

POLIMORFISMO EN JAVA

Charly Cimino

Polimorfismo en Java

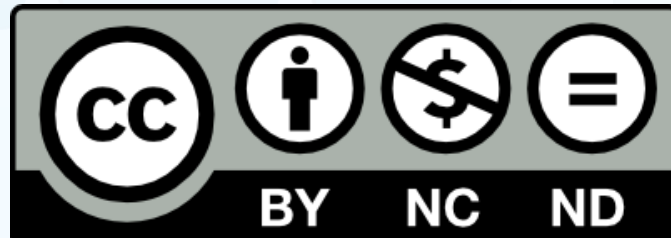
Charly Cimino

Este documento se encuentra bajo Licencia Creative Commons 4.0 Internacional (CC BY-NC-ND 4.0). Usted es libre para:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

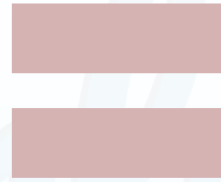
Bajo los siguientes términos:

- **Atribución** — Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.
- **No Comercial** — Usted no puede hacer uso del material con fines comerciales.
- **Sin Derivar** — Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted no podrá distribuir el material modificado.



Definición

“Polimorfismo”



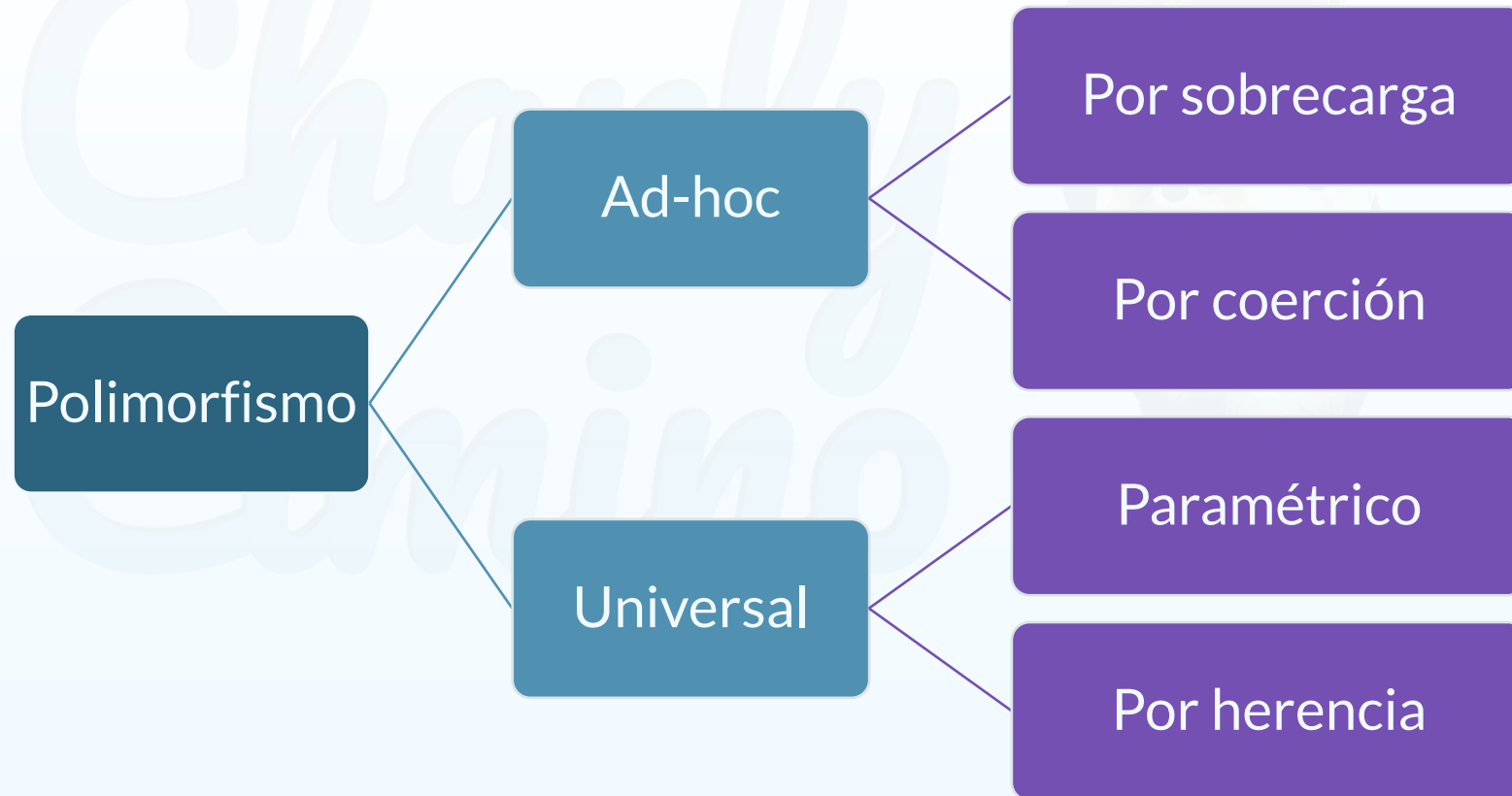
“Muchas Formas”

