



Clase 1 Continuacion

Instalacion

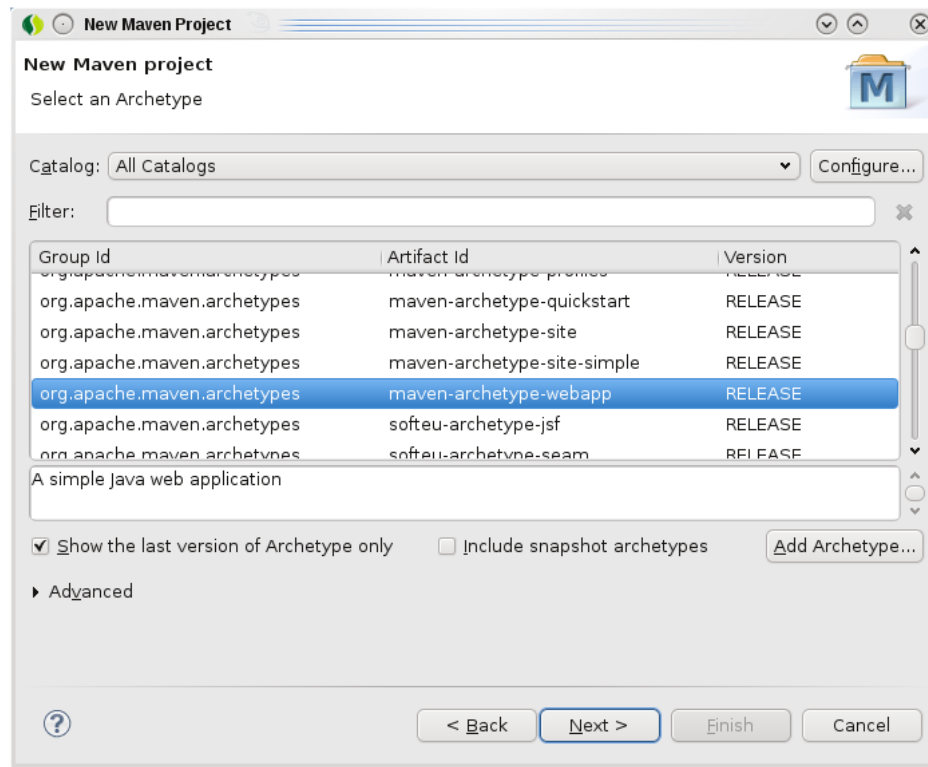
Para empezar necesitamos instalar:

