



DESARROLLO DE SOFTWARE I

Eric Gustavo Coronel Castillo

gcoronelc.blogspot.com

ecoronel@uch.edu.pe



TEMA: Spring MVC – Parte 1

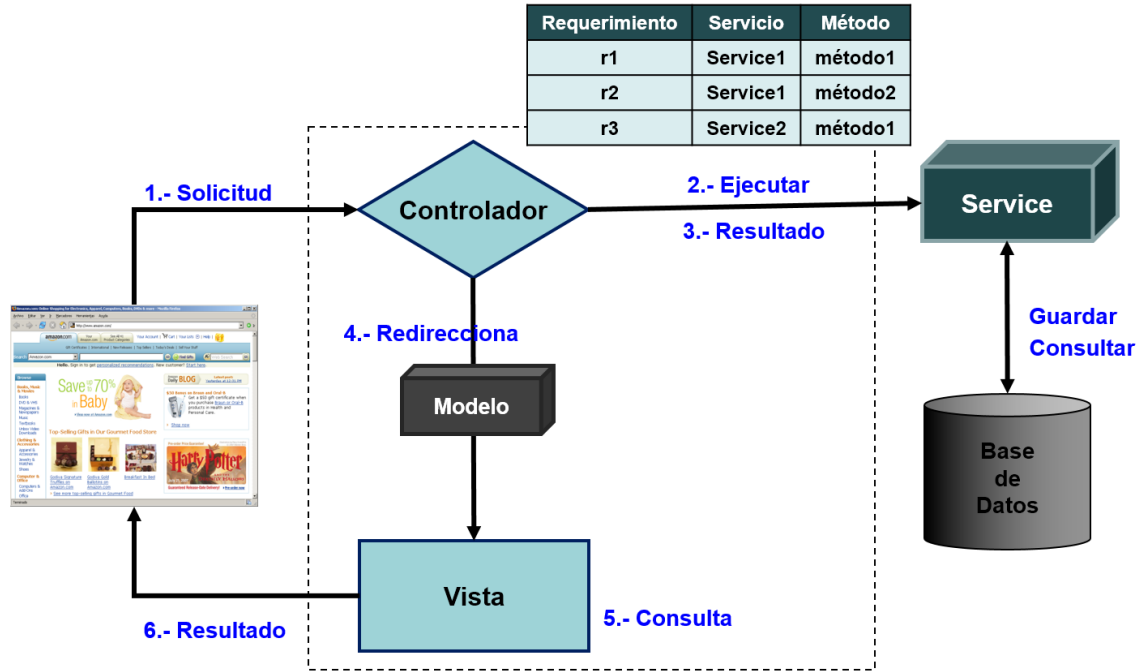


En esta sesión aprenderás a aplicar Spring MVC para el desarrollo de aplicaciones web.



