

Patrones de diseño

# BRIDGE

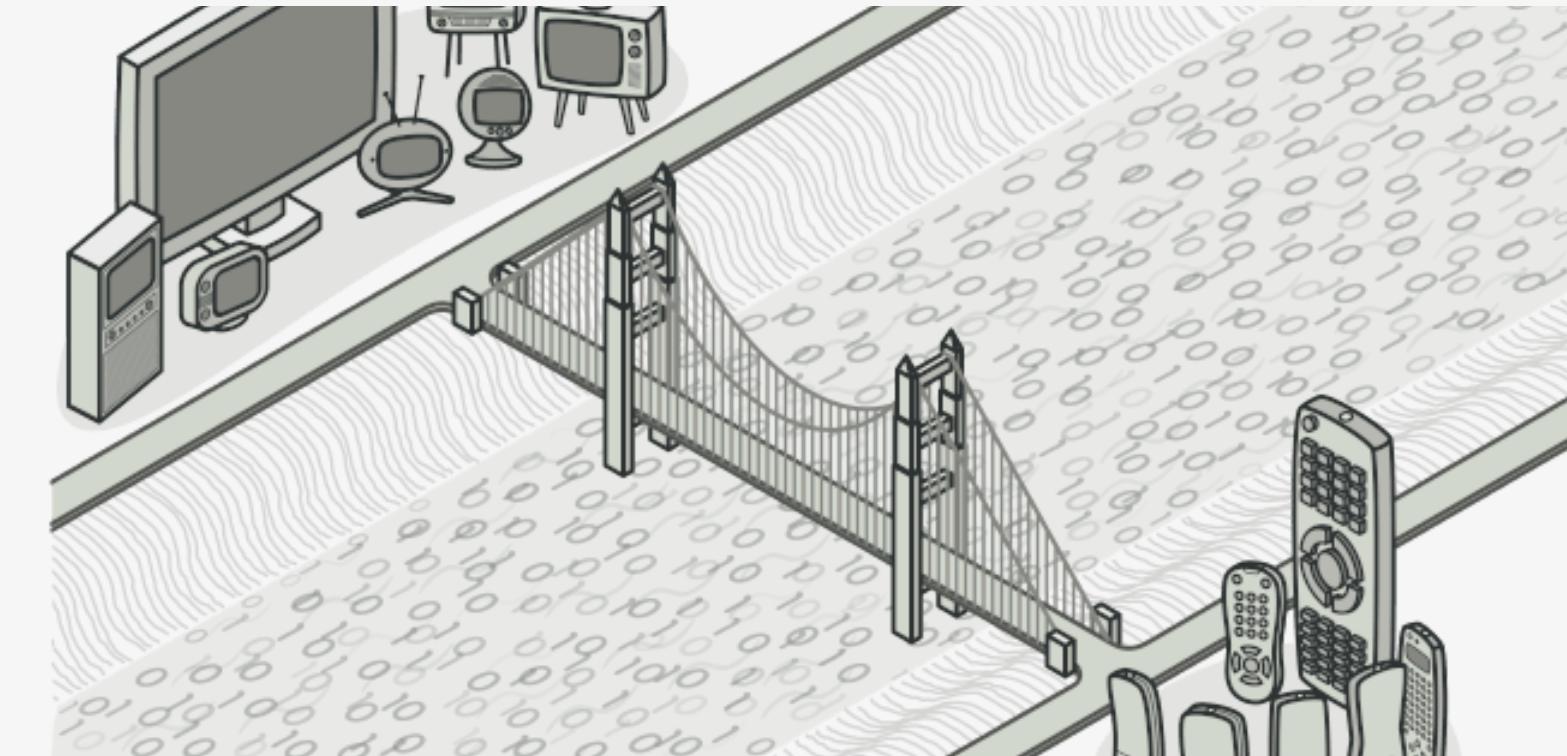
*Dylan Montiel Zúñiga*



# INTRODUCCIÓN

La **Abstracción (interfaz)** es una capa de control de alto nivel para una entidad. Esta capa no tiene que hacer ningún trabajo real por su cuenta, sino que debe delegar el trabajo a la capa de **implementación (plataforma)**.