El pilar fundamental de la POO.

Nos permite programar de forma genérica y con clases desacopladas.
El software se vuelve escalable, mantenible y reutilizable.

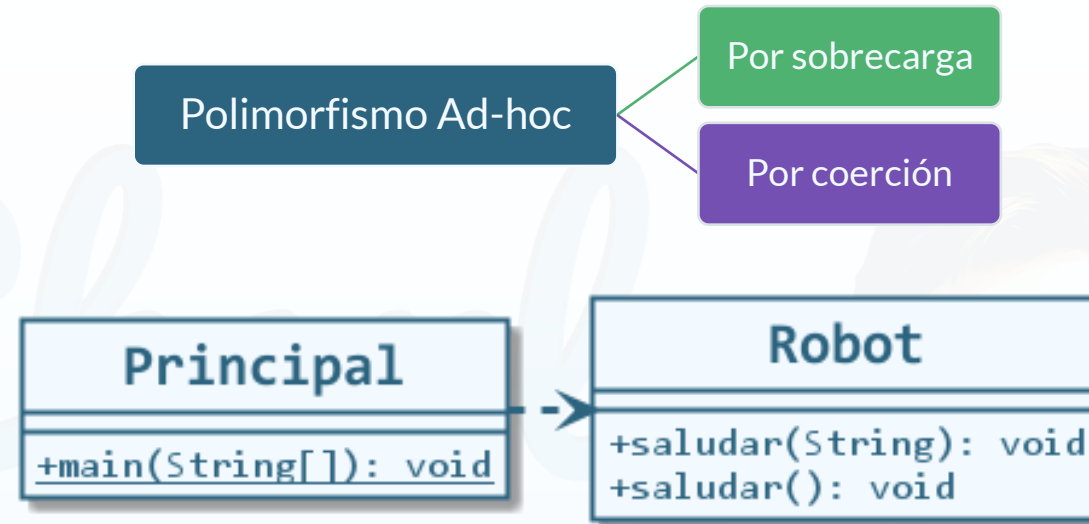
Clasificación

Cardelli y Wegner* clasifican al polimorfismo en diferentes categorías



* Cardelli, L. & Wegner, P. "On Understanding Types, Data Abstraction, and Polymorphism". En: Computing Surveys (Diciembre, 1985). Vol. 17, n. 4, p. 471

Ejemplo de polimorfismo por sobrecarga



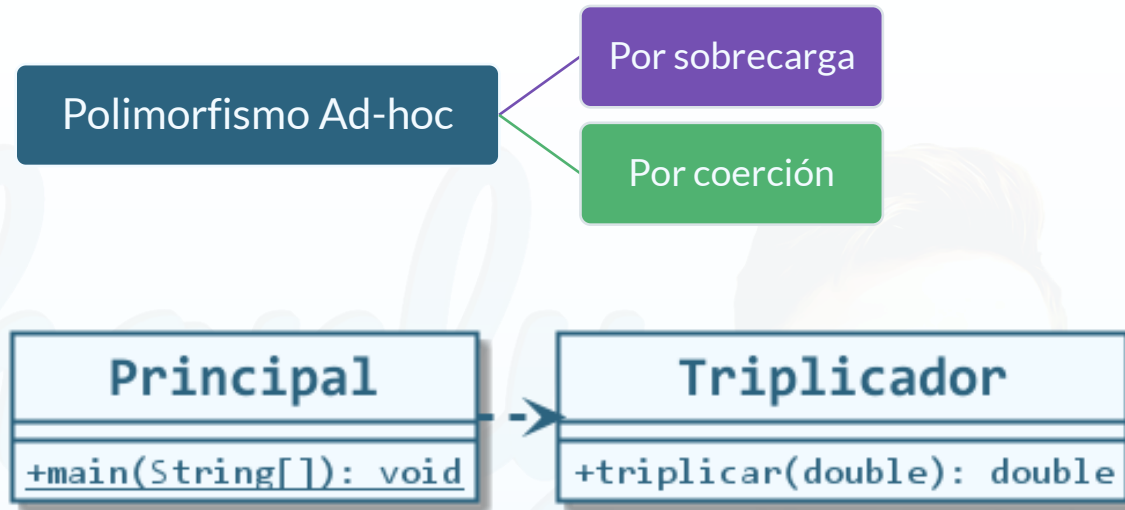
Principal.java

```
public class Principal {  
    public static void main(String[] args) {  
        Robot r = new Robot();  
        r.saludar("Juan"); // "Hola Juan"  
        r.saludar(); // "Hola extraño"  
    }  
}
```

