Programación 2

Lenguaje Java - Patrones de Diseño (*3era parte*)
Típos de Clases

Profesor: Eduardo Godoy.

eduardo.gl@gmail.com

Escuela de Ingeniería Civil Informática.

Universidad de Valparaíso.

25 de octubre de 2017

Contenido

Introducción.

- Se sabe que cada Aplicación tiene particularidades.
- Pero en general siempre se repiten los mismos problemas:

Problemas recurrentes al desarrollar una aplicación

- ¿Cómo persisto los datos?
- ¿Cómo autentifico a los usuarios?
- ¿Cómo separo la presentación, la lógica y control?.

- En lugar de reinventar continuamente una solución para cada nuevo proyecto. Es mucho mas productivo aplicar estrategias que ya hayan funcionado anteriormente.
- Esta idea es lo que lleva al concepto de **Patrón de Diseño** .

Patrón -¿ Un Modelo

- En ingeniería del software, un patrón es una solución ya probada y aplicable a un problema que se presenta una y otra vez en el desarrollo de distintas aplicaciones y en distintos contextos.
- Un patrón no es en general una solución en codificada y lista para usar, sino más bien un **modelo** que desceribe cómo resolver el problema y de ante qué circunstancias es aplicable.

Patrón - Ventajas

- **Están Probados:** son soluciones que han sido utilizadas con anterioridad de manera repetida y se ha comprobado que funcionan.
- **Son reutilizables:** corresponden con problemas que no son específicos de un caso concreto, sino que se presentan una y otra vez en distintas aplicaciones.
- **Son expresivos:** cuando un equipo de desarrolladores tiene un vocabulario común de patrones, se puede comunicar de manera fluida y precisa las ideas fundamentales sobre el diseño de una aplicación.

Patrón - Desventajas

- Uso sin justificación que aplique a un problema. (Simplemente porque son buenos)
- El análisis y experiencia debe dictar si es necesario o no aplicarlos.

DAO - *Data Acceso Object*

- DAO Es un patrón de diseño del tipo creacional.
- Está basado directamente en el concepto de *Separación de Responsabilidades*.

Separación de Responsabilidades.

- Cada clase es creada bajo un único fin o responsabilidad asignada.
- La responsabilidad debe ser cumplida en su totalidad por la clase.
- Este concepto define que una responsabilidad debe estar asociado de forma directa a solo un clase.

DAO - Responsabilidad

- En una aplicación empresarial, DAO tiene la responsabilidad de separar la persistencia de datos del resto de funcionalidades.
- Esto proporciona *independencia de la fuente de datos*, al tener dicha responsabilidad encapsulada dentro de la misma clase.

Nota Relevante

- Un principio básico de un buen diseño de sistema es identificarlos aspectos de la aplicacion que cambian o pueden cambiar y separarlos de los que van a permanecer siempre fijos. Muchos patrones de diseño se basan en encapsular de alguna forma la parte que cambia para hacer mas fácil la extensión del sistema.
- DAO encapsula la parte que puede cambiar, que es la interacción con la fuente de datos.

Titulo

- bla.