

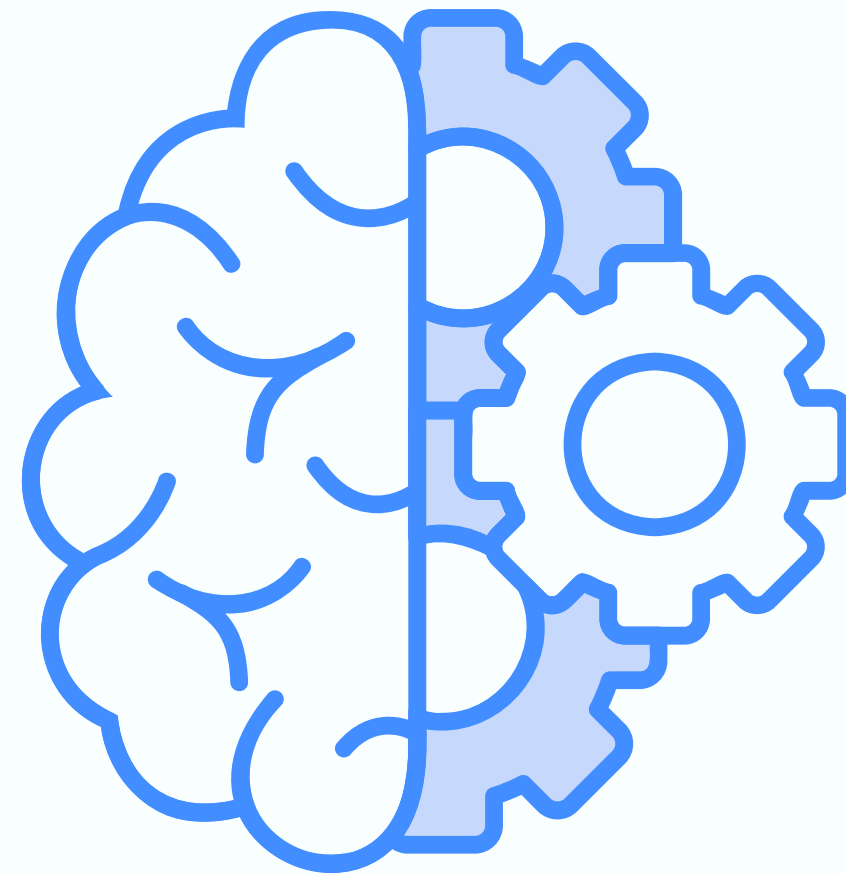
Octubre 2030

PATRÓN COMPOSITE

Bruno Lago

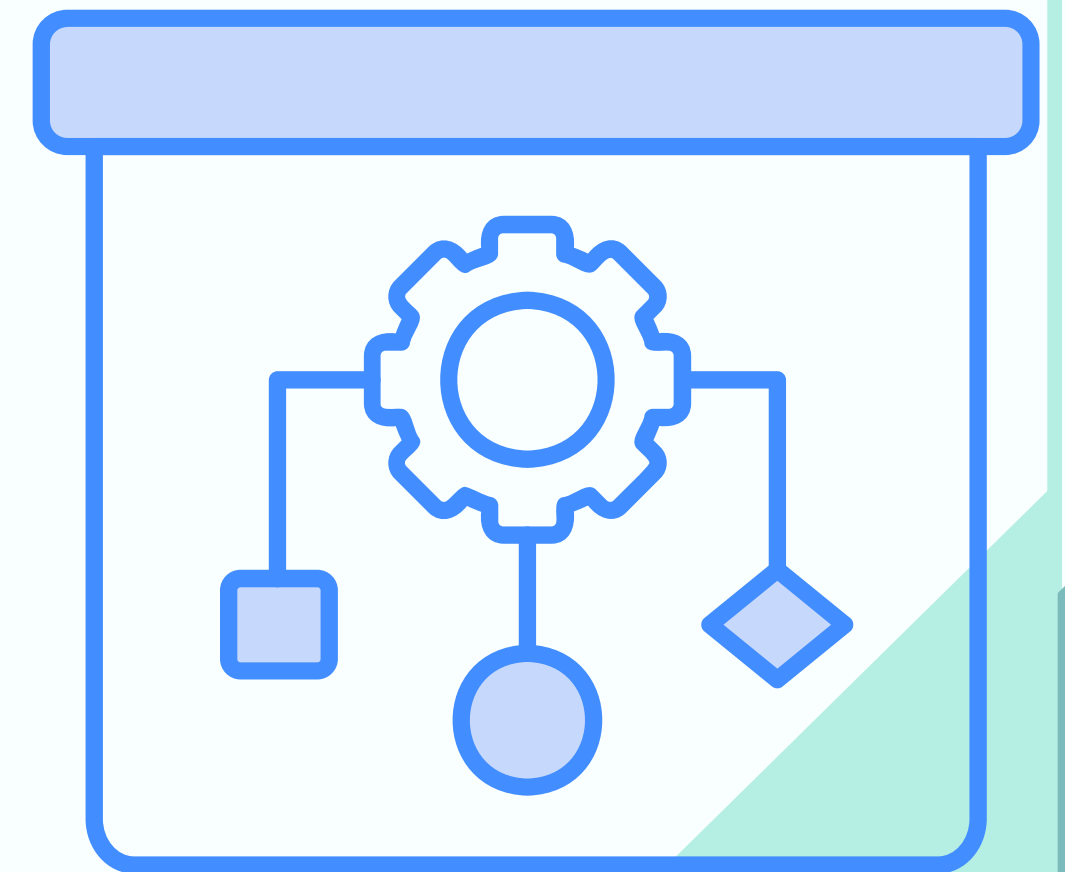
INTRODUCCIÓN

- El patrón Composite es un patrón de diseño estructural
- Organiza objetos en estructuras similares a árboles
- Permite tratar objetos individuales y composiciones de forma uniforme
- ¿Qué ejemplos del mundo real se te ocurren que usen estructuras compuestas?



PROPÓSITO

- Simplifica programas con composiciones recursivas
- Maneja objetos con similitudes
- Crea objetos complejos a partir de otros más simples
- ¿Cómo crees que este patrón puede ser útil en el desarrollo de software?



