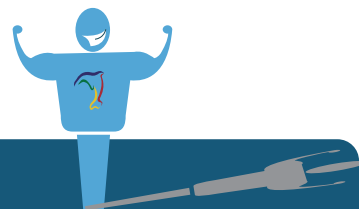


¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

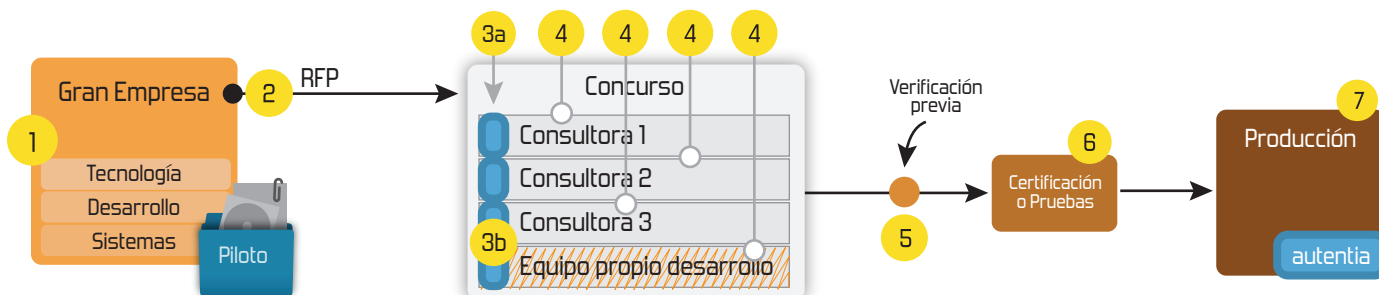
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

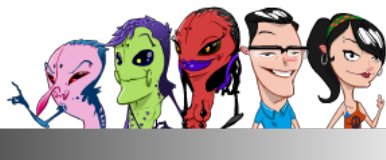
JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)


AdictosAlTrabajo

Final de Terrakas
¡¡No te lo pierdas!!
terrakas.com



autentia
Soporte a desarrollo informático

Hosting patrocinado por
enREDados

Entra en Adictos a través de 

E-mail

Contraseña

Entrar

[Deseo registrarme](#)
[Olvidé mi contraseña](#)



[Inicio](#)

[Quiénes somos](#)

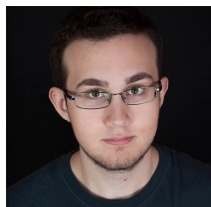
[Formación](#)

[Comparador de salarios](#)

[Nuestros libros](#)

[Más](#)

» Estás en: [Inicio](#) [Tutoriales](#) [Hola Mundo con Spring 3 MVC](#)



Daniel Diaz Suarez

Desarrollador Web en Autentia

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

[Ver todos los tutoriales del autor](#)

Fecha de publicación del tutorial: 2013-07-05

Tutorial visitado 7 veces [Descargar en PDF](#)

Hola Mundo con Spring 3 MVC

0. Índice de contenidos.

- [1. Entorno](#)
- [2. Introducción](#)
- [3. Preparación del entorno](#)
- [4. Entendiendo el esqueleto de la aplicación](#)
- [5. Probando la aplicación en un servidor local](#)
- [6. Conclusiones](#)

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.4 GHz Intel Core i7, 8GB DDR3 SDRAM)
- Mac OS X Lion 10.7.5
- Eclipse Juno

2. Introducción

Este tutorial es el primero de una serie de tutoriales prácticos que repasaran las principales posibilidades de Spring

En este tutorial vamos a ver paso por paso como crear nuestra primera aplicación de Spring y veremos las distintas partes que la forman, para ello vamos a utilizar una serie de herramientas Open Source que nos permiten crear rápidamente el esqueleto de lo que será nuestra aplicación.

Las herramientas que vamos a utilizar en el tutorial son las siguientes:

- [Eclipse Juno para desarrolladores Java EE](#)
- [Maven 3 a través del plugin para Eclipse](#)
- [Spring Source Tool Suite](#)
- [Apache Tomcat 7](#)

3. Preparación del entorno

Lo primero de todo es descargarnos la versión Juno de Eclipse, una vez esté descargada ejecutamos Eclipse y descargamos el las Spring Source Tool Suite (a partir de ahora SSTs).

