

Patrones de diseño

Sesión 08:

PATRONES ESTRUCTURALES: INTRODUCCIÓN A PATRONES ESTRUCTURALES PATRÓN ADAPTER Y PATRÓN FACADE.



**Universidad
Tecnológica
del Perú**

Logro de la sesión

Al finalizar la sesión, el estudiante elabora su programación usando patrones estructurales, patrón adapter y patrón facade empleando los diversos patrones de diseño aplicados a casos matemáticos, financieros y físicos.

Patrones de Diseño

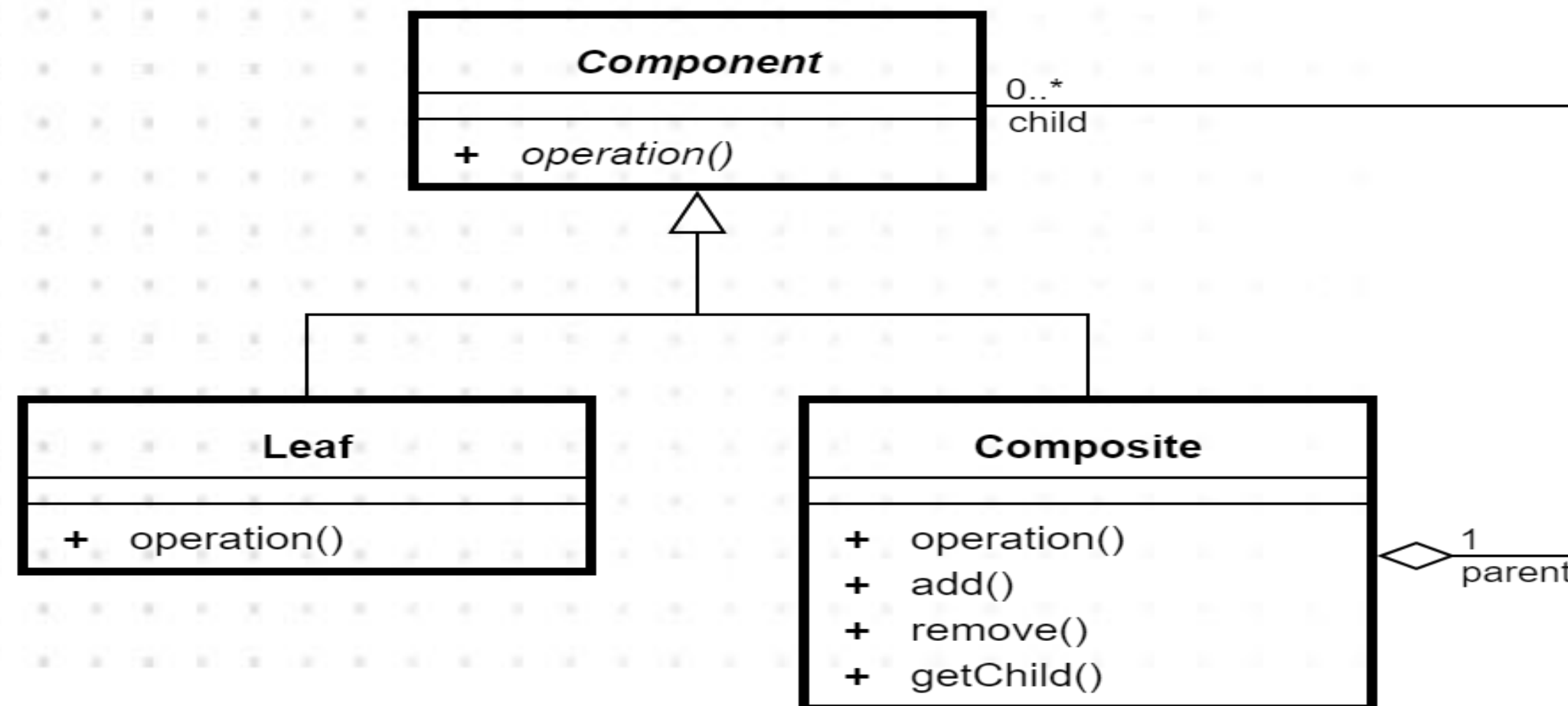


| | | PROPÓSITO | | |
|---------|--------|---|--|--|
| | | CREACIONAL | ESTRUCTURAL | COMPORTAMIENTO |
| ALCANCE | CLASE | Factory Method | Adapter | Interpreter Template Method |
| | OBJETO | Abstract Factory Builder Prototype Singleton | Adapter Bridge Composite Decorator Facade Proxy | Chain of Responsibility Command Iterator Mediator Memento Flyweight Observer State Strategy Visitor |

