

# ¿Qué es polimorfismo?



an NTT DATA Company

- Palabra de origen griego que significa “muchas formas”.
- Es uno de los 4 pilares de la programación orientada a objetos (***Abstracción, Encapsulación Herencia y Polimorfismo***).
- El polimorfismo está muy ligado al concepto de herencia

# Clase padre (abstracta o no), interfaz

The logo for everis, featuring the word "everis" in a white, lowercase, sans-serif font inside a green, rounded, irregular shape.

an NTT DATA Company

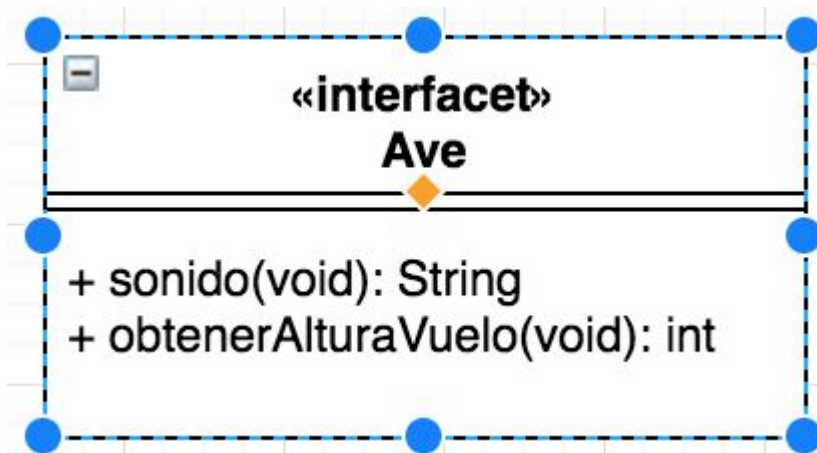
Se puede hacer polimorfismo con una clase “normal”, una clase abstracta (clase que puede contener métodos sin código y que han de ser implementados por cualquier clase que la extienda) o por una interfaz (clases donde sólo existen la declaración de métodos públicos que han de implementar cualquier clase que implemente la interfaz).

Deberemos usar una u otra opción según nuestra necesidad, teniendo en cuenta las ventajas y desventajas de cada una de ellas.

Por ejemplo una clase puede heredar de otra mientras que las interfaces sólo pueden heredar de interfaces.

En nuestro ejemplo usaremos una interfaz.

# La interfaz Ave

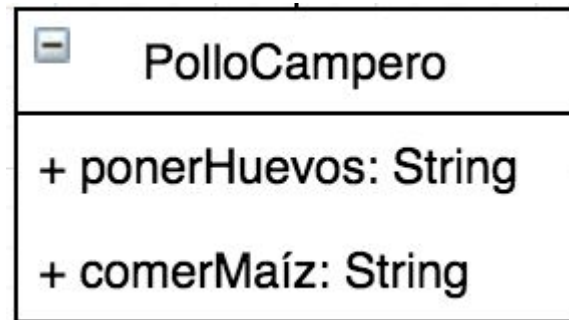


En nuestro ejemplo usaremos la <<**interfaz ave**>>, que obliga a implementar los métodos, **sonido** y **obtenerAlturaVuelo**.

