

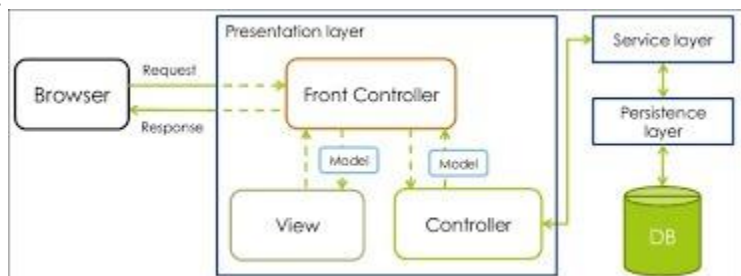
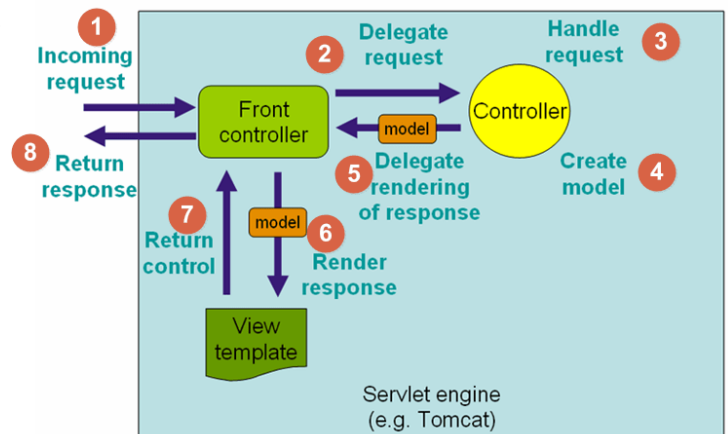
# MVC EN SPRING

*framework* modelo-vista-controlador

MVC es una arquitectura Modelo-Vista-Controlador, una solución de diseño comprobada para desarrollar aplicaciones basadas en web. Spring como un marco MVC proporciona componentes preconstruidos, para todas las capas de la arquitectura de la aplicación web, y el desarrollador solo tiene que integrarlos y puede concentrarse en la lógica empresarial central, que es el lema del marco Spring.

| PATRÓN MVC  |  |  |
|---|--|--|
| <b>Modelo:</b> el modelo no es más que un objeto que contiene datos comerciales, normalmente hablando, es POJO. | <b>Vista:</b> una vez que los datos del modelo se devuelven desde el back-end, View se encarga de representarlos en el navegador. La vista se puede implementar con tecnologías de interfaz de usuario como HTML, CSS, js frameworks. Entonces la entrada de View será modelo. | <b>Controlador:</b> si el usuario envía una solicitud, el controlador toma la solicitud, identifica el tipo de solicitud y le envía el método de servicio adecuado para generar los datos del modelo. Entonces la salida del controlador será Model. |

- Cualquier *solicitud entrante* que llegue a la aplicación web se enviará a Front Controller (Dispatcher Servlet)
- El *front controller* decide a quién (controller) tiene que entregar la solicitud, en función de los encabezados de la solicitud.
- El *controller* que tomó la solicitud, procesa la solicitud, enviándola a la clase de servicio adecuada.
- Una vez realizado todo el procesamiento, el *controller* recibe el modelo de la capa de *servicio o acceso a datos*.
- El *controller* envía el modelo al *front controller* (Servlet Dispatcher).
- El *servlet Dispatcher* encuentra la plantilla de vista, utiliza la resolución de vista y le envía el modelo.
- Usando la plantilla de vista, el modelo y la página de vista se construyen y se envían de vuelta al front controller.



# CONCEPTOS DE SPRING WEB MVC

## Controlador de resorte:

Spring proporciona muchos tipos de controladores, el desarrollador tiene que elegir uno en función de sus requisitos, ya sea que tenga un formulario en su interfaz de usuario, si tiene asistentes, etc.

## Modelo y vista:

ModelAndView es parte del paquete `org.springframework.web.servlet`. El controlador devuelve la instancia de ModelAndView y los valores del modelo se colocan en el objeto en forma de mapa.

## Objeto de comando:

Este es el objeto que se pasará una vez si el usuario envía cualquier formulario en la interfaz de usuario frontal, con todos los campos del formulario. El mismo objeto se pasará al validador para validar el contenido y verificar si se llena como lo necesitamos, y es una representación simple de estilo java bean.

## Validador:

- El validador se utiliza para validar los datos del formulario recibidos por el controlador.
- Se puede crear un validador personalizado para el formulario implementando la interfaz `org.springframework.validation.Validator`
- Tiene dos métodos `validar` y `errores`, para validar el objeto de comando pasado y errores para almacenar los errores.

## Spring tag library:

La biblioteca de tipos Spring Bind es muy simple y poderosa. Generalmente se usa en las páginas jsp usando la etiqueta `<spring:bind>`, que se usa para vincular campos de formulario al objeto de comando.