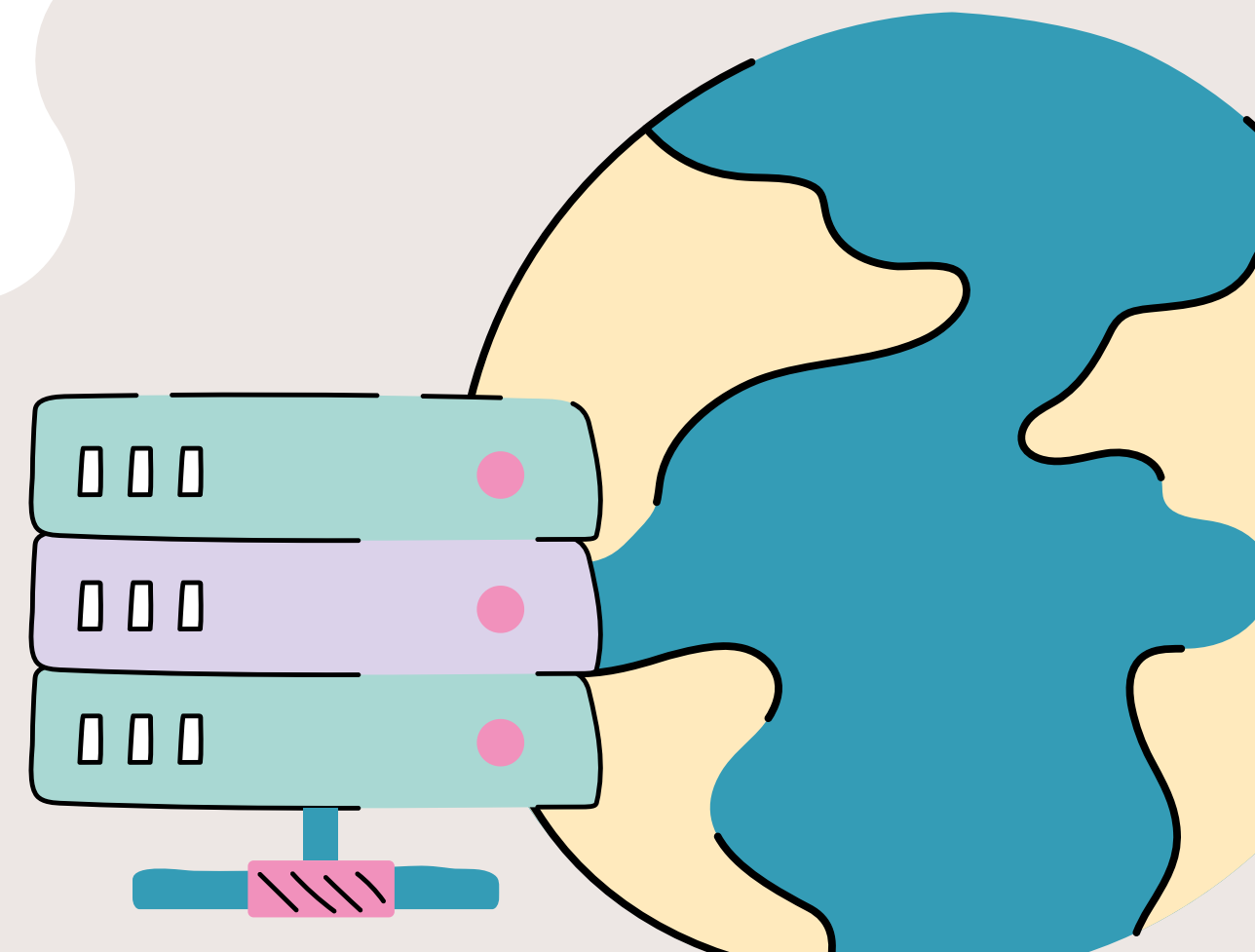
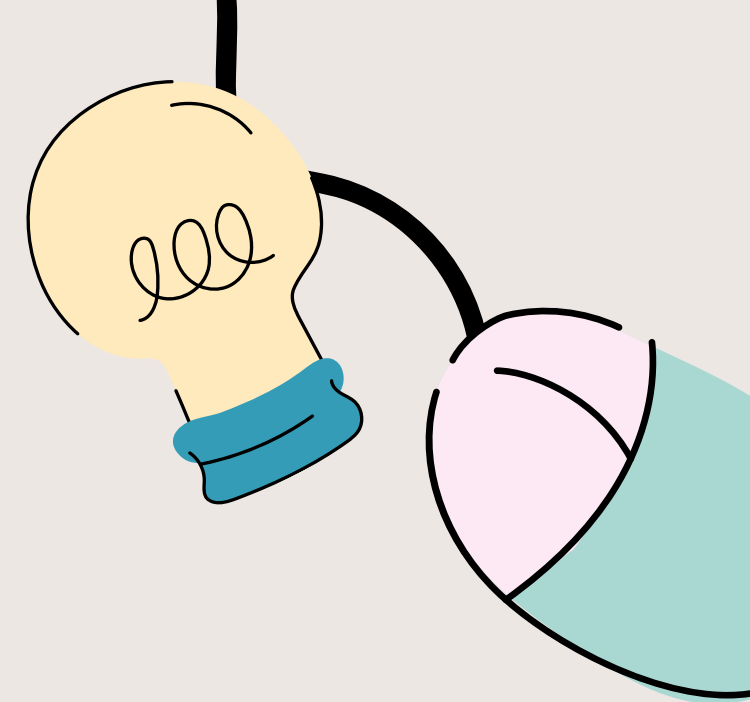
