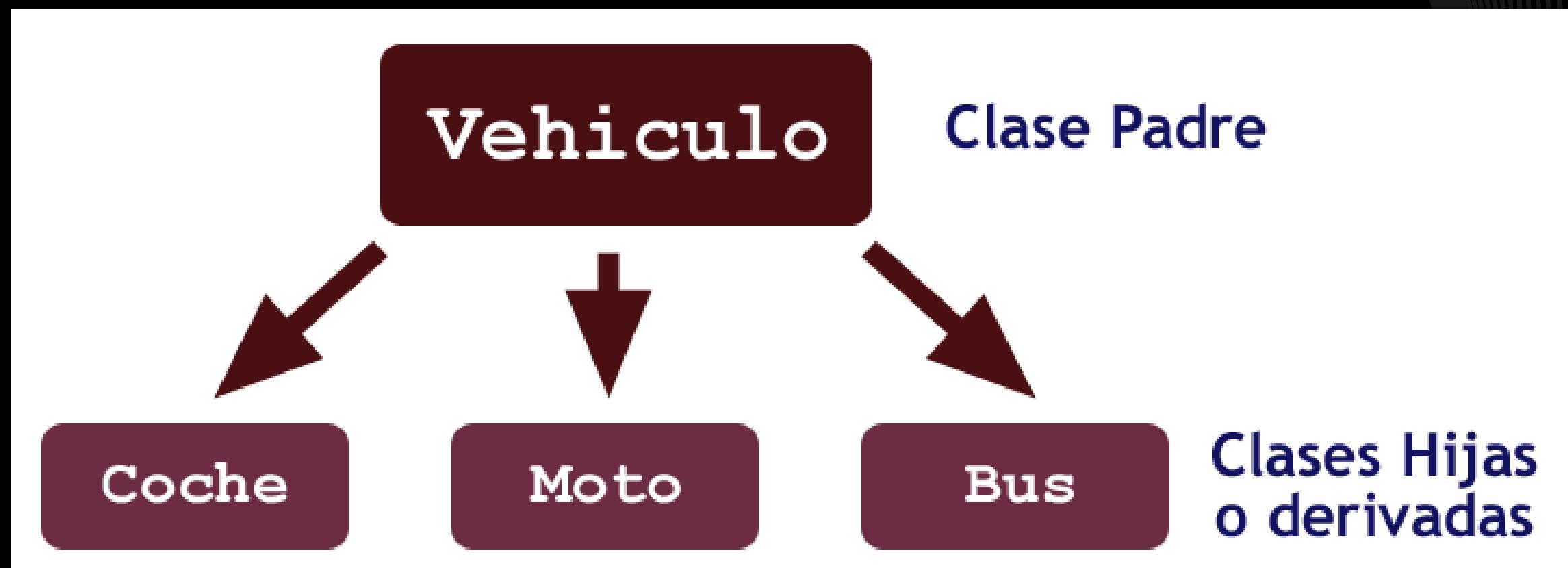
Nos permite crear nuevas clases a partir de otras. Si tuviéramos una clase "Autos" y quisiéramos crear unas clases "Auto deportivo" o "Auto clásico", podríamos tomar varias propiedades y métodos de la clase "Autos". Esto nos da una jerarquía de padre e hijo.



¿Qué es Polimorfismo?