- Java 8.
- Apache Maven.
- IDE IntelliJ o Eclipse.
- Tomcat 8



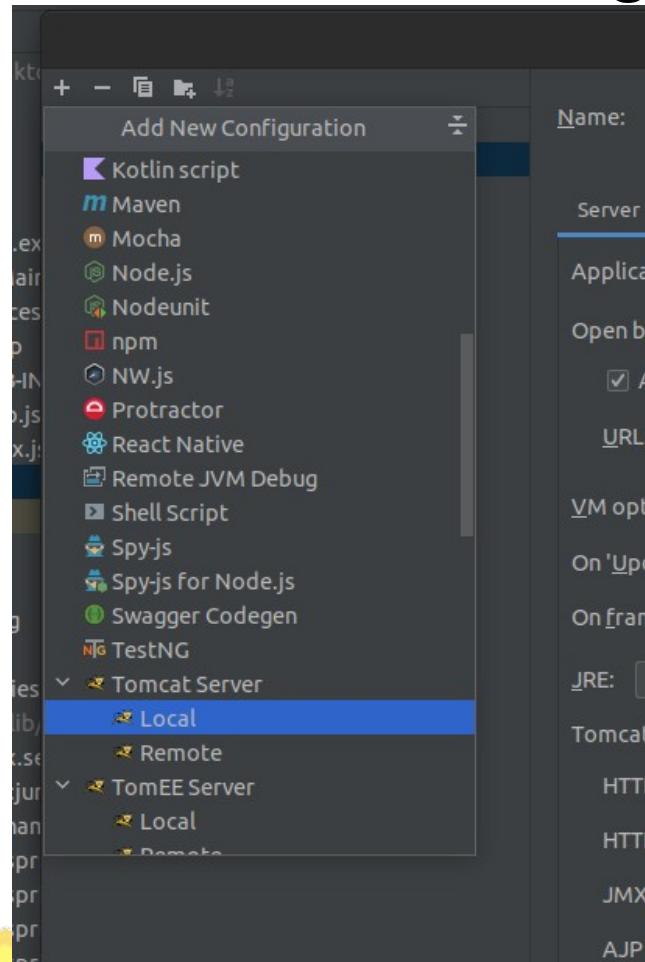
Instalacion

Creamos una aplicacion Maven



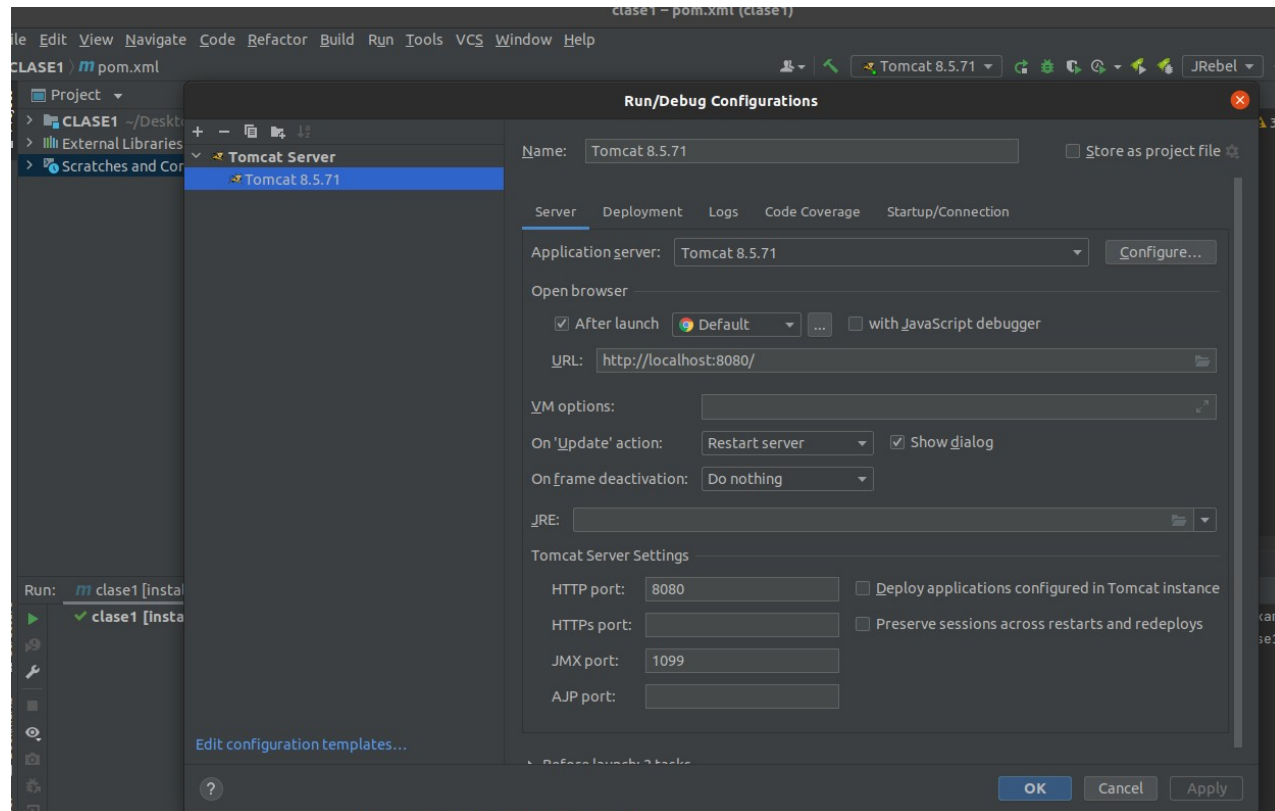
Instalacion

Agregamos una nueva Run Config usando Tomcat



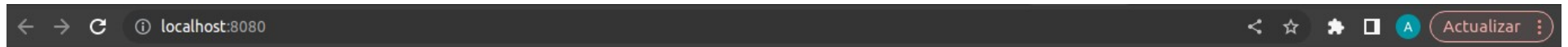
Instalacion

Seteamos la Run config apuntando al tomcat



Instalacion

Chequeamos que la aplicacion levanta



Hello World!



Instalacion

