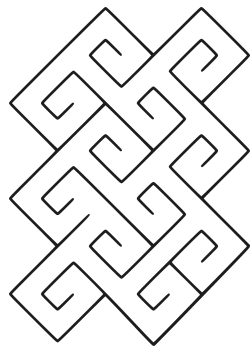


# Patrones de diseño



# Agenda

- Patrones estructurales
  - Facade
  - Proxy
  - Composite

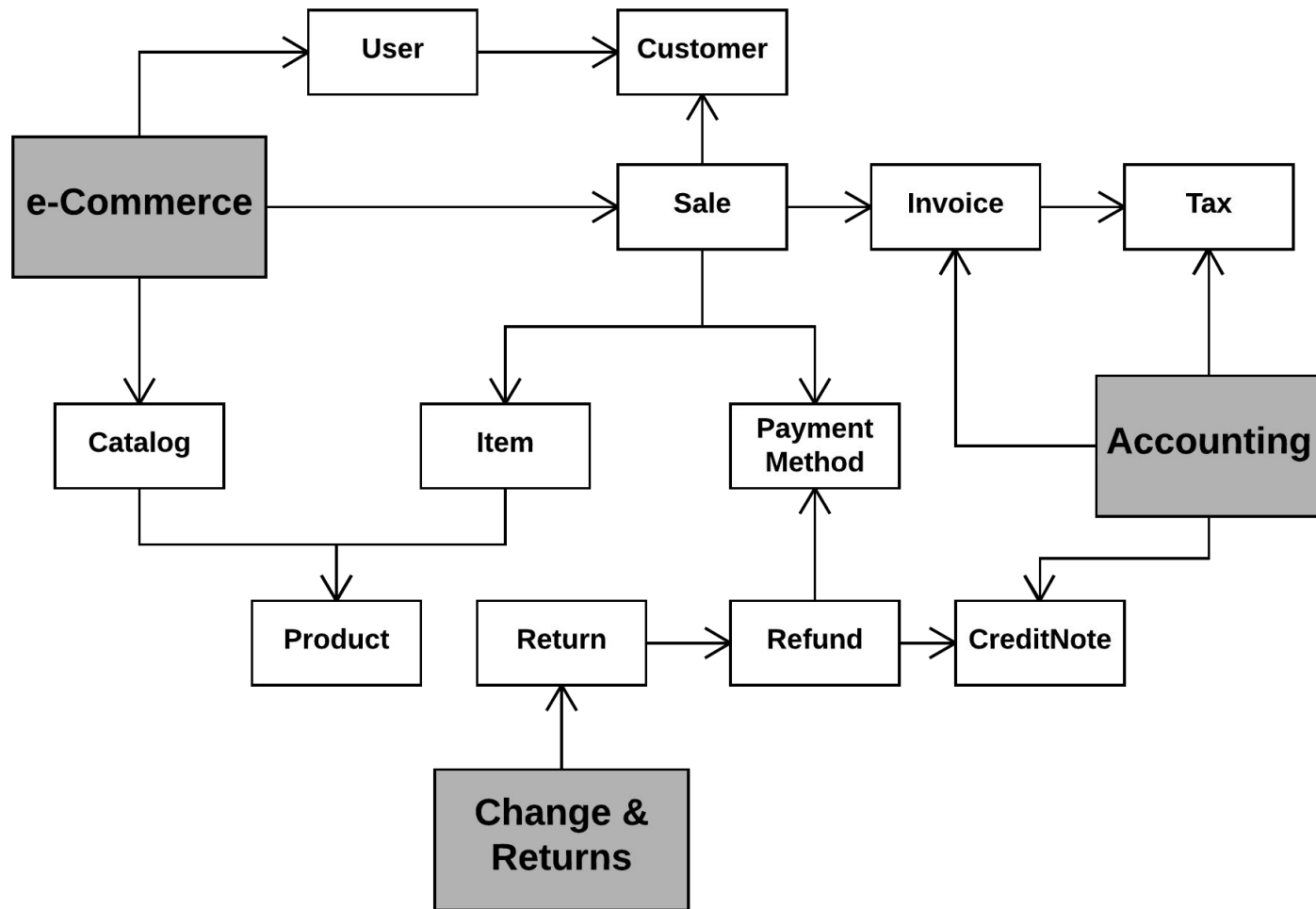


# Facade

Se tiene una interfaz altamente compleja, de la que se usa una porción muy reducida en la gran mayoría de los casos.

