

BRIDGE

TIPOS DE PATRONES



BRIDGE

Patrón de diseño estructural que te 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.

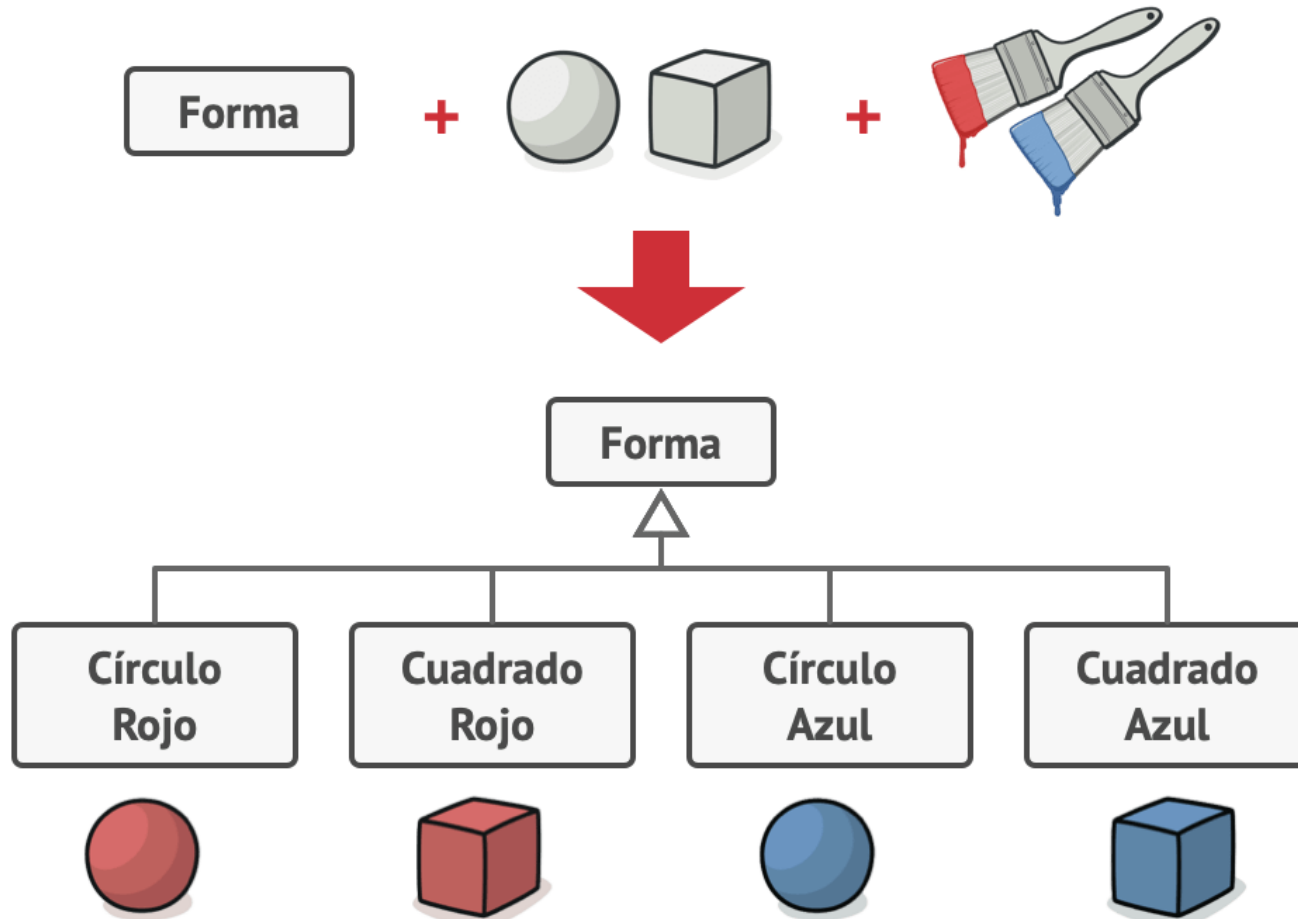
PROBLEMA

Digamos que tienes una clase Forma con un par de subclases: Círculo y Cuadrado.