Robot.java

```
public class Robot {  
    public void saludar(String nombre) {  
        System.out.println("Hola " + nombre);  
    }  
    public void saludar() {  
        System.out.println("Hola extraño");  
    }  
}
```

Ejemplo de polimorfismo por coerción



Principal.java

```
public class Principal {  
    public static void main(String[] args) {  
        Triplicador t = new Triplicador();  
        double num1 = 2.5;  
        int num2 = 5;  
        System.out.println(t.triplicar(num1)); // 7.5  
        System.out.println(t.triplicar(num2)); // 15.0  
    }  
}
```

Triplicador.java

```
public class Triplicador {  
    public double triplicar(double num) {  
        return 3 * num;  
    }  
}
```

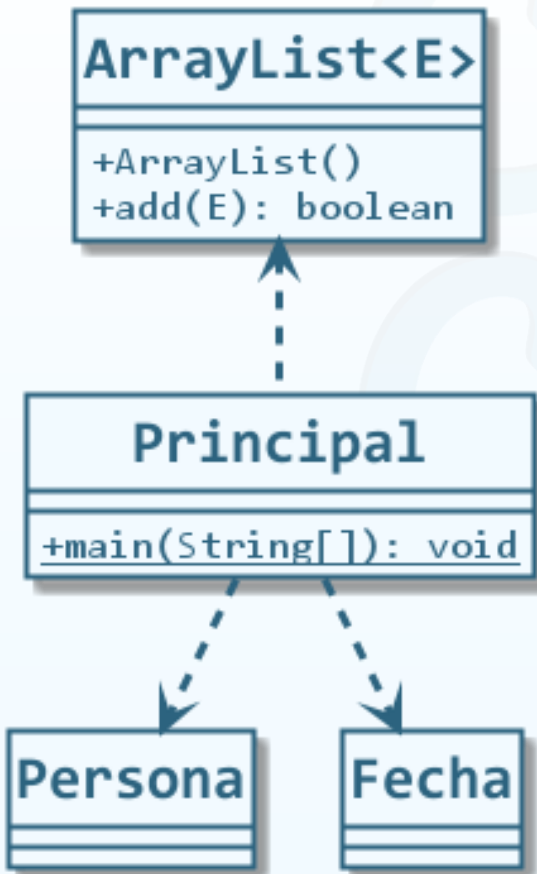
Ejemplo de polimorfismo paramétrico

Este tema será ampliado cuando veamos 'Generics'.

Polimorfismo Universal

Paramétrico

Por herencia



Principal.java

```
public class Principal {
    public static void main(String[] args) {
        ArrayList<Fecha> fechas = new ArrayList<Fecha>();
        fechas.add(new Fecha());
        ArrayList<Persona> personas = new ArrayList<Persona>();
        personas.add(new Persona());
    }
}
```

ArrayList.java

```
public class ArrayList<E> {
    public boolean add(E e) {
        // Guarda el elemento...
    }
}
```

Polimorfismo por herencia

“Polimorfismo por herencia”

=

“Polimorfismo”

Cuando decimos “polimorfismo” a secas, por lo general nos referimos particularmente al polimorfismo por herencia.

Polimorfismo por herencia

Mecanismo que permite enviar mensajes sintácticamente iguales a objetos de tipos distintos, cada uno con su propio comportamiento.