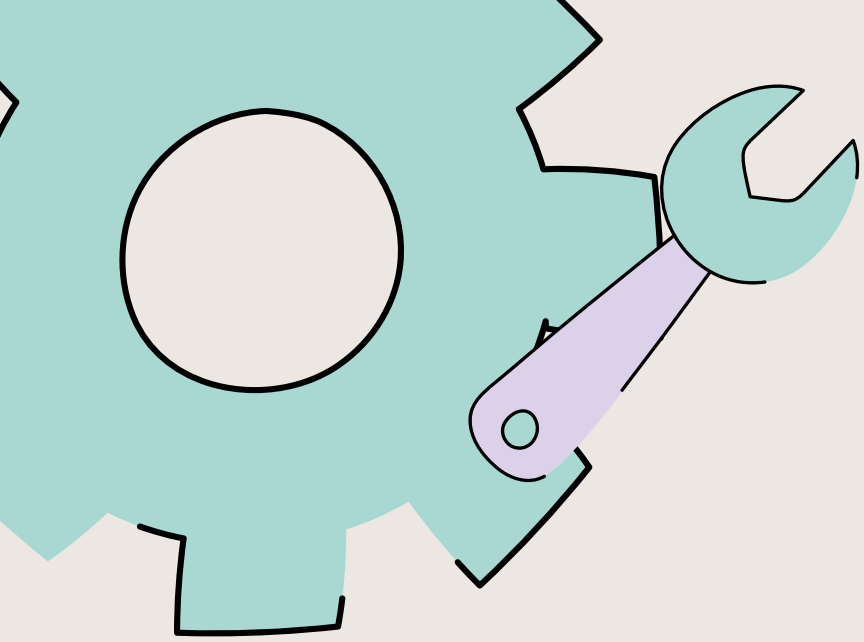