Tan solo hay que ir a la pestaña de Help (Ayuda) y entrar en el Eclipse Market.

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» [Comentando el libro: Agile Management de Angel Medinilla](#)

» [Final de temporada de Terrakas Autentia freaklances](#)

» [Atención, APLAZADO Estreno último capítulo de Terrakas](#)

» [Vendedor: Soy inseguro, filtra o elige por mi: si quieres que te compre.](#)

» [Comentando el libro: El arte de pensar, de Rolf Dobelli](#)

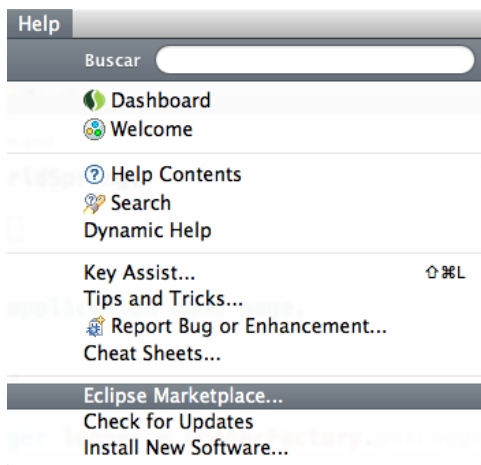
[Histórico de noticias](#)

Últimos Tutoriales

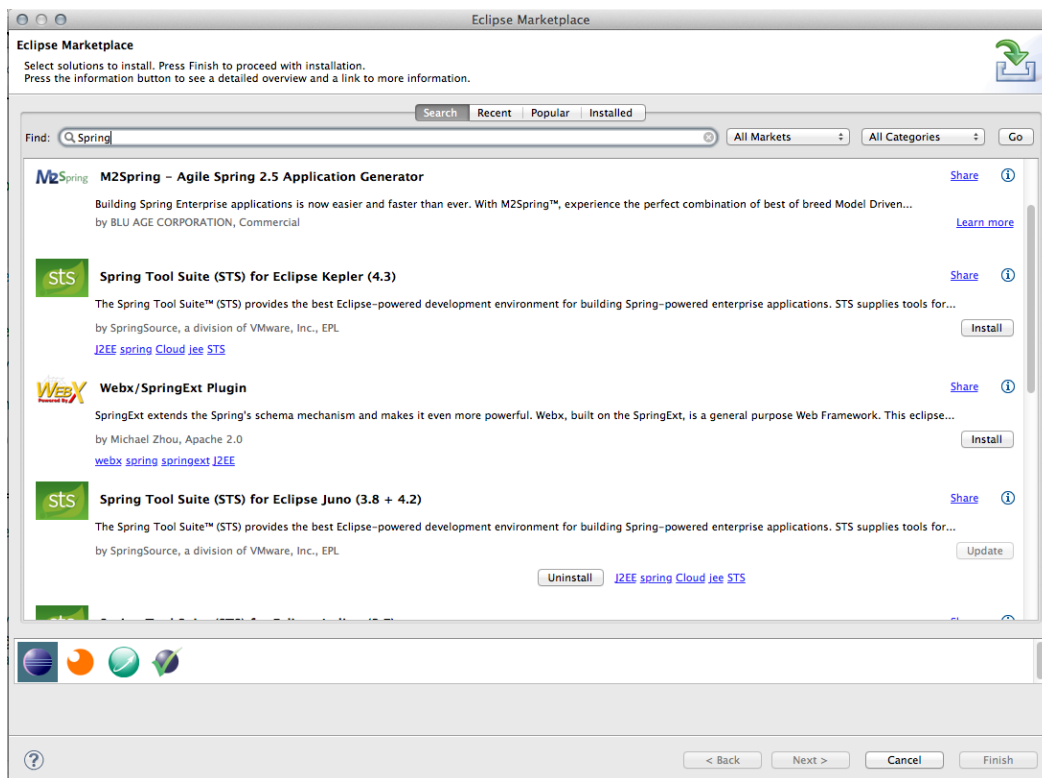
» [Ejecución de tests unitarios con junit en proyectos ant y su integración en jenkins y sonar para medir la cobertura.](#)

» [Comentando la AOS 2013](#)

» [Soporte en Sonar para un proyecto multi-lenguaje:](#)



