



III E T R I C A

CONSULTING

Patrones de diseño

Tema 1

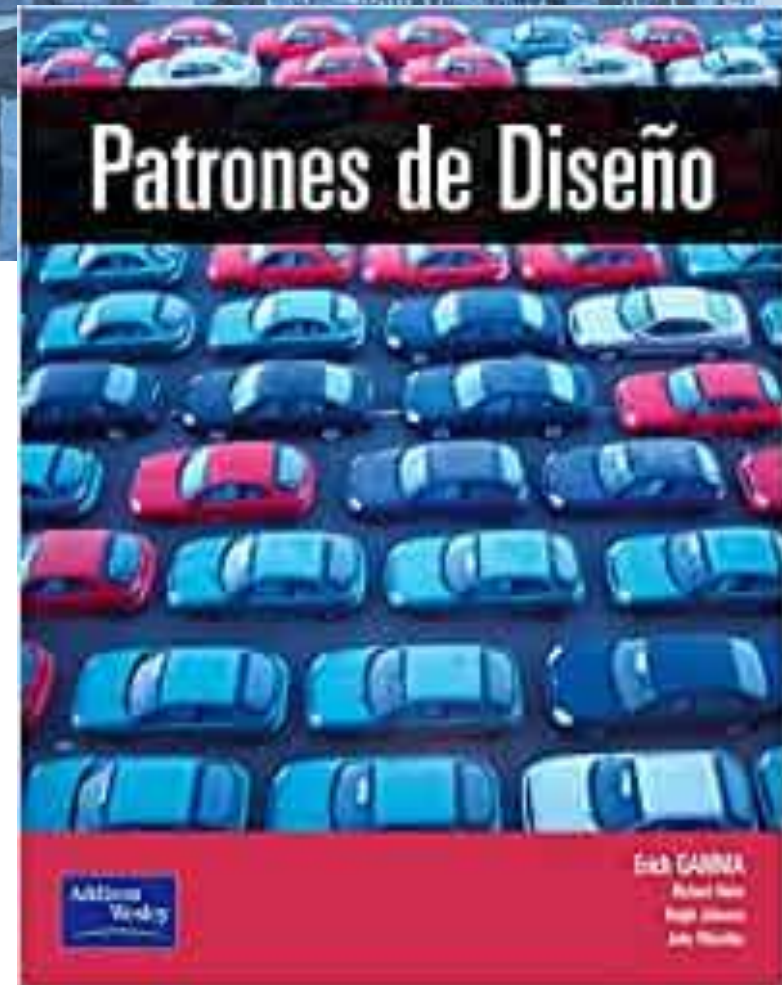
ÍNDICE

1. [PATRONES DE CREACIÓN](#)
2. [PATRONES ESTRUCTURALES](#)
3. [PATRONES DE COMPORTAMIENTO](#)
4. [PATRONES MODERNOS](#)
5. [PATRONES IMPORTANTES](#)
6. [MODELO VISTA CONTROLADOR](#)
7. [INYECCIÓN DE DEPENDENCIAS](#)
8. [DAO](#)
9. [SINGLETON](#)
10. [FACADE](#)
11. [OBSERVER](#)
12. [METHOD FACTORY](#)
13. [COMPOSITE](#)
14. [COMMAND](#)
15. [MÁS](#)