Agregamos dependencias de Spring en el pom.xml

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${org.springframework.version}</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${org.springframework.version}</version>
</dependency>

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>provided</scope>
</dependency>
```

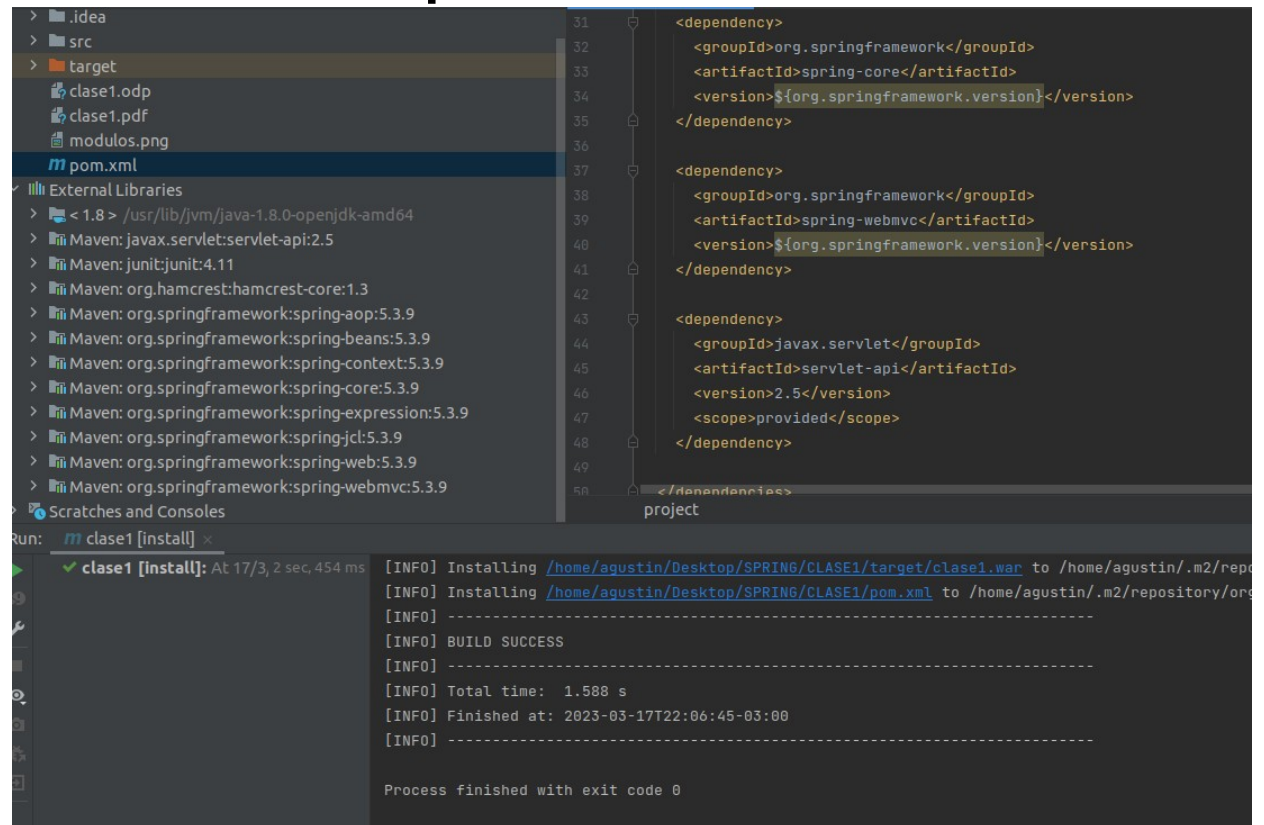
Y dentro de las properties , la version de Spring que vamos a utilizar