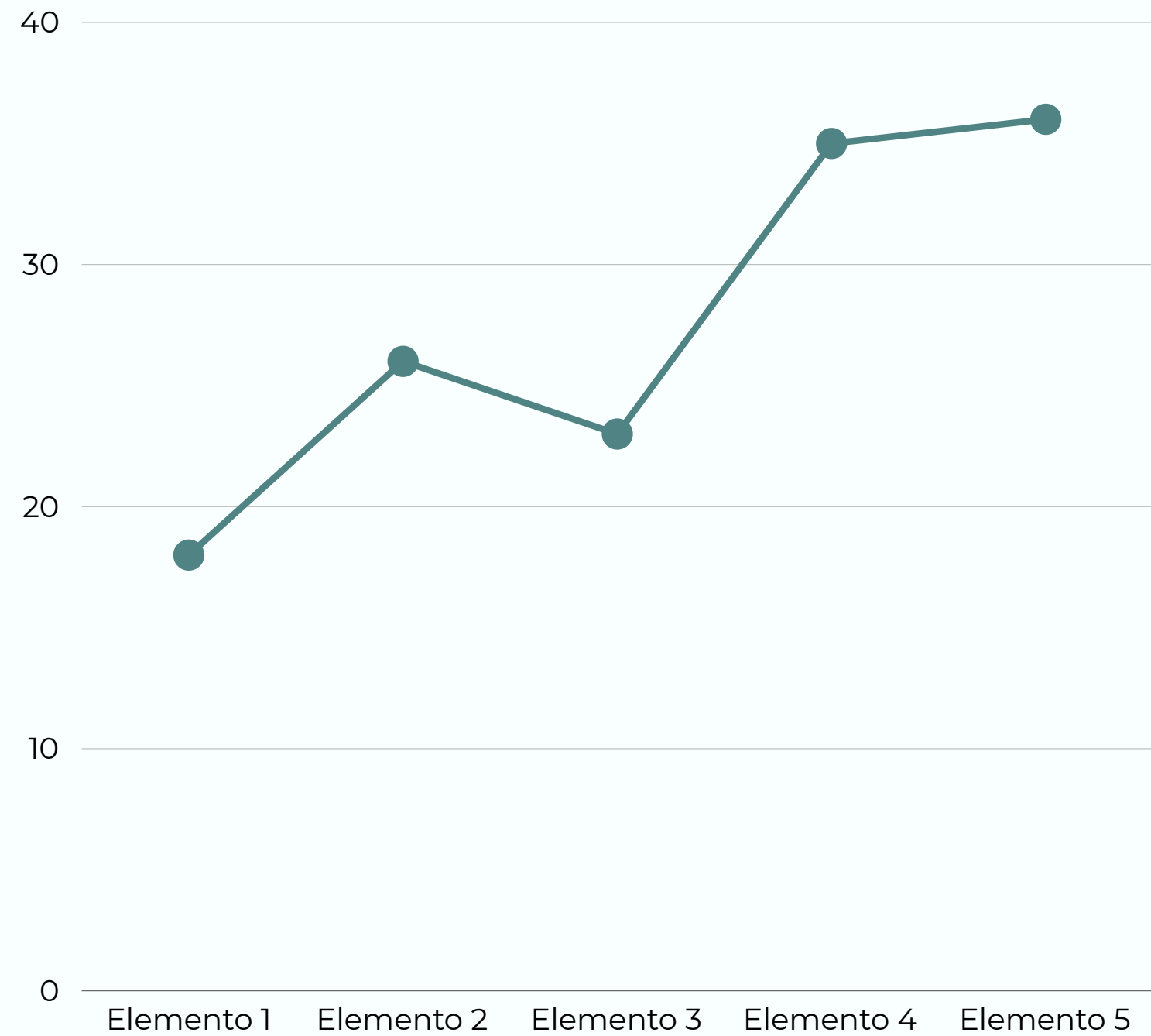
COMPONENTES CLAVE

- Componente: Interfaz o clase abstracta para todos los objetos
- Hoja (Leaf): Representa los objetos finales de una composición
- Composite: Contiene componentes hijos
- ¿Puedes pensar en una estructura jerárquica en tu vida diaria?



ESTRUCTURA JERÁRQUICA

- Se asemeja a una estructura en forma de árbol
- Raíz: Clase con características comunes
- Ramas/Hojas: Clases con características específicas
- ¿Cómo promueve esta estructura la organización del código?



HERENCIA

- Objetos con similitudes heredan de una base común
- Las clases derivadas agregan características específicas
- Simplifica el código y el manejo de objetos
- ¿Qué beneficios tiene usar herencia en este contexto?

R E C U R S I Ó N E N

- Los objetos compuestos pueden contener otros compuestos
- - Permite estructuras anidadas de cualquier profundidad
- - Hace posible composiciones de objetos complejos
- ¿Cómo podría ser útil la recursión para representar jerarquías reales?
-

VENTAJAS

- Facilita la representación de estructuras profundamente anidadas
- Mejora la escalabilidad y flexibilidad
- Simplifica la adición de nuevos objetos al sistema