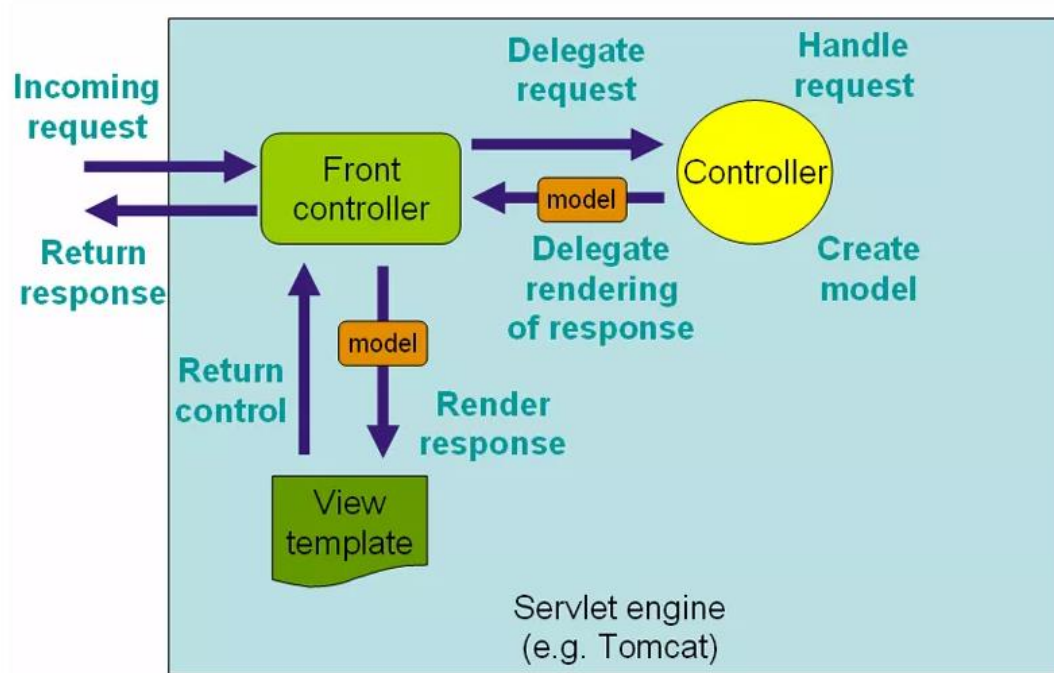


Fundamento



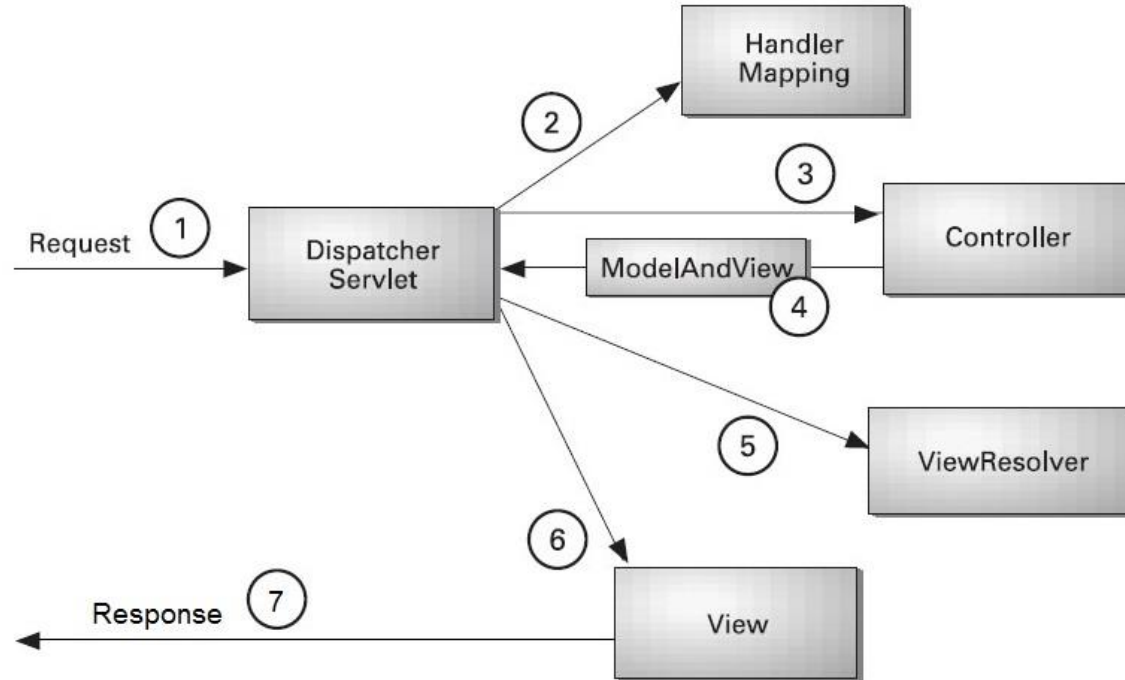


Arquitectura



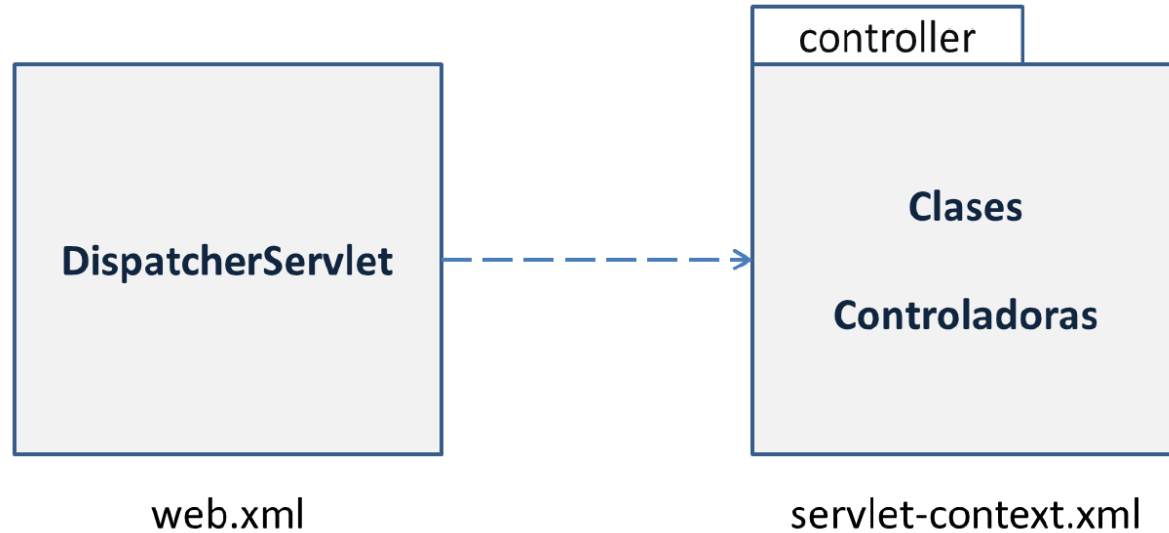


Arquitectura





Configuración - Contexto





Configuración - DispatcherServlet

```
<web-app . . . >

  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

</web-app>
```




Configuración - Servlet Context - ViewResolver



Nombre de la vista
↓
/WEB-INF/views/index.jsp
↑ ↑
Prefijo sufijo



Configuración - Servlet Context - ViewResolver

```
<beans:bean  
class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
  
    <beans:property name="prefix" value="/WEB-INF/views/" />  