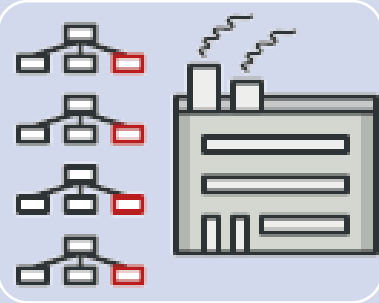




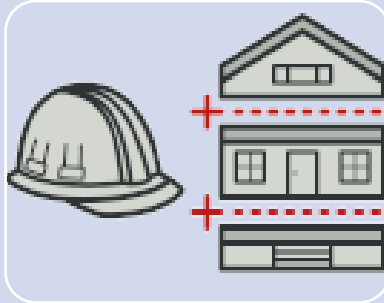
Patrones Erich Gamma



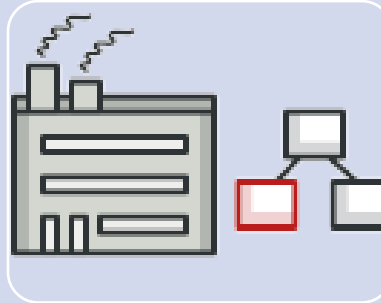
01 Patrones de creación



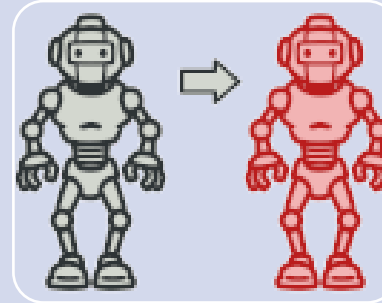
Abstract
Factory



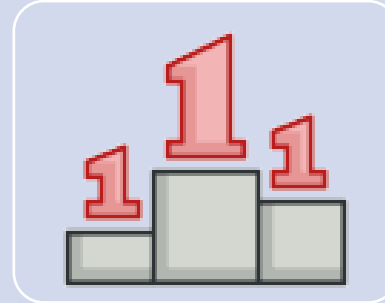
Builder



Factory
Method

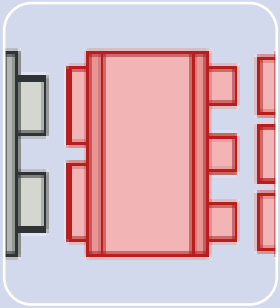


Prototype

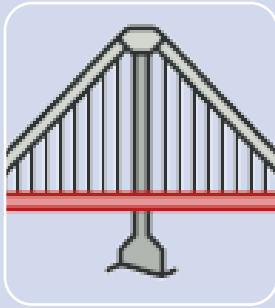


Singleton

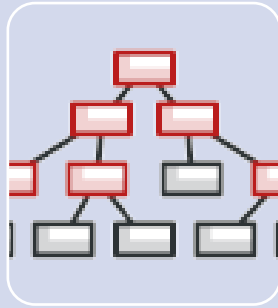
02 Patrones estructurales



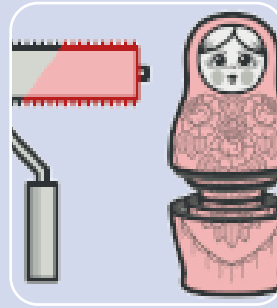
Adapter



Bridge



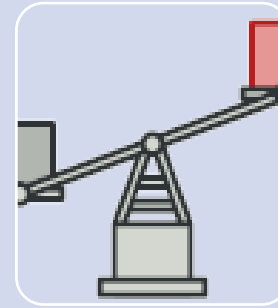
Composite



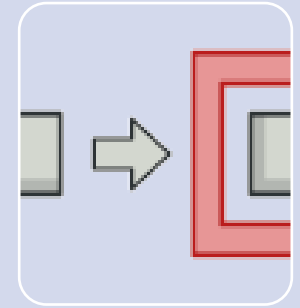
Decorator



Facade

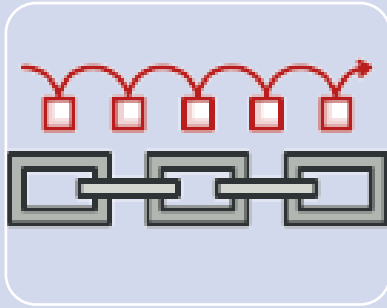


Flyweight



Proxy

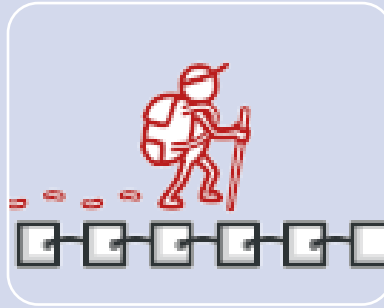
03 Patrones de comportamiento



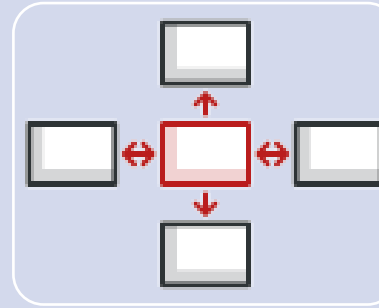
Chain of
responsibility



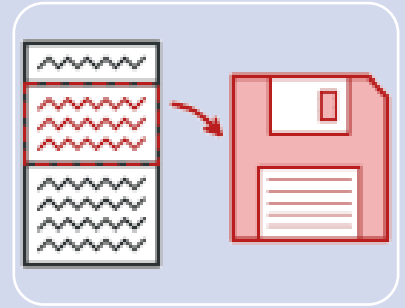
Command



Iterator

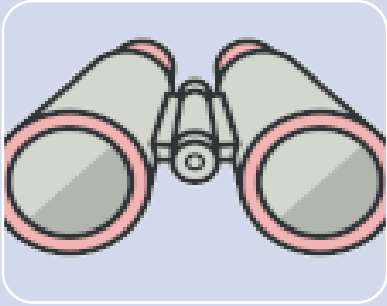


Mediator

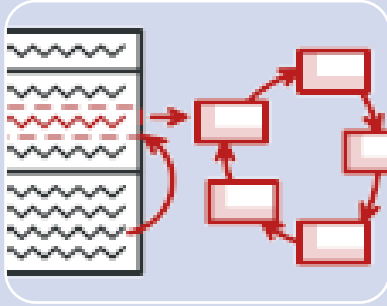


Memento

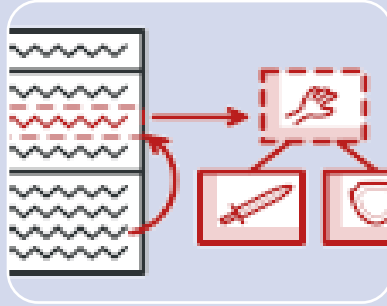
03 Patrones de comportamiento



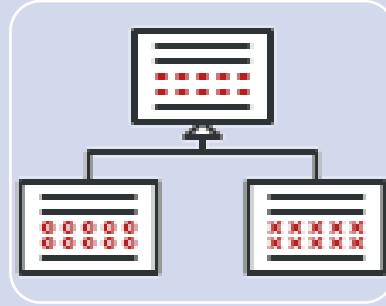
Observer



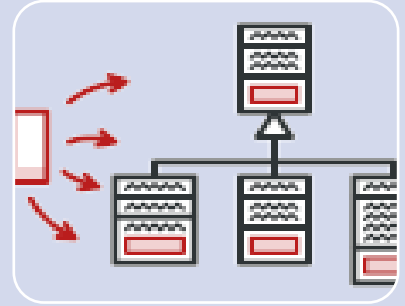
State



Strategy



Template



Visitor

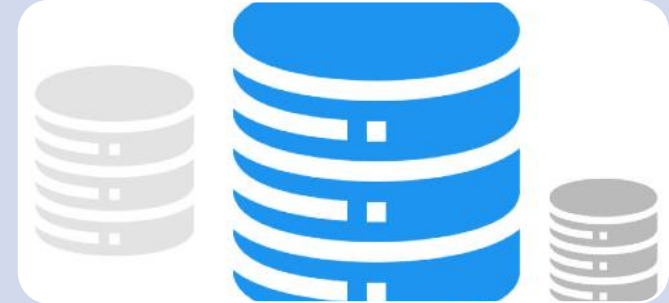
04 Otros patrones modernos (J2EE)