**Bridge** es un patrón de diseño estructural que permite dividir una clase grande, o un grupo de clases estrechamente relacionadas, en dos jerarquías separadas (abstracción e implementación) que pueden desarrollarse independientemente la una de la otra.



# PROBLEMA

1

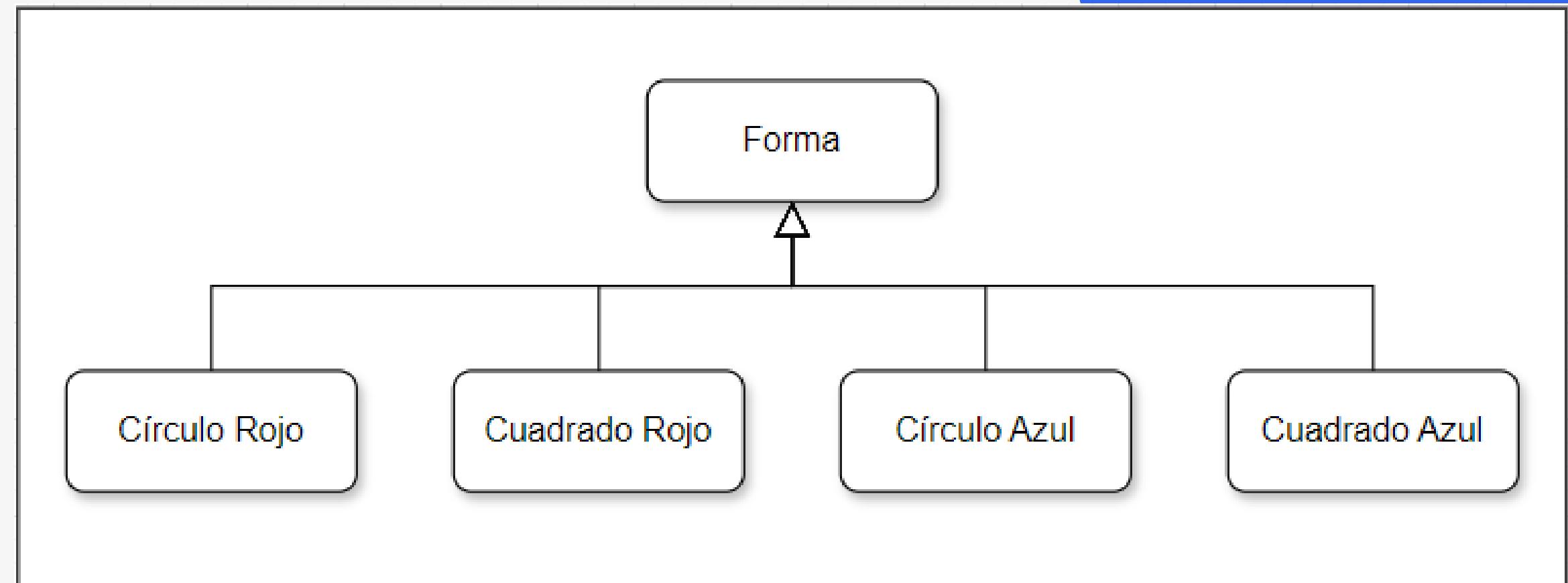
Tenemos una clase geométrica **Forma** con un par de subclases: **Círculo** y **Cuadrado**.

2

Se desea extender esta jerarquía de clase para que incorpore colores, creando las subclases de forma **Rojo** y **Azul**.

3

Sin embargo, como ya se cuenta con dos subclases, se tienen que crear cuatro combinaciones de clase, como **CírculoRojo** y **CuadradoAzul**.



