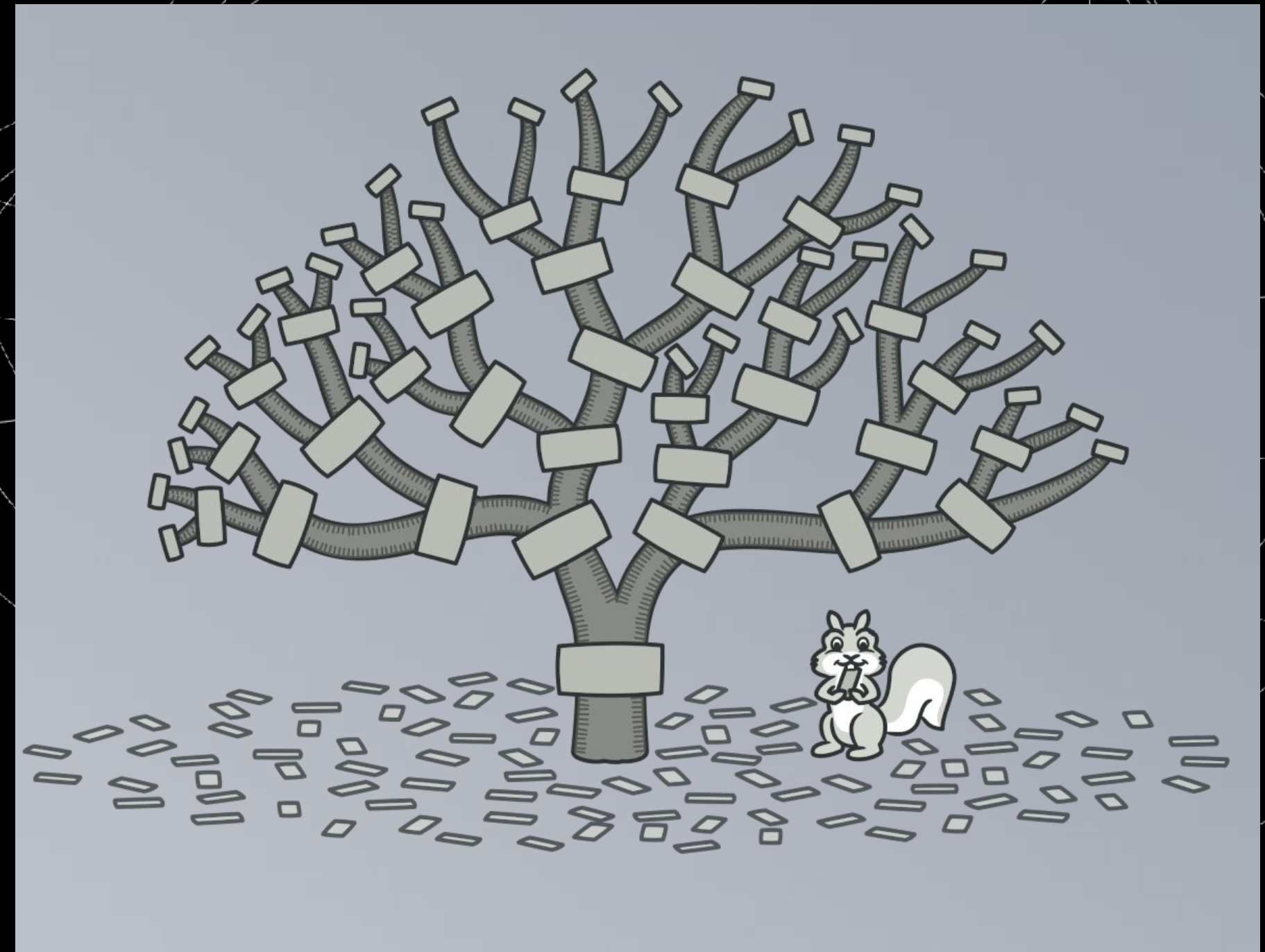


Patrón de diseño

COMPOSITE

Propósito

Composite es un patrón de diseño estructural que te permite componer objetos en estructuras de árbol y trabajar con esas estructuras como si fueran objetos individuales.



