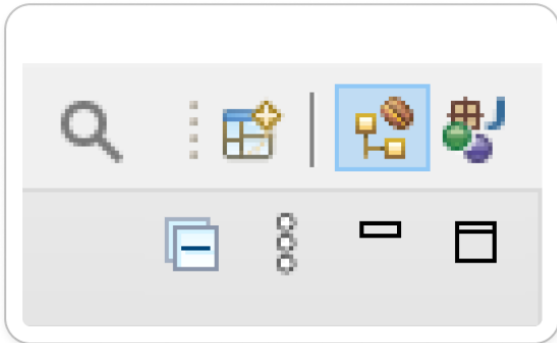
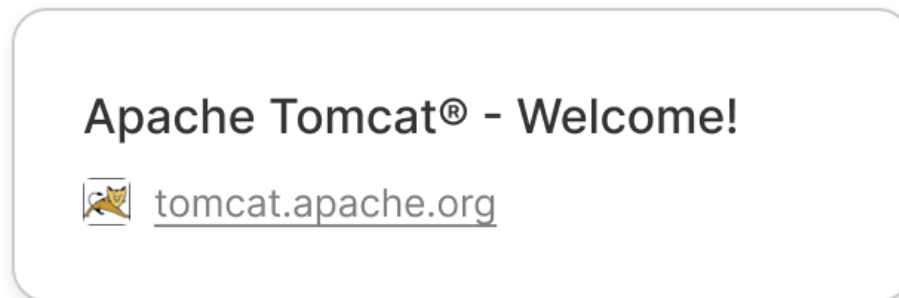


SPRING SET UP – pasos a seguir

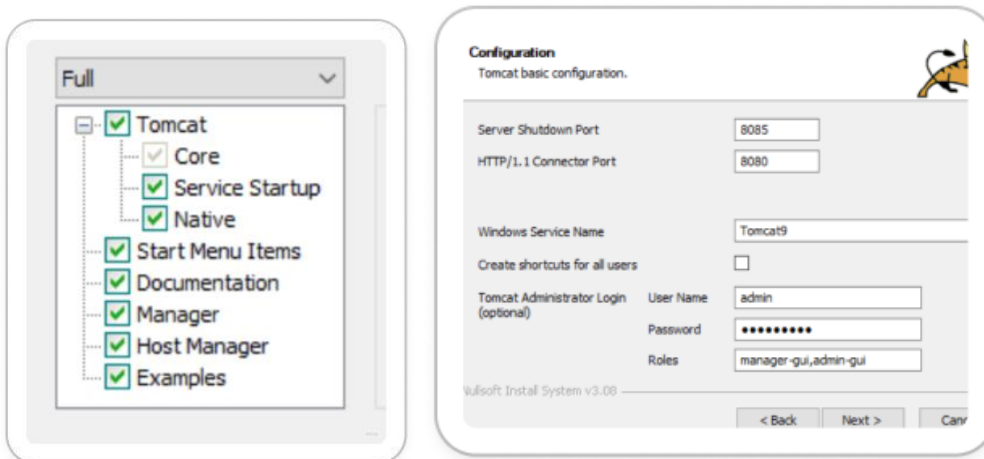
En Eclipse: activamos Java EE.



Instalamos Apache Tomcat (ver. 9.0).

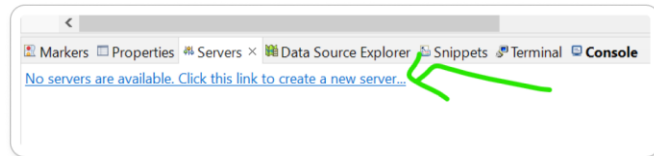
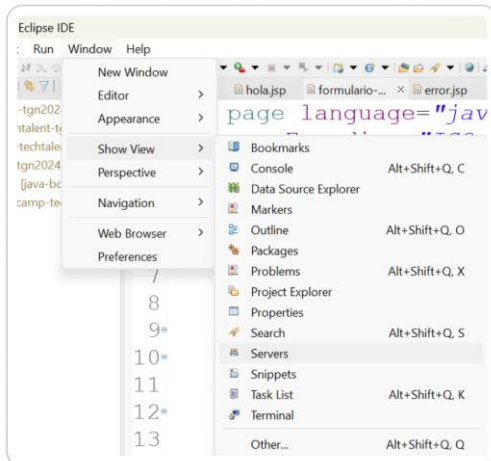


Es importante seleccionar todas las casillas. Seleccionamos un usuario y contraseña (admin/techtalent por ejemplo) y entonces se instala.

