

# Interfaces y la herencia

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Índice

1. [Interfaces](#)
2. [Interfaces y la herencia](#)

# 1 | Interfaces



Veamos más en detalle las interfaces y luego analizaremos su relación con la herencia.

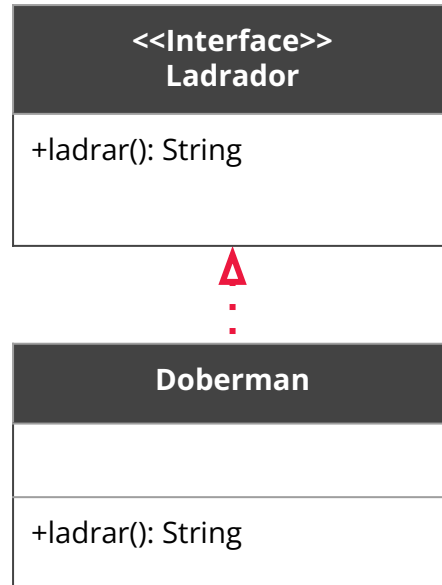


# Interfaces

Las interfaces son también relaciones del tipo “es un”, muy similares a las clases abstractas: se definen con la palabra clave **"interface"** en vez de "class". **Todos sus métodos son abstractos**, por lo cual, **no es necesaria la palabra "abstract"** y, al igual que las clases abstractas, los métodos no definen un cuerpo.

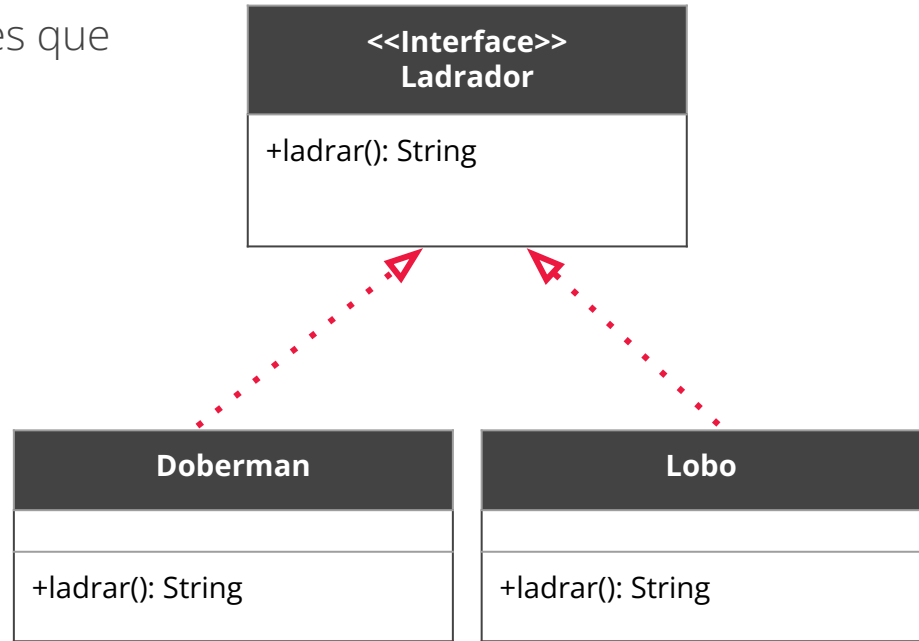
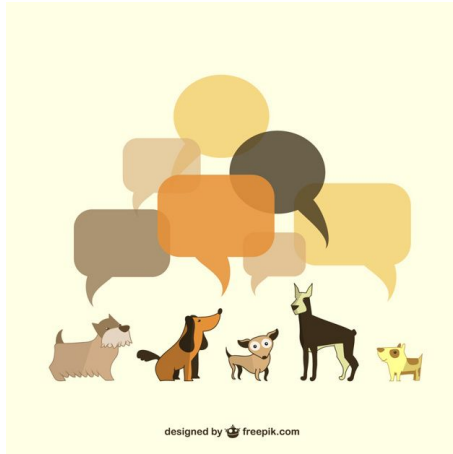


Doberman debe implementar sí o sí el método ladrar() por implementar la interface Ladrador.



Una **interface** establece un **contrato**. Toda clase que implemente una interface está obligada a implementar todos los métodos de esa interface.