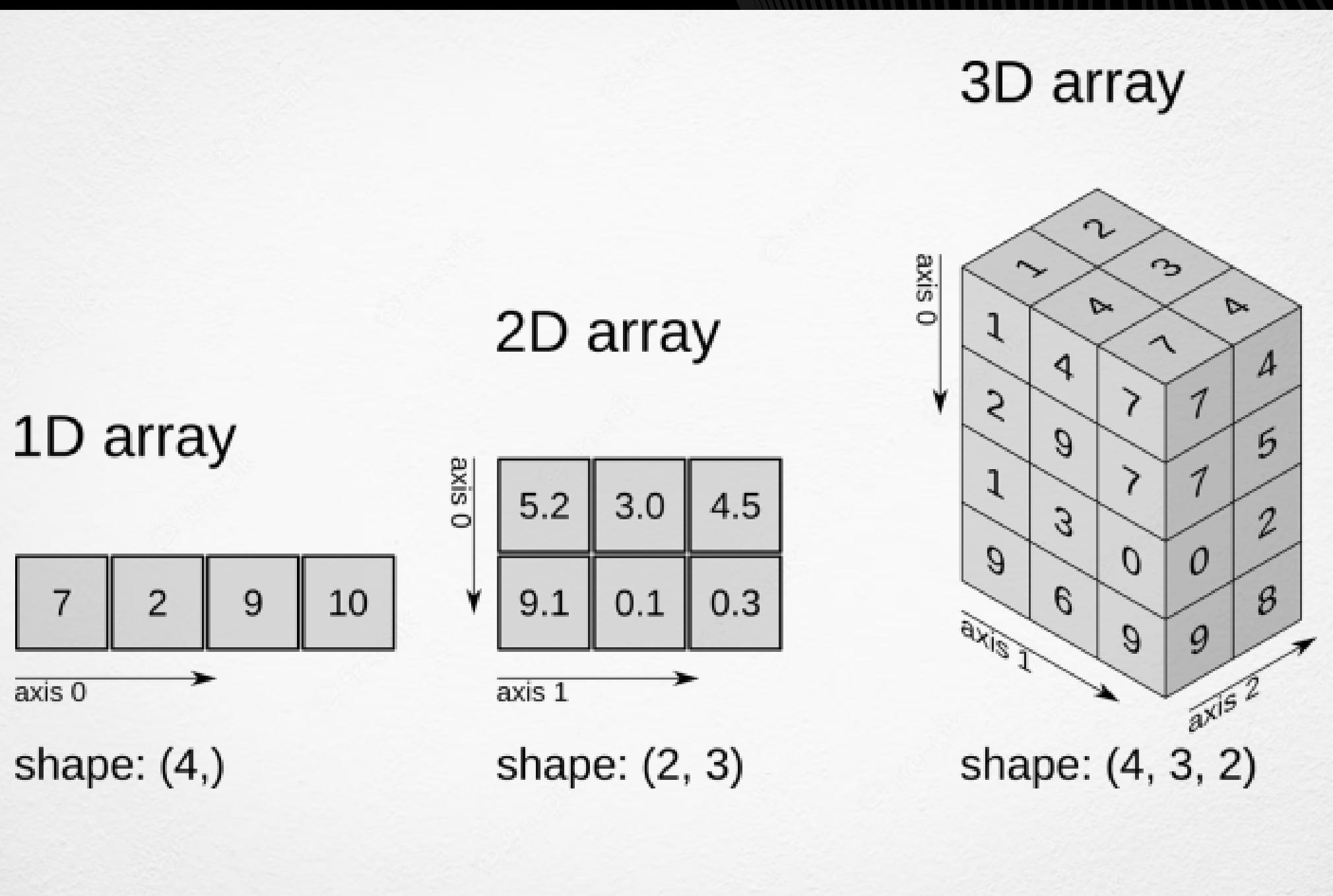
Proviene de Poli = muchas, morfismo = formas. Se utiliza para crear métodos con el mismo nombre pero con diferente comportamiento.

Tomando el mismo ejemplo de la cuenta bancaria, se podría ejemplificar al validar el monto máximo de transferencia al día. Donde una cuenta personal podría transferirse máximo 500S al día y a una cuenta empresarial 1500S.



¿Que es un ARRAY?

Es una estructura de datos que permite almacenar un conjunto de elementos del mismo tipo en una sola variable. Es como una lista o una tabla donde los elementos están dispuestos en una secuencia ordenada y se pueden acceder a ellos mediante un índice o posición numérica.