Buscamos por Spring y descargamos la versión de SSTS para Juno (en el momento de escribir este tutorial ya está fuera la siguiente versión de Eclipse llamada Kepler, aún así este tutorial se basa en la versión más estable Juno)



Una vez hayamos instalado todo y reiniciado Eclipse, creamos un nuevo "Spring Template Project"

[configuración desde Jenkins](#)

» [Integración de Selenium Grid con Jenkins](#)

» [Generación de informes de proyectos ágiles en TestLink](#)

Últimos Tutoriales del Autor

» [Integración de Selenium Grid con Jenkins](#)

» [TDD, BDD & Test de aceptación](#)

» [Haciendo BDD con Cucumber](#)

» [Como testear aplicaciones en Ember.js](#)

» [Introducción a Require.JS](#)

Últimas ofertas de empleo

2011-09-08

[Comercial - Ventas - MADRID.](#)

2011-09-03

[Comercial - Ventas - VALENCIA.](#)

2011-08-19

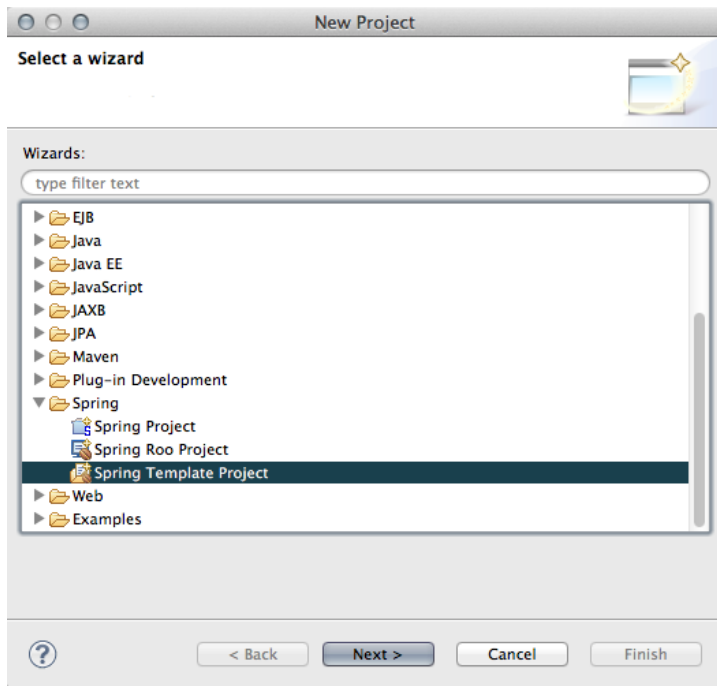
[Comercial - Compras - ALICANTE.](#)

2011-07-12

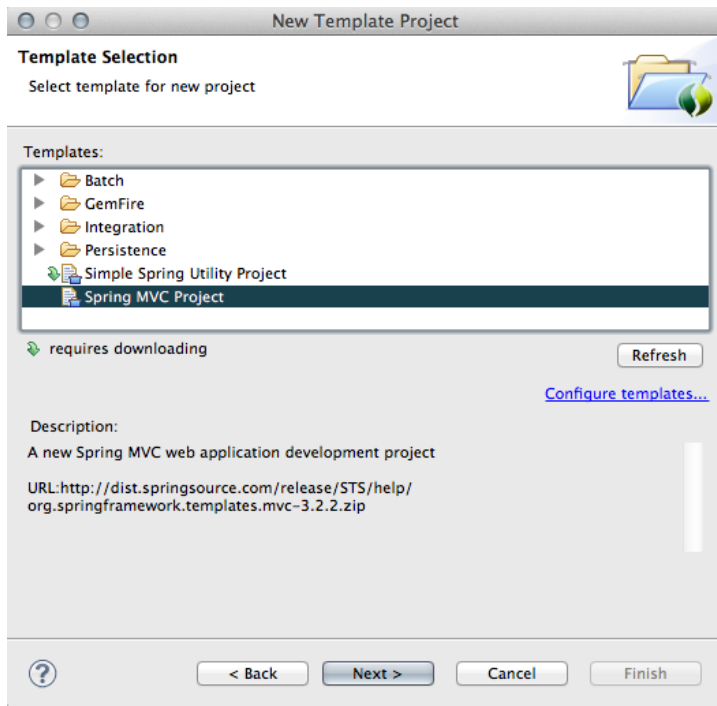
[Otras Sin catalogar - MADRID.](#)