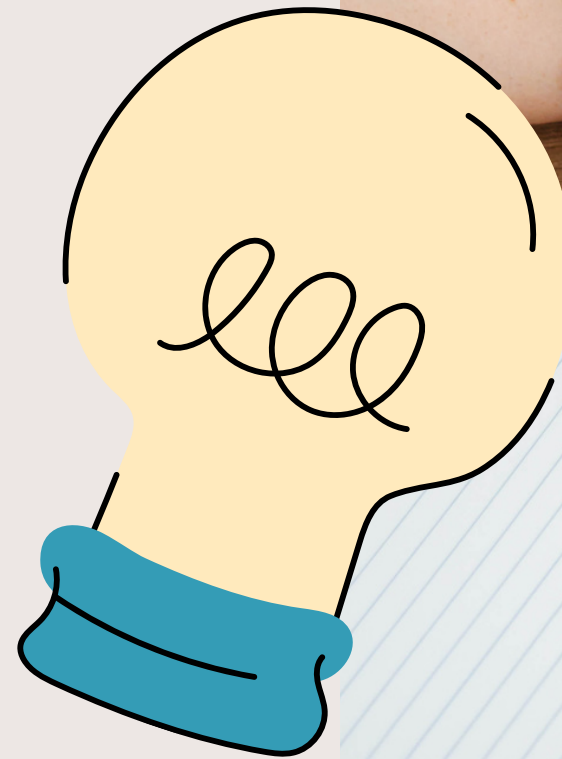
# Patrón Composite

Por Georlenys Martínez



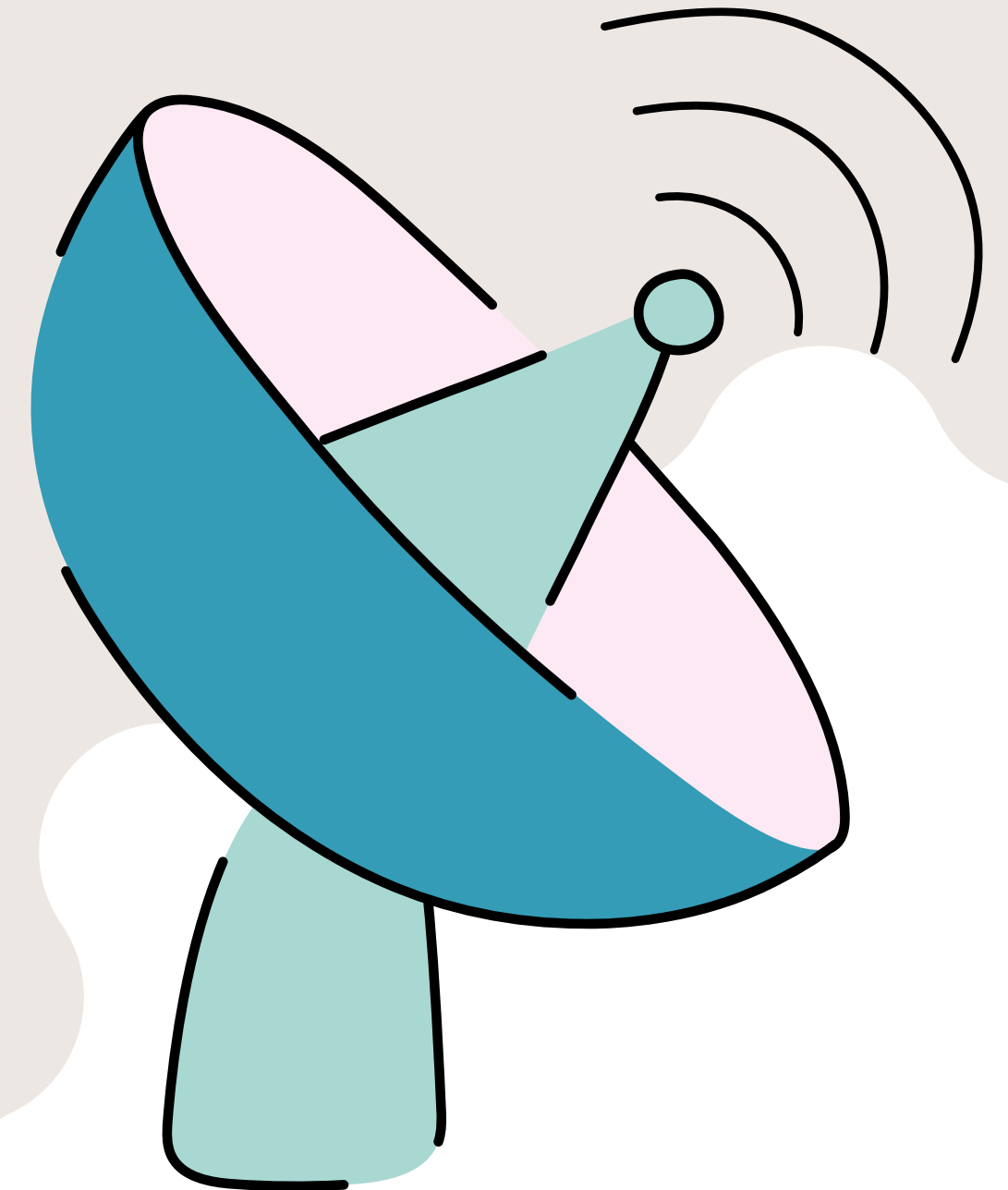
# Introducción:

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.