MVC



Inyección
de
dependencias



DAO

Patrones importantes 05

MVC

Inyección de
dependencias

DAO

Singleton

Facade

Observer

Factory

Composite

Command



Modelo vista controlador06

Propósito

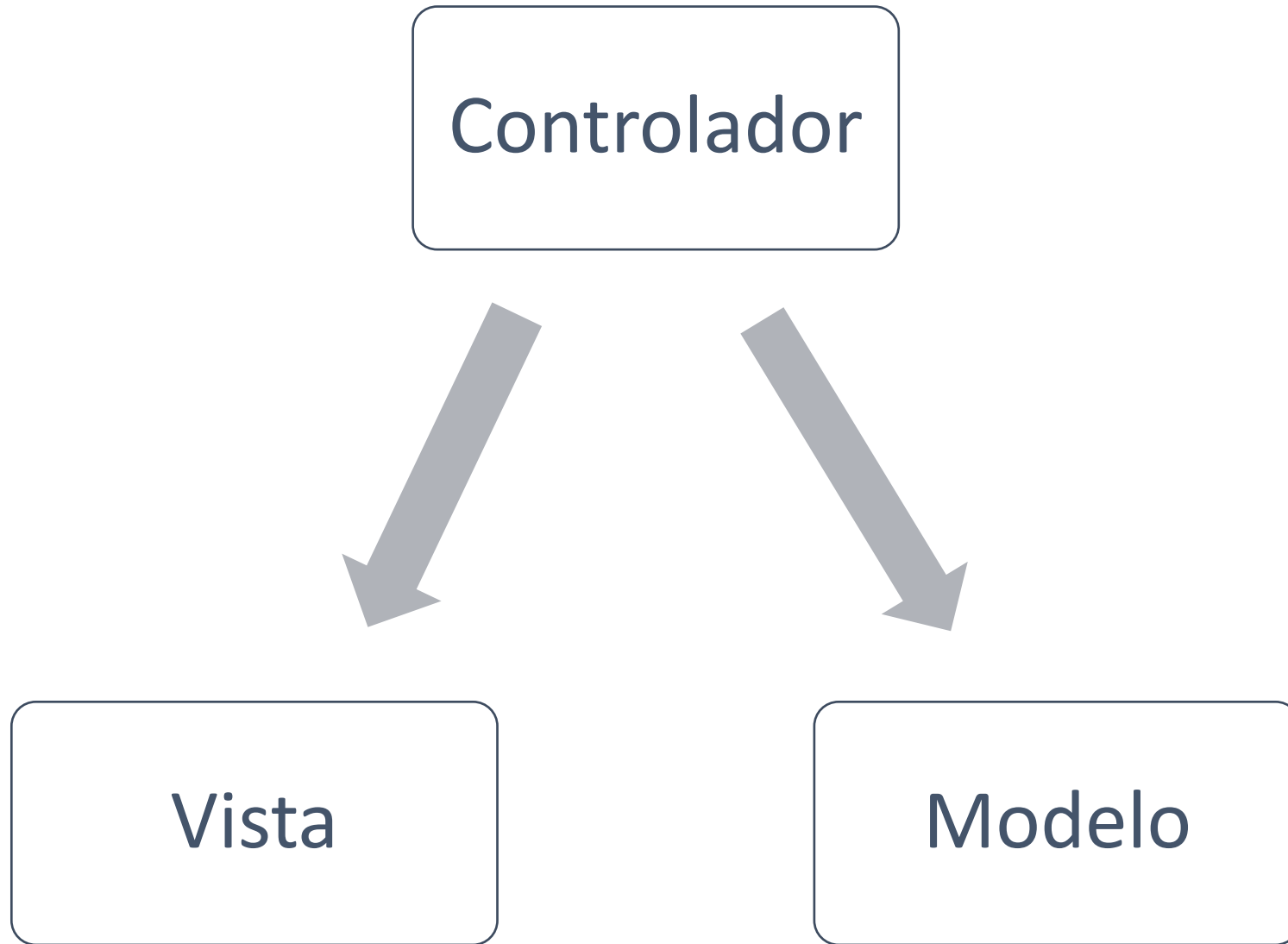
Motivación

Aplicación

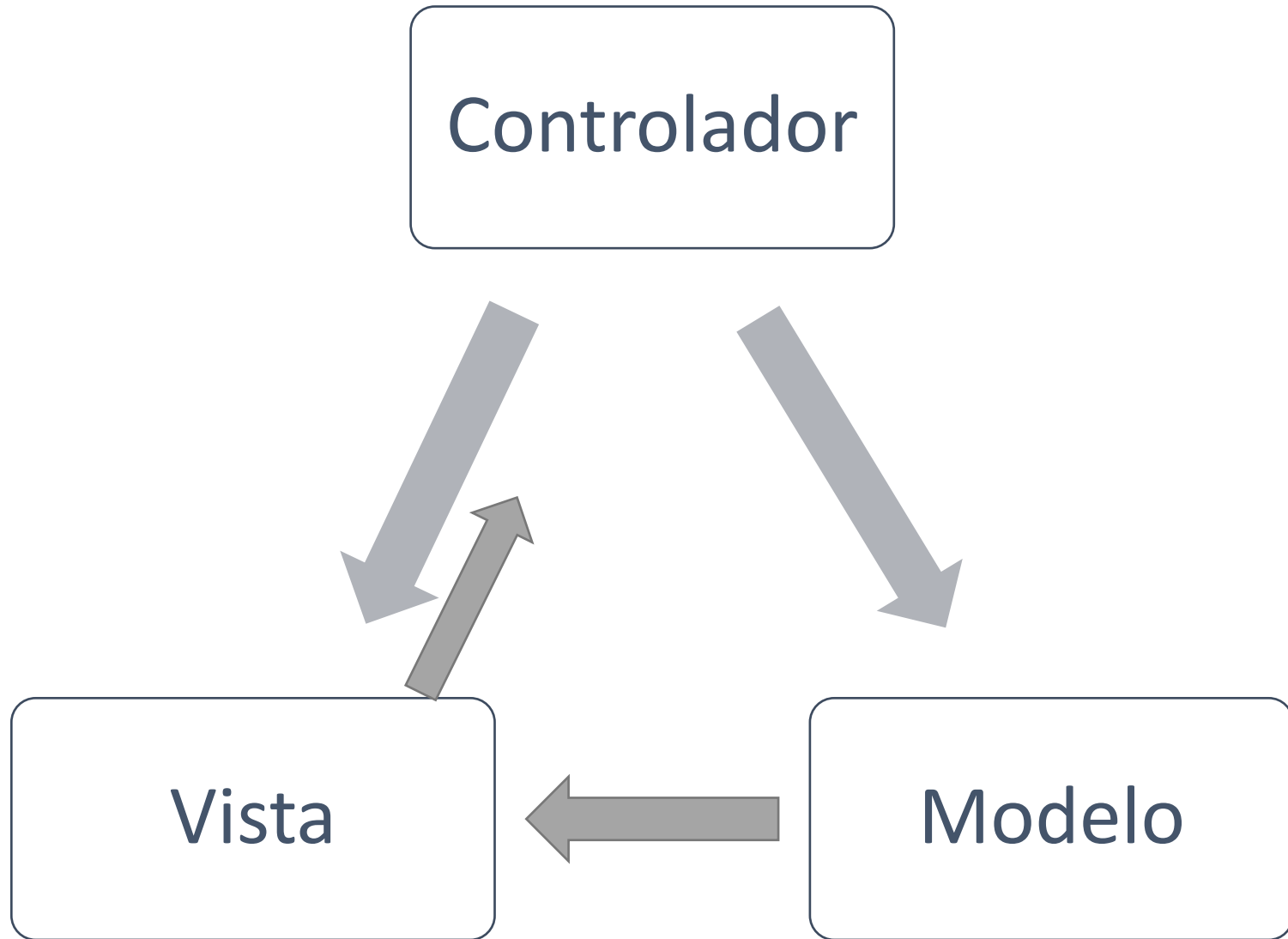
Estructura