# SOLUCIÓN

1

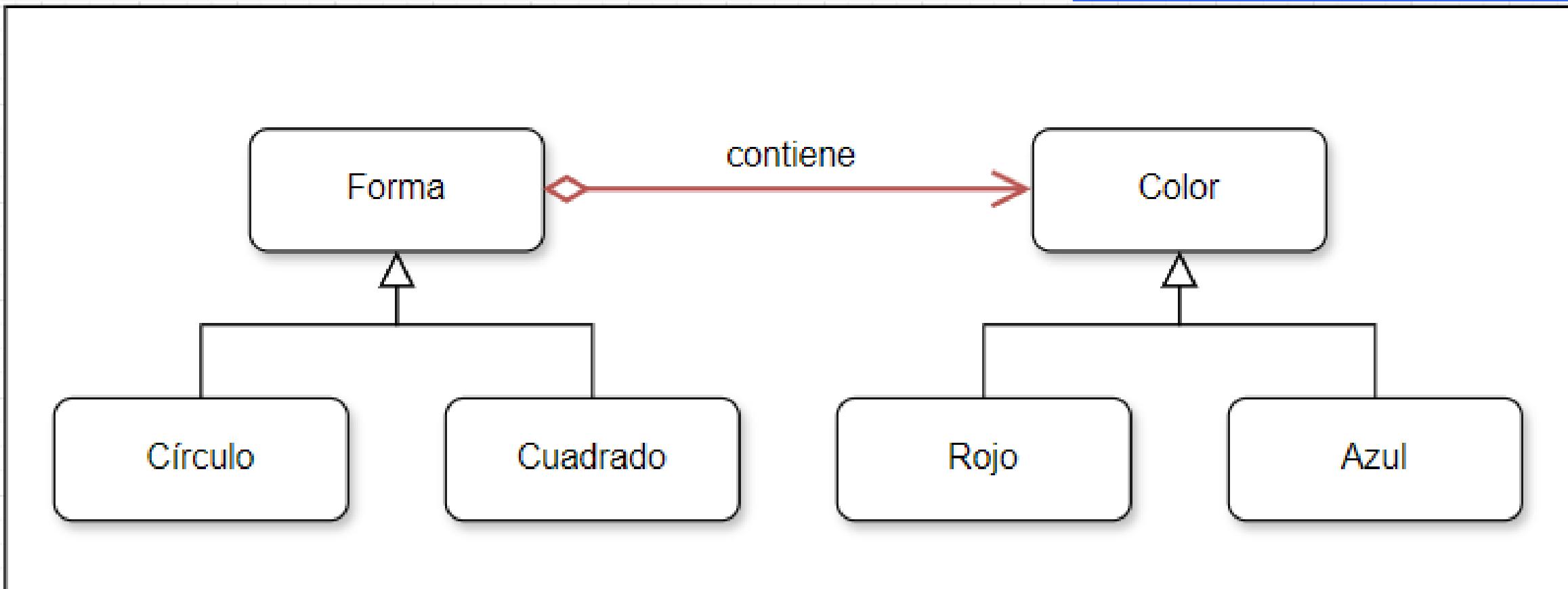
El patrón **Bridge** intenta resolver este problema pasando de la herencia a la composición del objeto.

2

Esto quiere decir que se extrae una de las dimensiones a una jerarquía de clases separada.