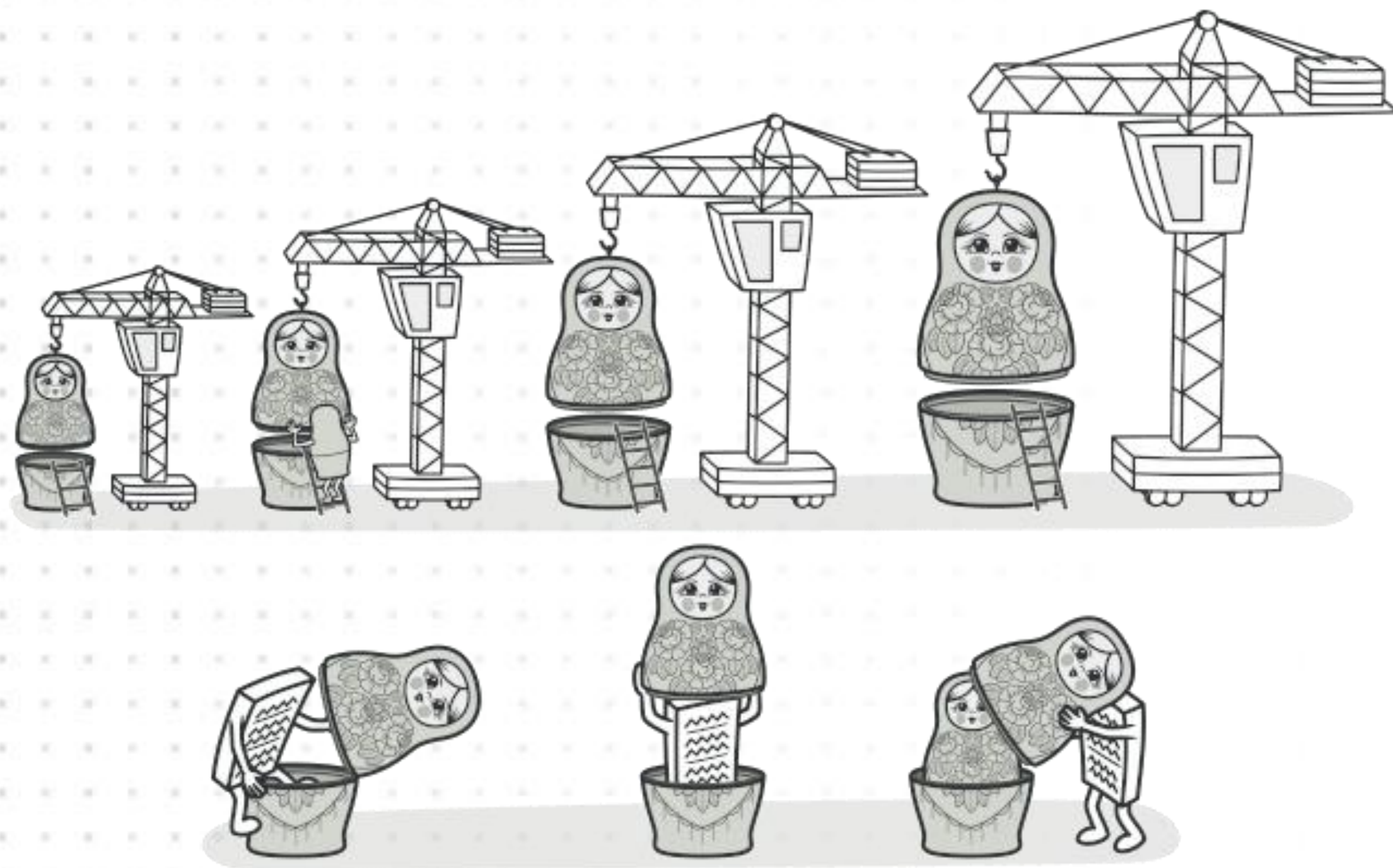
PATRON ESTRUCTURAL

Los patrones estructurales se centran en la composición de clases y objetos para crear estructuras más flexibles y eficientes.

Estos patrones son la clave para organizar y gestionar grandes sistemas de software, proporcionando soluciones elegantes a desafíos comunes en el diseño de software.