PROBLEMA

Deseas extender esta jerarquía de clase para que incorpore colores, por lo que planeas crear las subclases de forma Rojo y Azul.

PROBLEMA



PROBLEMA

Sin embargo, como ya tienes dos subclases, tienes que crear cuatro combinaciones de clase, como `CírculoAzul` y `CuadradoRojo`.

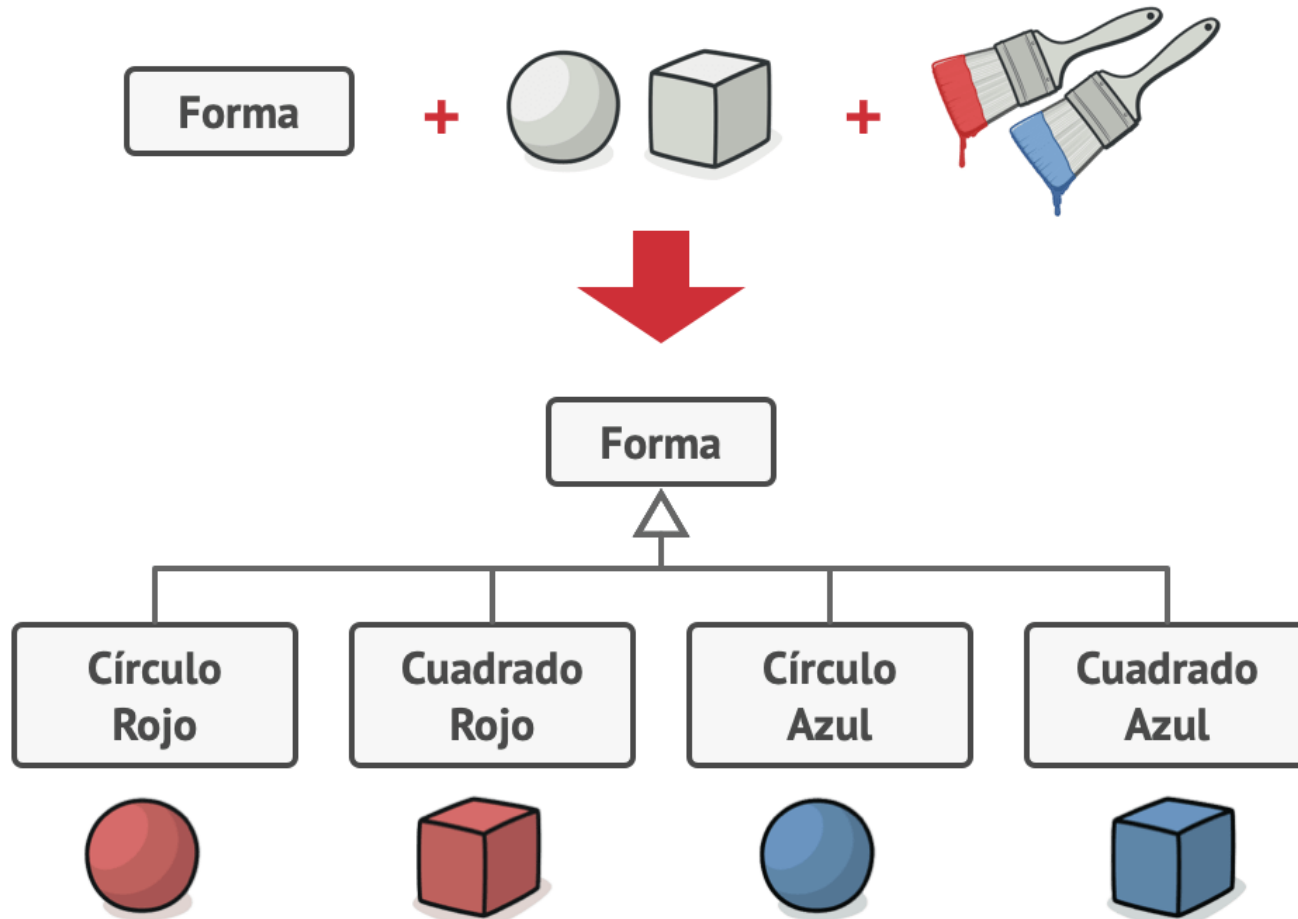
PROBLEMA

Añadir nuevos tipos de forma y color a la jerarquía hará que ésta crezca exponencialmente. Por ejemplo, para añadir una forma de triángulo deberás introducir dos subclases, una para cada color.