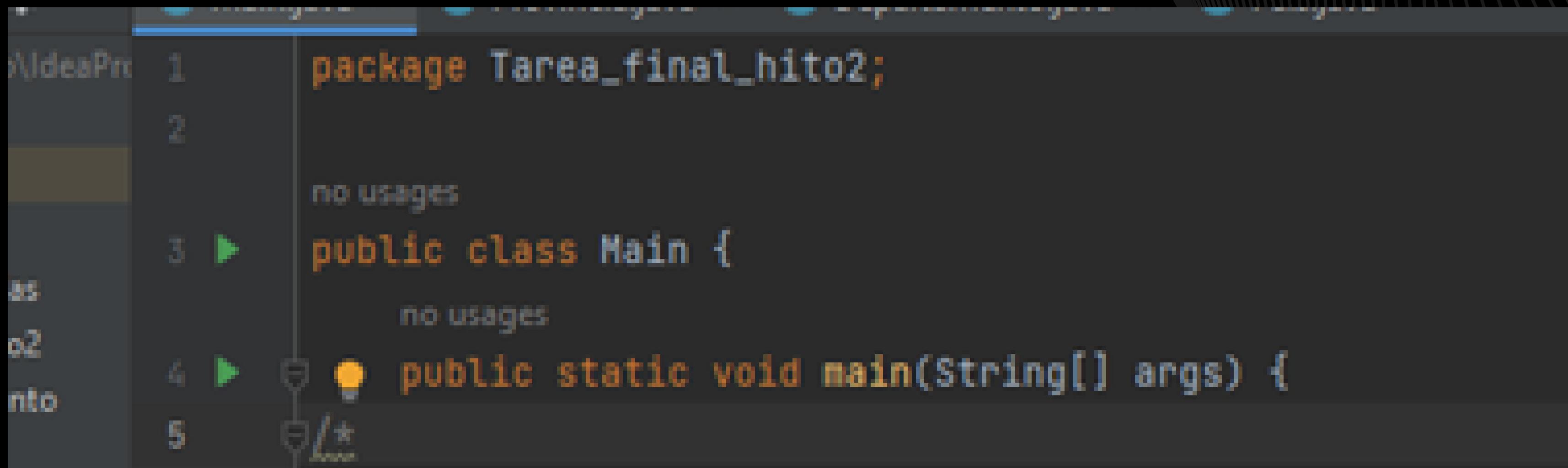


¿Qué son los paquetes en JAVA?

Los paquetes son el mecanismo que usa Java para facilitar la modularidad del código. Un paquete puede contener una o más definiciones de interfaces y clases, distribuyéndose habitualmente como un archivo. Para utilizar los elementos de un paquete es necesario importar este en el módulo de código en curso, usando para ello la sentencia import.



¿Cómo se define una clase main en JAVA y muestra un ejemplo?

