



POO/OOP

Programación Orientado a Objetos

Object-Oriented Programming



Vocabulario

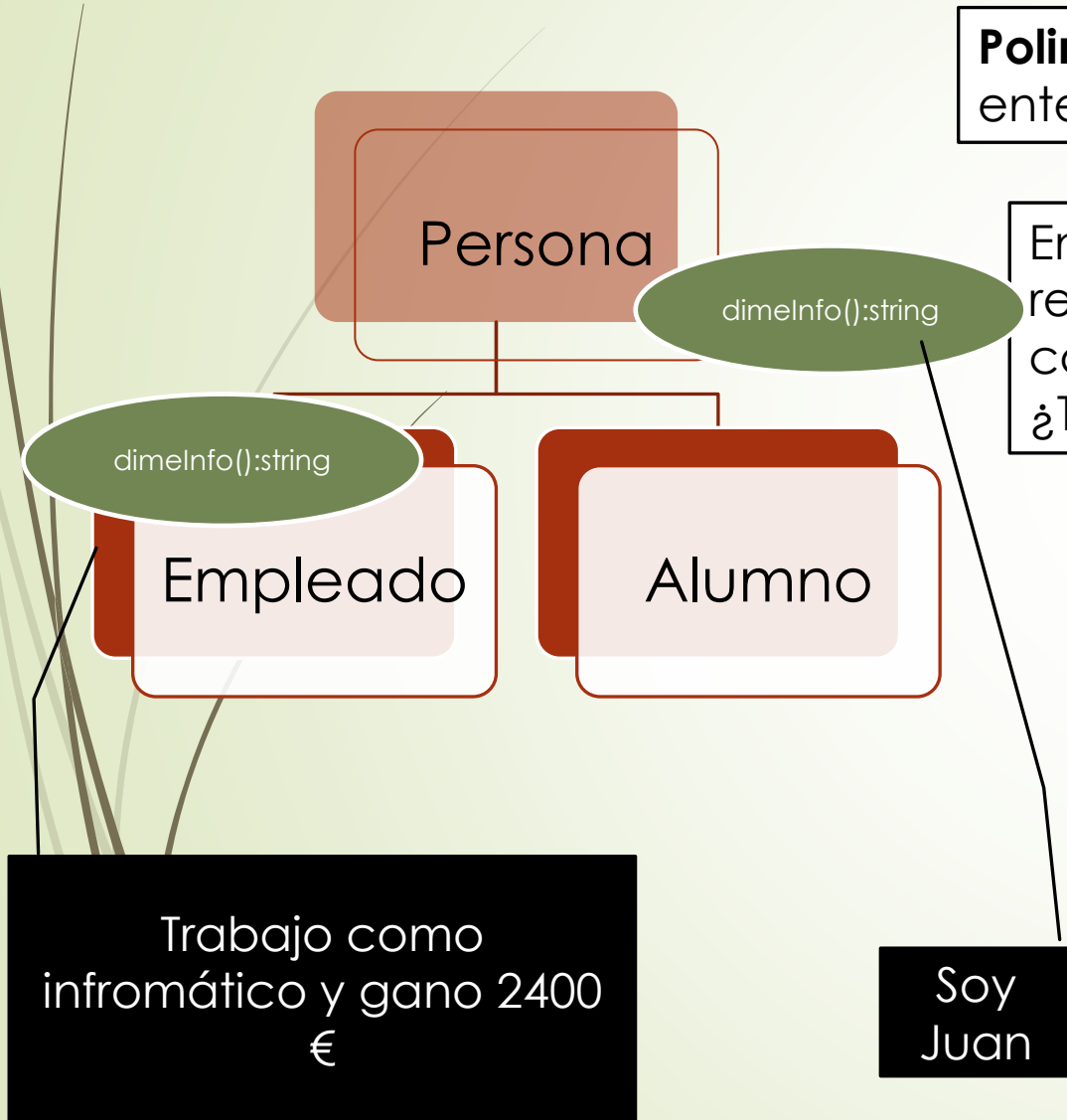


- Clase, objeto
- Ejemplar de clase, instancia de clase, ejemplarizar una clase, instanciar una clase
- Modularización
- Encapsulamiento / encapsulación
- Herencia
- **Polimorfismo**

Polimorfismo

Polimorfismo de forma genérica es considerar que un ente puede adoptar **múltiples formas**.

En **programación orientada a objetos** puede hacer referencia a que un mismo método puede representar código diferente.
¿Te suena lo de sobrescribir un método? 😊



El polimorfismo es la capacidad de una clase abstracta o un objeto determinado para responder a un mismo método con su propio comportamiento.

