

Patrón de diseño MVC

Patrón de diseño MVC	1
¿Qué aprenderás?	2
Introducción	2
MVC (Model View Controller)	2
Modelo	3
Vista	5
Controlador	6



¡Comencemos!

¿Qué aprenderás?

- Entender y diferenciar cada una de las capas del patrón de diseño MVC

Introducción

El modelo MVC (Model View Controller), es un patrón de diseño de software ampliamente utilizado por casi todos los frameworks más populares y eficientes del mercado, tales como Struts, Ruby on Rails, Angular, Django, Spring, entre otros. Debido a esto, es importante aprender cómo funciona este modelo, ya que casi todos los sistemas que se están desarrollando en la actualidad implementan el modelo vista controlador. MVC, adquiere su popularidad en base a su eficacia en cuanto a la reutilización de código y la separación de los componentes dependiendo la responsabilidad de los mismos.

¡Vamos con todo!



MVC (Model View Controller)

El Modelo Vista Controlador (MVC), es un patrón de diseño de software cuyo objetivo principal es lograr separar la Vista (Interfaces de usuario) con el Modelo del software (Acceso a datos), de tal forma que la lógica del sistema y el flujo del mismo sea orquestada por el controlador. De esta forma, se logra tener un software en que sus componentes están diseñados de manera independiente (desacople de objetos), por lo que un cambio en cualquier parte de estos componentes no debería afectar o impactar a otro componente.

Por ejemplo, si hay un cambio en el diseño de la base de datos, solo se deberían hacer cambios en el modelo, sin afectar la lógica implementada en la vista y en controlador, y viceversa.

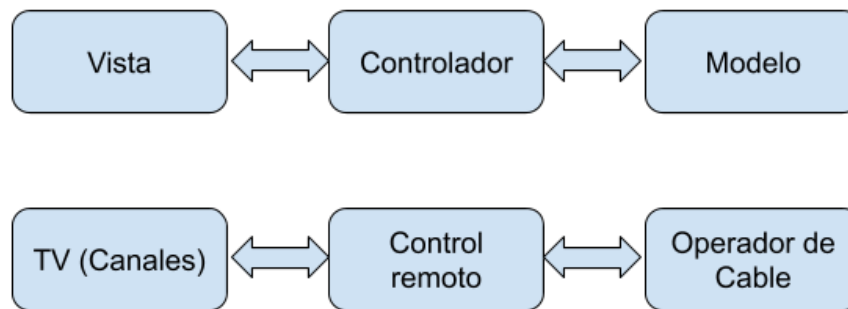


Imagen 1. Ejemplo de Modelo MVC con el televisor.

Fuente: Desafío Latam.

El Modelo Vista Controlador, se puede ejemplificar con el televisor. Los canales y la pantalla del televisor serían la Vista del sistema, ya que es lo que el usuario puede ver. El usuario, quien ve la televisión, pide a través del control remoto (Controlador) otro canal. El control remoto, identifica la acción del usuario, envía la señal para el cambio de canal al operador de cable (Modelo), quien tiene la información del número de canal, a que corresponde el canal, la categoría, etc. El operador de cable responde con la información del canal y la transmisión en vivo al controlador, quien finalmente devuelve el canal solicitado a la vista.

En resumen, si el operador de cable hace un cambio, por ejemplo el aumento de canales, no es necesario modificar la implementación del control remoto ni de la vista, ya que el cambio para estos componentes es transparente.

Modelo

Esta capa, es la encargada de obtener los datos, generalmente desde una base de datos e implementar la o las lógicas de negocio sobre los mismos.