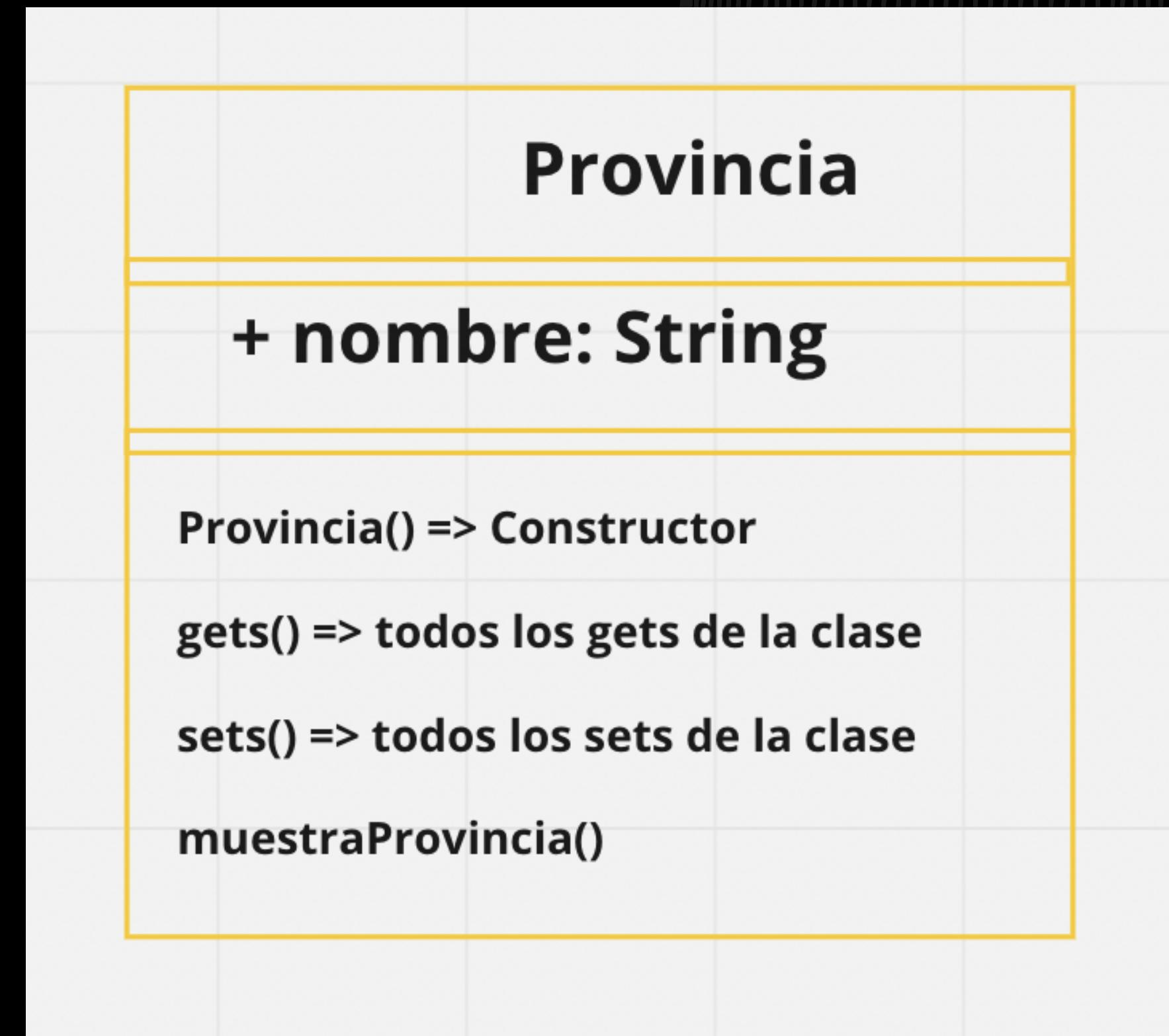


A screenshot of an IDE interface showing a Java code editor. The code defines a package named 'Tarea_final_hito2' and a public class named 'Main'. The 'Main' class contains a public static void main method that takes an array of strings as arguments. The code is numbered from 1 to 5 on the left side. The 'main' method is highlighted with a yellow lightbulb icon, indicating it is the entry point of the program.

```
1 package Tarea_final_hito2;
2
3 public class Main {
4     public static void main(String[] args) {
5 }
```

