

## Práctica No. 2. Navegación con Spring

### Preparación del Entorno

- Abrir el entorno de desarrollo *Eclipse*
- Crear un nuevo proyecto web dinámico llamado *NavegandoControladores*
- Descargar de la página web [http://academicos.azc.uam.mx/jfg/pags/tarea\\_taller\\_web.html](http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html) los archivos del Servidor *Apache Tomcat*
- Instalar, agregar y arrancar el servidor

### Descarga de Spring

De la siguiente página:

<https://repo.spring.io/release/org/springframework/spring/5.0.0.RELEASE/>

El archivo *spring-framework-5.0.0.RELEASE-dist.zip* que contiene todas las bibliotecas necesarias para utilizar Spring. Al descomprimir el archivo, se tiene una gran cantidad de jars, solo se utilizan algunas para el desarrollo de la práctica.

Se debe contar con las bibliotecas de Standard Taglib <https://tomcat.apache.org/taglibs/standard/> dar clic en la versión Standard 1.2.3 en la liga de *descargas*.

Descargar los cuatro archivos *Jar*:

- taglibs-standard-impl-1.2.5.jar
- taglibs-standard-spec-1.2.5.jar
- taglibs-standard-jstlel-1.2.5.jar
- taglibs-standard-compat-1.2.5.jar

Es necesaria la biblioteca JSP Api de la siguiente dirección <https://mvnrepository.com/artifact/javax.servlet/jsp-api/2.0> dando clic en la liga *Download (JAR)*.

Para simplificar el proceso de selección de atributos, se puede descargar el archivo *bibliotecas.zip* de la página web [http://academicos.azc.uam.mx/jfg/pags/tarea\\_taller\\_web.html](http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html)

- Se copiarán todas las bibliotecas en *Nombre/WebContent/WEB-INF/lib* para que estén disponibles en el proyecto, en donde **Nombre** es el nombre del proyecto.
- Agregarlas al proyecto.
- Crear un directorio llamado *jsp* en *WebContent*

## Desplegando una página de Bienvenida

Crear un archivo JSP llamado *index.jsp* en el directorio *jsp*

### *index.jsp*

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>Bienvenido a Spring MVC</h1>
</body>
</html>
```

Se configurará para que arranque al correr la aplicación en el servidor, este se hace en el archivo *web.xml*. De esta forma, solo con marcar el nombre del proyecto y el botón de ejecutar, se levantará en el servidor.

### *web.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">
  <display-name>NavegandoControladores</display-name>
  <welcome-file-list>
    <welcome-file>jsp/index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Se configurará el archivo *web.xml* para indicar que hay un *servlet*, la clase de *Spring* a la que corresponde y el tipo de mapeo que manejará, en este caso serán cadenas que incluyan la extensión *html*.

### *web.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">
```

```

<display-name>NavegandoControladores</display-name>
<welcome-file-list>
  <welcome-file>jsp/index.jsp</welcome-file>
</welcome-file-list>

<servlet>
  <servlet-name>controlador</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>controlador</servlet-name>
  <url-pattern>*.html</url-pattern>
</servlet-mapping>

</web-app>

```

En este caso se indica que el nombre será controlador, por lo que hay que crear un archivo llamado **controlador-servlet.xml** y se creará en el directorio **WebContent/WEB-INF**

### controlador-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">
</beans>

```

En este momento solo se tiene la información base, falta colocar los datos específicos. Esta información se puede descargar en la página web [http://academicos.azc.uam.mx/jfg/pags/tarea\\_taller\\_web.html](http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html) en el archivo **base-servlet.xml**, recordando que se debe ajustar el nombre según lo especificado en el archivo **web.xml**.

Antes de configurar es necesario crear un paquete en donde se colocarán los Controladores, el nombre del paquete será: **uam.tdaw.controladores**, esto en el directorio **src**. Esta ruta (nombre de paquete) hay que indicar que será aquella en la que se buscarán de manera automática los controladores. De esta forma el archivo **controlador-servlet.xml** queda de la siguiente manera:

### controlador-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.springframework.org/schema/mvc
           http://www.springframework.org/schema/mvc/spring-mvc.xsd
           http://www.springframework.org/schema/context
           http://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="uam.tdaw.controladores" />

</beans>
```

También hay que configurar la manera en que se re direccionarán las vistas una vez que sean invocadas por el controlador.

### controlador-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.springframework.org/schema/mvc
           http://www.springframework.org/schema/mvc/spring-mvc.xsd
           http://www.springframework.org/schema/context
           http://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="uam.tdaw.controladores" />

    <bean id="viewResolver"
          class="org.springframework.web.servlet.view.UrlBasedViewResolver">
        <property name="viewClass"
                  value="org.springframework.web.servlet.view.JstlView" />
        <property name="prefix" value="/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>

</beans>
```

En la propiedad **prefix** se está indicando lo que se colocará antes de la palabra (nombre de la página) cuando se invoque una vista. En **suffix** se coloca lo que estará después. En este caso se está indicando que se colocará **/jsp/** (la ruta en donde están las vistas) y que todas son **.jsp**.

## Creación del Controlador

Se creará una clase llamada `ControladorSaludoA` en el paquete ***uam.tdaw.controladores***, arriba del nombre de la clase, se agregará la etiqueta `@Controller` que indica a *Spring* que se trata de un controlador.

### ControladorSaludoA.java