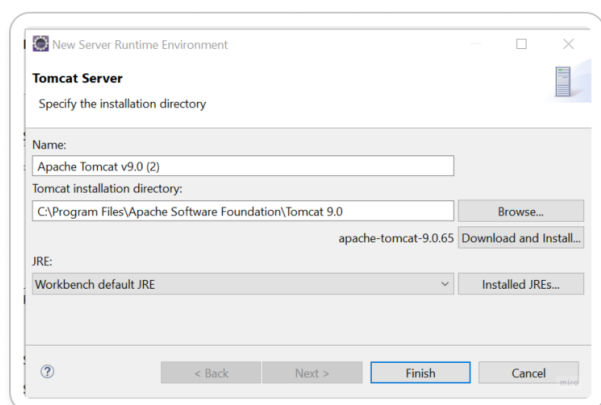
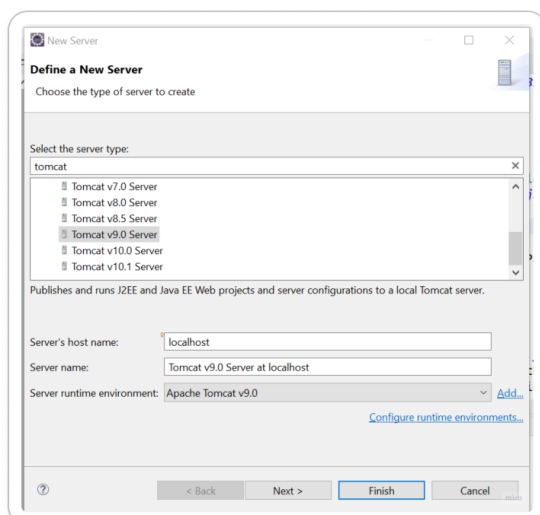


[!] importante: NO seleccionar las casillas ReadMe y Run.

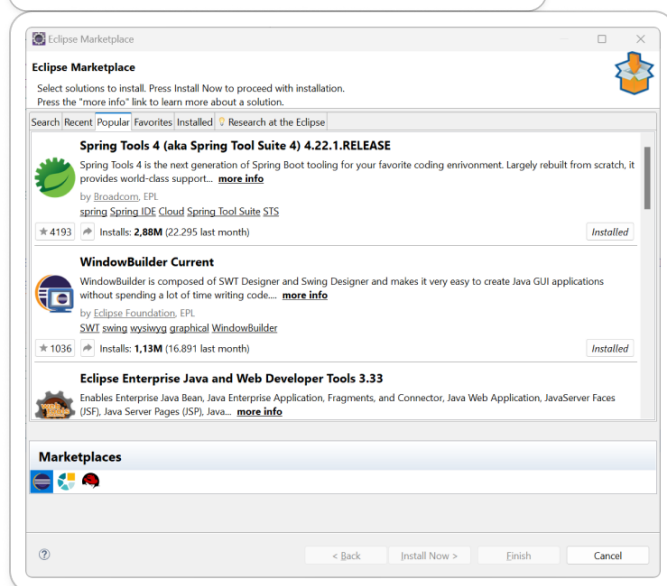
EN ECLIPSE: [carpeta proyecto](#)



Creamos *Nuevo servidor* y aparecerá la siguiente ventana. Seleccionamos la ver. 9.0 de Tomcat.



Y por último instalamos el primero que aparece en el pantallazo de la izquierda.



Para crear la página web generamos un proyecto Maven nuevo (proyecto simple). Nombramos al grupo *com.example*, al artifact *spring-mvc-pepito* y seleccionamos el packaging como *war*.

CONFIGURACIÓN:

Actualización de las dependencias del *pom.xml*

```
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>spring-mvc-demo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <dependencies>
    <!-- Spring MVC Dependency -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.3.20</version>
    </dependency>

    <!-- JSTL Dependency -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>

    <!-- Servlet API Dependency -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.2.3</version>
        <configuration>
          <failOnMissingWebXml>false</failOnMissingWebXml>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Creamos el siguiente archivo *web.xml* en *src/main/webapp/WEB-INF/web.xml*

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
version="3.1">

<display-name>Spring MVC Demo</display-name>

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

<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping> ...
```

Generamos archivo *spring-mvc-config.xml* en *src/main/webapp/WEB-INF/spring-mvc-config.xml*

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

  <!-- Configuración de escaneo de componentes -->
  <context:component-scan base-package="com.example.mvc" />

  <!-- Configuración de recursos estáticos -->
  <mvc:resources mapping="/resources/**" location="/resources/" />

  <!-- Soporte para anotaciones en Spring -->
  <mvc:annotation-driven />

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

También el controlador *HomeController.java* en `src/main/java/com/example/mvc`

```
package com.example.mvc;

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

@Controller
public class HomeController {

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String showHomePage() {
        return "hello-world";
    }
}
```

Creamos la vista JSP creando el archivo *hello-world.jsp* en `src/main/webapp/WEB-INF/views/` añadiendo lo siguiente:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Hello World</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

Vinculamos el proyecto `spring-mvc-demo` a TomCat.