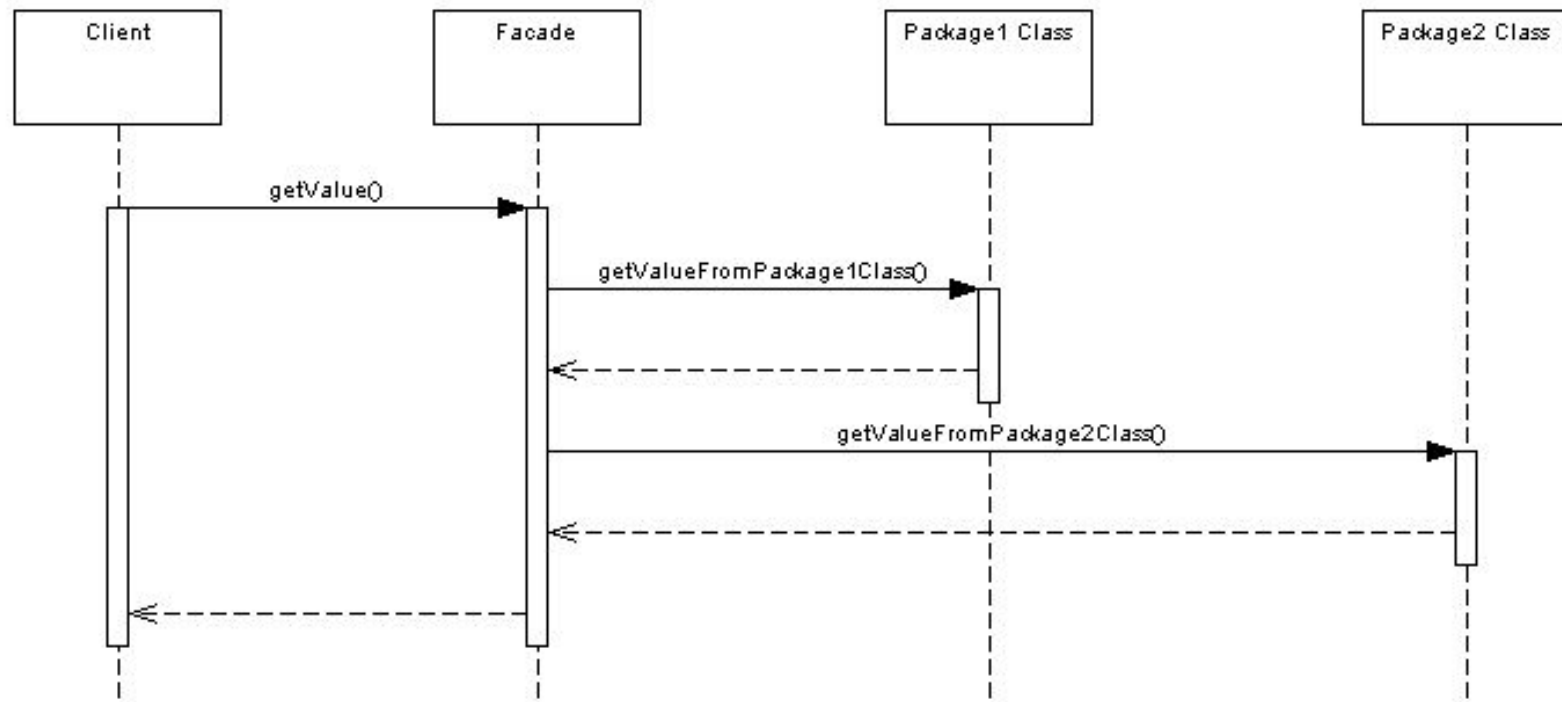
- Unifica el acceso a un conjunto de servicios o subsistemas
- Provee acceso de alto nivel para permitir un uso más simple
- Separa uso estándar y personalizado
- Reduce el acoplamiento con los subsistemas detrás de la fachada

# Facade



# Facade

¿Dudas?