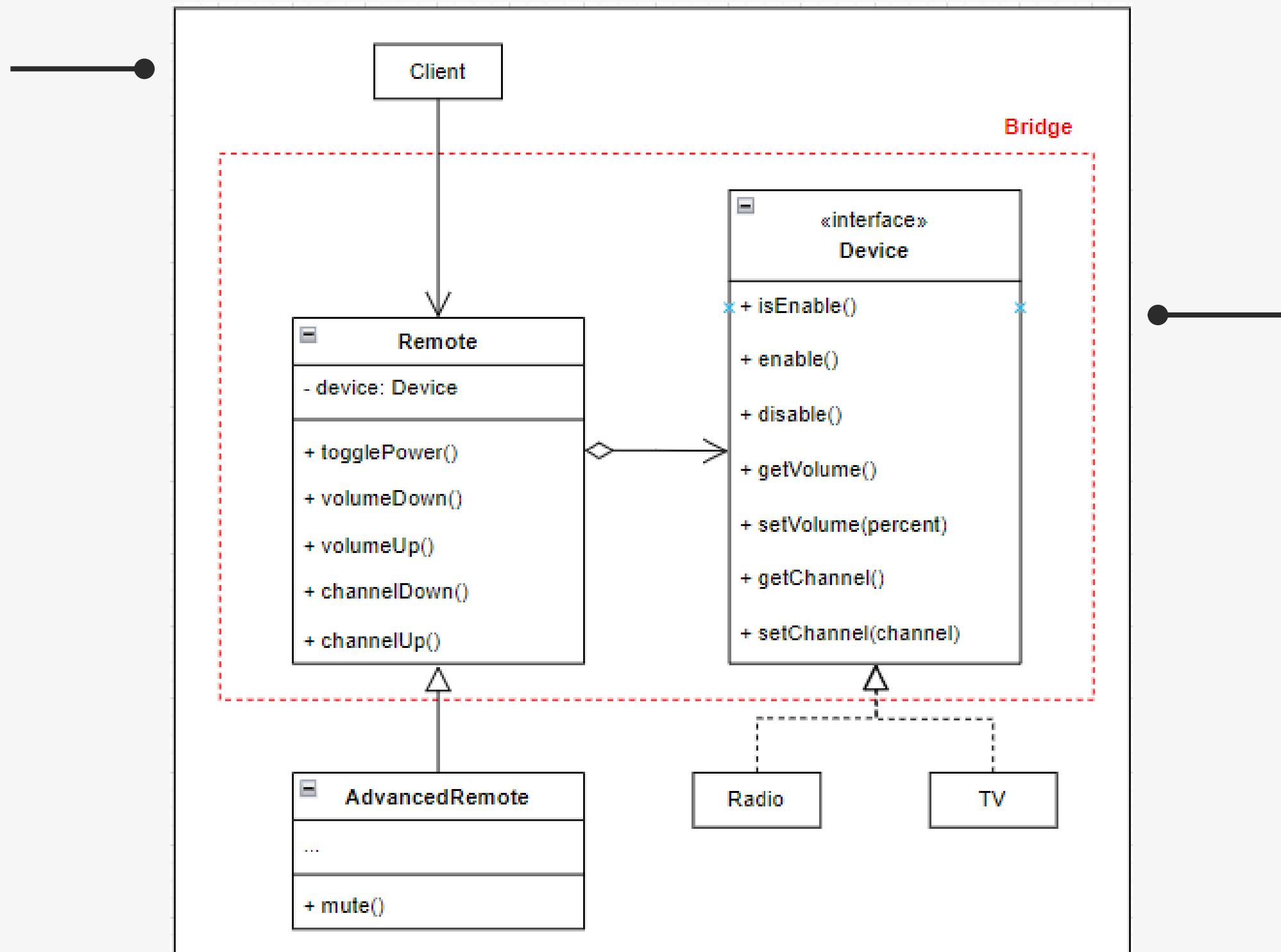
3

De modo que las clases originales referencian un objeto de la nueva jerarquía, en lugar de tener todo su estado y sus funcionalidades dentro de una clase.



# EJEMPLO

Normalmente, el Cliente sólo está interesado en trabajar con la abstracción. No obstante, el cliente tiene que vincular el objeto de la abstracción con uno de los objetos de la implementación.



Este ejemplo ilustra cómo puede ayudar el patrón **Bridge** a dividir el código de una aplicación que gestiona dispositivos y sus controles remotos. Las clases Dispositivo actúan como implementación, mientras que las clases Remoto actúan como abstracción.

# REFERENCIAS

- [1] [HTTPS://REFACTORING.GURU/ES/DESIGN-PATTERNS/BRIDGE](https://refactoring.guru/es/design-patterns/bridge)
- [2] [HTTPS://REACTIVEPROGRAMMING.IO/BLOG/ES/PATRONES-DE-DISEÑO/BRIDGE](https://reactiveprogramming.io/blog/es/patrones-de-diseño/bridge)
- [3] [HTTPS://WWW.YOUTUBE.COM/WATCH?V=6BIHZQMDGG&T=4S](https://www.youtube.com/watch?v=6BIHZQMDGG&t=4s)
- [4] [HTTPS://WWW.YOUTUBE.COM/WATCH?V=NDBOEDD8Q\\_4&T=236S](https://www.youtube.com/watch?v=NDBOEDD8Q_4&t=236s)