- Hace las consultas y el código más simples y concisos
- ¿Cuál de estas ventajas consideras más significativa?



DESVENTAJAS

- La implementación de la interfaz puede ser desafiante
- Requiere considerar cuidadosamente dónde ubicar las operaciones
- El diseño puede volverse demasiado general
- Puede ser difícil de entender en estructuras complejas
- ¿Cómo abordarías estas desventajas en tus propios diseños?

EJEMPLO: SISTEMA DE ARCHIVOS

- Archivos (objetos Hoja) y Carpetas (objetos Composite)
- Las carpetas pueden contener archivos y otras carpetas
- Permite un tratamiento uniforme de archivos y carpetas
 - ¿Cómo demuestra este ejemplo el patrón Composite?

IMPLEMENTACIÓN

- Define la interfaz del componente
- Crea clases hoja que implementen la interfaz
- Desarrolla clases compuestas que también implementen la interfaz
- Asegúrate de que las composiciones puedan manejar componentes hijos
- ¿Qué desafíos podrías enfrentar al implementar este patrón?

CUÁNDO USAR

- Cuando necesites representar jerarquías parte-todo
- Para estructuras que puedan anidarse recursivamente
- Para tratar objetos individuales y composiciones de forma uniforme
- ¿Puedes pensar en un proyecto donde aplicarías este patrón?



MUCHAS
GRACIAS