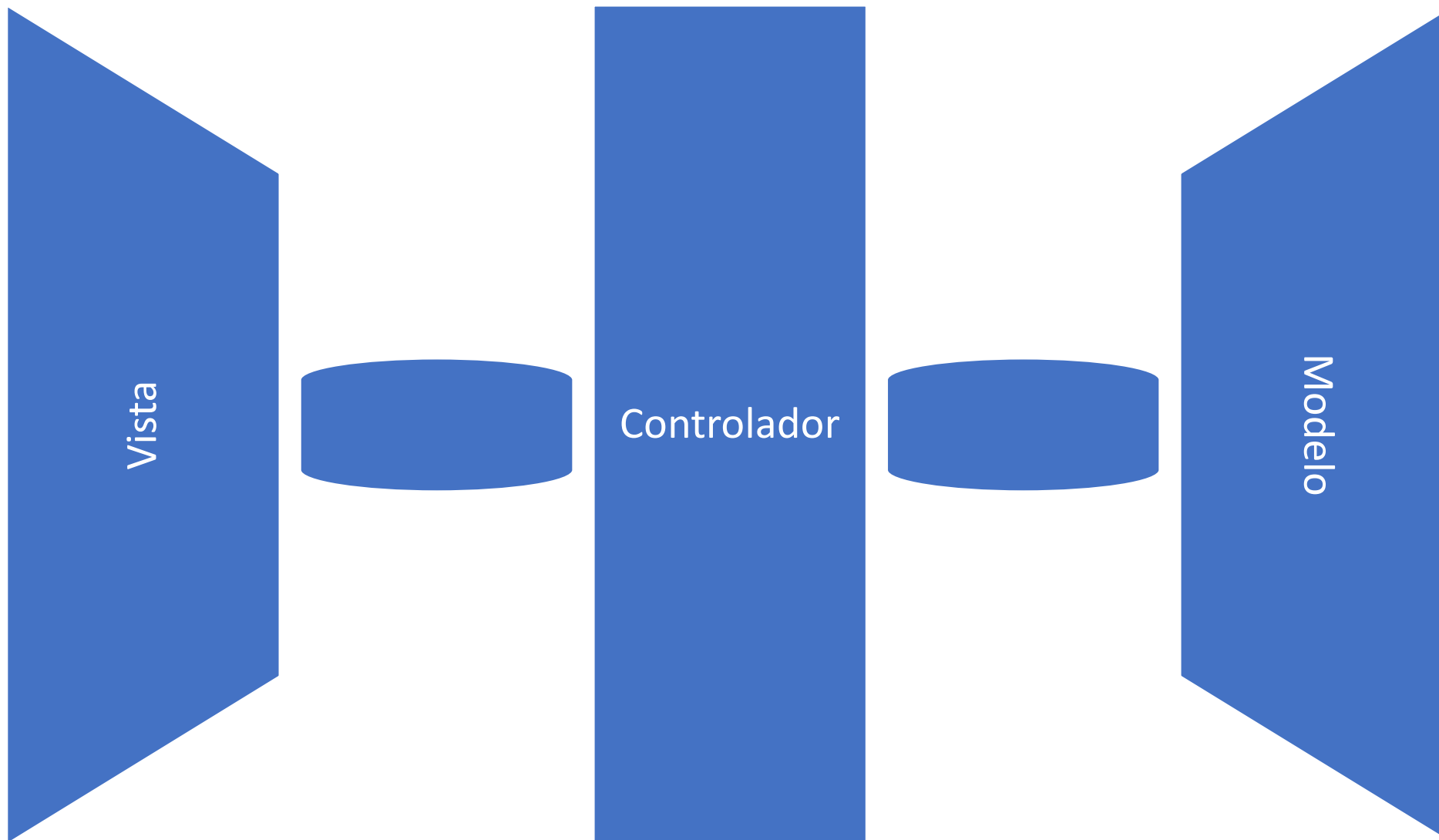
Estructura



Estructura flexible



Estructura por paquetes



Inyección de dependencias 07



Propósito

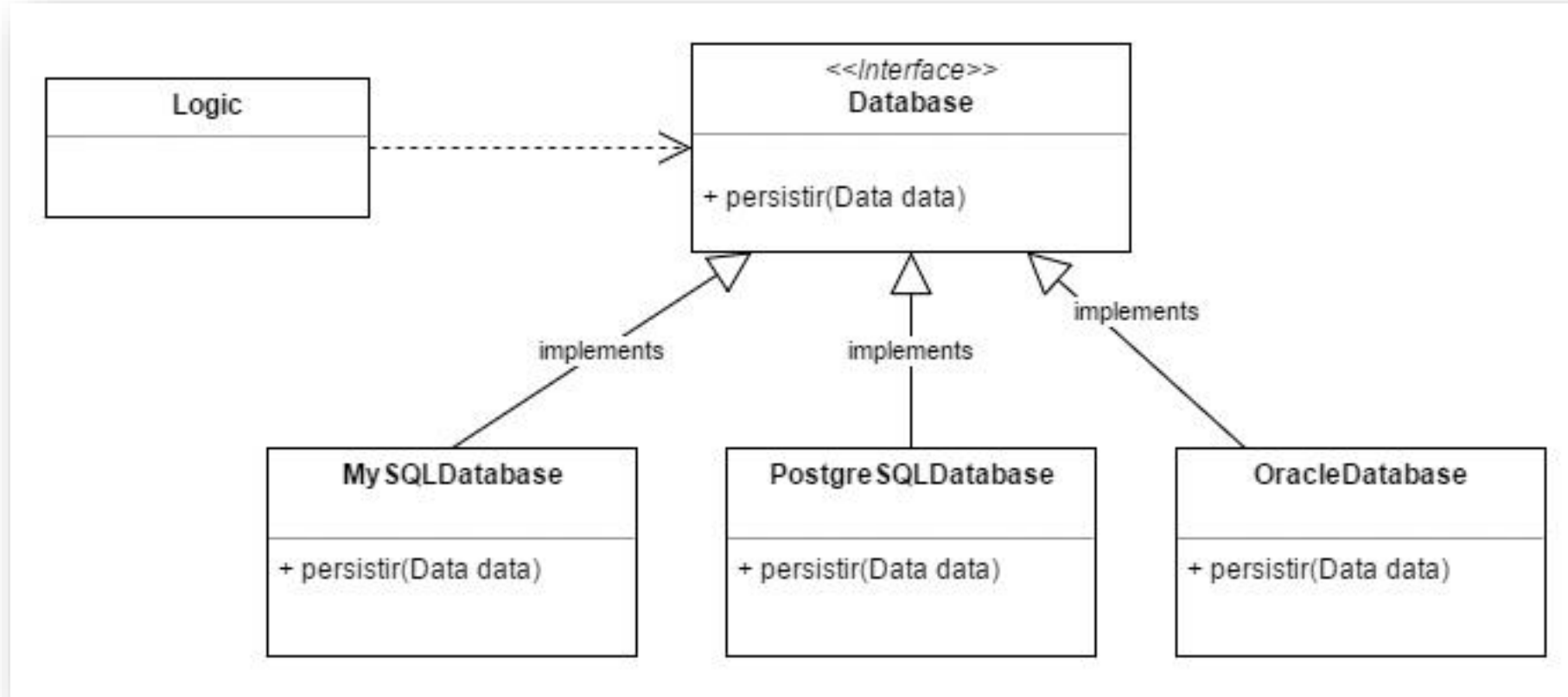
Motivación

Aplicación

Estructura



Estructura





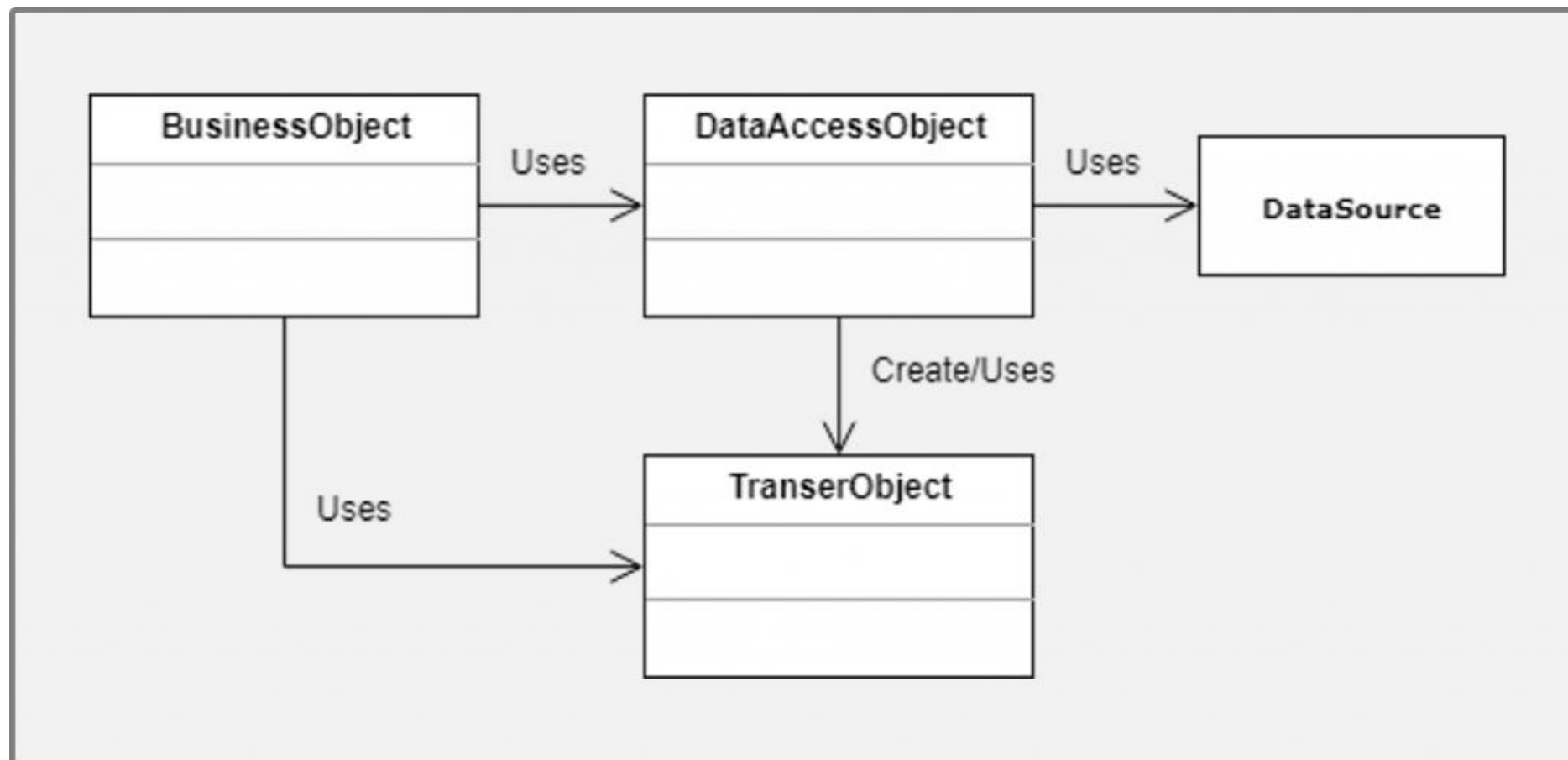
Propósito

Motivación

Aplicación