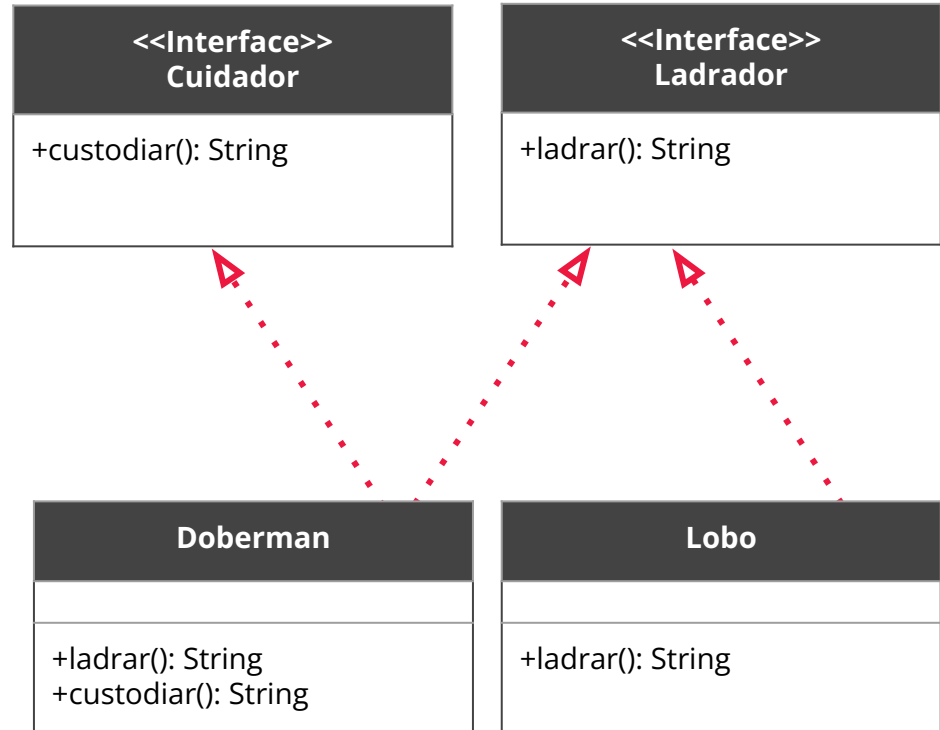
Por ejemplo, todas aquellas clases que implementen **Ladrador** deben implementar el método **ladrar**.



Una clase solo puede heredar de una sola clase, pero puede implementar múltiples interfaces.



Como vemos, **Doberman** implementa 2 interfaces, ya que además de ladrar() también tendrá la responsabilidad de custodiar por ser Cuidador.



# 2 | Interfaces y la herencia



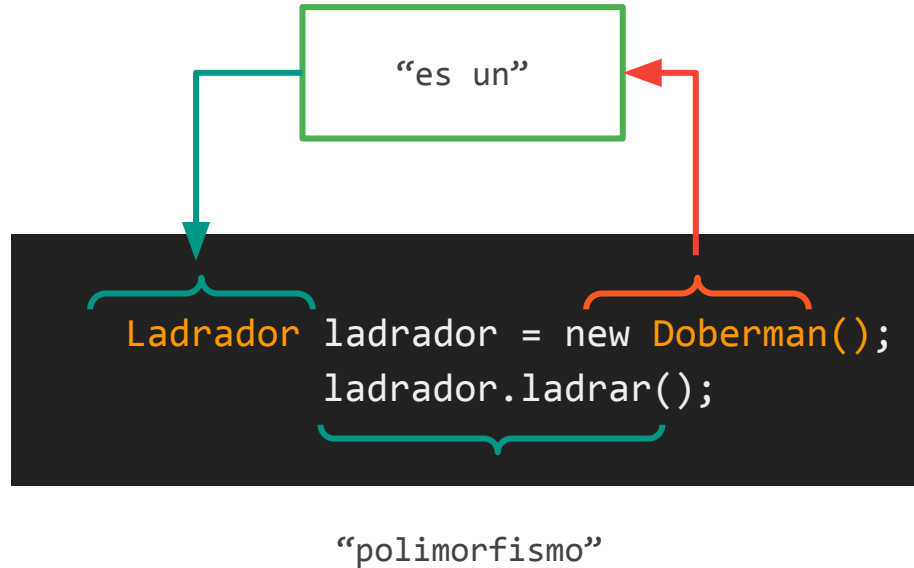
# Interfaces y la herencia

Cuando heredamos de una clase sumamos atributos y comportamientos de la clase padre, mientras que cuando implementamos una interface solo obligamos a la clase que la implementa a sobrecribir, es decir, implementar, los métodos de la misma.

