

HERENCIA y COMPOSICIÓN DE CLASES



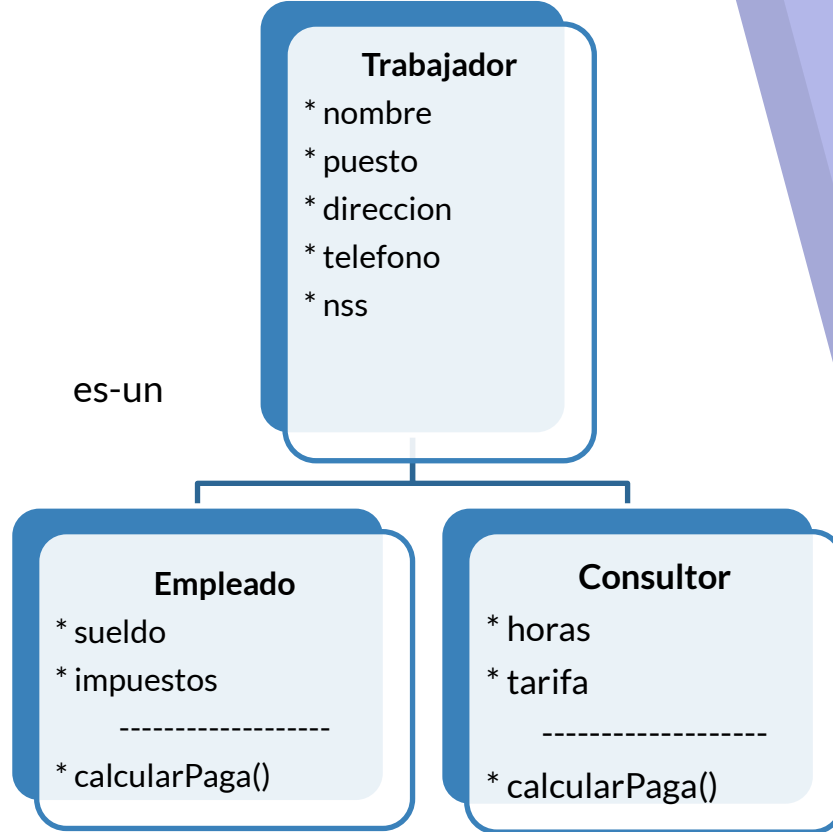


1.

HERENCIA DE CLASES

HERENCIA

- ▶ Una clase que extiende a otra hereda sus atributos y sus métodos (no constructores).
- ▶ Puede añadir atributos y métodos nuevos.



MODIFICADOR **PROTECTED**

- ▶ Si usamos `protected` en la clase base, tendremos acceso directo a los atributos.
- ▶ En otro caso, tendremos que acceder vía *getters/setters*.
- ▶ ¡OJO! Los constructores no se heredan aunque sean públicos.



2.

HERENCIA DE INTEFACES

HERENCIA DE INTERFACES

- ▶ También podemos establecer relaciones *jerárquicas* entre interfaces.
- ▶ Nos regimos por las mismas reglas que en el caso de las clases.



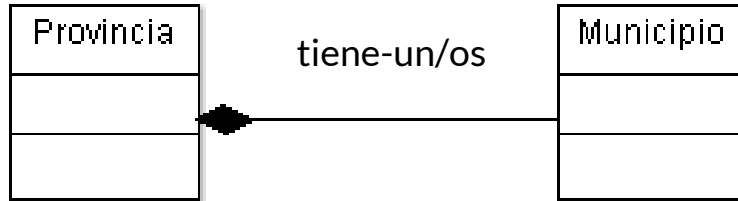
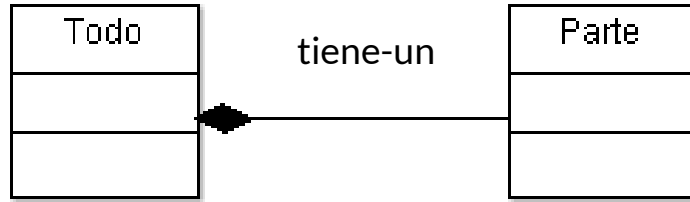
3.

COMPOSICIÓN DE CLASES

ASOCIACIONES ENTRE CLASES

- ▶ Normalmente, cuando representamos la estructura de un *sistema*, está formado por muchas clases.
- ▶ En este caso, no solamente importan las clases en sí, sino las *asociaciones*.
- ▶ Una de ellas es la **composición**.
- ▶ En UML, se representan de una forma especial.

ASOCIACIÓN DE COMPOSICIÓN



COMPOSICIÓN DE CLASES

- ▶ Dentro de la clase *Todo* tendremos una referencia a la clase *Parte*
- ▶ También es posible que la *multiplicidad* nos indique que debemos tener una colección (*Provincia* y *Municipio*).
- ▶ Normalmente hay dependencia de existencia entre la parte y el todo.