



Programación II

Clase 10

Programación Orientada
a Objetos

Patrones de Diseño

- Usos
- Tipos



01.

Patrones de Diseño

¿Qué son?

- Surgieron con la idea de realizar **desarrollo de software reusable**.
- Se los considera **buenas prácticas** (forma eficiente y escalable) para solucionar problemas conocidos y/o recurrentes.
- Se implementan en la etapa de diseño (antes del código).

Composición

- **Nombre**
Abstracción.
- **Problema**
Contexto en el que aplica el patrón.
- **Solución**
Esquemática. Define elementos y sus relaciones y responsabilidades.
- **Consecuencias**
popularidad complejidad extensibilidad

Tipos de Patrones

- **Creación**
Administrar la creación de objetos.
- **Comportamiento**
Interacción entre objetos.
- **Estructura**
Componer objetos en estructuras más grandes y complejas si perder flexibilidad y reutilización de código.



02.A

Composite

Objetos Compuestos

- **Objetivo**

Es un patón del tipo **estructural**.

Invocar de manera **polimórfica** tanto a un grupo de objetos (**composite**) como a los objetos individuales que lo componen (**leaf**).

- **Uso**

Interactuar de manera indistinta con objetos o grupos de objetos.

Útil en estructuras tipo árbol (**tree**) ya que implica recursión.

Objetos Compuestos

- **Requisitos**

Una clase debe agrupar objetos individuales, proveyendo los métodos para agregar o quitar objetos.

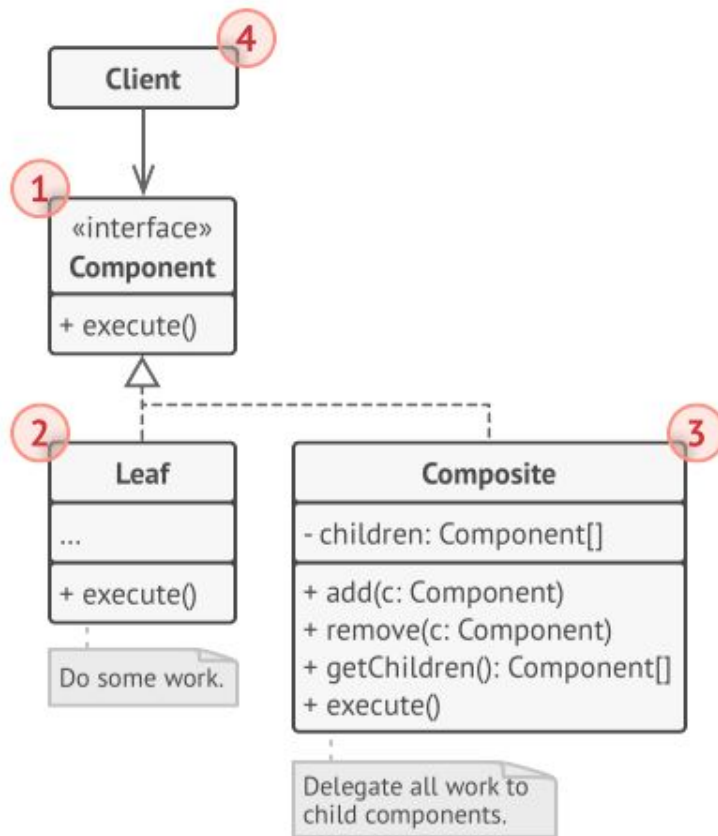
- **Retorno**

Según el método.

Si se invoca al objeto compuesto, invocará a cada uno de los objetos que lo componen.

Composite

UML



1. INTERFACE POLIMÓRFICA
2. OBJETO INDIVIDUAL
3. OBJETO COMPUESTO
4. INVOCADOR



02.B

Strategy

Estrategia

- **Objetivo**

Es un patón del tipo **comportamiento**.

Delegar a un objeto específico la implementación del algoritmo para realizar una tarea.

Este algoritmo puede cambiarse o ser parametrizado durante la invocación.

- **Uso**

Se utiliza cuando se desea especificar el mecanismo que se desea aplicar para resolver un determinado problema o cálculo.

Estrategia

- **Requisitos**

Un **contexto** que tenga como colaborador interno a la interface de las estrategias.

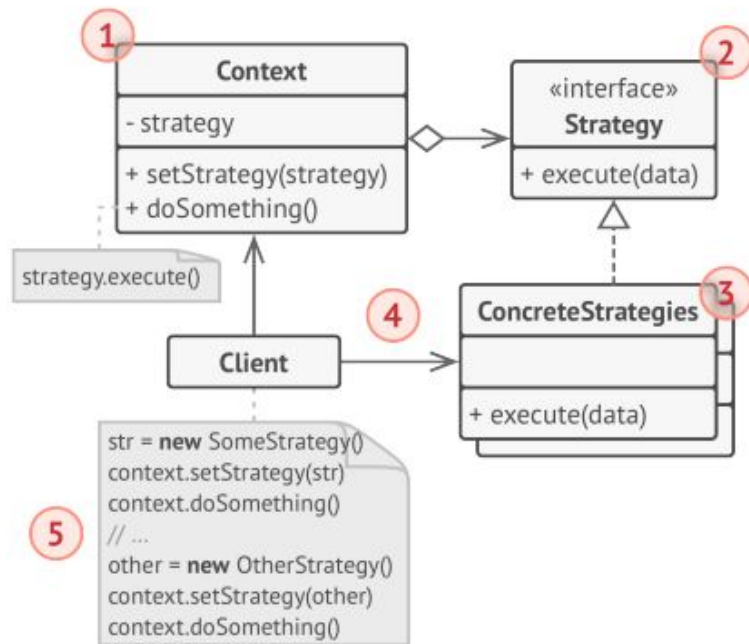
Dos o más **estrategias**, que implementen la misma interface y resuelvan la invocación al método de distinta manera.

- **Retorno**

Realiza la acción de la manera especificada.

Strategy

UML



1. OBJETO QUE DEBE REALIZAR UNA ACCIÓN
2. INTERFACE POLIMÓRFICA PARA LAS ESTRATEGIAS
3. ESTRATEGIAS CONCRETAS
4. INVOCADOR