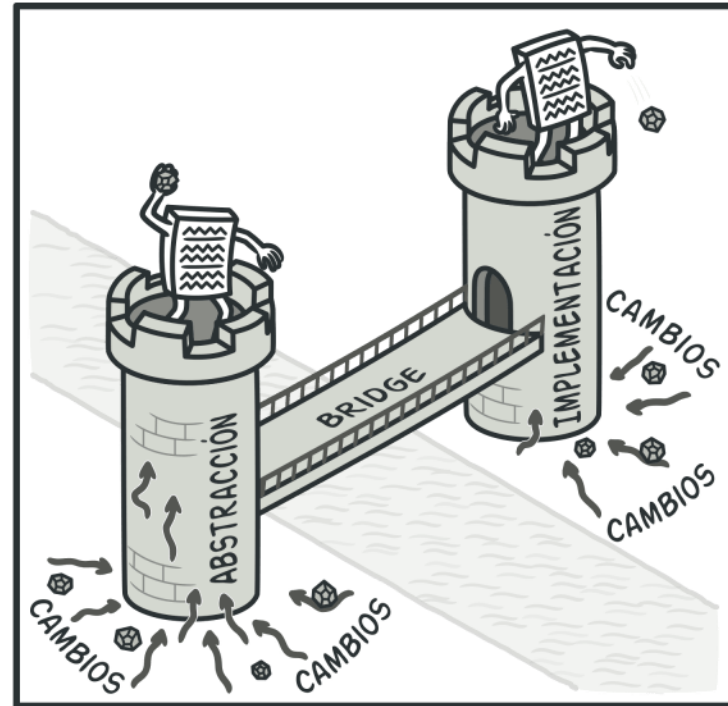
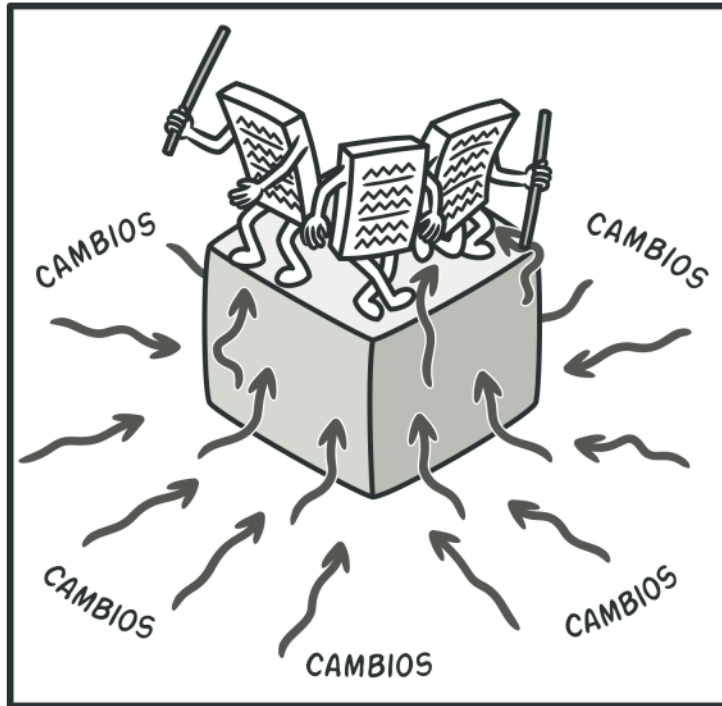
PROBLEMA

Para añadir un nuevo color habrá que crear tres subclases, una para cada tipo de forma. Cuanto más avancemos, peor será.