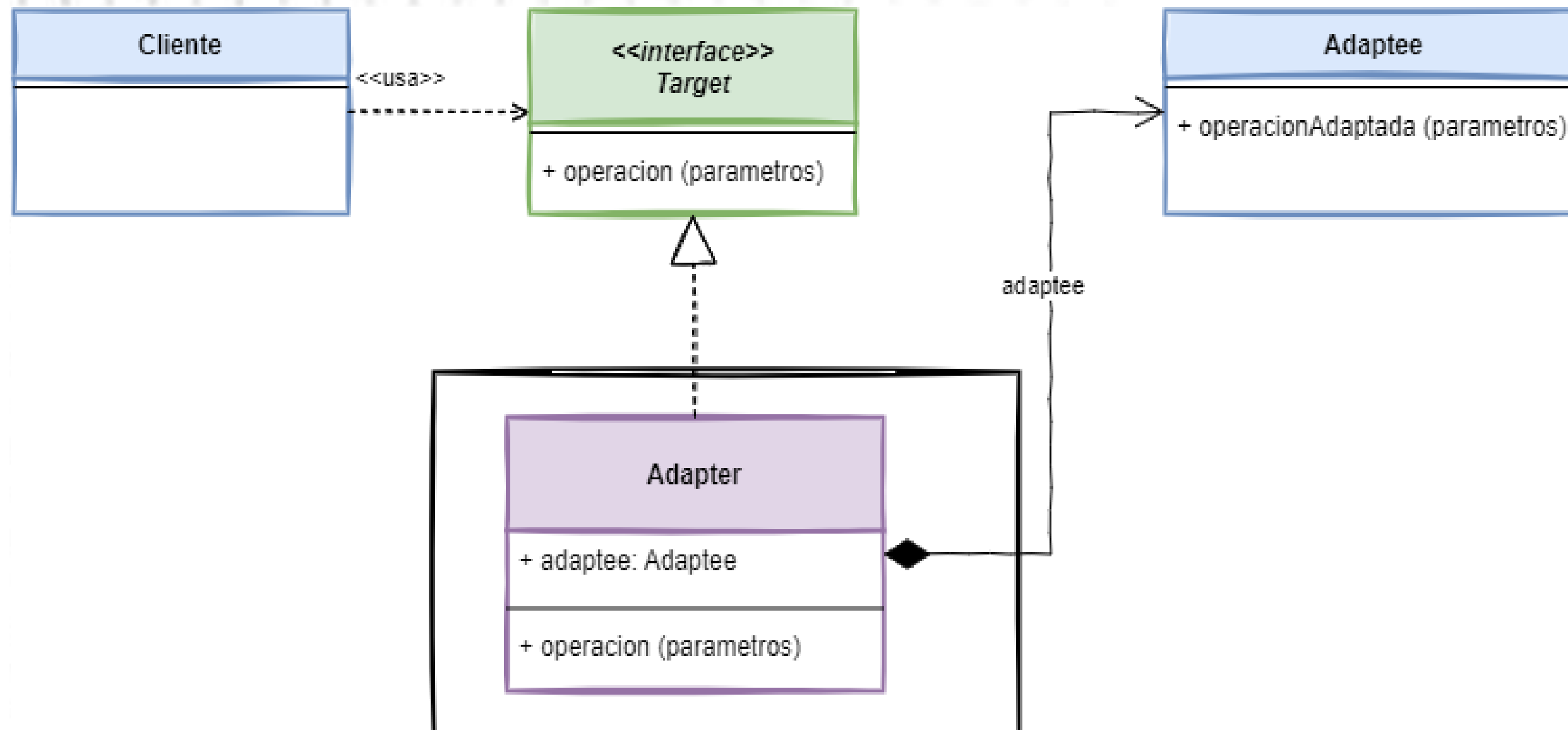


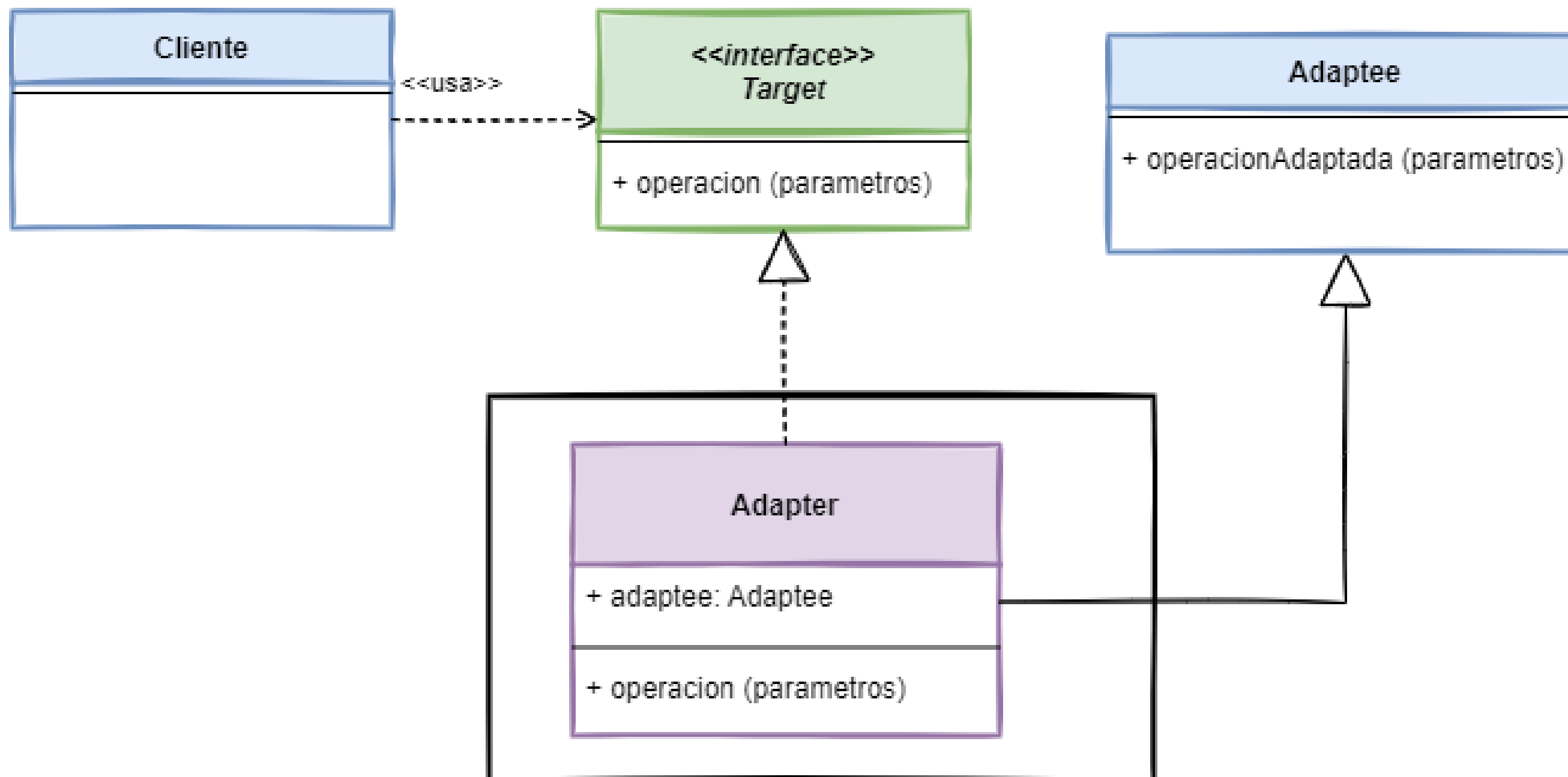
PATRON ESTRUCTURAL



PATRÓN ADAPTER

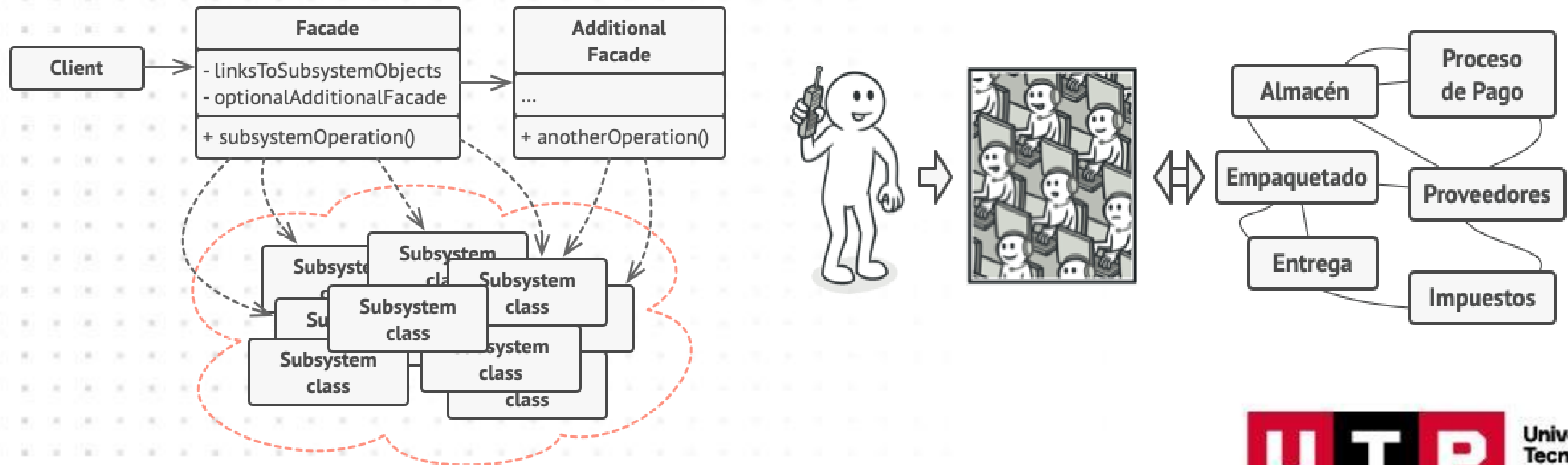
Adapter es un patrón de diseño estructural. Este es el único patrón que tiene los dos ámbitos, clases y objetos, ya que podemos implementarlos de dos formas diferentes. Este patrón se utiliza cuando tenemos dos elementos de una aplicación los cuales necesitamos que trabajen juntos, pero sus interfaces de comunicación no son compatibles. Para esos casos creamos una clase intermedia que facilita la comunicación entre ambas. Así, Adapter servirá como puente entre ambos elementos.



