

Ensayo :

Patrones de diseño y su importancia en el desarrollo web

Los patrones de diseño son soluciones a problemas de diseño no triviales que son efectivas y reusables, tales problemas son tan comunes que es altamente probable que alguien más (o una comunidad entera) ya haya probado su implementación en distintos ámbitos y escenarios concluyendo que es efectiva ante el problema de diseño en cada instancia que se presenta del mismo.

Existen de diversos tipos, para solventar distintas problemáticas tales como de creación de objetos genérica, estructuración de arquitecturas complejas, mejora de presentación de datos, cadenas de responsabilidades con mismo objeto o uso de objetos de manera compartida , por mencionar algunas las principales subcategorías que hay.

Al estar completos, es decir, dentro de la definición de un patron de diseño tenemos:

1. El **nombre del problema** el cual per se describe en pocas palabras qué es lo que resuelve el patrón.
2. El **problema** describe cuando aplicar el patrón de diseño y cómo ayuda al problema de manera específica.
3. La **solución** que detalla los elementos que conforman el diseño así como sus relaciones ,responsabilidades y colaboraciones.
4. Las **consecuencias** describen el impacto y los intercambios (trade-offs) que conlleva la aplicación del patrón del diseño.

Los patrones de diseño nos permiten delegar y administrar responsabilidades en los módulos de un sistema usando una de estructura de datos potencialmente reutilizable con un paradigma orientado a objetos, es por ello que dentro del desarrollo de aplicaciones es altamente efectivo necesario dado que al ser un ambiente con demasiados años en desarrollo, es altamente probable que alguien más ya haya enfrentado un problema similar.

Recordando aquí que el uso de patrones de diseño no va ligado a las implementaciones concretas de los problemas sino a los componentes y las relaciones de uso común de estos dentro de los diseños, y haya sido registrado algún patrón que nos ayude a resolverlo incluso de manera más sencilla que de la compleja variedad de opciones que se tiende a generar al atacar un problema desconocido y muy específico.

De manera específica, el modelo vista controlador nos permite la manipulación de datos (dentro del modelo) en su forma pura sin tener que sincronizar a mano la presentación y procesamiento de los mismos; es responsabilidad del controlador procesar los datos y señales de intercomunicación

para su presentación en diferentes posibles vistas del sistema sin afectar al modelo de datos trabajado en su estructura.

Esto en web representa que el desarrollo de nuestras aplicaciones puede ir segmentado en capas las cuales tienen una responsabilidad y objetivo específicos, podemos decir que podemos "separar" el desarrollo backend del frontend debido a la interdependencia de la información sin necesidad de preocuparnos en cada etapa del procesamiento, pudiendo procesar datos crudos⁷ encargando a algún patrón de diseño que se encargue de presentarlos de manera correcta al usuario final. Esto es un gran apoyo debido a que mediante el uso de distintos patrones podemos segmentar al sistema y asignar a cada nivel de encapsulamiento su trabajo a realizar de manera correcta, resolviendo nuestro problema original sin desperdiciar tiempo en reinventar algo ya con bases y pudiendo maximizar la modularidad a nivel de "dejar hacer su trabajo a cada quién y todos felices al hacer depuración y chequeo contra lógica de negocios".

Referencias

- [1] Erick Gamma et al *Design Patterns (Elements of Reusable Object-Oriented Software)*.