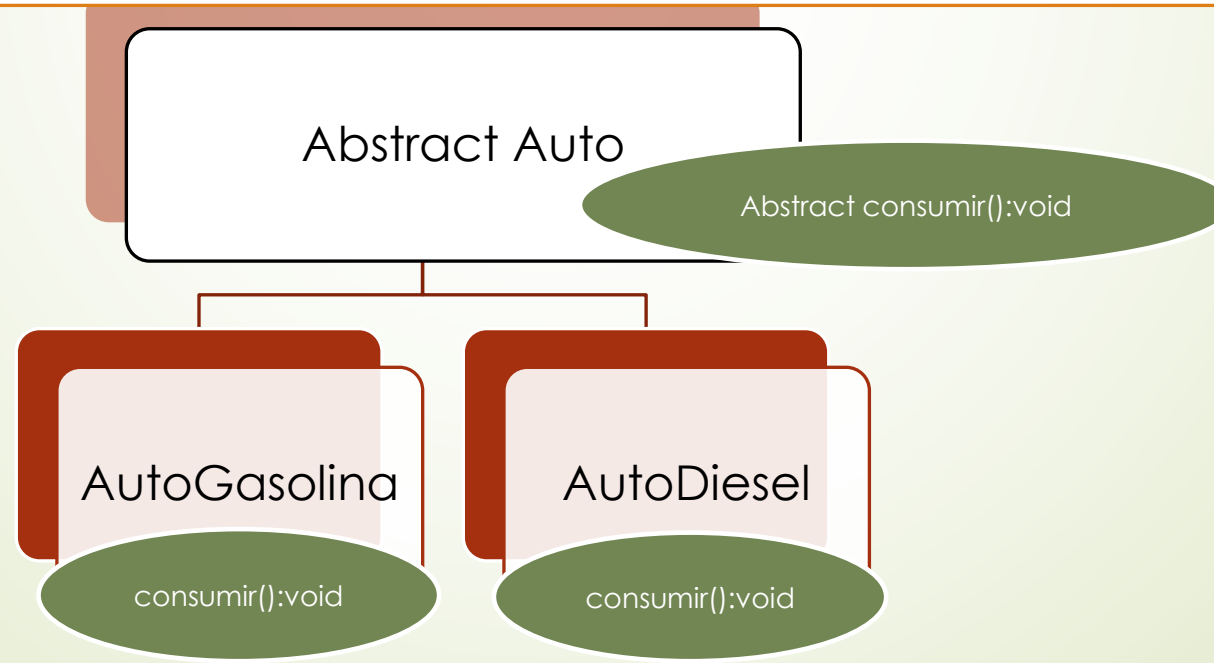
El polimorfismo especifica que un parámetro u objetos pasado o esperado puede adoptar múltiples formas y tener diferentes comportamiento con los mismos métodos

Polimorfismo 2

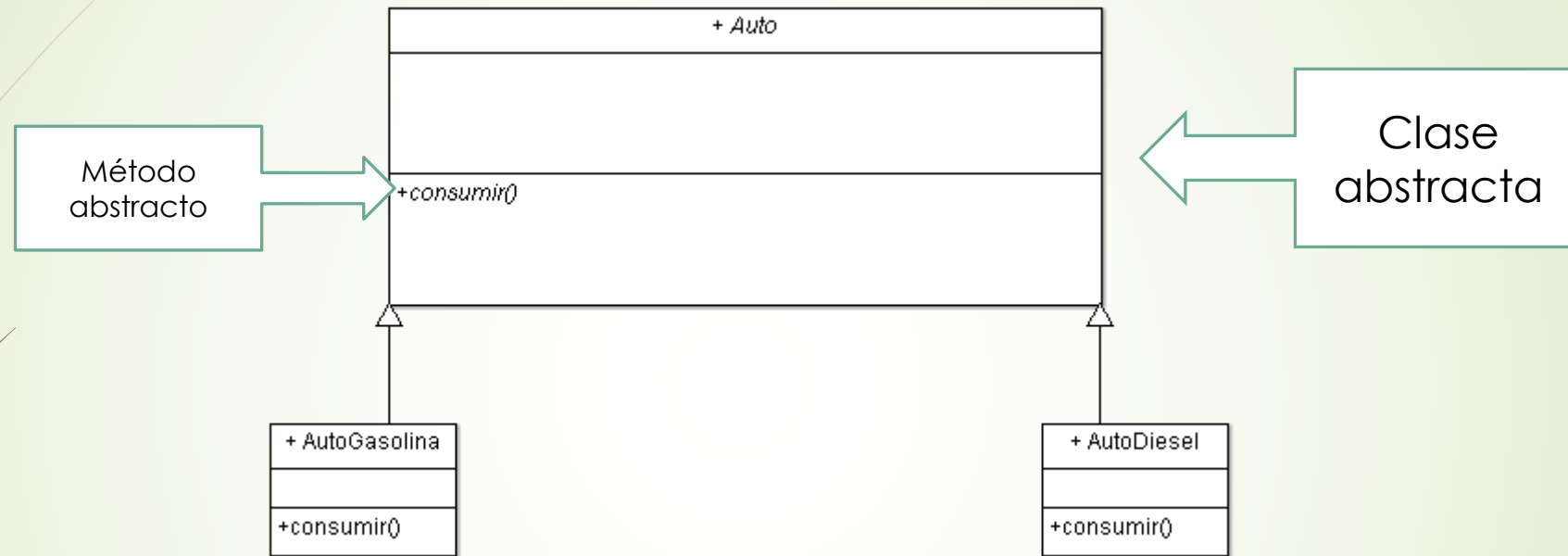
En **programación orientada a objetos** existe otra forma de **polimorfismo** que tiene que ver con el **tipado de datos**. Aunque como ya sabemos PHP es un lenguaje débilmente tipado pondremos un ejemplo pero antes una definición.

Definición:

El polimorfismo es una **relajación del sistema de tipos**, de tal manera que una referencia a una clase (atributo, parámetro o declaración local o elemento de un vector) acepta direcciones de objetos de dicha clase y de sus clases derivadas (hijas, nietas, ...).



Polimorfismo de tipado de datos



```
// Prueba de polimorfismo: La función mostrar() usa un objeto Auto como parámetro.
// Sin embargo, en tiempo de ejecución decidirá cuál función consumir() es apta.
function mostrar(Auto $auto){
    echo $auto->consumir();
}

// Prueba de los objetos y las funciones
$carroGasolina = new AutoGasolina();
$carroDiesel = new AutoDiesel();
```