PROBLEMA



SOLUCIÓN



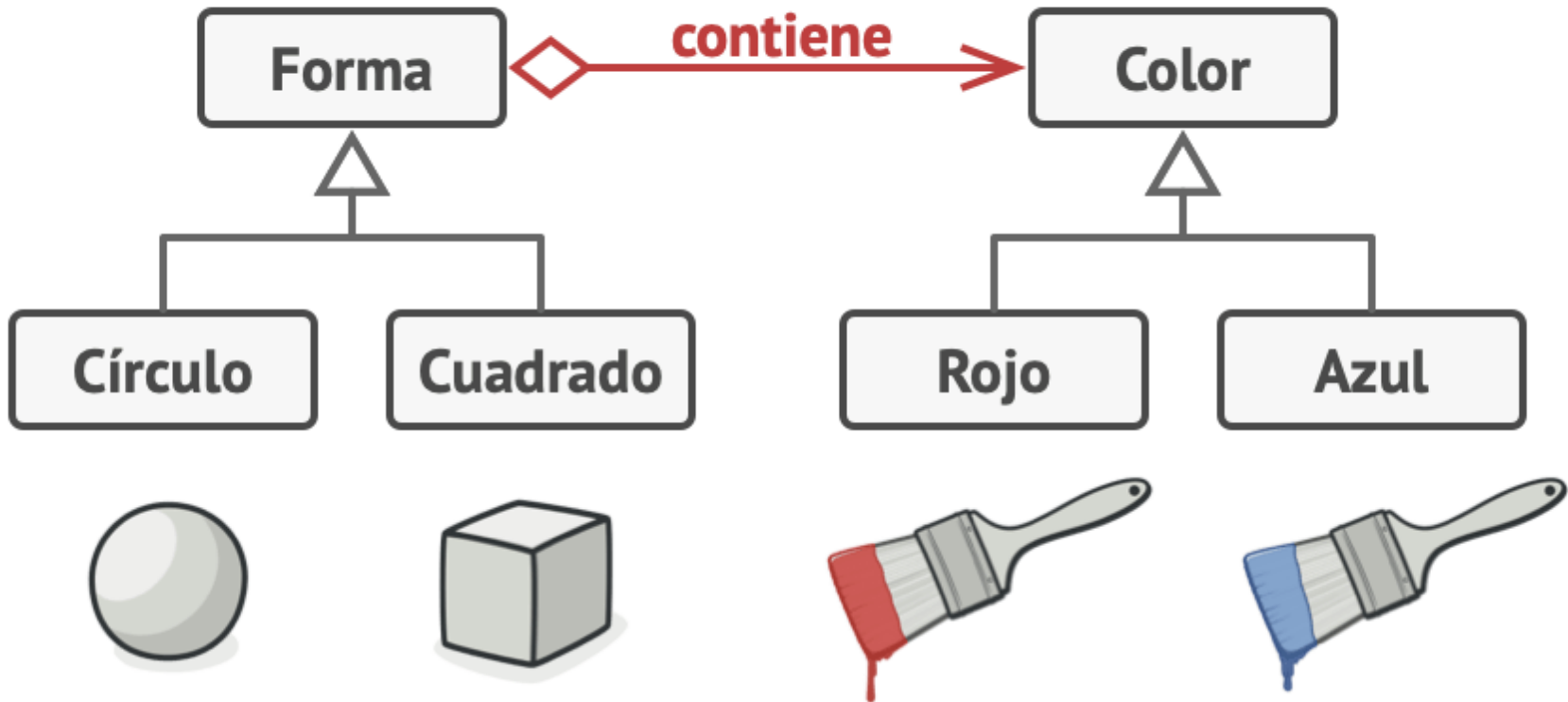
SOLUCIÓN

Usar composición del objeto en vez de herencia.