```
package uam.tdaw.controladores;

import org.springframework.stereotype.Controller;

@Controller
public class ControladorSaludoA {

}
```

Se debe importar la clase correspondiente `org.springframework.stereotype.Controller`

Se creará un método que regresa un tipo de dato `ModelAndView` que es el que invoca a la vista siguiente, a este método se le coloca la etiqueta `@RequestMapping` cuyo parámetro atraparé la llamada a una vista en particular, en este caso ***saludameControl***.

### ControladorSaludoA.java

```
package uam.tdaw.controladores;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ControladorSaludoA {

    @RequestMapping("saludameControl")
    public ModelAndView saludandoControlA() {

        System.out.println("LLEGUÉ AL CONTROLADOR
ControladorSaludoA!!");
        return new ModelAndView("bienvenidoControlA");

    }

}
```

El valor que se regresa corresponde a una vista, en este caso llamada `bienvenidoControlA`.

Para probar la navegación, se modificará el archivo `index.jsp` colocando una liga hacia una página, en este caso el nombre será la del parámetro `@RequestMapping`

### `index.jsp`

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>Bienvenido a Spring MVC</h1>

<a href="saludameControl.html">Si me das clic te saludará el
Controlador</a>

</body>
</html>
```

Se debe crear también una página (JSP) que corresponda al nombre de retorno del método creado, en este caso `bienvenidoControlA`

### `bienvenidoControlA.jsp`

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>HOLA! Soy el Control A</h1>
</body>
</html>
```

Es posible agregar otro método que redirija a una página diferente, en este caso el método `saludandoControlB`, mapeado a interceptar las peticiones a `saludameControlB`

### ControladorSaludoA.java

```
package uam.tdaw.controladores;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ControladorSaludoA {

    @RequestMapping("saludameControl")
    public ModelAndView saludandoControlA() {
        System.out.println("LLEGUÉ AL CONTROLADOR
ControladorSaludoA!!");
        return new ModelAndView("bienvenidoControlA");
    }

    @RequestMapping("saludameControlB")
    public ModelAndView saludandoControlB() {

        System.out.println("LLEGUÉ AL CONTROLADOR ControladorSaludoA!!
pero a otro método");
        return new ModelAndView("bienvenidoControlB");
    }

}
```

Se redirige a `saludameControlB` que es otro archivo JSP, en este caso con el nombre `bienvenidoControlB`

### saludameControlB.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>HOLA! Soy el Control B</h1>
</body>
</html>
```

También es posible crear otra clase controladora con sus propios métodos, sin embargo una misma dirección no puede ser “*atrapada*” por dos métodos diferentes sin importar que estén en distintas clases. Se creará la clase **ControladorSaludoB** en el método **uam.tdaw.controladores**

### ControladorSaludoB.java

```
package uam.tdaw.controladores;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ControladorSaludoB {

    @RequestMapping("saludameControlC")
    public ModelAndView saludandoControlA() {
        System.out.println("LLEGUÉ AL CONTROLADOR
ControladorSaludoB!!");
        return new ModelAndView("bienvenidoControlA");
    }
}
```

Se agregará una nueva liga al archivo **index.jsp**

### index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>Bienvenido a Spring MVC</h1>

<a href="saludameControl.html">Si me das clic te saludará el
Controlador</a>
<br><br>
<a href="saludameControlB.html">Si me das clic te saludará el Controlador
B</a>
<br><br>
<a href="saludameControlC.html">Si me das clic te saludará el Controlador
de Otra Clase</a>
</body>
</html>
```



En este caso se invoca a una página *JSP* ya existente, pero podría crearse una nueva para desplegar.

Todas las ligas se han colocado en el archivo *index.jsp*, pero también es posible colocarlas en cualquier página *JSP*.

En este caso se colocará una liga en el archivo *bienvenidoControlA.jsp* que será atrapado por un método mapeado a *saludameControlD* que se creará en la clase *ControladorSaludoB* el cuál regresará a una página creada en un sub directorio (*otro*) que está en el directorio *jsp* (*/otro/bienvenidoControlC*).

### **bienvenidoControlA.jsp**

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>HOLA! Soy el Control A</h1>

<a href="saludameControlD.html">Si me das clic te saludará el Controlador
de Otra Clase</a>

</body>
</html>
```

### **ControladorSaludoB.java**

```
package uam.tdaw.controladores;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ControladorSaludoB {

    @RequestMapping("saludameControlC")
    public ModelAndView saludandoControlA() {

        System.out.println("LLEGUÉ AL CONTROLADOR
ControladorSaludoB!!");
        return new ModelAndView("bienvenidoControlA");
    }
}
```

```

@RequestMapping("saludameControlD")
public ModelAndView saludandoControlB() {
    System.out.println("LLEGUÉ AL CONTROLADOR
ControladorSaludoB!!");
    return new ModelAndView("/otro/bienvenidoControlC");
}
}

```

Crear el directorio *otro* en *jsp*, y crear el archivo **bienvenidoControlC.jsp**

### **bienvenidoControlC.jsp**

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>HOLA! Soy el Control C y estoy un un sub directorio</h1>
</body>
</html>

```