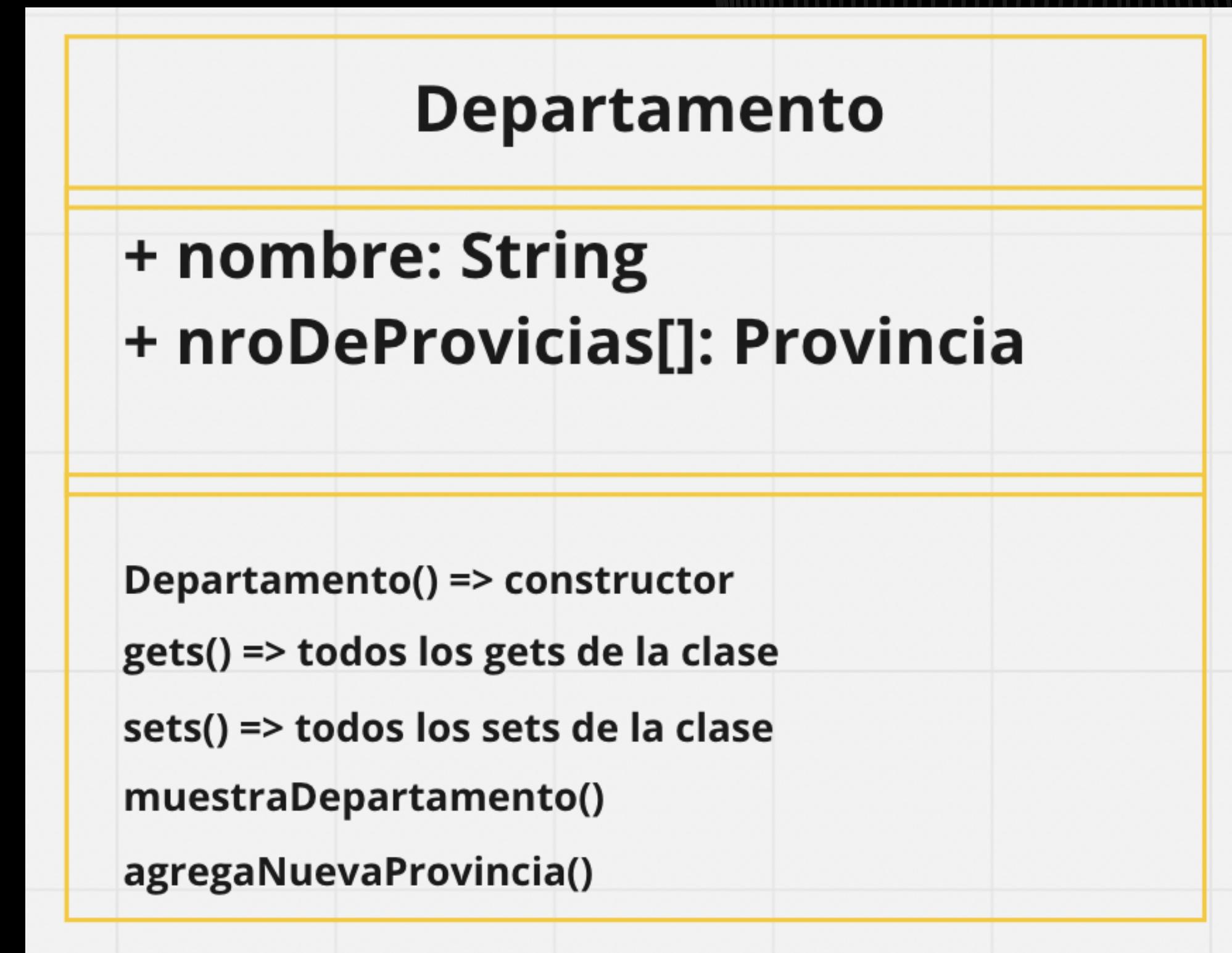
Generar la clase Provincia.