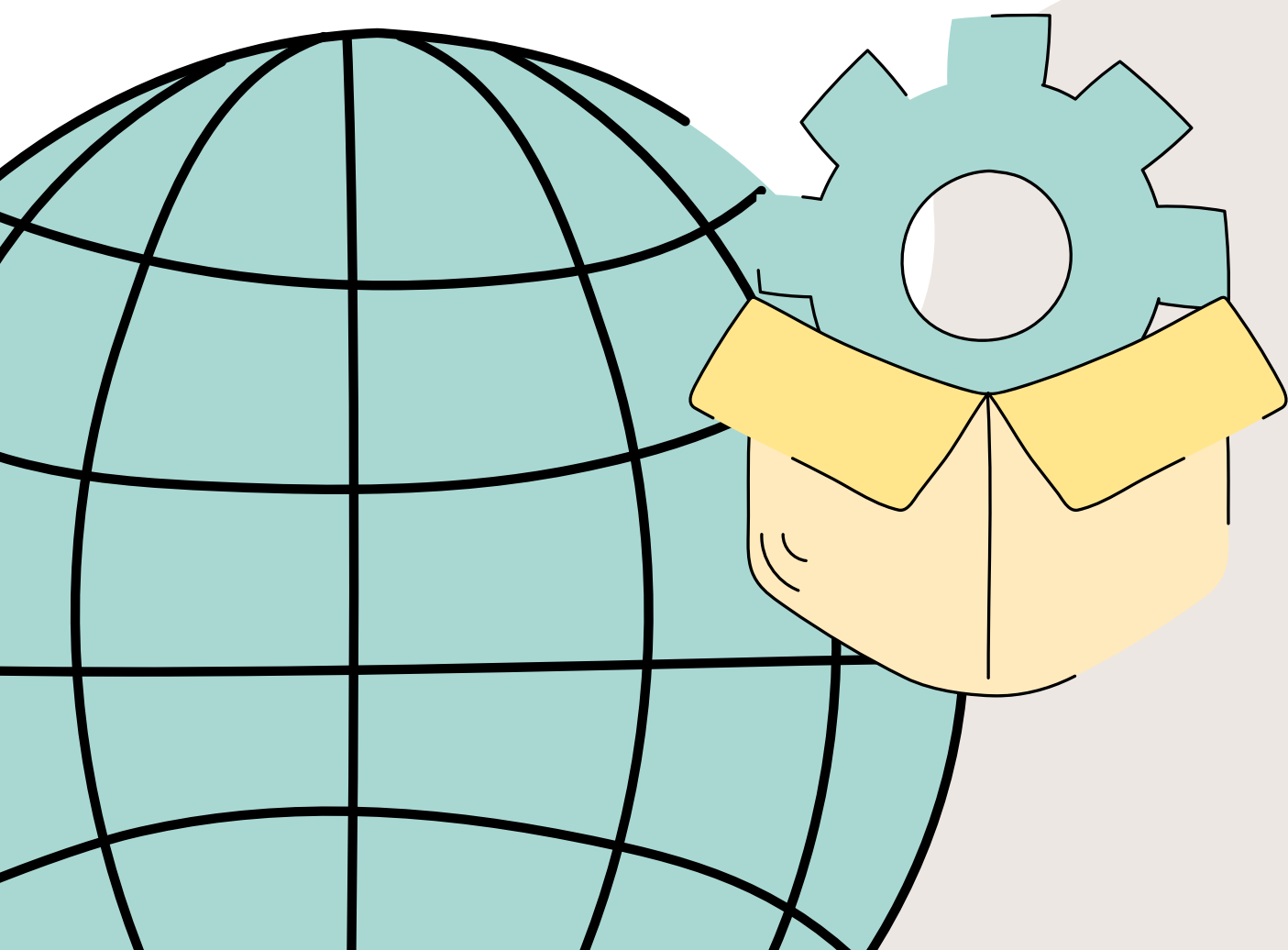


# ¿Qué es el patrón Composite?

El concepto básico del patrón Composite consiste en representar objetos simples y sus containers en una clase abstracta de manera que puedan ser tratados uniformemente. Este tipo de estructura se conoce como jerarquía parte-todo, en la que un objeto es siempre, o una parte de un todo, o un todo compuesto por varias partes.



# ¿Qué problemas resuelve un diagrama UML de patrón Composite?



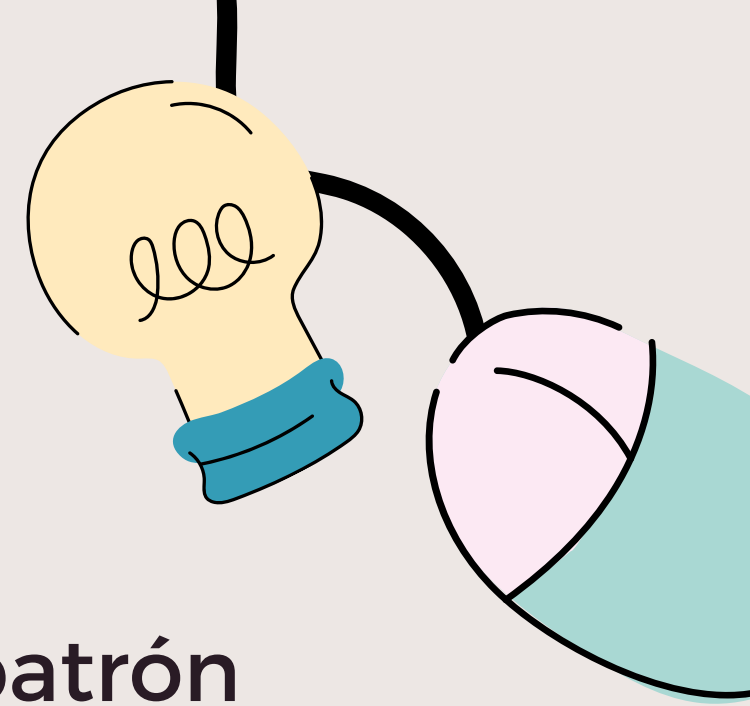
La finalidad del patrón Composite es, mejorar la gestión de problemas de diseño recurrentes en la programación orientada a objetos. El resultado deseado es un software flexible, caracterizado por objetos fácilmente implementables, intercambiables, reutilizables y testeables.

A tal efecto, el patrón de diseño Composite describe un método según el cual los objetos simples y complejos pueden ser tratados de la misma manera. De este modo, se puede crear una estructura de objetos fácilmente inteligible que permite al cliente un acceso altamente eficiente. Además, también se minimiza la probabilidad de error en el código.