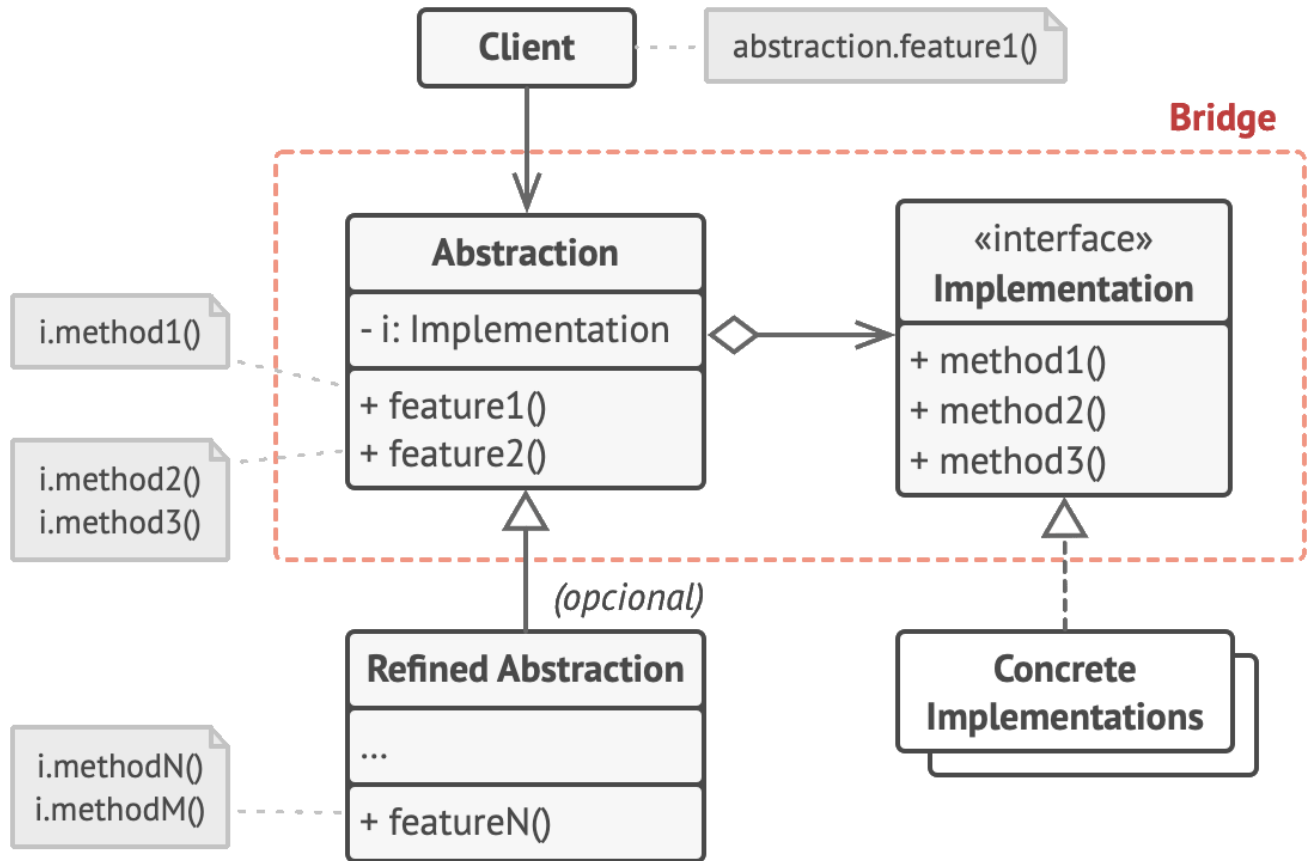
SOLUCIÓN

Se extrae una de las dimensiones a una jerarquía de clases separada, de modo que las clases originales referencian un objeto de la nueva jerarquía.