- Crear una clase MAIN
- Crear todos los gets y sets de la clase.
- Crear una instancia de la clase Provincia
- Mostrar los datos de una provincia



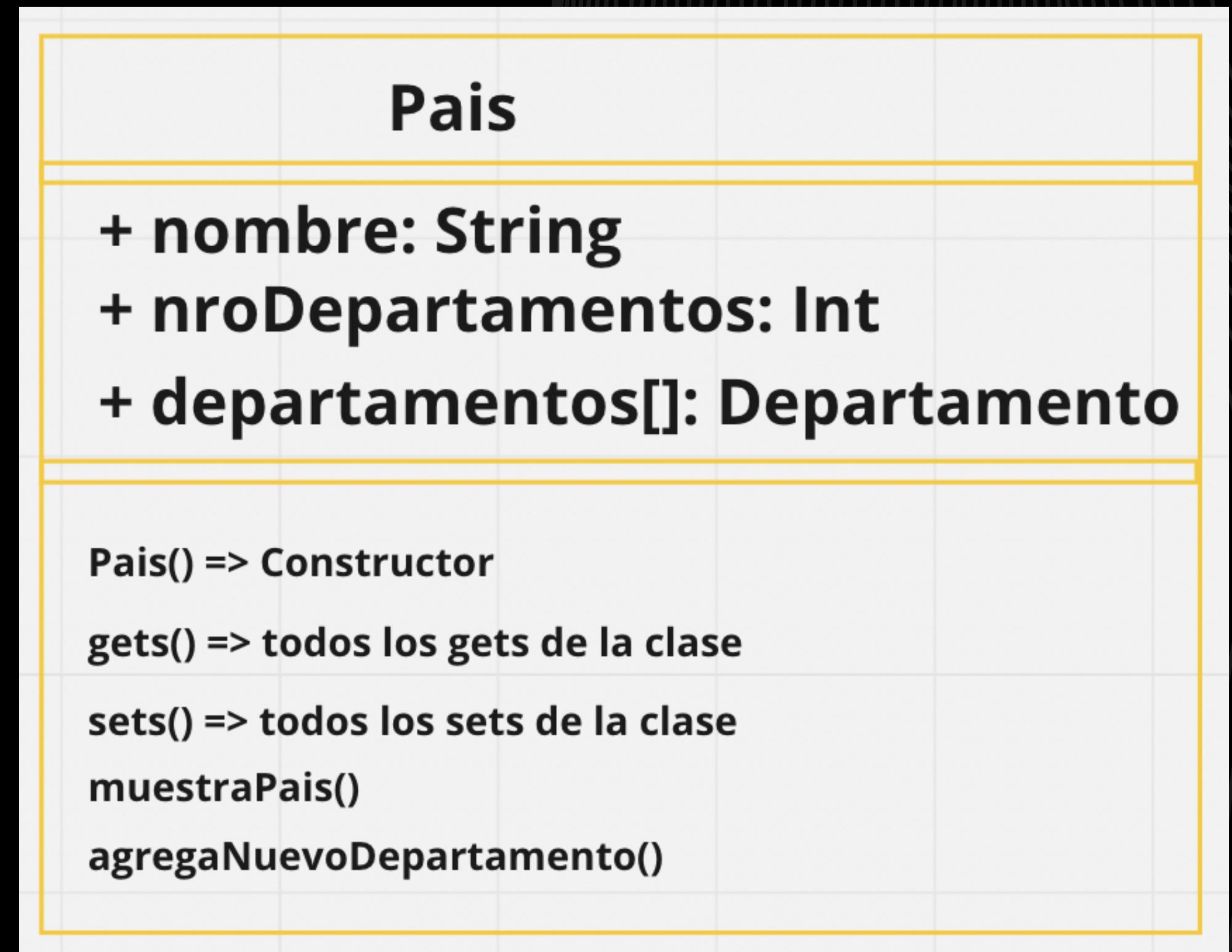
Generar la clase Departamento.

- Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)
- Crear todos los gets y sets de la clase.
- Crear una instancia de la clase Departamento.
- Omitir el método `agregaNuevaProvincia()`
- Mostrar los datos de los departamentos.



Generar la clase País.

- Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)
- Crear una instancia de la clase País
- Crear una instancia de la clase Departamento.
- Omitir el método agregaNuevoDepartamento()
- Mostrar los datos del País.



Gracias!!