```
<properties>
  <org.springframework.version>5.3.9</org.springframework.version>
</properties>
```



Instalacion

Luego de correr el Build de Maven, deberiamos notar las dependencias descargadas



The screenshot displays an IDE interface with three main panels. The left panel shows a project structure with folders like .idea, src, and target, and files like class1.odp, class1.pdf, and modules.png. The middle panel lists external libraries, including Java 1.8.0, Maven, and various Spring Framework artifacts. The right panel shows the pom.xml file with three dependency entries for spring-core, spring-webmvc, and javax.servlet-api. The bottom panel shows the Maven build output, indicating a successful installation of the project.

```
> .idea
> src
> target
  class1.odp
  class1.pdf
  modules.png
  pom.xml
External Libraries
  < 1.8 > /usr/lib/jvm/java-1.8.0-openjdk-amd64
  Maven: javax.servlet:servlet-api:2.5
  Maven: junit:junit:4.11
  Maven: org.hamcrest:hamcrest-core:1.3
  Maven: org.springframework:spring-aop:5.3.9
  Maven: org.springframework:spring-beans:5.3.9
  Maven: org.springframework:spring-context:5.3.9
  Maven: org.springframework:spring-core:5.3.9
  Maven: org.springframework:spring-expression:5.3.9
  Maven: org.springframework:spring-jcl:5.3.9
  Maven: org.springframework:spring-web:5.3.9
  Maven: org.springframework:spring-webmvc:5.3.9
Scratches and Consoles
Run: m class1 [install] x
  class1 [install]: At 17/3, 2 sec, 454 ms
    [INFO] Installing /home/agustin/Desktop/SPRING/CLASE1/target/class1.war to /home/agustin/.m2/repository/org/springframework/spring-webmvc/5.3.9/class1.war
    [INFO] Installing /home/agustin/Desktop/SPRING/CLASE1/pom.xml to /home/agustin/.m2/repository/org/springframework/spring-webmvc/5.3.9/pom.xml
    [INFO] -----
    [INFO] BUILD SUCCESS
    [INFO] -----
    [INFO] Total time: 1.588 s
    [INFO] Finished at: 2023-03-17T22:06:45-03:00
    [INFO] -----
    Process finished with exit code 0
```


Instalacion

Luego vamos a modificar el `web.xml` en el directorio '`src/main/webapp/WEB-INF`'

Vamos a definir un **DispatcherServlet** (tambien llamado 'Controlador Frontal'). Su mision sera controlar hacia donde seran enrutadas todas nuestras solicitudes.

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

<servlet-mapping>
  <servlet-name>springapp</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```



Instalacion

Luego vamos crear un archivo de configuracion de Spring llamado **app-config.xml** y lo vamos a guardar en **/WEB-INF/spring**

```
<?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">

  <!-- Scans the classpath of this application for @Components to deploy as beans -->
  <context:component-scan base-package="com.example.springapp.web" />

  <!-- Configures the @Controller programming model -->
  <mvc:annotation-driven/>

</beans>
```



Instalacion

Bueno, ahora creamos los resource packages llamados **src/main/java** y **src/main/resources**

En src/main/java creamos el paquete **com.example.springapp.web.controller** y alli es donde almacenaremos nuestros controllers. En este caso agregaremos MainController.java

```
package com.example.springapp.web.controller;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class MainController {

    protected final Log logger = LogFactory.getLog(getClass());

