



Tecnicatura Universitaria
en Programación

LABORATORIO DE COMPUTACIÓN III

Anexo N°4:
Patrones

Material de Estudio
2° Año – 3° Cuatrimestre



Índice

Patrones de Diseño	2
Patrones	2
Anti-Patrones.....	2
GOF (The Gang of Four)	2
Lista de patrones de GOF	3
Más clasificaciones de los patrones de diseño:.....	4
Explicación	5
BIBLIOGRAFÍA	6

Patrones de Diseño

Patrones

Los patrones de diseño son soluciones probadas y comprobadas para problemas comunes en el diseño de software. Son un conjunto de mejores prácticas y enfoques para resolver problemas de diseño de software de manera efectiva y eficiente. Además, proporcionan un lenguaje común y una estructura para los desarrolladores de software, lo que les permite comunicarse y colaborar de manera más efectiva en proyectos de desarrollo de software.

Los patrones de diseño se basan en la idea de que muchos problemas de diseño de software son recurrentes y pueden ser resueltos de manera similar en diferentes contextos. Estos patrones encapsulan soluciones probadas y comprobadas para estos problemas, lo que permite a los desarrolladores reutilizar soluciones ya existentes en lugar de reinventar la rueda cada vez que se enfrentan a un problema similar.

Anti-Patrones

Además de los patrones de diseño, también existen los anti patrones, que son soluciones ineficientes o incorrectas para problemas de diseño de software. Los anti patrones son patrones de diseño mal aplicados o enfoques inadecuados para resolver problemas de diseño, lo que puede resultar en código de baja calidad, difícil de mantener y propenso a errores.

Los anti patrones pueden surgir de diversas causas, como la falta de comprensión adecuada del problema a resolver, la presión por cumplir plazos ajustados o la falta de experiencia en el diseño de software. Algunos ejemplos de anti patrones incluyen el "Objeto todopoderoso" (**God Object**), que concentra demasiada lógica en un solo objeto, el "código espagueti" (**Spaghetti Code**), que se refiere a un código fuente complejo y desorganizado que es difícil de entender y mantener.

GOF (The Gang of Four)

El libro "Design Patterns: Elements of Reusable Object-Oriented Software", también conocido como **GOF**, es un libro de referencia escrito por cuatro autores reconocidos en el campo de la programación orientada a objetos. Estos son Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, y el libro fue publicado por primera vez en 1994. Este libro se considera uno de los clásicos en el campo de la ingeniería de software, y ha sido ampliamente utilizado y citado en la comunidad de desarrollo de software, se enfoca en los patrones de diseño y presenta un catálogo de 23 patrones de diseño diferentes, que se dividen en tres categorías principales:

Patrones de diseño creacionales: Estos patrones se refieren a la creación de objetos y manejo de su ciclo de vida. Ejemplos de patrones creacionales incluyen el patrón Singleton, el patrón Factory Method, y el patrón Abstract Factory.

Patrones de diseño estructurales: Estos patrones se refieren a la organización y composición de clases y objetos. Ejemplos de patrones estructurales incluyen el patrón Decorator, el patrón Adapter, y el patrón Composite.

Patrones de diseño de comportamiento: Estos patrones se refieren a la interacción y comunicación entre objetos. Ejemplos de patrones de comportamiento incluyen el patrón Observer, el patrón Strategy, y el patrón Template Method.

Lista de patrones de GOF

1. Creacionales:
 - a. Abstract Factory
 - b. Builder
 - c. Factory Method
 - d. Prototype
 - e. Singleton
2. Estructurales:
 - a. Adapter
 - b. Bridge
 - c. Composite
 - d. Decorator
 - e. Facade
 - f. Flyweight
 - g. Proxy
3. Comportamentales:
 - a. Chain of Responsibility
 - b. Command
 - c. Interpreter
 - d. Iterator
 - e. Mediator
 - f. Memento
 - g. Observer
 - h. State
 - i. Strategy
 - j. Template Method
 - k. Visitor

Más clasificaciones de los patrones de diseño:

Además de la clasificación de los patrones de diseño propuesta por el libro GOF, existen otras clasificaciones que también son ampliamente aceptadas por la comunidad de desarrollo de software.

Patrones de optimización: Estos patrones se utilizan para optimizar el rendimiento y la eficiencia de un sistema. Estos patrones buscan identificar y abordar cuellos de botella, redundancias y otras áreas problemáticas en el diseño o implementación de un sistema, con el objetivo de mejorar su desempeño y eficiencia.

- Patrón Caché (Cache).
- Patrón Prefetching.
- Patrón Indexación (Indexing).
- Patrón Pool de Objetos (Object Pool).
- Patrón Lazy Initialization.
- Patrón Load Balancing.
- Patrón Código en Línea(Inline Code).

Patrones de diseño arquitectónicos: Los patrones arquitectónicos son soluciones probadas y comprobadas para el diseño y organización de la estructura de un sistema de software. Son enfoques de alto nivel que ayudan a los arquitectos de software a tomar decisiones sobre la distribución, interacción y organización de los componentes de un sistema, con el objetivo de lograr un sistema robusto, escalable y mantenible. Los patrones arquitectónicos proporcionan un marco conceptual para abordar problemas comunes en el diseño de software, y se utilizan para definir la arquitectura global de una aplicación o sistema.

- Patrón MVC (Model-View-Controller)
- Patrón MVP (Model-View-Presenter)
- Patrón MVVM (Model-View-ViewModel)
- Patrón Hexagonal (o Ports and Adapters)
- Patrón CQRS (Command Query Responsibility Segregation)
- Patrón Repositorio (Repository)
- Patrón Inyección de Dependencias (Dependencies injections)

Patrones de diseño para concurrencia y paralelismo: Se utilizan en la programación de software para manejar y aprovechar eficientemente la ejecución concurrente y paralela de tareas en sistemas informáticos. La concurrencia se refiere a la capacidad de un sistema para manejar múltiples tareas concurrentemente, mientras que el paralelismo implica la ejecución simultánea de tareas en diferentes núcleos o procesadores. Estos patrones son especialmente importantes en sistemas

modernos, donde la computación en paralelo se ha vuelto cada vez más común para aprovechar el poder de procesamiento de los sistemas multi-core y distribuidos.

- Patrón Monitor (Monitor)
- Patrón Barrera (Barrier)
- Patrón Productor-Consumidor (Producer-Consumer)
- Patrón Pool de Hilos (Thread Pool)
- Patrón Actores (Actor)
- Patrón MapReduce
- Patrón Parallelism
- Patrón Locking

Explicación

Se provee el libro GOF en su versión original en inglés y en español para el estudio de cada patrón.

BIBLIOGRAFÍA

Libro “Design Patterns - Elements of Reusable Object-Oriented Software” de Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.



Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.