

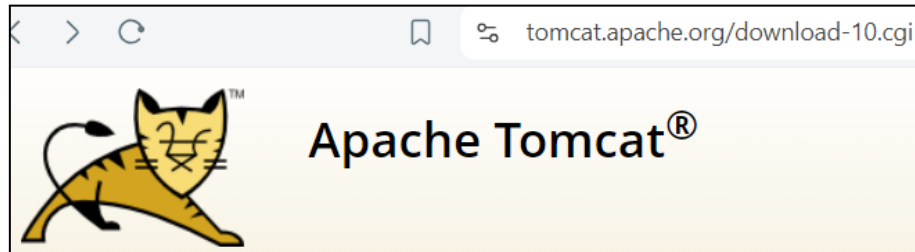


Programación con Java

TAREA 23

Índice.....	2
Introducción.....	3
Compilación y ejecución de un programa en Java.....	4
Webgrafía.....	5

En esta práctica instalaremos tomcat V.9 o V.10 para crear un proyecto en maven en el que añadiremos las dependencias de Spring y crearemos una página web. Dicha instalación la haremos a través de la web oficial de apache:



Para utilizar tomcat deberemos definir el plugin correspondiente en el archivo pom.xml

```
<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <version>2.2</version>
  <configuration>
    <port>8080</port>
    <path>/</path>
  </configuration>
</plugin>
```

Al tener Maven y Tomcat en nuestra máquina podemos empezar creando un proyecto maven en el que tendremos que cambiar el archivo pom con el plugin comentado antes y tres dependencias para la utilización de Spring.

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>6.1.8</version>
  </dependency>
  <dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
    <version>3.0.0</version>
  </dependency>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

Crearemos y definiremos el archivo de “despliegue descriptor” llamado web.xml, este archivo contiene información sobre cómo debe comportarse el contenedor servlet al ejecutar la aplicación web.

```
<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 Junior TGN</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>
</web-app>
```

Junto a este crearemos otro archivo xml de configuración de Spring donde se configuraran aspectos específicos:

```
<?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>
```

Por último, crearemos dos archivos: el primero será el controlador del siguiente archivo; este segundo archivo será la página web que se visualizará en el navegador.

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

@Controller
public class Controlador {

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

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

- El copiloto de confianza:
[ChatGPT](#)
- Figma explicación:
[FIGMA](#)