Patrones de Diseño

Se basan fundamentalmente en retomar un conjunto de soluciones que han sido comprobadas y luego aplicarlas para la resolución de problemas en el desarrollo de un software.

Para poder aplicar los patrones es sumamente importante tener delimitados los problemas y el contexto en el desarrollo de este.

Clases de Patrones

Las clases de patrones abarcan un gran abanico de oportunidades en el diseño de software, entre los cuales se encuentran:

* Patrones arquitectónicos.
* Patrones estructurales.
* Modelado de datos dentro de un sistema.
* Patrones creacionales
* Patrones de datos.
* Patrones conductuales.

Los patrones más utilizados para la resolución de problemas relacionados con la interacción entre interfaz de usuario, la lógica de negocio y los datos son:

- Modelo vista controlador
  - Plantea el uso de 3 capas para separar la interfaz de usuario de los datos y la lógica de negocio, dichas capas son
- Modelo vista presentador
  - Plantea el uso de 3 capas y toda la lógica de la presentación de la interfaz reside en el presentador, dichas capas son

- ✓ **Modelo**: define la estructura de datos y su función principal es el almacenamiento y persistencia de datos.
- ✓ **Vista**: contiene la interfaz de usuario y maneja la interacción del usuario con la interfaz de usuario para enviar peticiones al controlador.
- ✓ **Controlador**: responde a eventos, capturandolos por la interacción de usuarios en la interfaz, procesa la información y solicita o modifica datos en el modelo.

- ✓ **Modelo**: gestiona los datos y su función principal es el almacenamiento y persistencia de datos.
- ✓ **Vista**: es una interfaz de comportamiento de lo que se puede realizar con la vista.
- ✓ **Presentador**: envía las acciones de la vista hacia el modelo.

Comparando

MVC

* El controlador puede tener múltiples vistas para representar los datos.
* Contiene la lógica de negocio del sistema y sirve de intermediario entre el modelo y la vista.

MVP

* Cada vista tiene asignada su presentador responsable de gestionar su lógica de representación.
* Unicamente se encarga de la lógica representación de la vista, así como de hacer de intermediario entre el modelo y la vista.

Los patrones de diseño mencionados anteriormente podrían interpretarse como:

El controlador y modelo de MCV sería el modelo MVP, por otra parte, la vista de MVC encapsula al presentador y la vista de MVP, en otras palabras MVP estaría enfocado a la vista al dedicar dos capas a esta y su representación.

Patrón Observer

El patrón observer se basa en dos objetos con una responsabilidad bien definida; por una parte los objetos observables y por otra, los observadores.

- Objetos observables

Son objetos con un estado concreto, capaces de informar a los suscriptores subscritos al observable y que desean ser notificados sobre cambios de estado de estos objetos.

- Objetos observadores.

Son objetos que se subscriben a los objetos observables y que solicitan ser notificados cuando el estado de estos observables cambie.

Al empezar a diseñar se tiene que tomar en cuenta desde el primer momento que al crear un nuevo diseño, se utilicen los patrones existentes en todo momento.

↓

Estos patrones no se limitan solo al diseño puro, sino que pueden aplicarse en todas las fases del desarrollo, como:

- La planificación de la arquitectura.
- El tipo de solución al problema.
- Características o funciones específicas.
- La forma en abordar los problemas.

Cada pequeña solución de la abstracción de cada problema debe funcionar en conjunto con el problema principal.

↓

es decir:

Si un problema general se aborda abstrayendo en pequeños subproblemas, estas pequeñas soluciones deben funcionar de forma fluida con el conjunto de todos los pequeños problemas que conforman uno general.