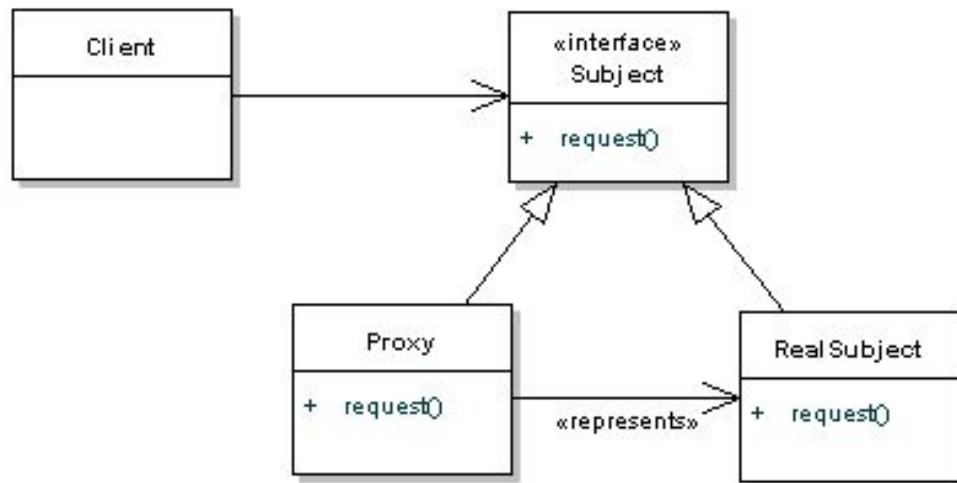


# Proxy

Cuando se necesita postergar operaciones costosas, controlar el acceso a determinados tipos de recursos, acortar el tiempo de acceso a un objeto que lleva mucho tiempo conseguir, un proxy nos es de mucha ayuda.

- Permite postergar operaciones costosas y evitables
- Casos de uso
  - Remoto - para objetos separados de lugar - embajador
  - Virtual - para objetos costosos
  - Protección - para objetos que necesitan manejar permisos o niveles de acceso
  - Referencia Inteligente - para usarse como puntero inteligente - manejo de lockeos
- Mocks

# Proxy



# Composite

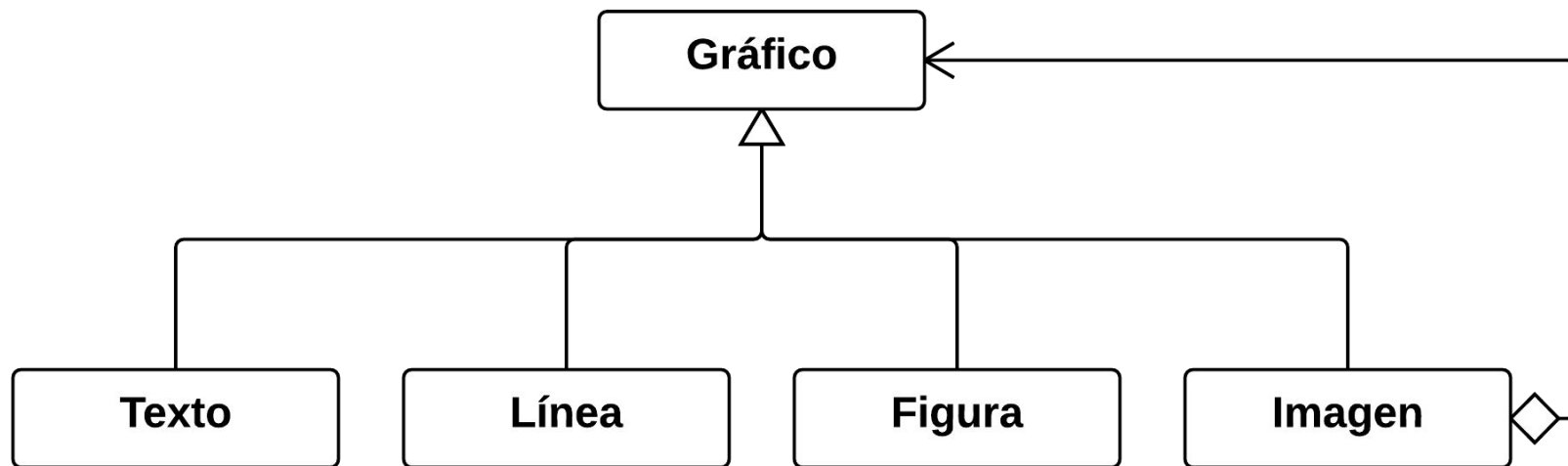
Tratar de forma unificada objetos simples y objetos compuestos, organizando estructuras jerarquizadas con interfaz uniforme.

- Simplifica el código cliente
- Unifica el modo de uso de objetos compuestos de individuales



# Composite

```
for (Grafico grafico : misGraficos) {  
    grafico.dibujar()  
}
```



¿ ?



# Bibliografía

- Design Patterns CD - Gamma, Helm, Johnson, Vlissides