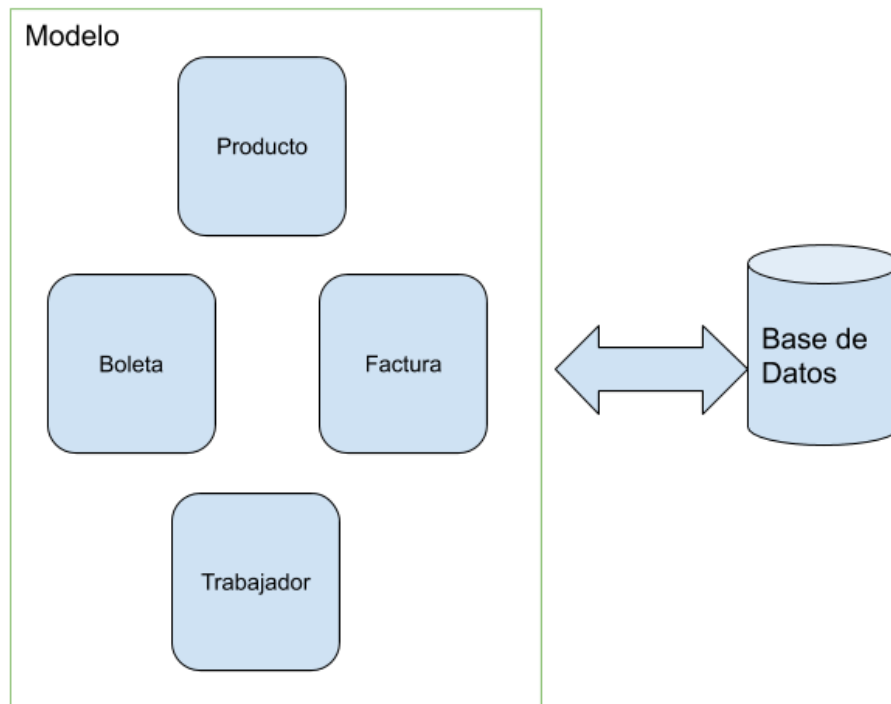


Imagen 2. Modelo.
Fuente: Desafío Latam.

El modelo corresponde a todos los objetos del sistema que tienen alguna relación con la base de datos. En la imagen, se ven algunos ejemplos de objetos de un modelo, tal como Producto, Boleta, Factura, Trabajador. Estos objetos tienen la responsabilidad de buscar los registros en la base de datos correspondiente a su tipo, es decir Factura debe tener la responsabilidad de tener los datos de las facturas, Boleta los datos de las boletas, y así sucesivamente.

Vista

Esta capa, se encarga o tiene la responsabilidad de presentar, de manera gráfica, los datos provenientes del modelo, mediante solicitudes al controlador.

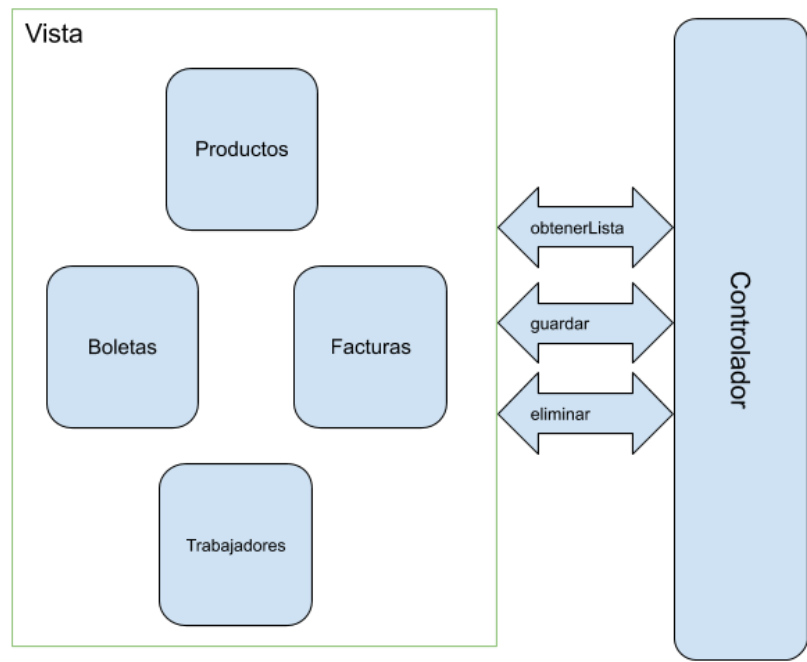


Imagen 3. Vista.
Fuente: Desafío Latam.

