



Ciclo 2

Semana 2

Modelamiento y Modelo Vista Controlador (MVC)

Lectura 4 - Patrón de diseño MVC: Modelo – Vista – Controlador

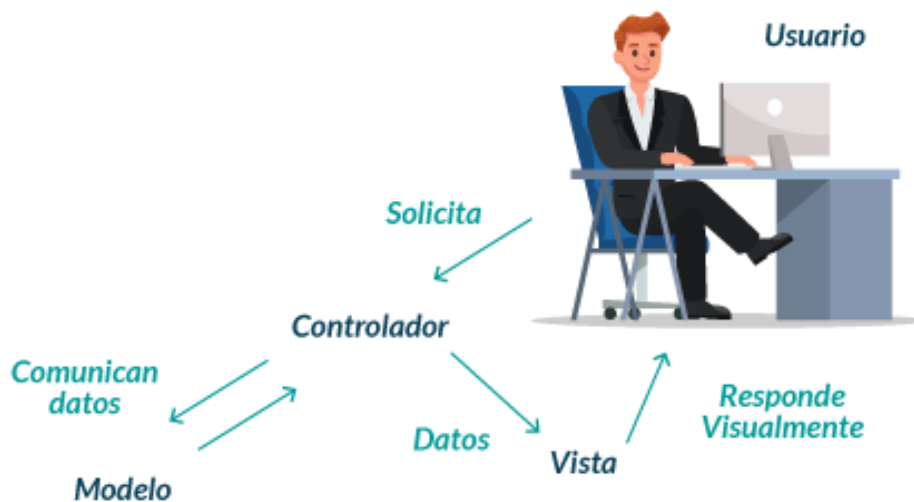


| Patrón de diseño MVC: Modelo – Vista – Controlador

Los patrones de diseño se pueden entender como el esqueleto de la solución a un problema cuando se hace el desarrollo de software, dando solución probada y documentada a problemas en contextos similares, facilitando la reutilización de arquitecturas y diseños de software.

El patrón de arquitectura ofrece solución a problemas de arquitectura de software dando una descripción de los elementos y la relación entre estos además de las restricciones de cómo usarlos, con subsistemas, responsabilidades e interrelaciones y ofrecen un nivel mayor de abstracción.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que divide los datos de un programa, la interfaz de usuario, y la lógica de control en tres componentes conocidos como Modelo, Vista y Controlador, separando por un lado los componentes que representan la información y por el otro la interacción del usuario. Este patrón se basa en la reutilización de código y la separación de conceptos para facilitar la tarea de desarrollo y su mantenimiento.



Ahora definamos cada componente por separado:

Modelo:

Representa la información o datos con la que el sistema opera, su lógica de negocio, y sus mecanismos de persistencia como un conjunto de clases que representan la información del mundo real que el sistema debe reflejar. Esta lógica de datos da integridad a los datos y permite derivar datos nuevos; por tanto, en el modelo se operan los datos y las reglas de negocio asociadas al sistema, el análisis sintáctico y el procesamiento de los datos de entrada y de salida.

A través del modelo se accede al almacenamiento de datos ya que lo ideal es que el modelo sea independiente del almacenamiento, además define la funcionalidad o reglas del sistema y lleva un registro de las vistas y controladores notificando a las vistas los cambios que en los datos ocasionen los agentes externos

El modelo gestiona el acceso a los datos, consultas y actualizaciones, implementando privilegios de acceso descritos en la lógica de negocio. Además, envía a la vista la información solicitada para que sea mostrada.

Vista:

Es la interface de usuario y acomoda la información enviada al cliente y los medios que interactúan con este. La vista es la interface gráfica, la capa que ve el usuario en un formato que sea adecuado para su clara interacción. La vista recibe los datos que genera el modelo y se los muestra al usuario, llevando un registro de su controlador asociado y ofrece un servicio de actualización que se invoca por el controlador o por el modelo.