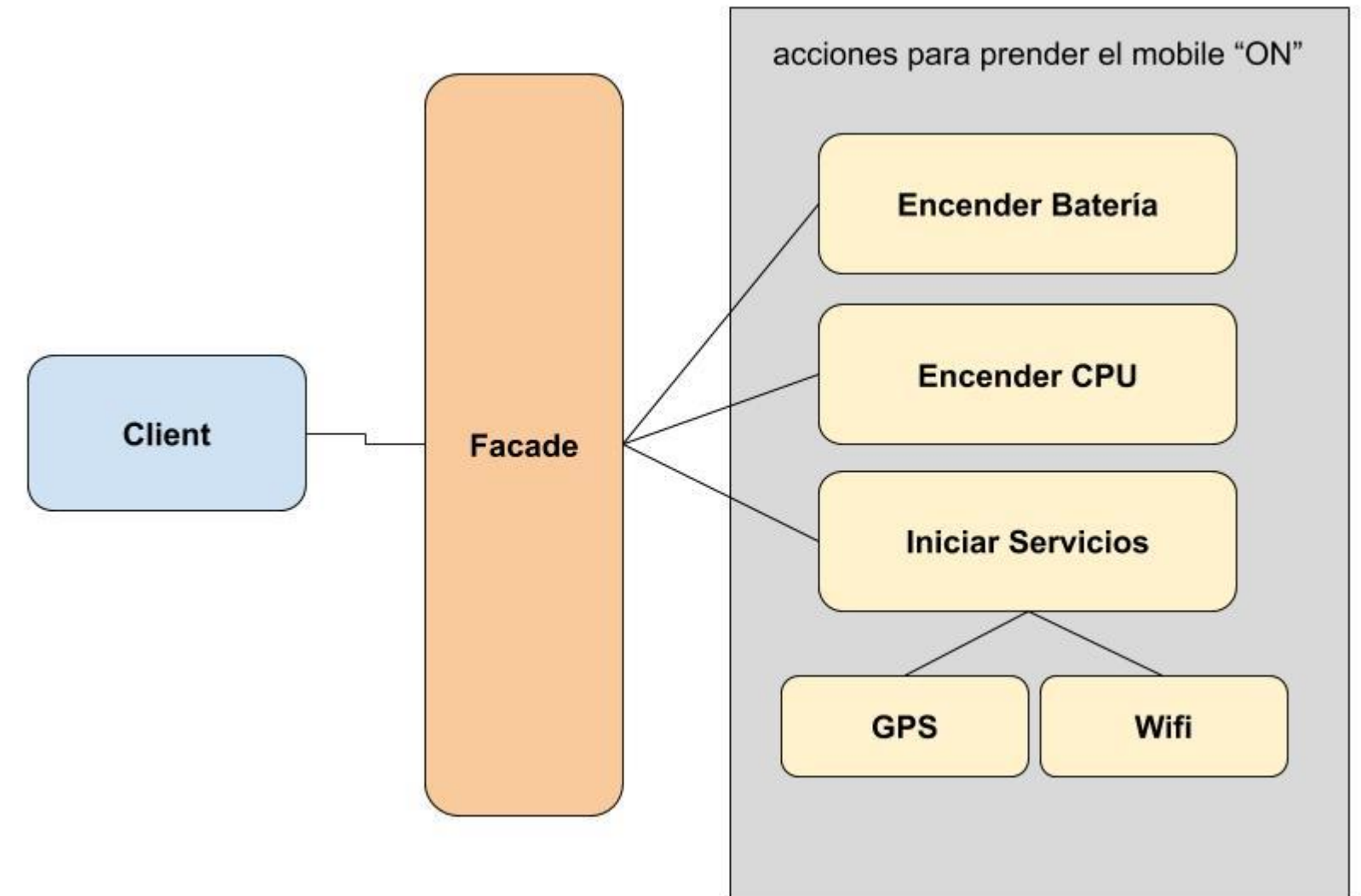