Estructura







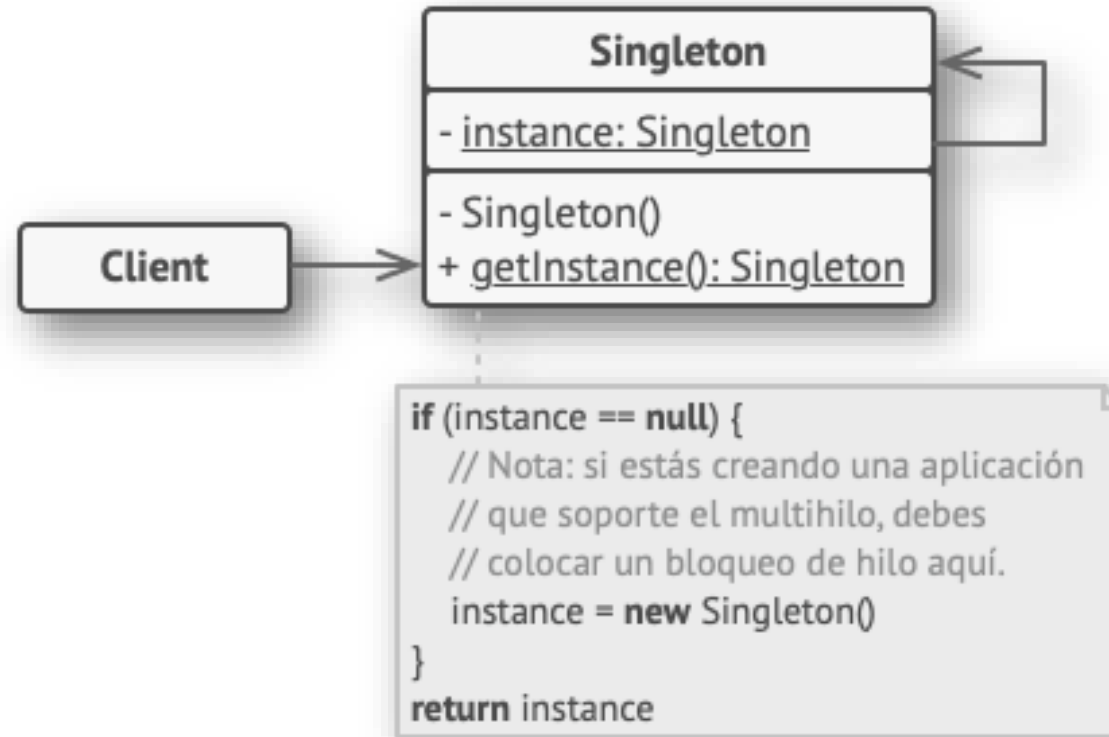
Propósito

Motivación

Aplicación

Estructura







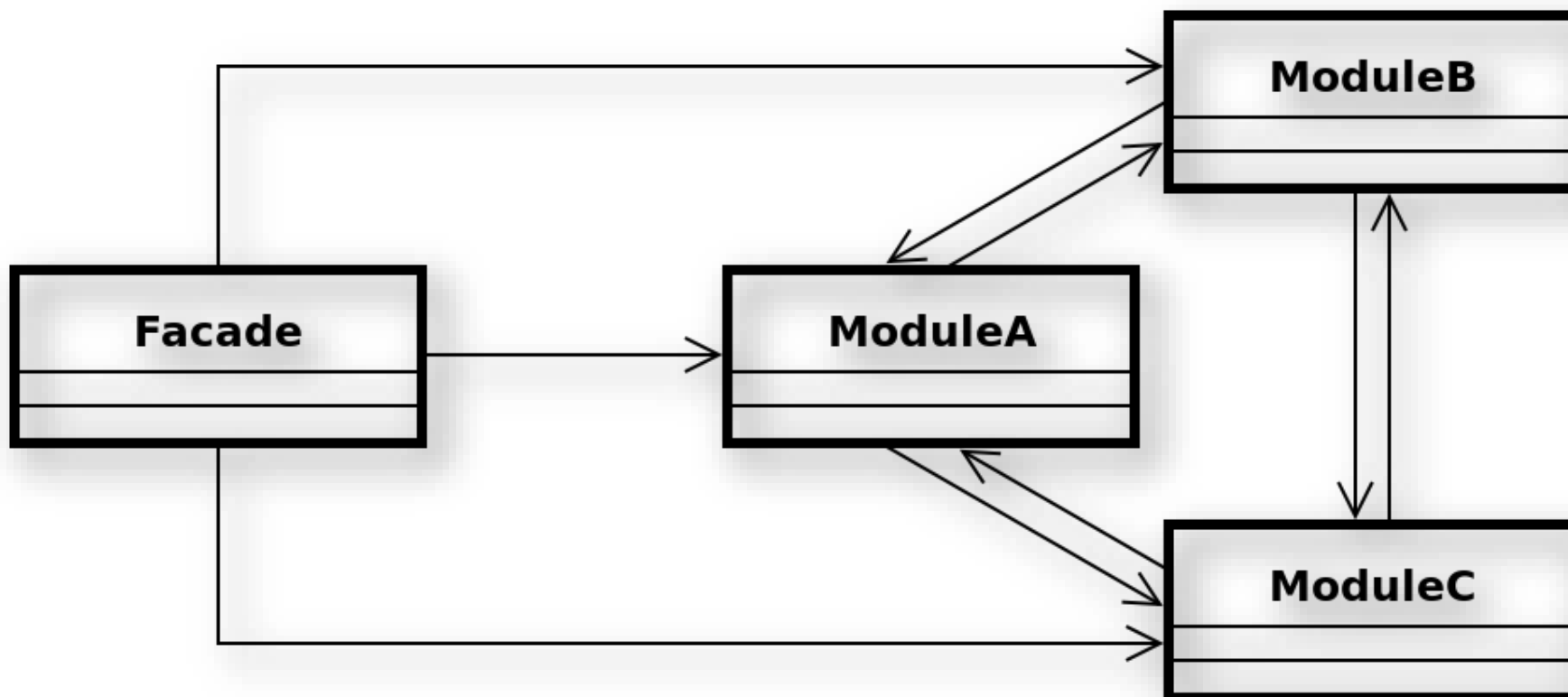
Propósito

Motivación

Aplicación

Estructura







Propósito

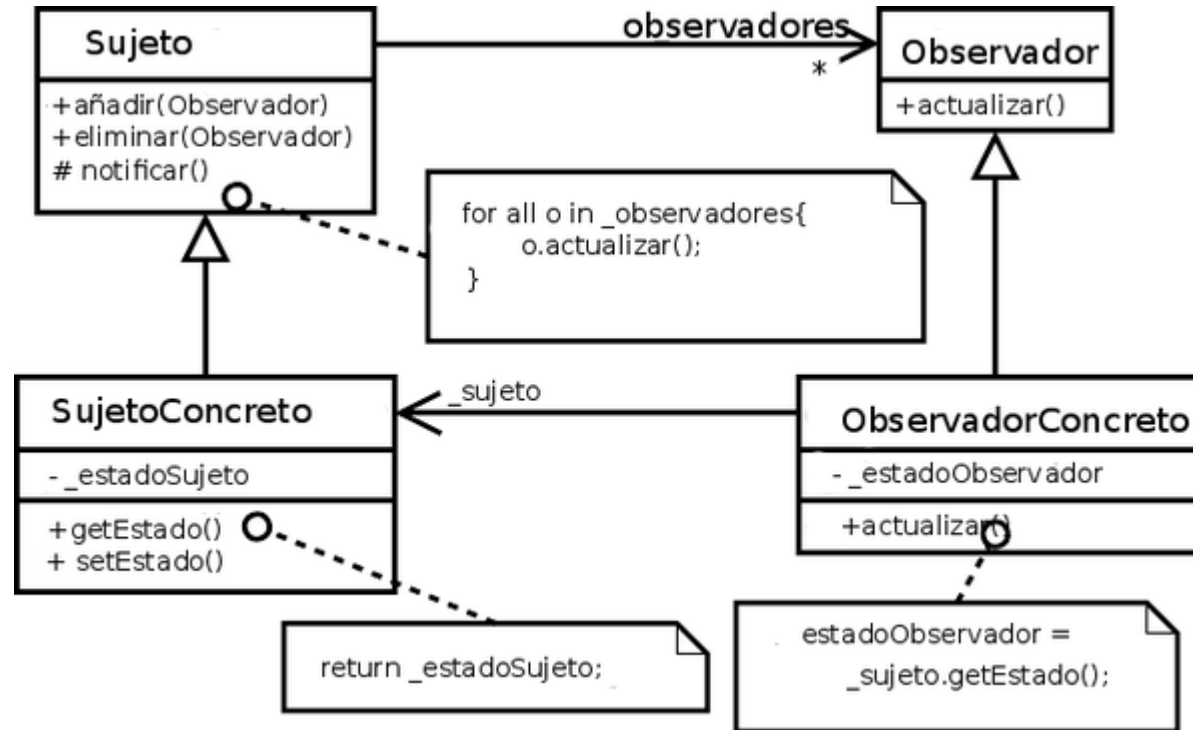
Motivación

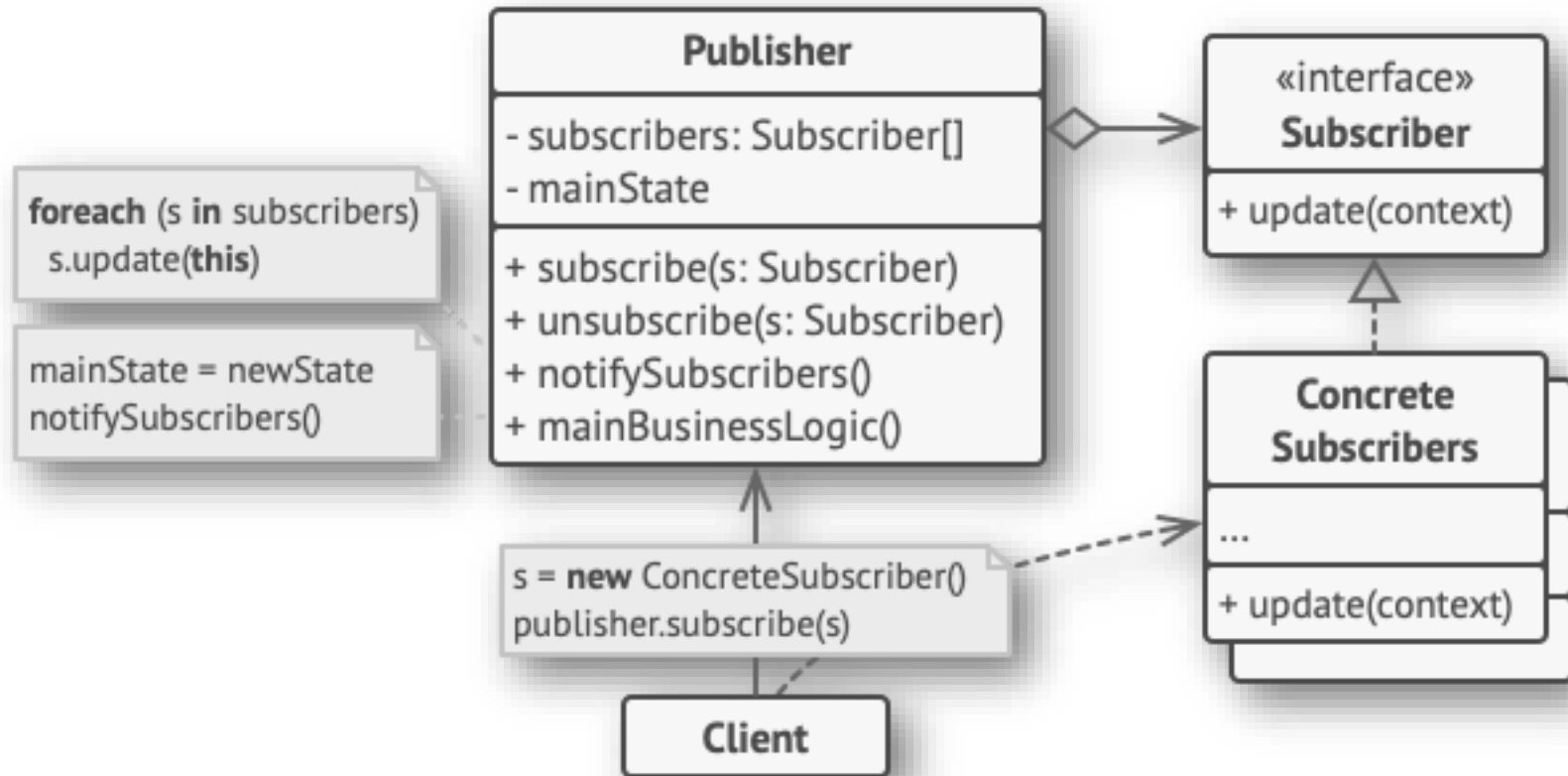
