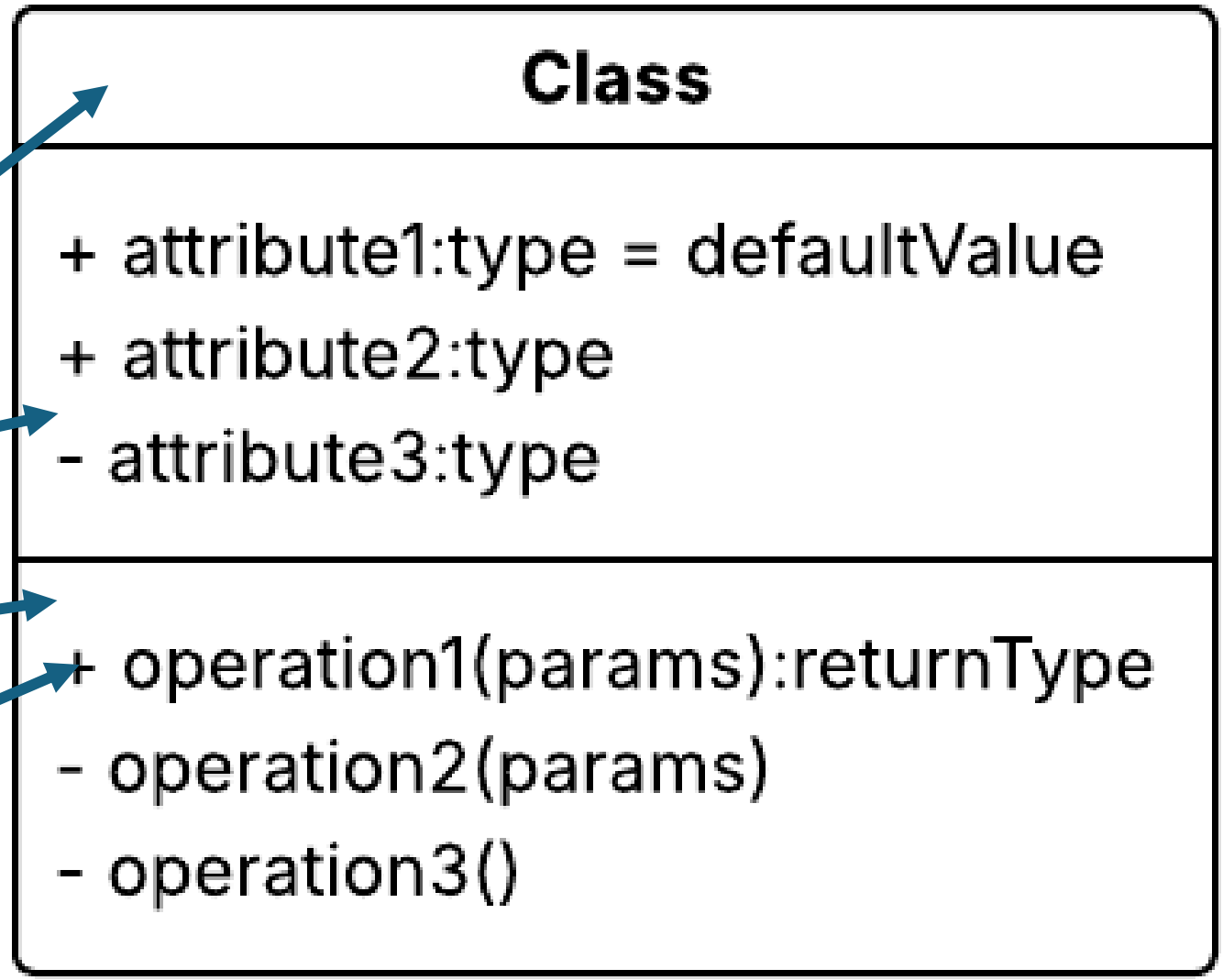
Diagrama de clases

Christian Jaimes

Clase

- Nombre de la clase
- Atrributos
- Metodos
- Modificadores de acceso

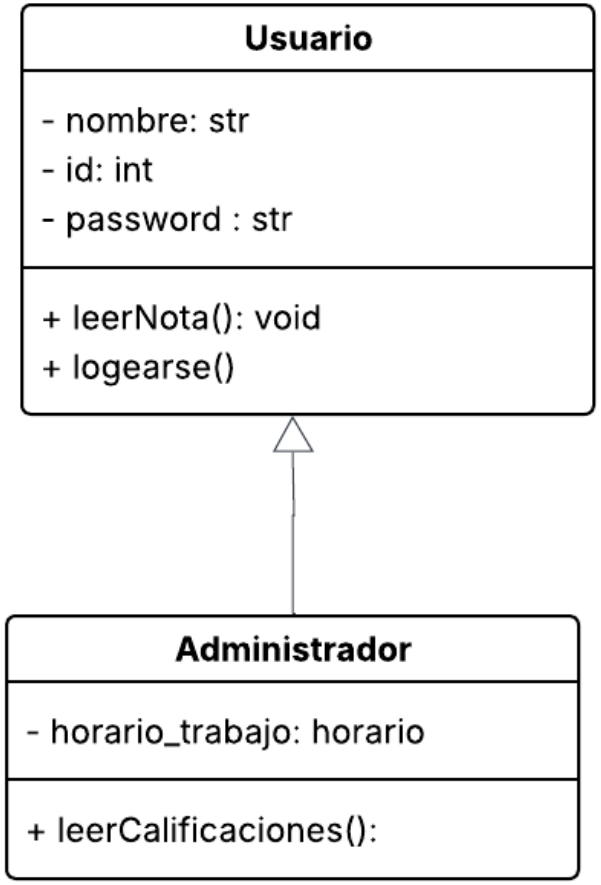


Herencia

Concepto:
La **herencia** permite que una clase (subclase o hija) adquiera los atributos y métodos de otra (superclase o padre).

Se utiliza para **reutilizar código** y **modelar jerarquías**.

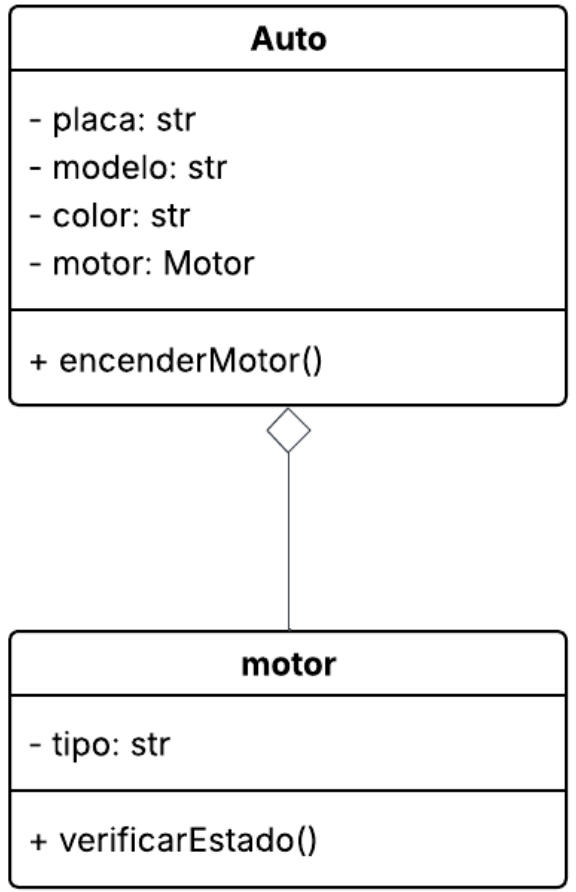
En UML:
