



**Tecnológico Nacional de México**

**Instituto Tecnológico de Tijuana**

**SEMESTRE:**

Agosto-Diciembre 2023

**CARRERA:**

Ingeniería en Sistemas Computacionales

**MATERIA:**

Patrones De Diseño De Software

**Nombre del trabajo**

1.1 Concepto de Patrones de diseño

**UNIDAD A EVALUAR:**

Unidad 1

**NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:**

González Guzmán María José #19211650

**NOMBRE DEL MAESTRO (A):**

Jose De Jesus Parra Galaviz

Fecha de entrega: 02 de septiembre del 2023

## Concepto de Patrones de Diseño.

- a) Un patrón de diseño es una descripción general de cómo resolver un problema comúnmente recurrente dentro de un contexto dado en el diseño de software. Es una solución template que puede ser adaptada y aplicada a problemas similares, con modificaciones mínimas o ninguna en absoluto. Los patrones de diseño permiten una comunicación efectiva entre los diseñadores de software, promueven la reutilización de código y mejoran la calidad y la flexibilidad del diseño del software.

(Gamma et al., 1994)

- b) Un patrón de diseño resuelve un problema general que ocurre una y otra vez en el diseño de aplicaciones y sistemas. Un patrón no es una solución terminada que se puede transformar directamente en código. Es una descripción general de cómo resolver un problema. Puede acelerar el proceso de diseño proporcionando ejemplos probados y efectivos de cómo hacerlo.

(Freeman et al., 2004)

## Los cuatro componentes de los patrones de diseño

En general, un patrón tiene cuatro elementos esenciales:

1. El nombre del patrón es un identificador que podemos utilizar para describir un problema de diseño, sus soluciones y consecuencias en una o dos palabras. Nombrar un patrón aumenta de inmediato nuestro vocabulario de diseño. Nos permite diseñar a un nivel más alto de abstracción. Tener un vocabulario para los patrones nos permite hablar de ellos con nuestros colegas, en nuestra documentación e incluso con nosotros mismos. Facilita pensar en diseños y comunicarlos, junto con sus compensaciones, a otros. Encontrar buenos nombres ha sido una de las partes más difíciles de desarrollar nuestro catálogo.

2. El problema describe cuándo aplicar el patrón. Explica el problema y su contexto. Puede describir problemas de diseño específicos, como cómo representar algoritmos como objetos. Puede describir estructuras de clases u objetos que son sintomáticas de un diseño inflexible. A veces, el problema incluirá una lista de condiciones que deben cumplirse antes de que tenga sentido aplicar el patrón.

3. La solución describe los elementos que componen el diseño, sus relaciones, responsabilidades y colaboraciones. La solución no describe un diseño o implementación concretos, porque un patrón es como una plantilla que se puede aplicar en muchas situaciones diferentes. En cambio, el patrón proporciona una

descripción abstracta de un problema de diseño y cómo una disposición general de elementos (clases y objetos en nuestro caso) lo resuelve.

4. Las consecuencias son los resultados y las compensaciones de aplicar el patrón. Aunque a menudo no se mencionan cuando describimos decisiones de diseño, son fundamentales para evaluar alternativas de diseño y comprender los costos y beneficios de aplicar el patrón. Las consecuencias para el software a menudo se refieren a compensaciones entre espacio y tiempo. También pueden abordar cuestiones de lenguaje e implementación. Dado que la reutilización suele ser un factor en el diseño orientado a objetos, las consecuencias de un patrón incluyen su impacto en la flexibilidad, extensibilidad o portabilidad de un sistema. Enumerar estas consecuencias de manera explícita le ayuda a comprenderlas y evaluarlas.

(Gamma et al., 1994)

## Bibliografía

- Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). *Head First Design Patterns*. O'Reilly Media.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.