Metodología de la programación II

Practica IX

Emiliano Salvatori

Octubre 2019

1. Practica IX

Framework

Responda las siguientes preguntas e indique las fuentes de información que utilizó.

¿Cómo podría definir lo que es un Patrón?

Un patrón de diseño, en el ámbito del desarrollo del software, resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. [1]

¿Cuál es su utilidad?

Los Patrones de Diseño, facilitan la reutilización de diseños y arquitecturas de software que han tenido éxito. Son soluciones que han sido testeadas en ocasiones anteriores y pueden ser reutilizables para poder dar solución a la misma calidad de problemas.

¿Qué características tiene?

Un patrón:

- Tiene un nombre.
- Es una solución a un problema en un contexto particular recurrente.

Describa brevemente cuáles serían sus motivaciones para utilziar un patrón

Alguna de las motivaciones para utilizarlos pueden ser las siguientes:

- Capturan la experiencia y la hacen accesible a los no expertos.
- Los nombres de patrones forman un vocabulario que ayuda a los desarrolladores a comunicarse mejor.
- Un sistema es comprendido más rápidamente cuando está documentado con los patrones que usa.
- Los patrones pueden ser la base de un manual de ingeniería de software.
- Facilitan la reestructuración de un sistema tanto si fue o no concebido con patrones en mente.

¿Qué es lo que diferencia a un Patrón de un Framework?

Se listan a continuación las características de los Patrones de Diseño:

- Las descripciones de patrones suelen ser independientes de los detalles de implementación o del lenguaje de programación (salvo ejemplos usados en su descripción).
- Los patrones de diseño son elementos arquitecturales más pequeños que los frameworks.
  - Un framework incorpora varios patrones
  - Los patrones se pueden usar para documentar frameworks.

[1] https://es.wikipedia.org/wiki/Patrónn_de_diseño

Se listan a continuación las características de los Frameworks:

- Los frameworks están implementados en un lenguaje de programación, y pueden ser ejecutados y reutilizados directamente.
- Los patrones de diseño están menos especializados que los frameworks.
  - Los frameworks siempre se aplican a un dominio de aplicación particular.
- Un Framework ofrece un conjunto integrado de funcionalidad específica de un dominio.
- Un Framework es una aplicación medio acabada.
- Las aplicaciones completas se desarrollan mediante herencia, e instanciando componentes parametrizados del framework.

¿Qué tipos de Patrones Existen?

Según el nivel de abstracción los patrones pueden clasificarse de la siguiente manera

- Patrones arquitecturales.
- Patrones de *Diseño.*
- Patrones elementales.

¿Cómo se Clasifican los Patrones de Diseño?

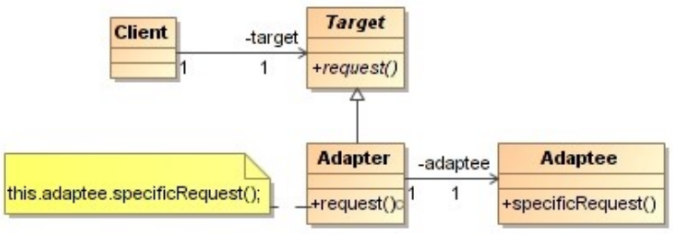
Los Patrones de Diseño se clasifican en:

- **Patrones de Creación**: Tratan sobre la inicialización y configuración de clases y objetos
- **Patrones Estructurales**: Tratan de desacoplar la interfaz e implementación de clases y objetos.
- **Patrones de Comportamiento**: Tratan las interacciones dinámicas entre sociedades de clases y objetos. Responden a la pregunta *¿Cómo interaccionan y se distribuyen responsabilidades los objetos?*.

Seleccione un patrón y presente un ejemplo en el que se pueda observar su uso a través de un diagrama de clases

El patrón **Adapter o Adaptador** se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pueda utilizar la primera haga uso de ella a través de la segunda. [2]
Es recomendable utilizar el patrón adaptador cuando:

- Se desea usar una clase existente, y su interfaz no sea igual a la necesitada.
- Cuando se desea crear una clase reutilizable que coopere con clases no relacionadas. Es decir, que las clases no tienen necesariamente interfaces compatibles.

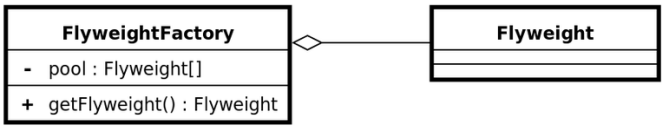


[2] https://es.wikipedia.org/wiki/Adaptador_(patrón_de_diseño)

Seleccione un patrón y describa alguna situación de la vida cotidiana en la que se podría aplicar

El patrón **Flyweight (u objeto ligero)** sirve para eliminar o reducir la redundancia cuando tenemos gran cantidad de objetos que contienen información idéntica, además de lograr un equilibrio entre flexibilidad y rendimiento (uso de recursos). Este permite el uso de un gran número de objetos de grano fino de forma eficiente mediante utilizando el uso compartido.

El diagrama para éste patrón sería el siguiente:



Un ejemplo de la vida real podría ser el de la red telefónica pública. En el cual hay diversos componentes, por ejemplo nodos de conmutación, que se deben compartir entre los distintos usuarios. Los usuarios no conocen cuántos componentes de cada tipo hay disponibles al momento de realizar la llamada. Lo único por lo que se preocupan los usuarios es por obtener tono para marcar y efectuar la llamada.