En la gráfica, se muestran ejemplos de vistas, tales como una pantalla, en donde se muestre la lista de Productos, otra para mostrar la lista de Facturas y así sucesivamente. Cada vista tiene la responsabilidad, en este caso, solicitar al controlador las listas de datos para ser mostradas en la pantalla al usuario. Además, estas vistas podrían solicitar también al controlador, guardar un producto, borrar un Trabajador u otra acción que se requiera.

Controlador

Este se encarga de recibir las peticiones de los usuarios o de la Vista, controlando así estas peticiones para ir a buscar los datos al Modelo. El controlador recibe los datos desde el Modelo, para así devolver estos a la capa Vista del software. A grandes rasgos, esta capa se encarga de controlar, valga la redundancia, el flujo de datos e información entre la capa Vista y la capa del Modelo propiamente tal.

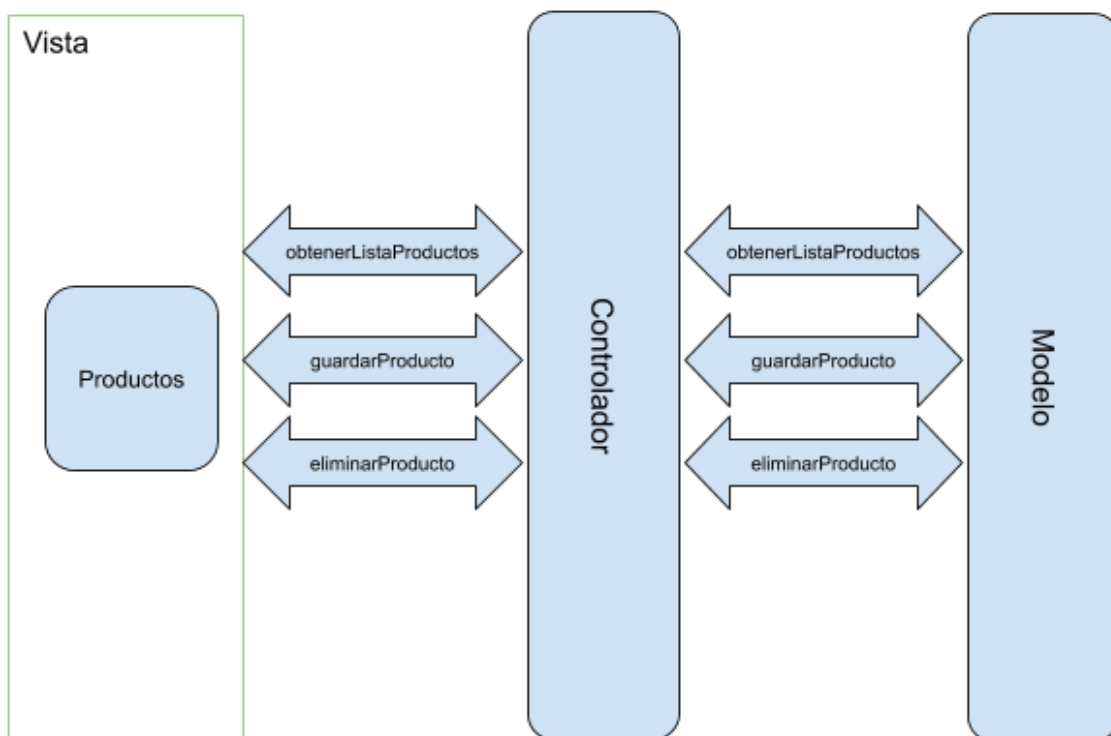


Imagen 4. Relación de la vista con el controlador y modelo.

Fuente: Desafío Latam.

En la imagen 4, se muestra como la vista Productos tiene la responsabilidad de solicitar al controlador la lista de productos, guardar un producto o bien eliminar un producto. El controlador, por su parte, recibe estas peticiones y evalúa cómo interactuar con el modelo correspondiente. También podemos apreciar como funciona esto con Trabajadores.