Aplicación

Estructura



Estructura





FACTORY METHOD 12



Propósito

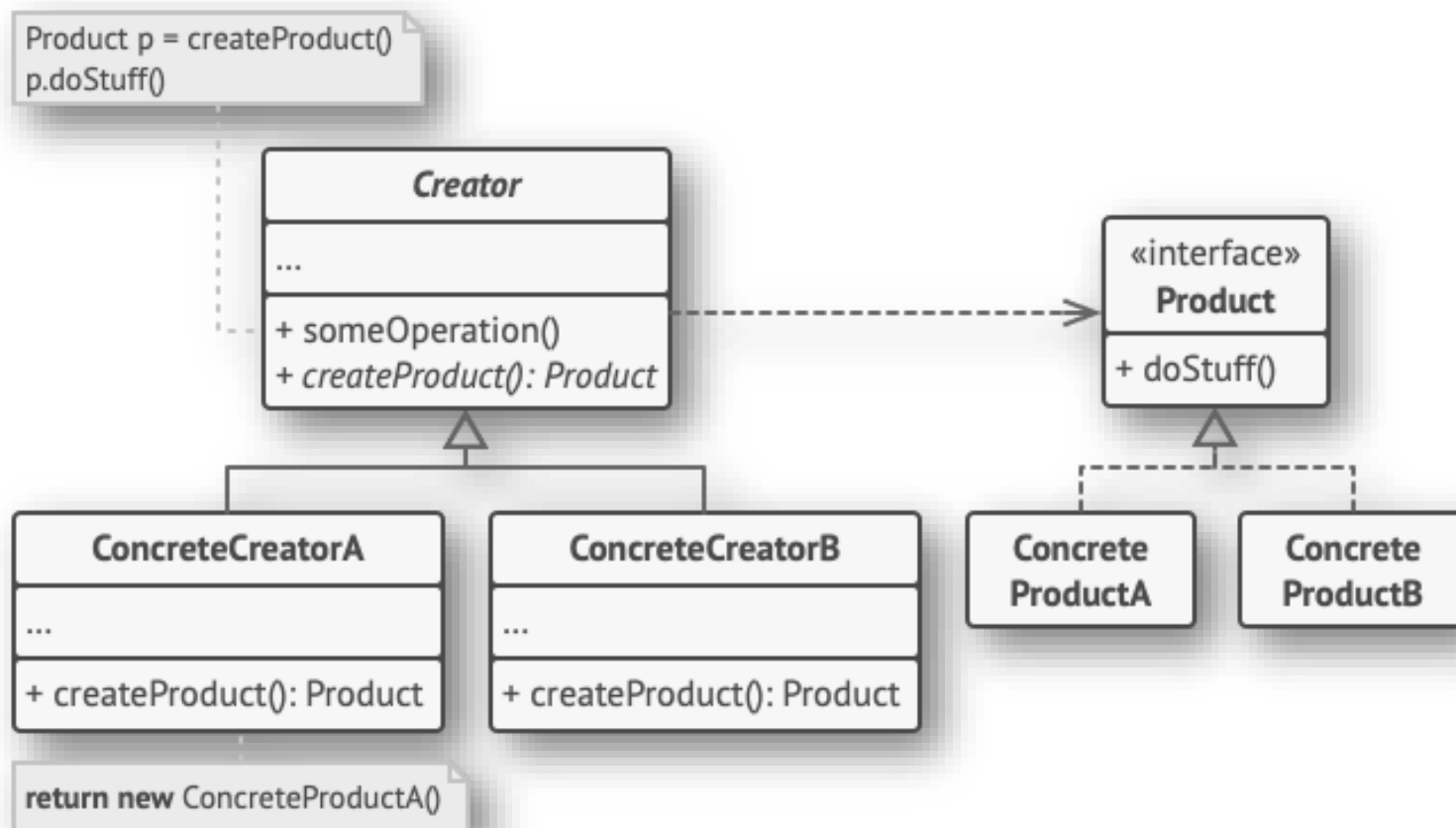
Motivación

Aplicación

Estructura



Estructura



COMPOSITE 13



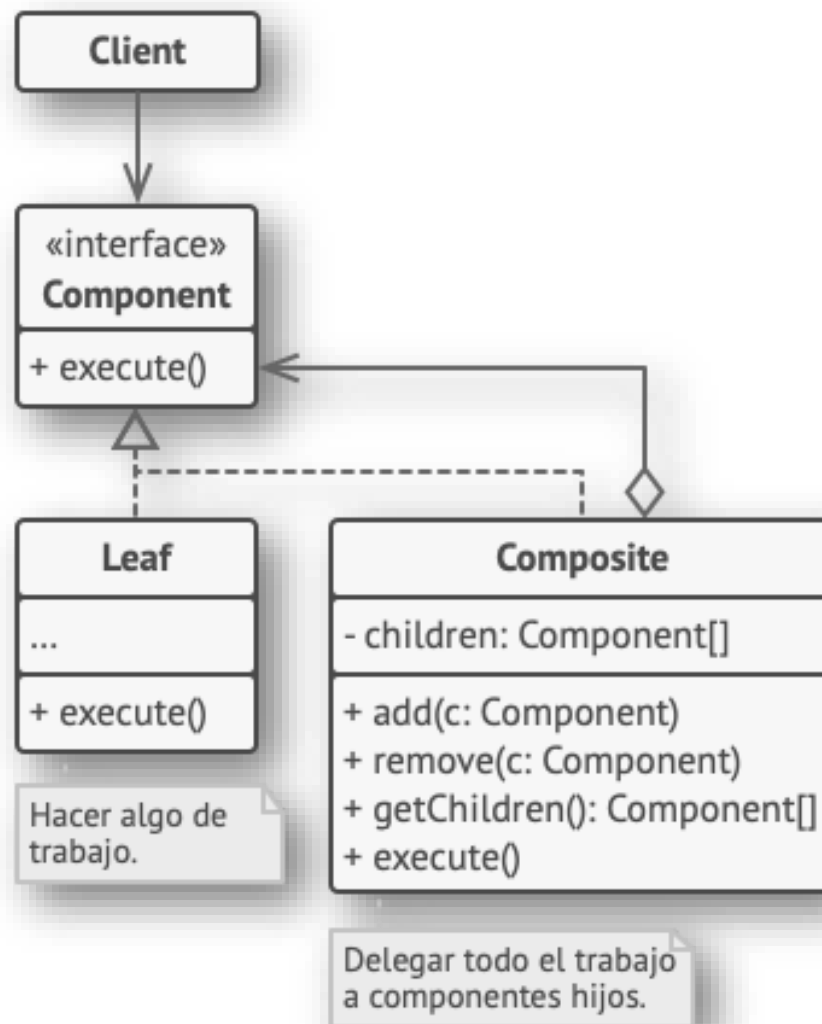
Propósito

Motivación

Aplicación

Estructura





COMMAND 14



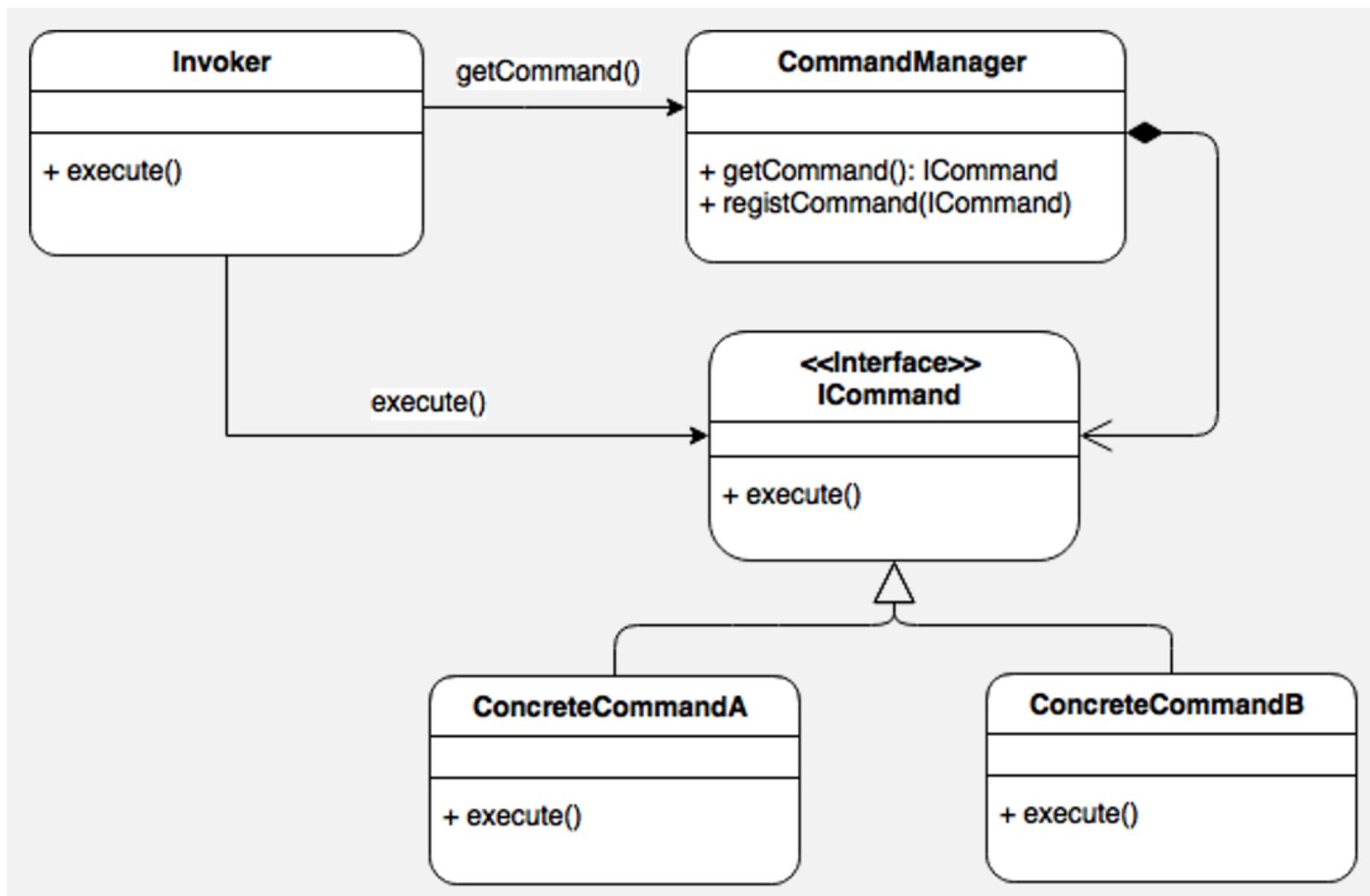
