

Patron de diseño

MVC

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del MVC) para la arquitectura.

- **Modelo:** Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va aquí, en el modelo.
- **Vista:** Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.
- **Controlador:** Se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

DDD

El Modelo se refiere al conocimiento que el equipo tiene sobre el dominio. Este incluye solo la parte relevante para los casos de uso que se implementan en la aplicación. La idea es que el conocimiento se vaya refinando y adaptando constantemente a medida que se aprenda sobre el dominio. El objetivo es ahorrar energía en soluciones técnicas y sobrecargo de los desarrolladores a aprender los principios del negocio. El Modelo también se define como el lenguaje común usado por los desarrolladores y los expertos del dominio, lo que facilita la comunicación con los expertos y evita malos entendidos.

Ahora que ya sabemos lo que es DDD, necesitas cumplir con algunos requisitos indispensables:

- El desarrollo debe ser iterativo. Esto será necesario para ir refinando el modelo del dominio continuamente a medida que aprendemos más sobre este y avanzamos.
- Debe existir una estrecha relación entre los desarrolladores y los expertos del dominio. Como dije antes, el conocimiento profundo del dominio es esencial, al igual que la colaboración con los expertos de desarrollo durante la vida del proyecto; esto evitará malos entendidos entre las partes del equipo y ofrecerá la oportunidad de obtener un conocimiento más profundo del dominio.