Se representa con una **flecha vacía** que apunta **desde la subclase hacia la superclase**.
Etiqueta: *"is a"* (es un tipo de...).



Agregación

La **agregación** representa una relación débil del tipo **“tiene un”**.
El objeto contenido puede **existir independientemente** del contenedor.

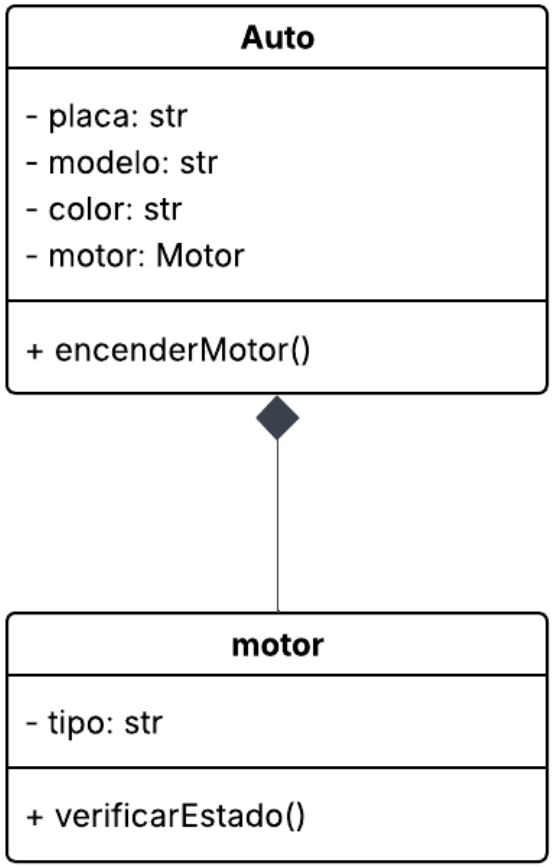
En UML:
Se representa con un **rombo blanco** en el lado del **todo (contenedor)**.
Indica una **relación débil de dependencia**.