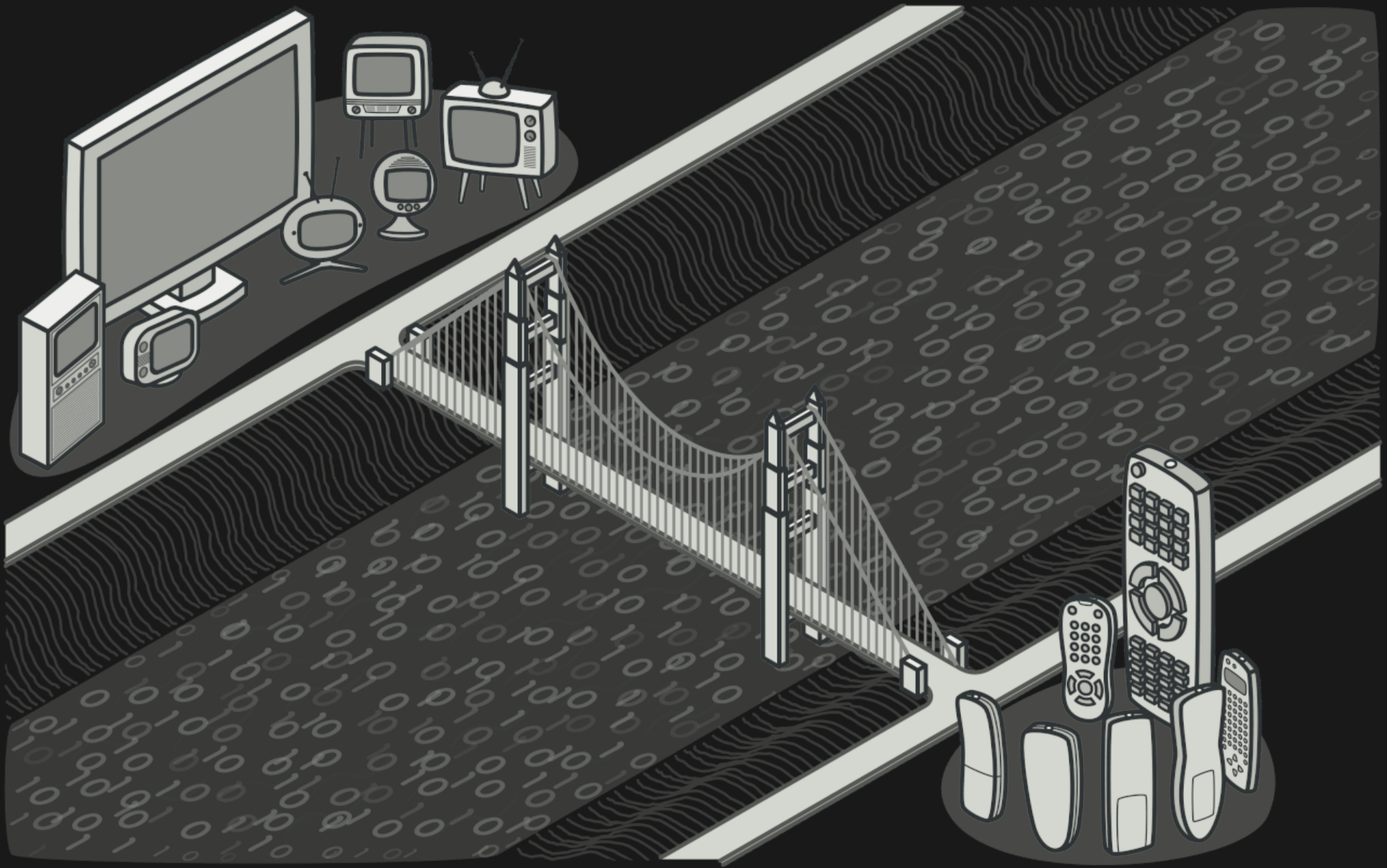
SOLUCIÓN



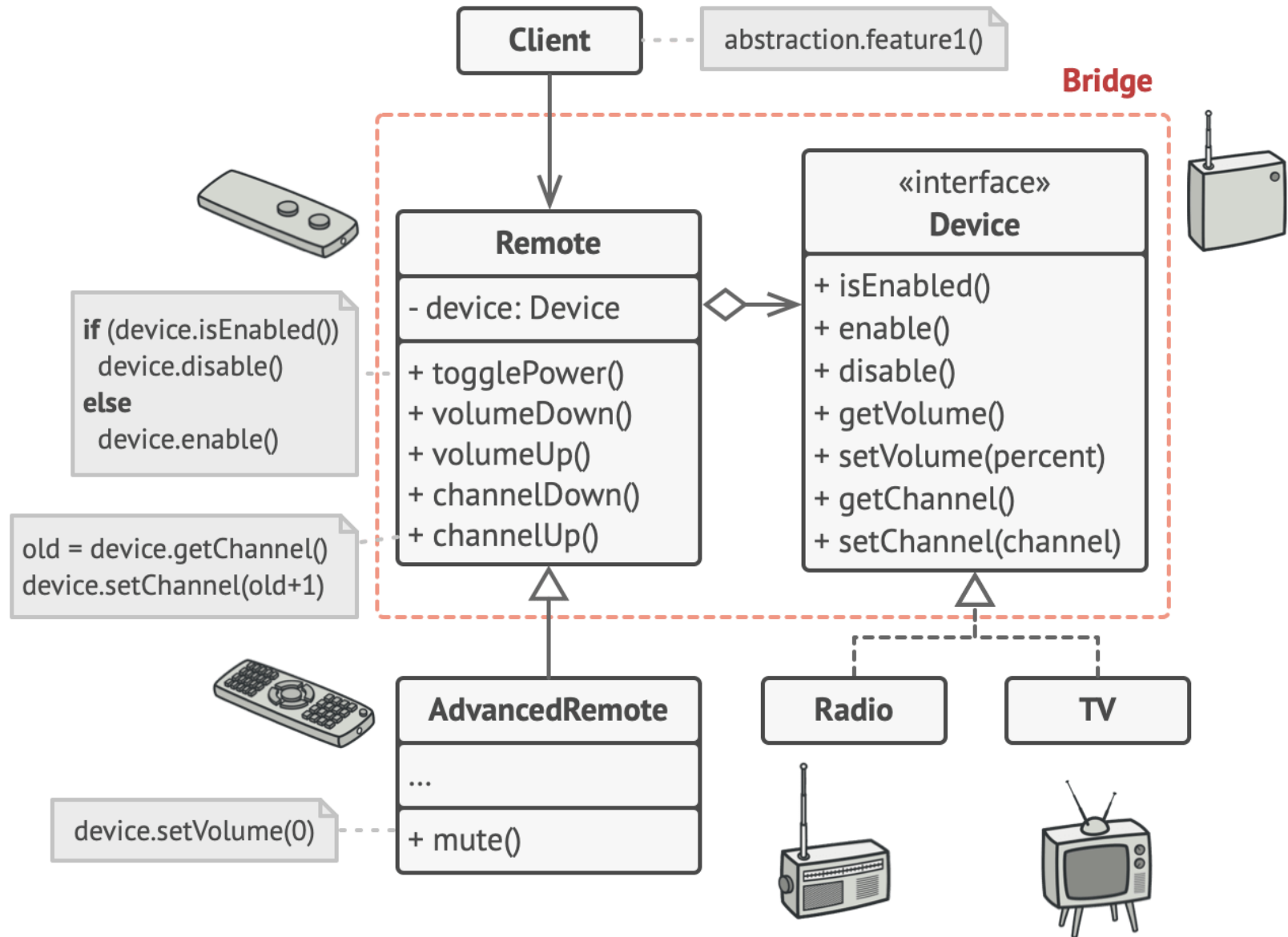
ESTRUCTURA



EJEMPLO



EJEMPLO



VENTAJAS

- Puedes crear clases y aplicaciones independientes de plataforma.
- El código cliente funciona con abstracciones de alto nivel.
- Puedes introducir nuevas abstracciones e implementaciones independientes entre sí.
- Principio de responsabilidad única.

DESVENTAJAS

Puede ser que el código se complique si aplicas el patrón a una clase muy cohesionada.