# Creando las clases que extenderán la interfaz.

everis

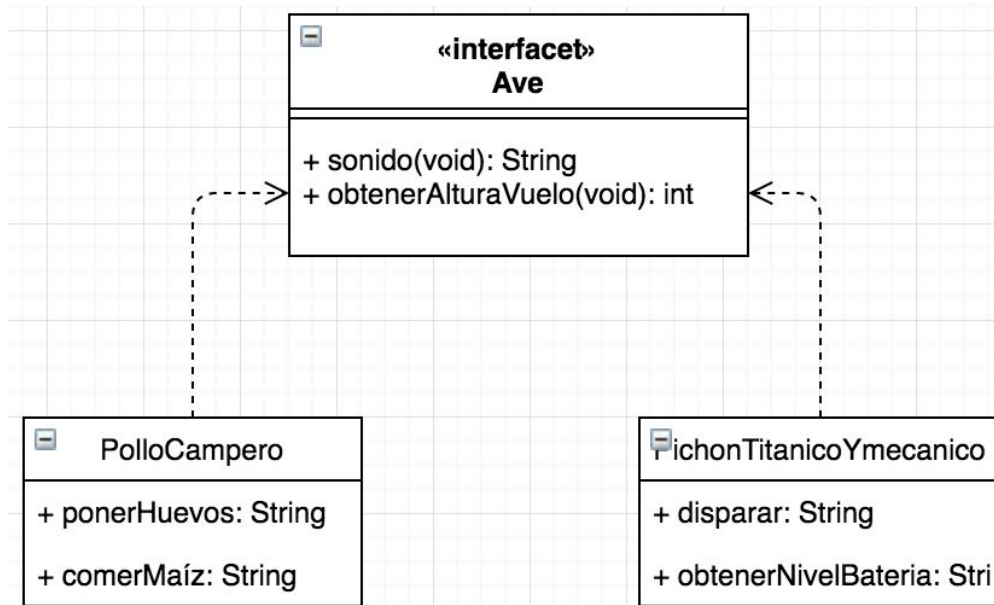
an NTT DATA Company



# Implementamos las interfaces.



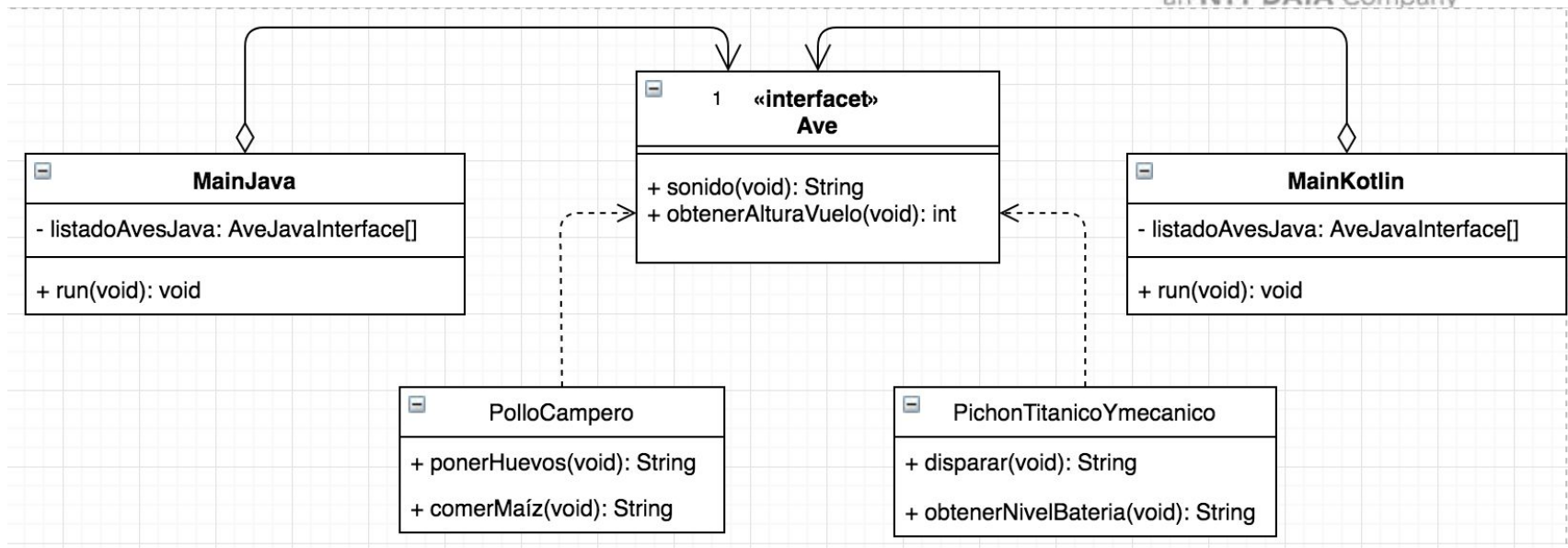
DATA Company



# Polimorfismo en acción, por fin...!!!

everis

an NTT DATA Company



# Polimorfismo en acción, por fin.....!!!

The Everis logo consists of a green, irregular, cloud-like shape. Inside this shape, the word "everis" is written in a white, lowercase, sans-serif font.

Para el ejemplo crearemos las clases **MainJava** **MainKotlin**. an NTT DATA Company

En esas clases creamos como atributo un array de elementos ***AveJavaInterface*** y ***AveKotlinInterface*** respectivamente.

Agregaremos al array un objeto de cada tipo (***PolloCampero*** y ***PichonTitánico***) cada uno en la versión que corresponda.

Iteremos sobre la lista y comprobamos de que tipo es cada uno, según el tipo ejecutamos los métodos que correspondan. Es posible gracias a que ***PolloCampero*** y ***PichonTitanicoYmecanico*** implementan la interfaz ***AveJavaInterface*** y por lo tanto son objetos de ese tipo.

# Notas sobre el polimorfismo.



- Si usas los objetos usando el tipo de la clase padre, sólo tendrás acceso a los métodos públicos de dicha clase padre o interfaz.
- Para usar los métodos públicos de las clases hijas, te verías obligado a castear al tipo de objeto en cuestión.
- En lenguajes como Java para saber si un objeto es de un tipo determinado se puede usar la palabra reservada instanceof.
- Si es necesario para identificar el tipo de un objeto se puede usar un enum con los tipos de los objetos, un método getType o similar al que se puede agregar lógica y que indique el tipo del objeto etc.