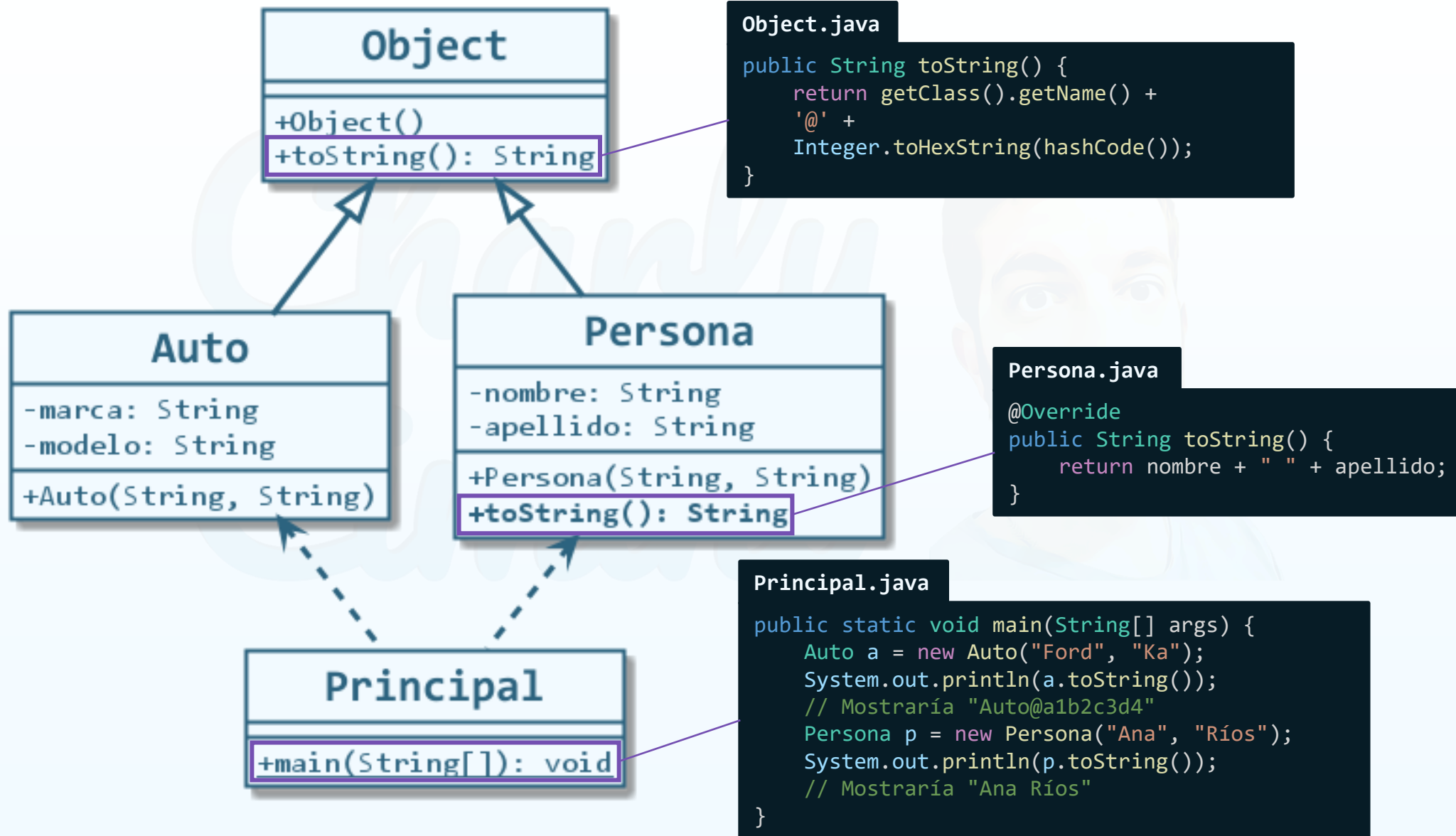


“Encender”