# Patrón de diseño Composite:

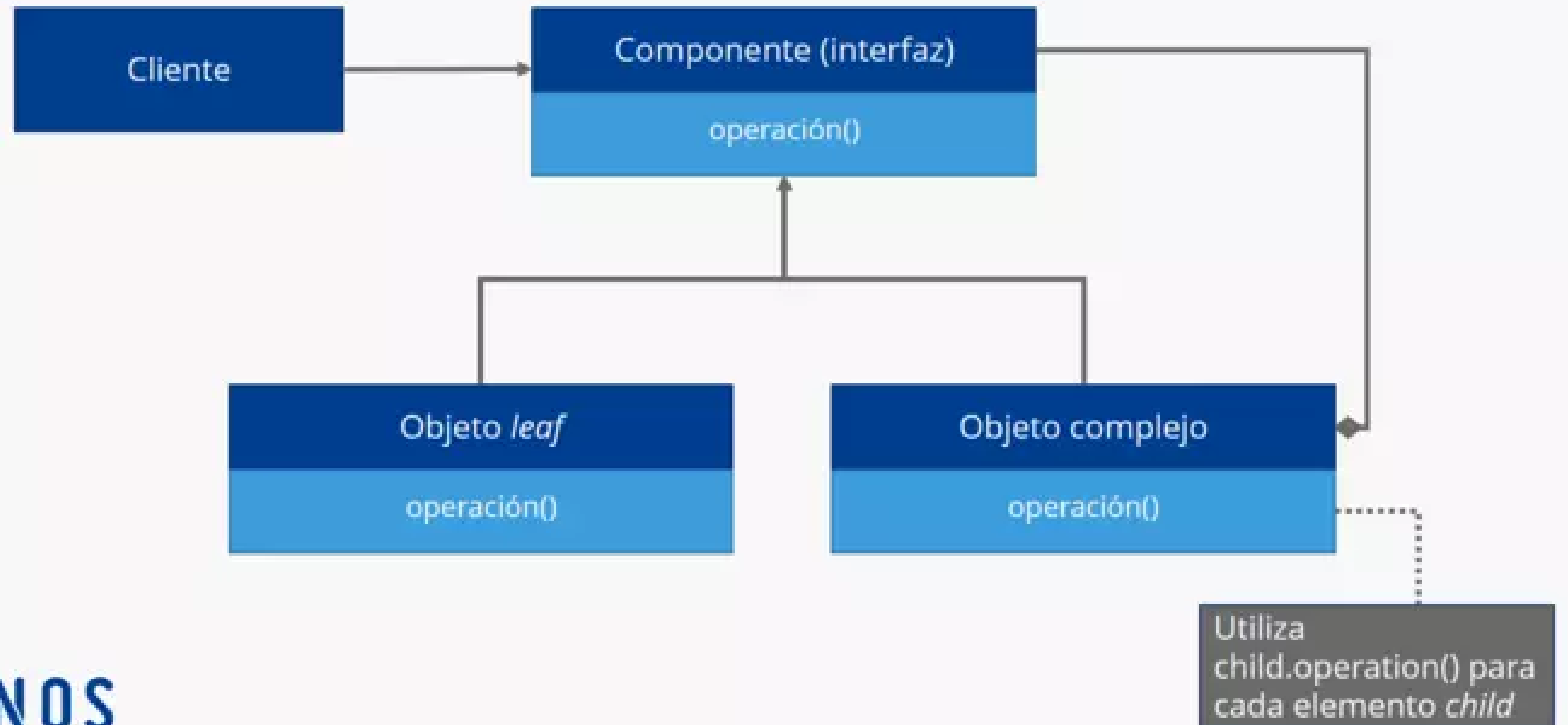


Para implementar jerarquías parte-todo, el patrón Composite proporciona una interfaz de componentes unificada para los objetos simples, también llamados objetos hoja, y los objetos complejos. Los objetos hoja simples integran esta interfaz directamente, mientras que los objetos complejos envían peticiones específicas del cliente automáticamente a la interfaz y a sus componentes subordinados. Para el cliente, resulta totalmente indiferente de qué tipo de objeto se trate (parte o todo) ya que solo se comunica con la interfaz.