Problema

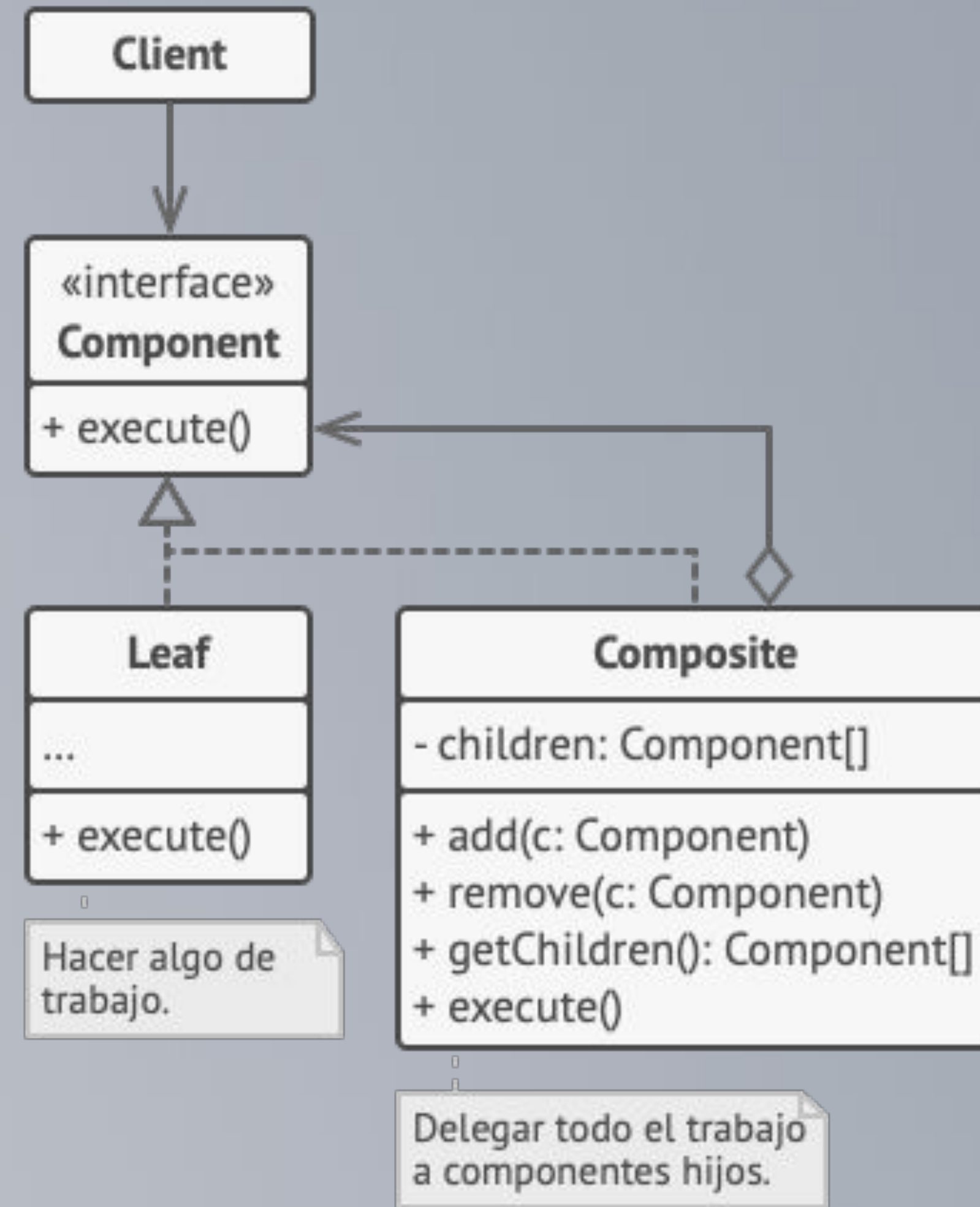
- El uso del patrón Composite sólo tiene sentido cuando el modelo central de tu aplicación puede representarse en forma de árbol.