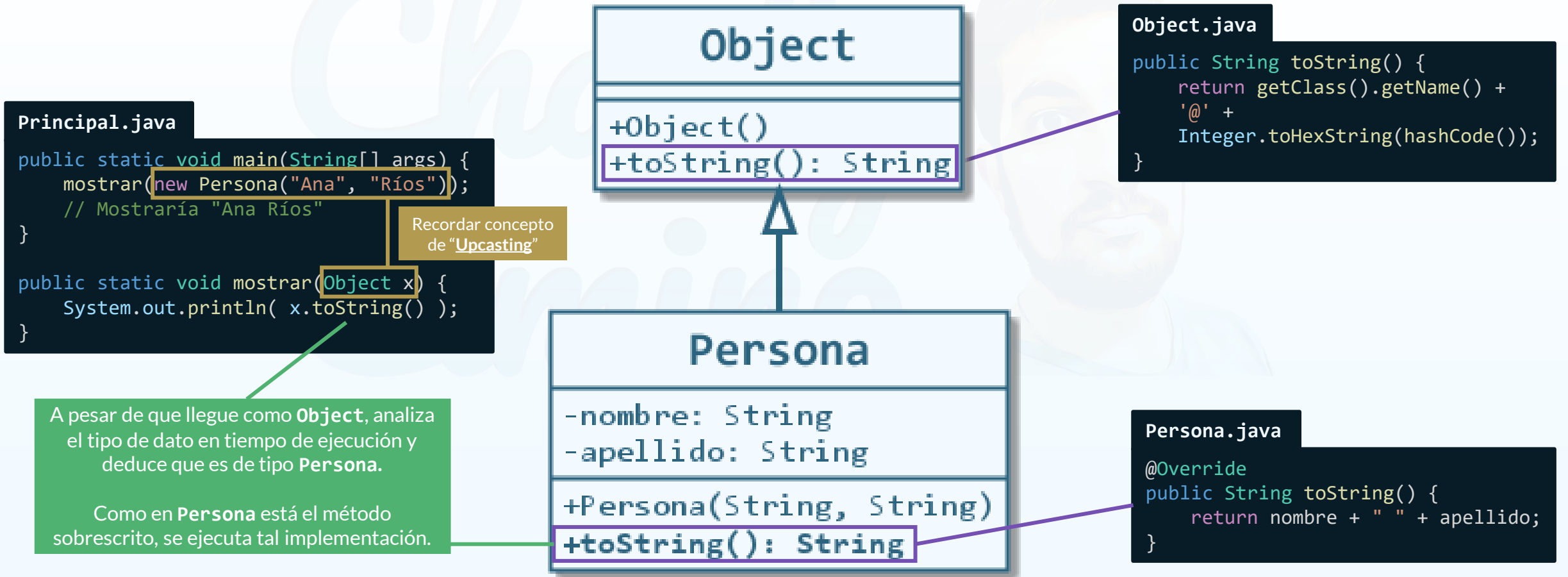
Ambos entienden el mensaje, pero lo responden de maneras diferentes.