# Patrón de diseño Composite:

Diagrama UML: Patrón Composite



# **Campos de Aplicación del Patrón Composite**

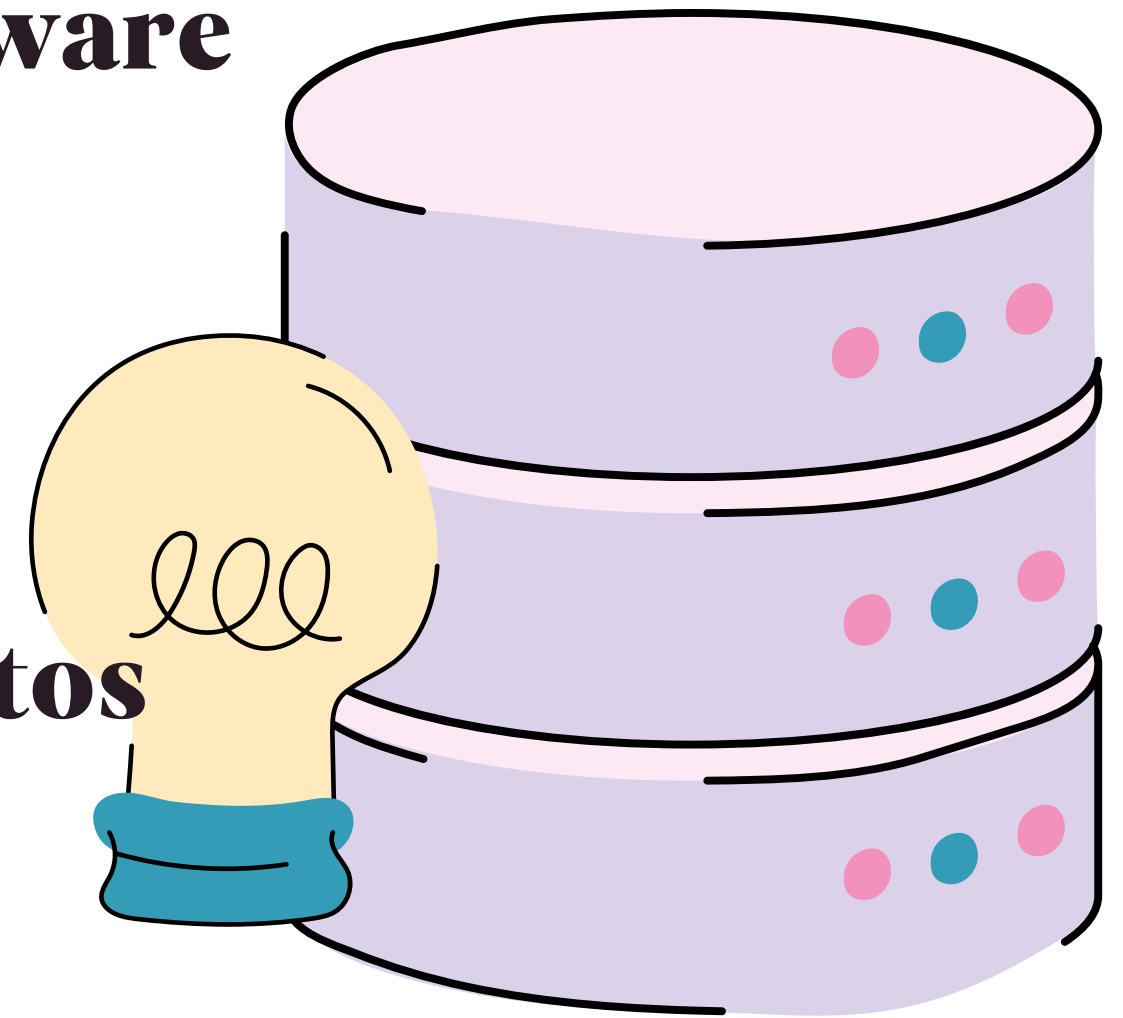
Utilizar el patrón Composite compensa cuando existe la necesidad de realizar operaciones en estructuras de datos dinámicas con jerarquías complejas. Esto también se conoce como estructura de árbol binaria, que resulta especialmente interesante para una amplia variedad de programas. Algunos ejemplos típicos:



**Interfaces gráficas  
de usuario (GUI)**

**Menús de software**

**Sistemas de datos**



# Ventajas

- Facilita la representación de estructuras altamente anidadas
- Código simple y conciso
- Gran escalabilidad