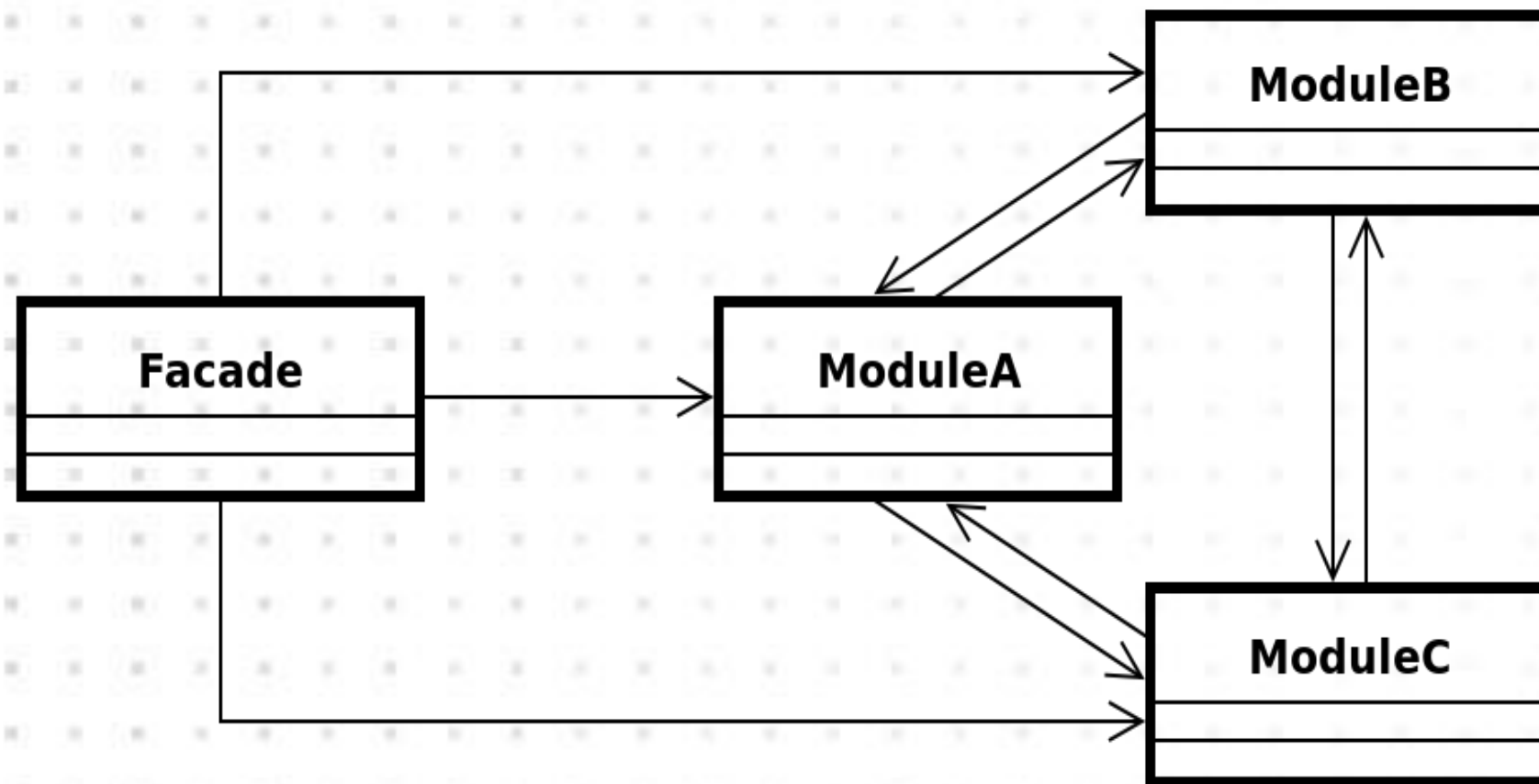