Es la representación los datos que contiene el modelo mostrados al usuario. Esta relación entre vista y modelo es de muchas a uno pues cada vista se asocia a un modelo, aunque existen casos donde muchas vistas se asocian al mismo modelo.

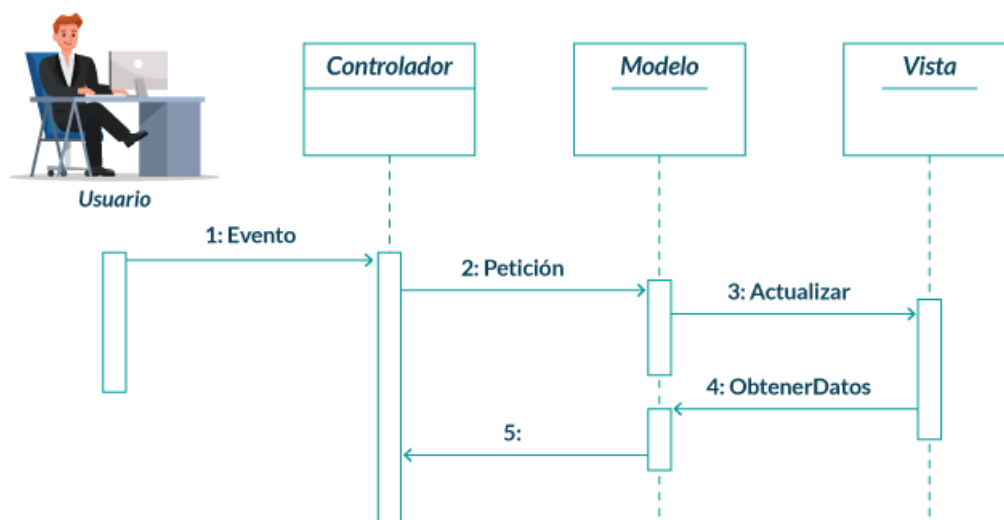
La vista toma la información requerida del modelo y hace el despliegue. Si este despliegue implica un cambio en el modelo de dominio, la vista cambia con notificaciones que genera el modelo.



Controlador:

Es el intermediario entre la vista y el modelo y gestiona su flujo de información y sus cambios para adaptar los datos según las necesidades de cada uno. En este componente se controla todo lo que puede hacer la aplicación respondiendo a las acciones que realiza el usuario e invoca los cambios en el modelo y en la vista. Se compone de acciones representadas con funciones en la clase. Las peticiones de acceso o uso de información van al modelo a través del controlador.

El controlador se encarga de recibir eventos de entrada que genera el usuario, contiene las reglas de gestión de dichos eventos como peticiones al modelo o la vista.



- El usuario ingresa su petición a través de un evento.
- El Controlador toma este evento y ejecuta una petición al modelo o a la vista si así corresponde.
- El modelo llama la vista para su actualización si así corresponde y para hacer esta actualización la vista podría solicitar datos al Modelo.
- El Controlador recibe el control.

Este patrón de diseño MVC tiene una implementación modular dando a las aplicaciones que lo implementan extensibilidad y un mantenimiento fácil y único, aunque el patrón modelo



Semana 2

Modelamiento y Modelo
Vista Controlador (MVC)

vista controlador requiere más tiempo en la etapa inicial de desarrollo pues exige al programador desarrollar más clases que en otros entornos de desarrollo.

La vista de la aplicación se actualiza siempre sin que el programador deba solicitar que la vista se actualice, ya que el proceso es una tarea automática del modelo. Además, toda modificación que afecte al dominio, implicará una modificación solamente en el modelo y en las interfaces de este con la vista y las modificaciones a la vista no afectaran al modelo de dominio, sino a la representación de la información. MVC es un patrón de diseño orientado a objetos por lo que su implementación suele ser costosa y difícil en lenguajes que no sean orientados a objetos.

Un ejemplo de MVC es:

