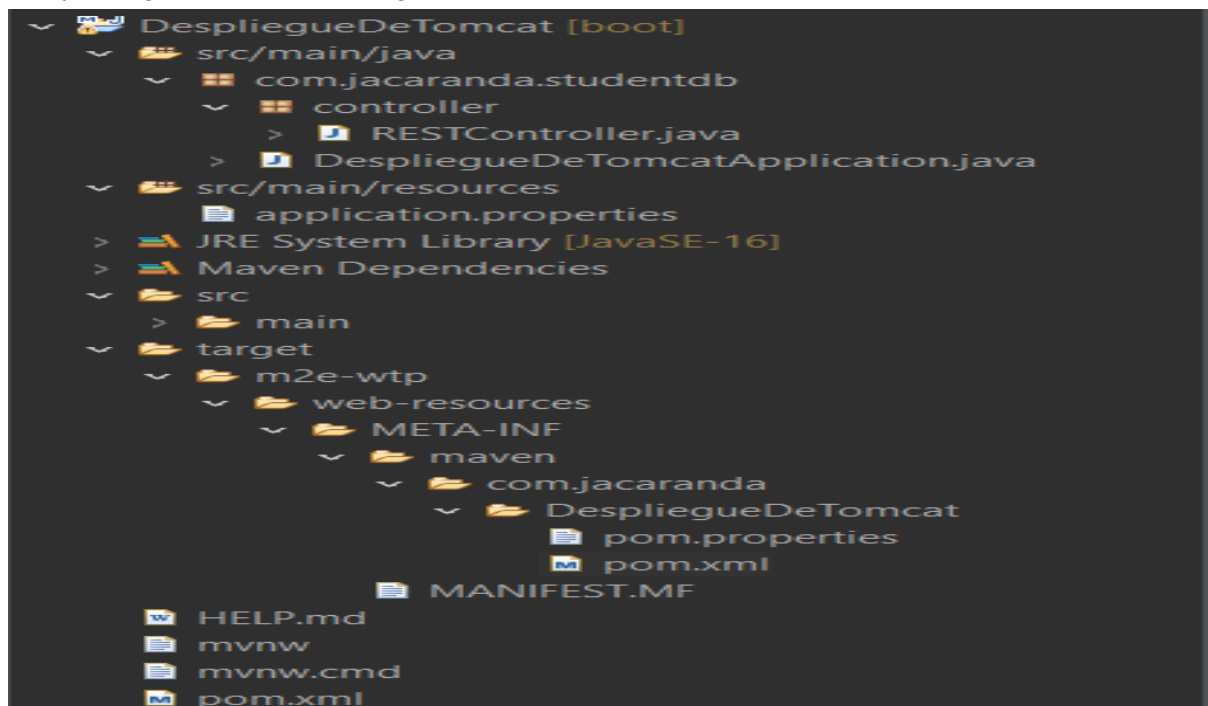


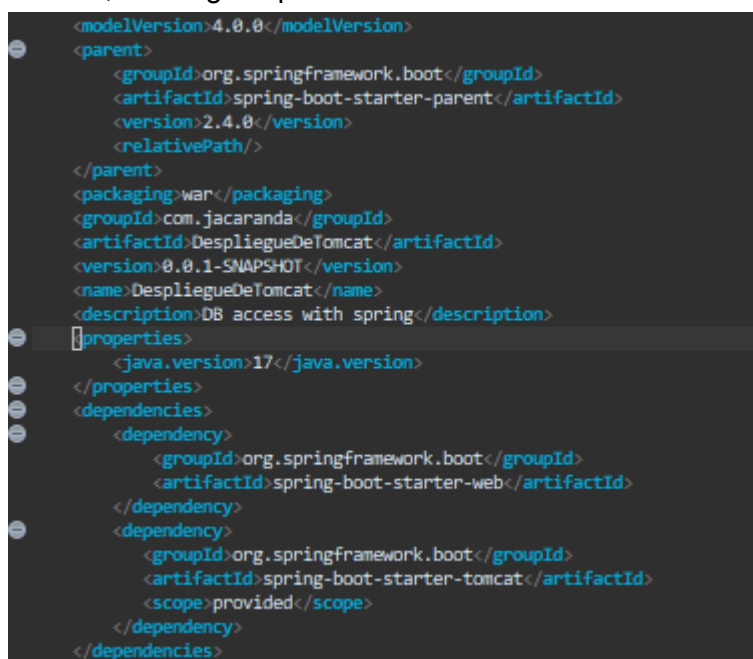
Despliegue de .war en Tomcat

Antonio González Luque

1- Debemos crear un nuevo proyecto de Spring en Eclipse para poder crear el .war, a no ser que ya tengamos uno por ahí guardado.



2- Modificaremos el pom.xml del proyecto para ejecutarlo en la configuraciones por defecto, es decir, sin ningún tipo de modificación.



3- El siguiente paso será añadir al pom.xml las dependencias pertenecientes al .war, para poder crearlo.

```
<build>
  <finalName>${artifactId}</finalName>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

Después debemos añadir una interfaz al ejecutable de la aplicación:

```
import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class DespliegueDeTomcatApplication extends SpringBootServletInitializer {

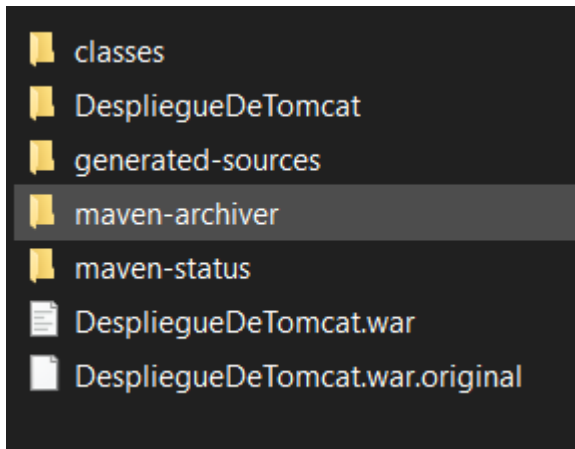
    public static void main(String[] args) {
        SpringApplication.run(DespliegueDeTomcatApplication.class, args);
    }
}
```

Y por último creamos el controlador para poder ver que la aplicación funcione:

```
@RestController
public class RESTController {

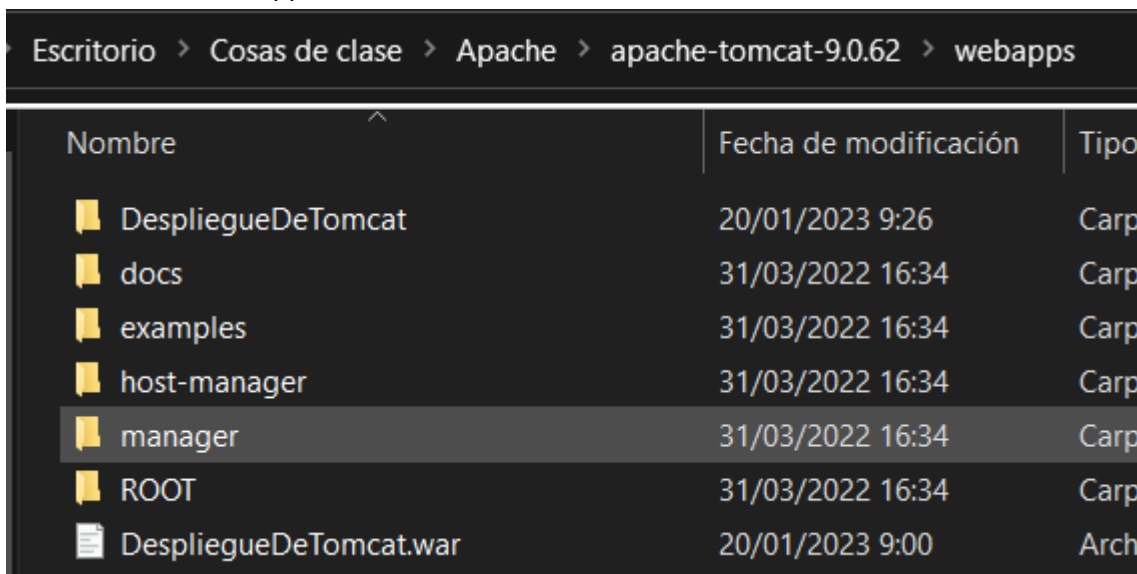
    @GetMapping("/hello")
    public String sayHello() {
        return "Holaaaaa";
    }
}
```

4- Una vez esto, ahora podemos desplegar el proyecto para ejecutar y crear el .war. Una vez hecho eso, en el directorio .target tendremos nuestro .war

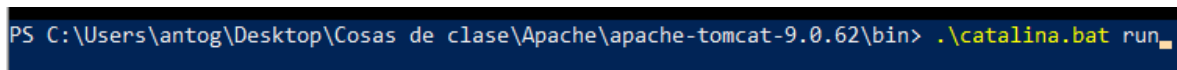


*NOTA: En caso de que no se genere deberás instalar MVN.

5- Ahora que tenemos el .war vamos a descargar TomCat y luego meteremos nuestro .war al directorio de webapps

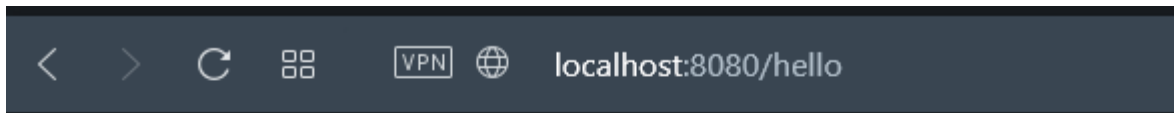


6- Ahora abrimos un terminal y navegaremos hasta el /bin de apache



Y tras usar este comando ya podremos entrar a nuestro proyecto desplegado en Tomcat.

En mi caso, se vería así:



Holaaaa