2011-07-06

[Otras Sin catalogar - LUGO.](#)

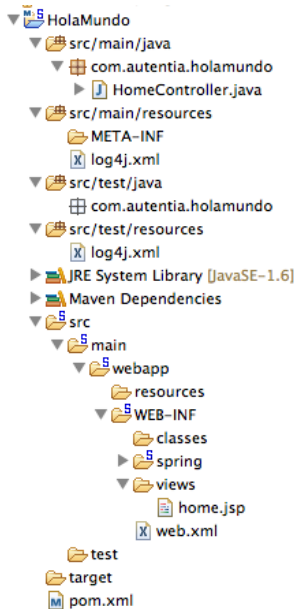


En la siguiente opción seleccionamos "Spring MVC Project", nos dir  que tiene que descargar X datos, le decimos que si y esperamos a que se descargue.



4. Entendiendo el esqueleto de la aplicaci n

Una vez est  descargado repasamos las carpetas y archivos que nos ha instalado:



Lo primero que vamos a mirar es el pom.xml, este es el archivo central de Maven, este archivo contiene Meta-Información del proyecto, como el nombre, la organización y lo mas importante, las dependencias, además también podemos añadir plugins para Maven, como por ejemplo el plugin de Eclipse, que permite crear proyectos para Eclipse a través de Maven.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.autentia</groupId>
  <artifactId>holamundo</artifactId>
  <name>Ho laMundo</name>
  <packaging>war</packaging>
  <version>1.0.0-BUILD-SNAPSHOT</version>
  <properties>
    <java-version>1.6</java-version>
    <org.springframework-version>3.1.1.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
  </properties>
  <dependencies>
    <!-- Spring -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${org.springframework-version}</version>
      <exclusions>
        <!-- Exclude Commons Logging in favor of SLF4j -->
        <exclusion>
          <groupId>commons-logging</groupId>
          <artifactId>commons-logging</artifactId>
        </exclusion>
      </exclusions>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${org.springframework-version}</version>
    </dependency>

    <!-- AspectJ -->
    <dependency>
      <groupId>org.aspectj</groupId>
      <artifactId>aspectjrt</artifactId>
      <version>${org.aspectj-version}</version>
    </dependency>

    <!-- Logging -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>${org.slf4j-version}</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>jcl-over-slf4j</artifactId>
      <version>${org.slf4j-version}</version>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
</project>
```

Maven además de todo esto nos permite hacer muchas más cosas, pero eso esta fuera del objetivo de este tutorial.

El siguiente paso es fijarnos en la estructura de carpetas que nos ha creado, vamos a detenernos en estas cuatro carpetas:

- ▶ src/main/java
- ▶ src/main/resources
- ▶ src/test/java
- ▶ src/test/resources

La primera es donde irán todas nuestras clases Java que formarán nuestra aplicación, como estructuramos esas clases ya depende mas de cada uno.

La siguiente carpeta que vemos es src/main/resources, en está carpeta se encuentran todos los recursos de configuración que requiera nuestra aplicación, en este momento solo tenemos el archivo de configuración del modulo de loggin log4j.xml

Las siguientes dos carpetas son las que conforman la parte de testing de nuestra aplicación, en `src/test/java` tendremos las clases que serán los test unitarios, y en `src/test/resources` tendremos los recursos de configuración que necesitemos para que nuestra aplicación pueda ser testeada.

Como podemos observar SSRS nos ha creado una clase Java llamada `HomeController`:

```
/**
 * Simply selects the home view to render by returning its name.
 */
@RequestMapping(value = "/", method = RequestMethod.GET)
public String home(Locale locale, Model model) {
    logger.info("Welcome home! The client locale is {}.", locale);

    Date date = new Date();
    DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG, DateFormat.LONG, locale);