    <beans:property name="suffix" value=".jsp" />  
  
</beans:bean>
```



Configuración - Servlet Context – Habilitar Anotaciones

```
<annotation-driven />
```

Configuración - Servlet Context – Clases Controladoras

```
<context:component-scan base-package="pe.egcc.demomvc.controller" />
```



Configuración - Servlet Context – Clase Controladora

```
package pe.egcc.demomvc.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

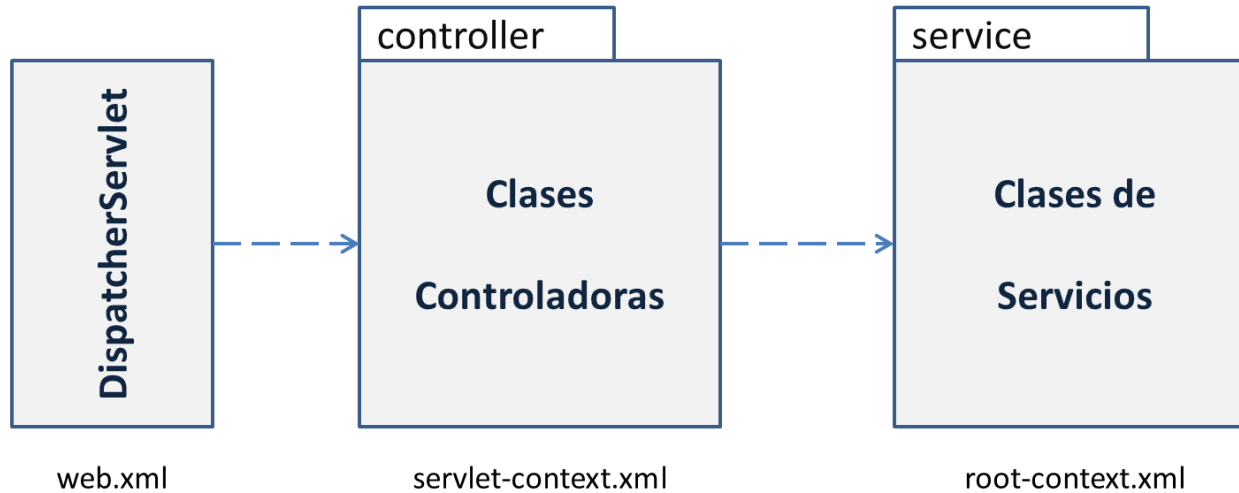
@Controller
public class HomeController {

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home( Model model ) {
        model.addAttribute("mensaje", "Hola GUSTAVO CORONEL." );
        return "home";
    }

}
```