Propósito

Motivación

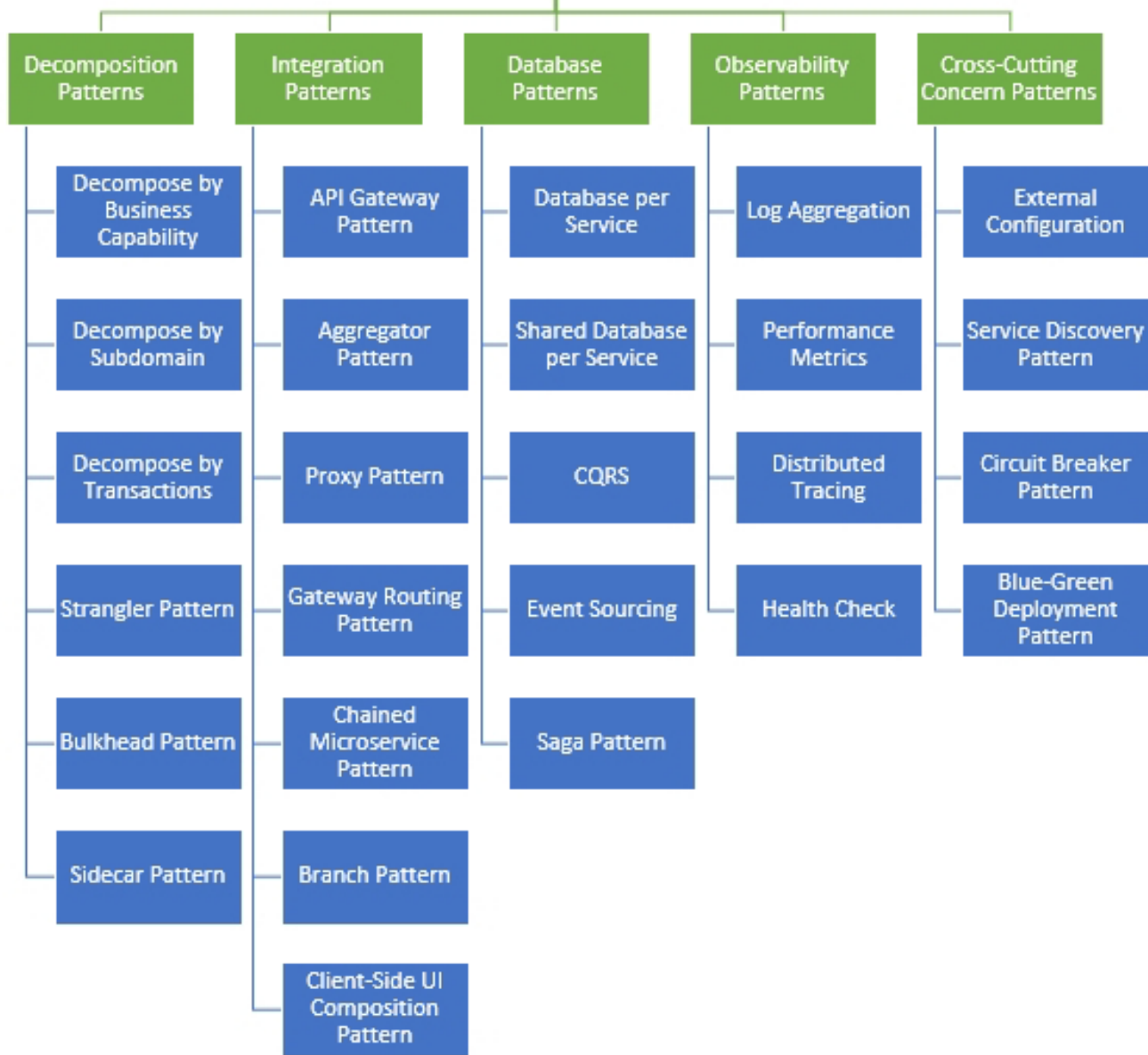
Aplicación

Estructura





Design Patterns for Microservices



15

MÁS

Desde fuera de un singleton, ¿se puede sobrescribir la instancia?

¿Es posible implementar la interfaz `java.util.Observer` cambiando el tipo de argumento a `String`?

El patrón DAO se puede usar en combinación con otros patrones. Dar ejemplos de ello.

Busca alguna clase que hayas usado que se comporte como Factory Method

Dar un ejemplo de uso de una estructura "Composite"

¿Se puede crear un clon de un Singleton?

Revisar estas preguntas de entrevistas a modo de resumen [aquí](#)