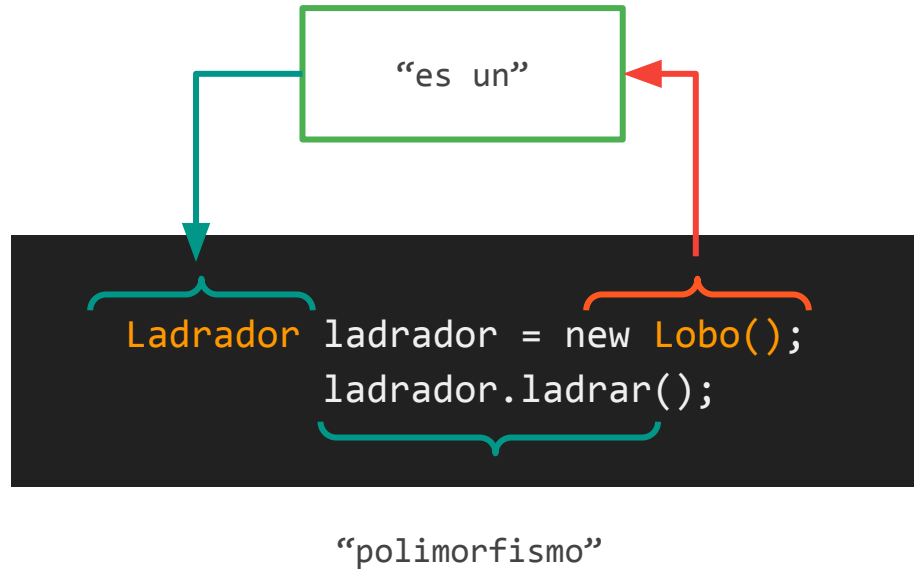


Tienen en común que son relaciones del tipo “es un”, con lo cual las interfaces también nos permite realizar vinculación dinámica y, por ende, polimorfismo.

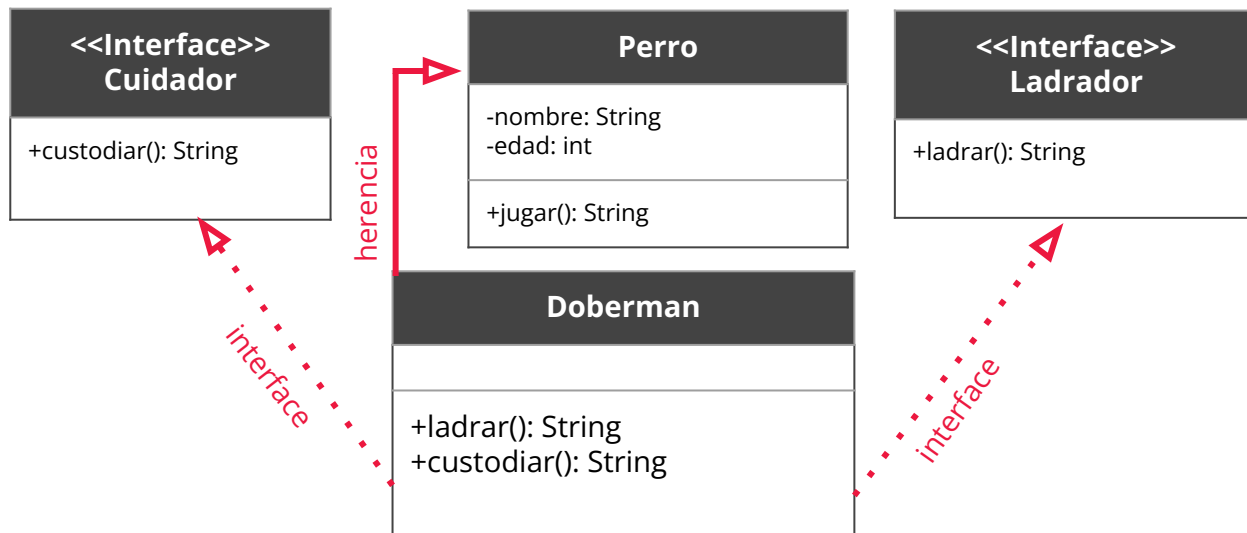
# Ejemplo 1



## Ejemplo 2



Lo que permiten las interfaces es independizarse de una jerarquía, permiten agregar comportamiento a una clase que no se obtenga desde un nivel superior en la jerarquía, se “enchufa” lateralmente a la jerarquía. **Incluso podríamos hasta mezclar ambos mecanismos.**



DigitalHouse>  
Coding School