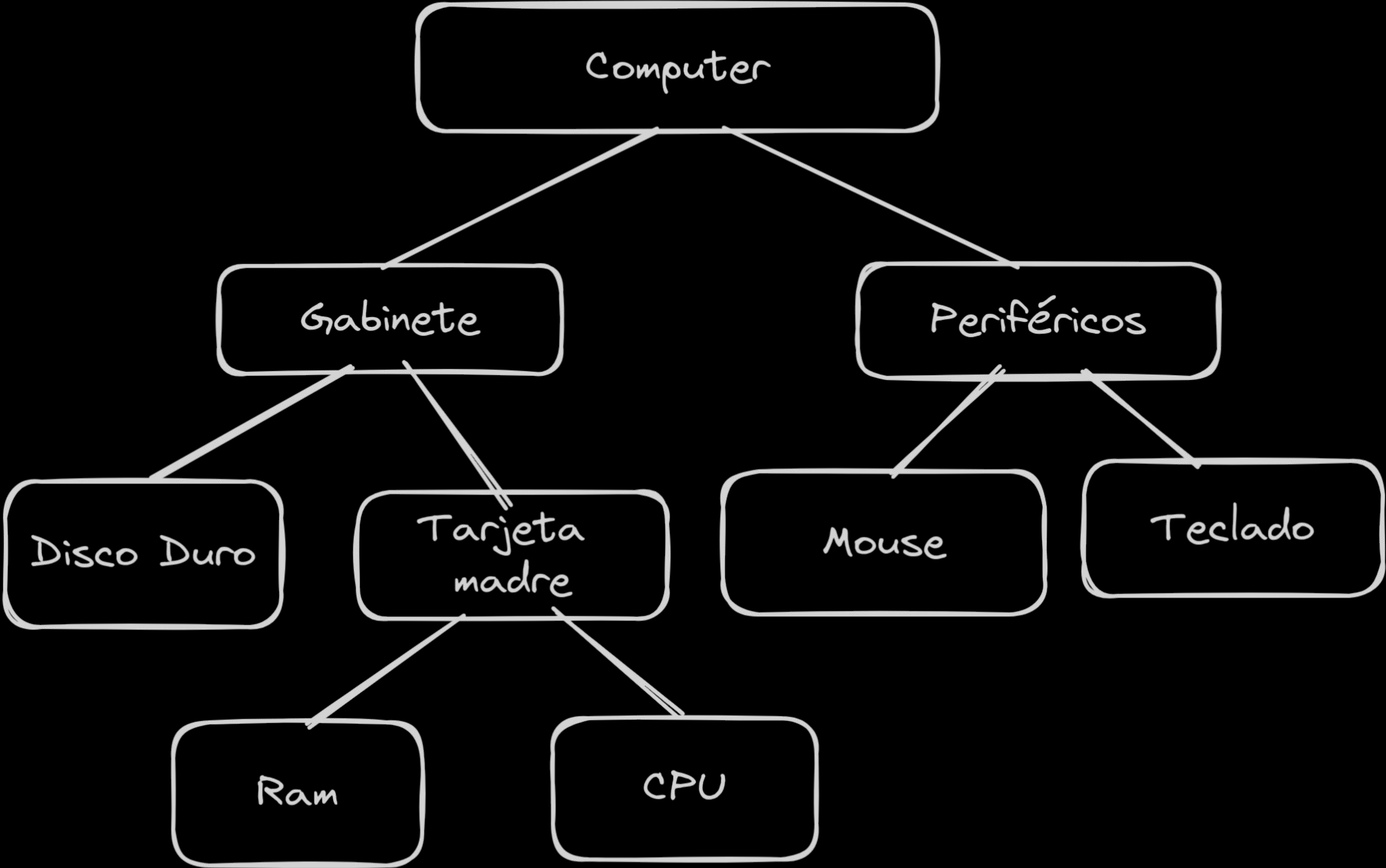
Solución

- El patrón Composite sugiere que trabajes con Productos y Cajas a través de una interfaz común que declare un método para calcular el precio total.
- ¿Cómo funcionaría este método?
Para un producto, sencillamente devuelve el precio del producto. Para una caja, recorre cada artículo que contiene la caja, pregunta su precio y devuelve un total por la caja. Si uno de esos artículos fuera una caja más pequeña, esa caja también comenzaría a repasar su contenido y así sucesivamente, hasta que se calcule el precio de todos los componentes internos. Una caja podría incluso añadir costos adicionales al precio final, como costos de empaquetado.





Estructura



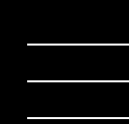
Aplicabilidad

-Utiliza el patrón Composite cuando tengas que implementar una estructura de objetos con forma de árbol.

El patrón Composite te proporciona dos tipos de elementos básicos que comparten una interfaz común: hojas simples y contenedores complejos. Un contenedor puede estar compuesto por hojas y por otros contenedores. Esto te permite construir una estructura de objetos recursivos anidados parecida a un árbol.

-Utiliza el patrón cuando quieras que el código cliente trate elementos simples y complejos de la misma forma.

Todos los elementos definidos por el patrón Composite comparten una interfaz común. Utilizando esta interfaz, el cliente no tiene que preocuparse por la clase concreta de los objetos con los que funciona.



¡GRACIAS!