    String formattedDate = dateFormat.format(date);

    model.addAttribute("serverTime", formattedDate );

    return "home";
}
```

Este método atenderá a las peticiones de tipo GET que se hagan en `localhost:8080/HolaMundo/` (dando por hecho que estamos ejecutando este código en nuestro ordenador local bajo Apache Tomcat), y responde ejecutando el archivo JSP llamado `home` que se encuentra en `src/main/webApp/views` (La carpeta que contiene las vistas se declara en la configuración de Spring)

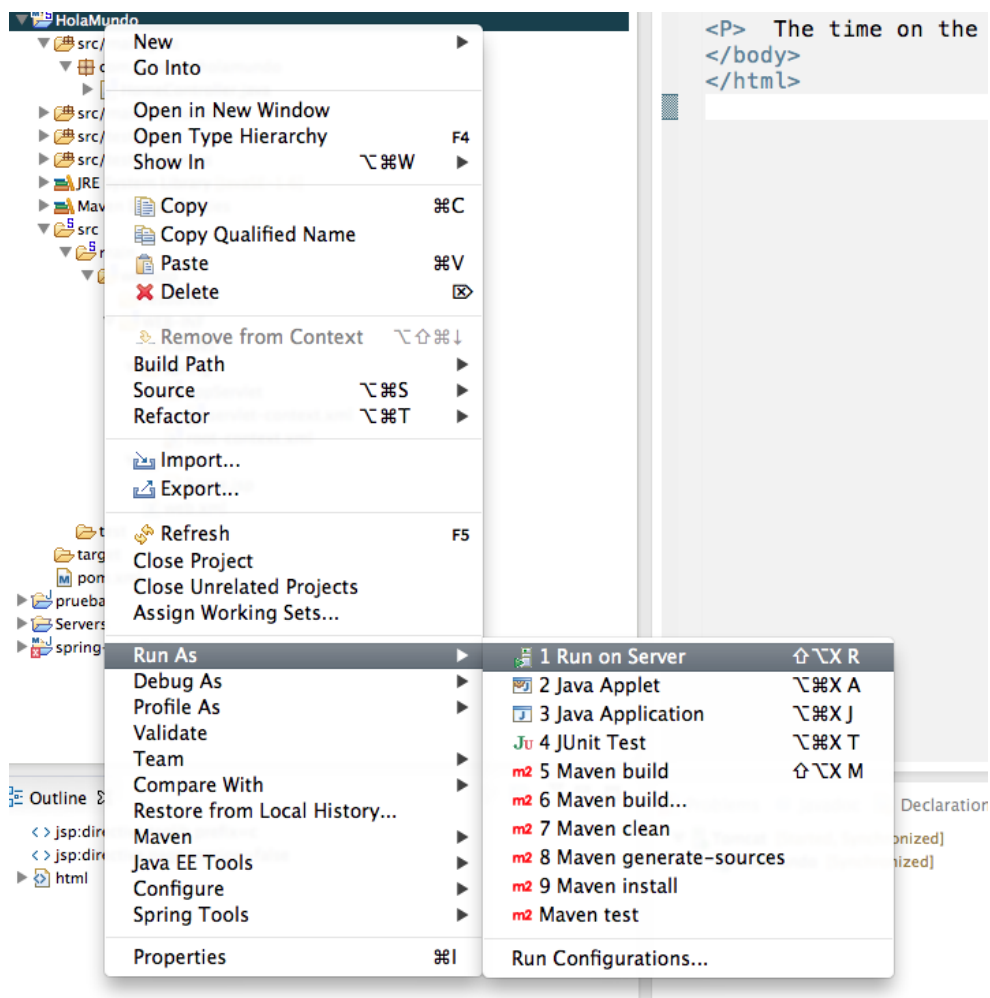
Si abrimos el archivo `home.jsp` podemos ver los contenidos:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>
    Hello world!
</h1>

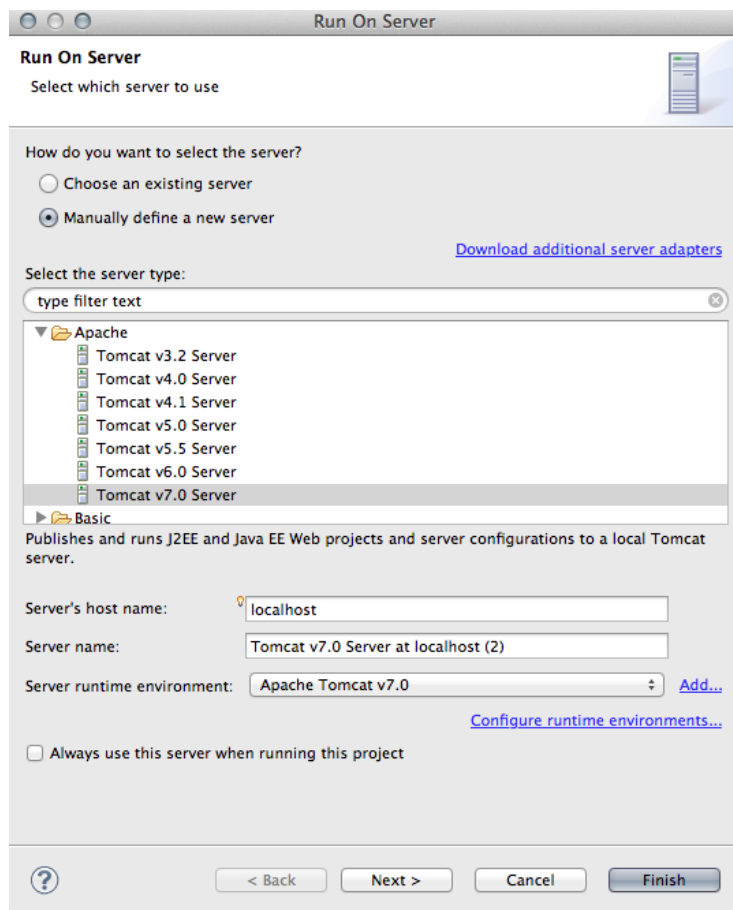
<P> The time on the server is ${serverTime}. </P>
</body>
</html>
```