PATRÓN FACADE

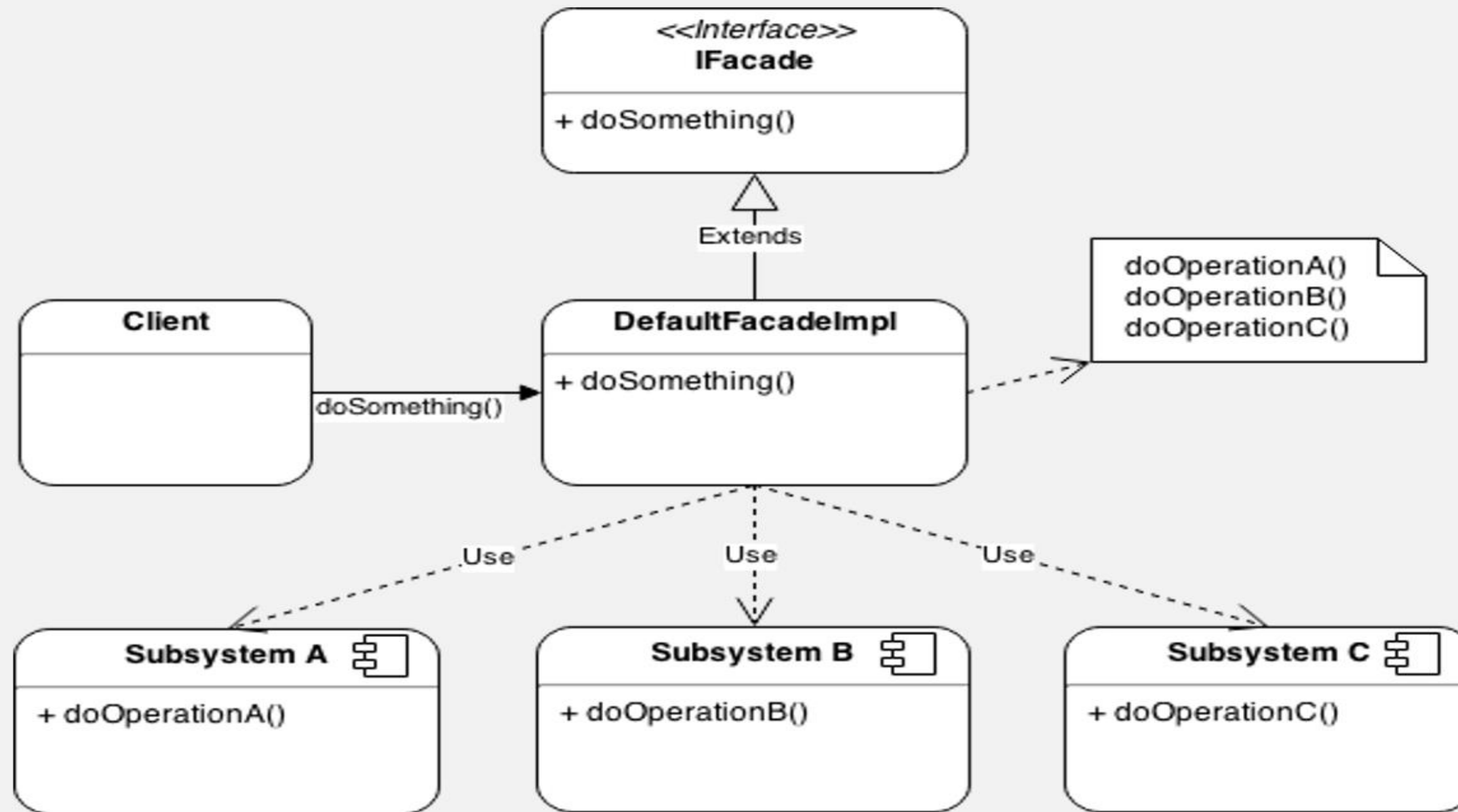
El patrón de diseño Facade simplifica la complejidad de un sistema mediante una interfaz más sencilla. Mejora el acceso a nuestro sistema logrando que otros sistemas o subsistemas usen un punto de acceso en común que reduce la complejidad, minimizando las interacciones y dependencias.



Se trata de un patrón de diseño bastante útil en vista de que también desacopla los sistemas .



Facade pattern – Class diagram



¿Consultas o dudas?



Actividad



Resolver la actividad planteada en la plataforma.

Cierre

¿Qué hemos aprendido hoy?



Elaboramos nuestras conclusiones sobre el tema tratado



**Universidad
Tecnológica
del Perú**