Polimorfismo en acción



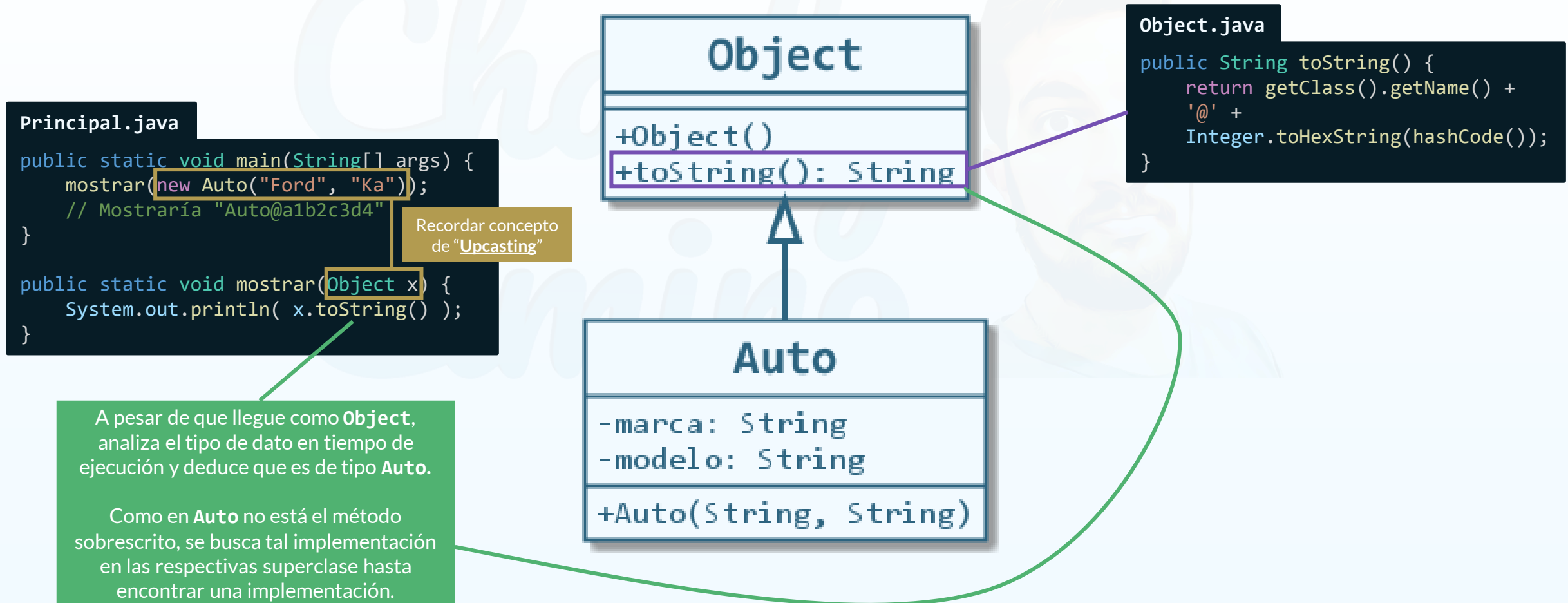
Ligadura dinámica

No se sabe exactamente qué método se va a ejecutar.
Se liga en tiempo de ejecución el llamado al método con su respectiva implementación.

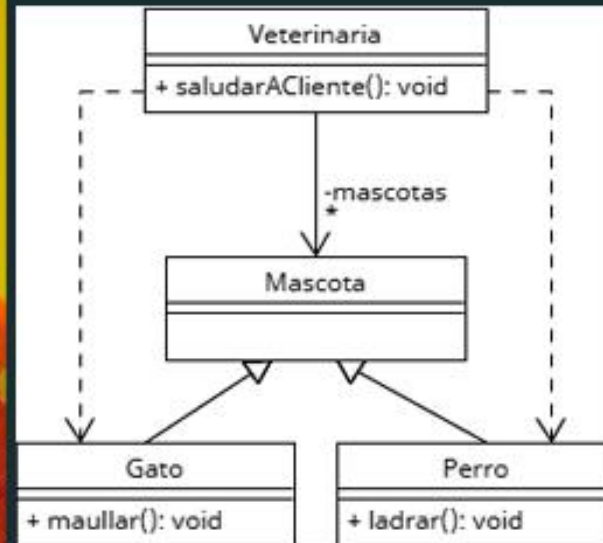


Ligadura dinámica

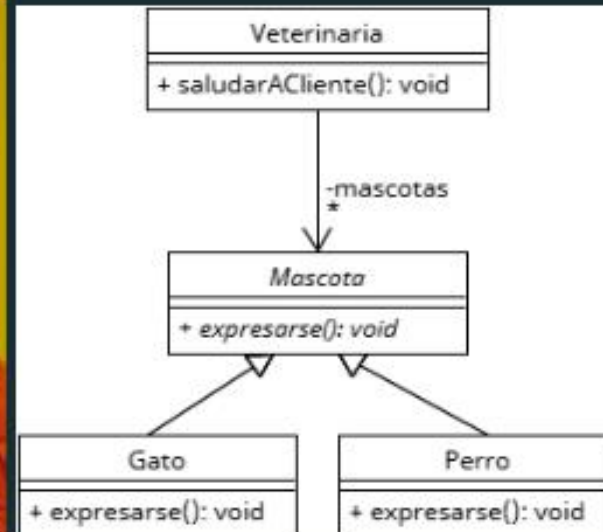
No se sabe exactamente qué método se va a ejecutar.
Se liga en tiempo de ejecución el llamado al método con su respectiva implementación.



¿Por qué usar polimorfismo?



```
// Método que NO usa polimorfismo
public void saludarACliente() {
    for (Mascota mascota : mascotas) {
        if (mascota instanceof Perro) {
            ((Perro) mascota).ladrar();
        } else if (mascota instanceof Gato) {
            ((Gato) mascota).maullar();
        }
    }
}
```



```
// Método que usa polimorfismo
public void saludarACliente() {
    for (Mascota mascota : mascotas) {
        mascota.expresarse();
    }
}
```

FIN DE LA PRESENTACIÓN

Encontrá más como estas en mi [sitio web](#).