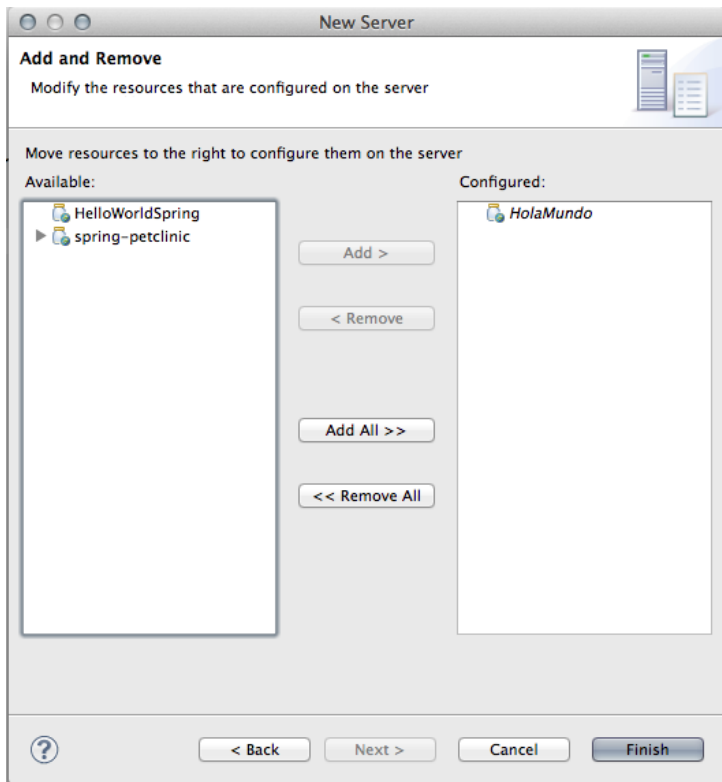
5. Probando la aplicación en un servidor local

Para probar la aplicación en un servidor local (en este caso Apache Tomcat 7), hacemos click con el botón derecho sobre el nombre de nuestro proyecto, Run As -> Run on server .



Seleccionamos la opción de crear un nuevo servidor y le añadimos nuestro proyecto:





Una vez le damos a finalizar Eclipse arrancará el servidor y momentos después el navegador interno mostrándonos el siguiente mensaje (con la fecha actual):



6. Conclusiones

Hemos podido comprobar lo fácil que es crear una aplicación de Hola Mundo con Spring y ejecutarla, gracias a las SSRS podemos crear proyectos de Spring ya preparados para empezar a añadirle nuestras clases, lo cual nos ahorra bastante trabajo repetitivo.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) | [0](#)

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA Impulsores Comunidad ¿Ayuda?

---- 0 personas han traído clicks a esta página
sin clicks + + + + + + + +

powered by [karmacracy](#)

Copyright 2003-2013 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