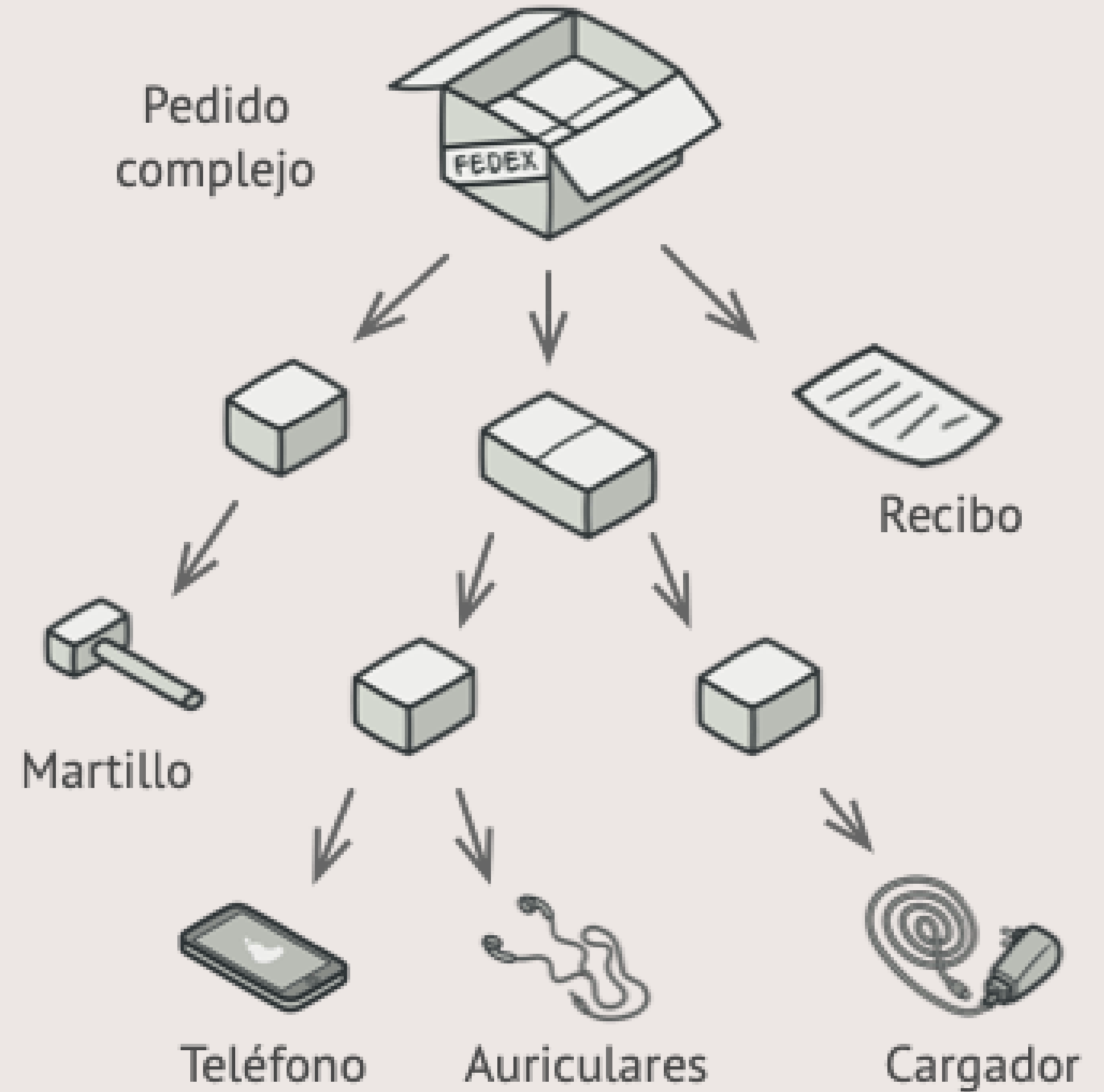
**versus**

# Desventajas

- Implementación de interfaz de componentes complicada
- Ajustes posteriores de las propiedades Composite complicados



# Ejemplo



# Solución

El patrón Composite sugiere que trabajes con Productos y Cajas a través de una interfaz común que declara un método para calcular el precio total.



The background is a solid light beige color. In the center is a white, cloud-like shape with a scalloped border. To the left, there are several overlapping geometric shapes: a pink rectangle with diagonal lines, a teal rectangle with a white circle, a purple triangle, a yellow triangle, and a teal circle with three vertical lines. To the right, there is a teal globe with a grid pattern, a purple cloud-like shape, and two large arrows, one pink and one yellow, pointing upwards and to the right.

# Gracias