    @RequestMapping(value="/hello")
    public ModelAndView handleRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        logger.info("Returning hello view");

        return new ModelAndView("hello.jsp");
    }
}
```

Instalacion

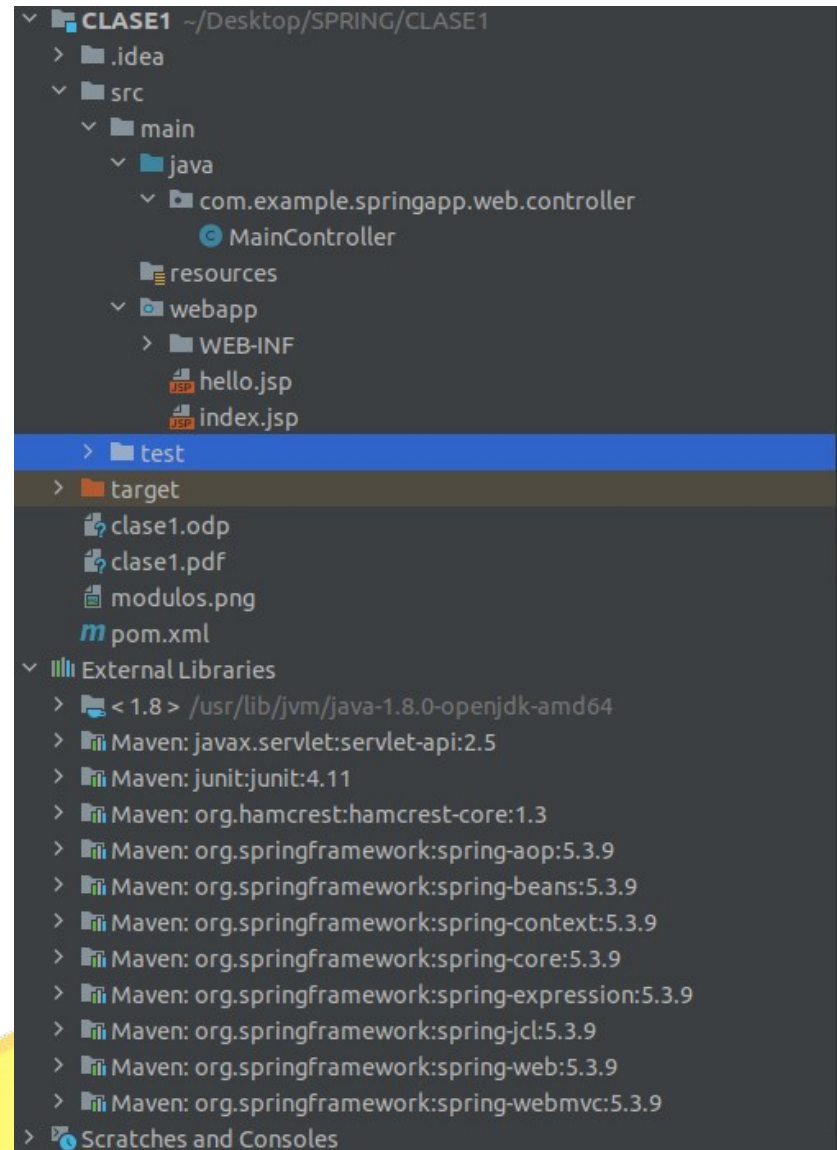
Por ultimo deberiamos agregar nuestra Vista **hello.jsp** , a la altura de nuestro index.jsp

```
<%--  
  Created by IntelliJ IDEA.  
  User: agustin  
  Date: 17/3/23  
  Time: 22:48  
  To change this template use File | Settings | File Templates.  
--%>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head><title>Hello :: Spring Application</title></head>  
<body>  
<h1>Hello - Spring Application</h1>  
<p>Greetings.</p>  
</body>  
</html>
```



Instalacion

La estructura de directorios y archivos deberia verse de la siguiente manera



Instalacion

Una vez finalizado, buildidiamos y deployamos la aplicacion y deberiamos poder acceder a nuestro nuevo Endpoint /hello



Hello - Spring Application

Greetings.