Composición

La **composición** es una relación fuerte del tipo **“está compuesto por”**.
Si el objeto contenedor se destruye, los objetos contenidos **también se destruyen**.

En UML:
Se representa con un **rombo negro** en el lado del **todo (contenedor)**.
Indica **dependencia total**.



Comparativa general

Relación	Tipo de vínculo	Existencia independiente	Representación UML	Ejemplo
Herencia	“es un”	Sí	Flecha vacía (Δ)	Estudiante es una Persona
Agregación	“tiene un”	Sí	Rombo blanco (◇)	Departamento tiene Profesores
Composición	“está compuesto por”	No	Rombo negro (◆)	Casa tiene Habitaciones

Interfaz

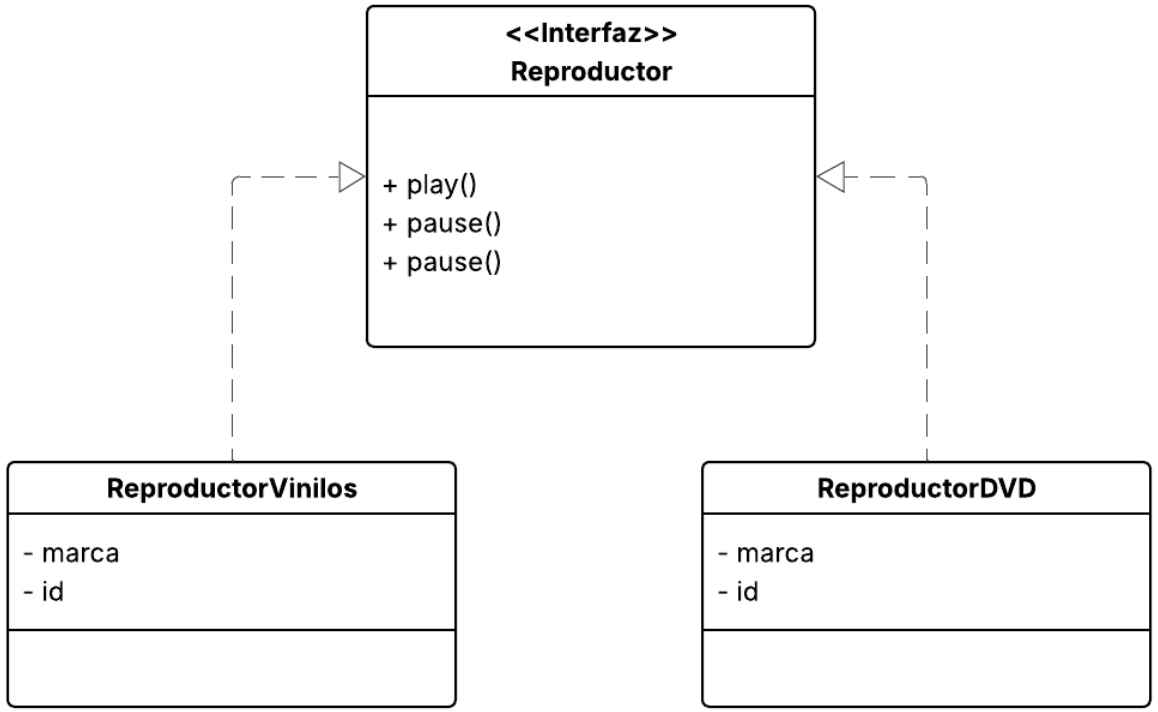
Concepto:
Una **interfaz** define **un conjunto de métodos** que **deben ser implementados** por las clases que la usen.

Sirve para **establecer contratos** entre clases sin definir la implementación concreta. Las interfaces permiten **abstracción**, **desacoplamiento** y **polimorfismo**.

En UML:
Se representa con:

- El estereotipo <<interface>>, o
- Una **línea discontinua con un círculo** (lollipop) que conecta con las clases que la implementan.

La relación se llama **realización** (no herencia).



Dependencia

Concepto:
La **dependencia** es una relación temporal y débil entre clases:
una clase **usa a otra** para realizar una tarea, pero **no la almacena como atributo permanente**.
Es una relación de tipo "**usa a**".

En UML:
Se representa con una **línea discontinua con una flecha abierta**.
Indica que una clase **depende del comportamiento** de otra, **solo mientras se ejecuta** un método.