26/09/2025

PATRONES DE DISEÑO 2025

Patrones a desarrollar por grupo:

- ✔ GRUPO 1: Abstract Factory
- ✔ GRUPO 2: Composite
- ✔ GRUPO 3: Proxy
- ✔ GRUPO 4: Decorator
- ✔ GRUPO 5: Adapter
- ✔ GRUPO 7: State
- ✔ GRUPO 8: Strategy
- ✔ GRUPO 9: Facade - Command
- ✔ GRUPO 10: Builder
- ✔ GRUPO 12: Bridge
- ✔ GRUPO 13: Visitor
- ✔ GRUPO 14: Null Object
- ✔ GRUPO 15: Factory Method
- ✔ GRUPO 16: Template Method
- ✔ GRUPO 17: Memento
- ✔ GRUPO 18: DTO
- ✔ GRUPO 19: Mediator
- ✔ GRUPO 20: Chain of Responsability - Prototype
- ✔ GRUPO 21: Flyweight
- ✔ GRUPO 22: Iterator
- ✔ GRUPO 23: MVC

Fecha de presentación para revisión: 08/10. Documento con lo investigado. Dudas sobre cada patrón. Dudas sobre el ejemplo.

Fecha de presentación de la clase: 15/10

Requisitos de la Presentación

1. Introducción (Qué es y para qué sirve):
    - Definir el patrón asignado.
    - Indicar su categoría (Creacional, Estructural, Comportamiento o de Arquitectura).
    - Mencionar su propósito principal en una sola frase (ej: "El patrón Adapter permite que interfaces incompatibles trabajen juntas").
    - Explicar qué principio(s) SOLID ayuda a implementar.
2. El Problema (El "Antes"):
    - Presentar un escenario simple y común que sea problemático (difícil de mantener, inflexible, con código duplicado, etc.).
    - Mostrar un diagrama UML simple de esta mala solución.
3. La Solución (El "Después"):
    - Introducir el patrón como la solución elegante a ese problema.
    - Mostrar el diagrama UML de la solución aplicando el patrón. Explicar las responsabilidades de cada clase (el *Context*, la *Strategy*, el *Component*, el *Decorator*, etc.).

4. Ventajas y Desventajas:
    - Resumir en 2-3 puntos clave cuál es la gran ventaja de usar el patrón.
    - Mencionar si tiene alguna desventaja (ej: mayor complejidad, más clases).

Otras consideraciones a tener en cuenta:

• El objetivo de la clase es que los oyentes puedan tener una visión general acerca del patrón estudiado, y más importante aún, puedan comprender con un sencillo ejemplo cuál es su finalidad y ventaja de aplicación. Deben tener en cuenta que el resto de la clase no posee conocimientos acerca del patrón presentado.
• Tiempo aproximado de clase: 10 minutos.
• No es necesario que incluyan código fuente para explicar el ejemplo. Pueden hacerlo si consideran que es necesario, pero cuidado con distraer el objetivo principal, que es la claridad de conceptos.

Qué debo entregar?

Documento con lo investigado de los patrones asignados
Presentación de la clase.

- **Documento de Investigación:** Documento en PDF con la investigación, diagramas. No olvidar indicar la bibliografía en caso de usar alguna distinta a la sugerida.
- **Presentación:** Diapositivas utilizadas en la clase.

Bibliografía sugerida

- Introducción a los Patrones de Diseño - Oscar Blancarte . 2016
- Introducción a las Arquitecturas de Software - Oscar Blancarte . 2020
- Orientación a Objetos. Java y UML – Carlos Fontela – Nueva Librería – 2011 (Capitulo 20)
- Head First. Design Patterns – Freeman – O'Relly – 2004
- Software Architecture Design Patterns in Java - Partha Kuchana – AuerbachPublications – 2004
- https://refactoring.guru/es/design-patterns/catalog

Modo y Criterios de Evaluación:

Tendrán una nota final grupal para la cual se tendrá en cuenta: completitud de los puntos requeridos, claridad de los conceptos y sobre todo claridad y pertinencia de los ejemplos. Asimismo, recibirán una devolución acerca del desempeño durante la exposición.

Criterio	Puntaje Máximo	Descripción
Claridad Conceptual	2	Define el patrón, su propósito y categoría de forma precisa y fácil de entender.
Análisis del Problema	2	Explica claramente el escenario "sin el patrón" y por qué es problemático.
Calidad del Ejemplo	3	Explica con un ejemplo claro, que comprenden y facilita entender como funciona el patrón.
Presentación y Didáctica	3	La exposición es clara, respeta el tiempo y está orientada a compañeros sin conocimiento previo.