Contexto de la Aplicación





HttpServletRequest

Un controlador de Spring MVC soporta como parámetro HttpServletRequest, de esta manera tienes acceso a por ejemplo los parámetros que recibe similar a como se hace en un Servlet.

```
@RequestMapping(value="procesarFactura.htm", method=RequestMethod.POST)
public String sumar(HttpServletRequest request, Model model){

}
```



HttpServletResponse

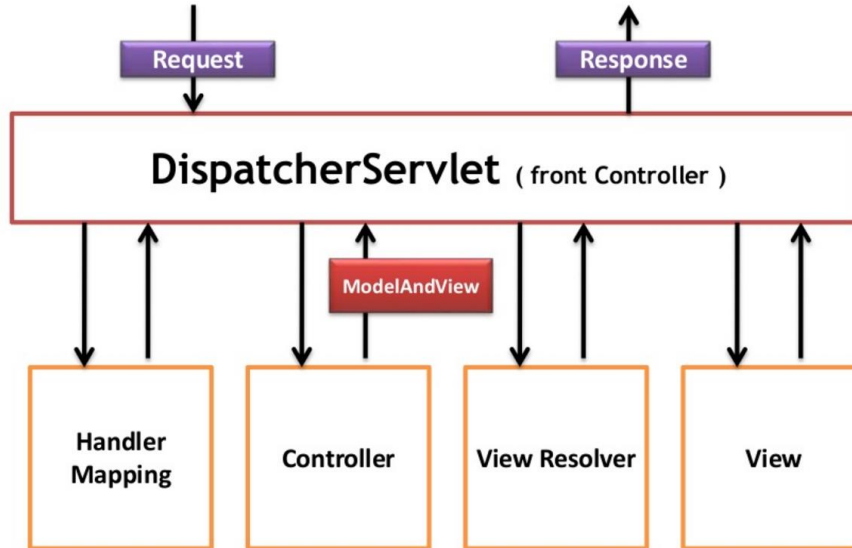
Un controlador de Spring MVC soporta como parámetro HttpServletResponse, de esta manera tienes acceso a generar una salida de manera directa hacia el navegador.

```
@RequestMapping(value="procesarFactura.htm", method=RequestMethod.POST)
public String sumar(HttpServletRequest request, HttpServletResponse response){

}
```



Componentes Model y View





Interface Model

Normalmente, esta interfaz se utiliza como parámetro de un controlador para comunicar el modelo de datos que se debe enviar a la vista.

```
@RequestMapping(value = "procesar.htm", method = RequestMethod.POST)
public String sumar(HttpServletRequest request, Model model) {

    . . .

    return "nombreVista";
}
```



Clase ModelAndView

Esta clase se debe utilizar cuando se quiere retornar el nombre del view y el modelo de datos como un solo objeto.

```
@RequestMapping(value = "procesar.htm", method = RequestMethod.POST)
public ModelAndView sumar(HttpServletRequest request) {

    ModelAndView mav = new ModelAndView("nombreVista");

    . . .

    return mav;
}
```