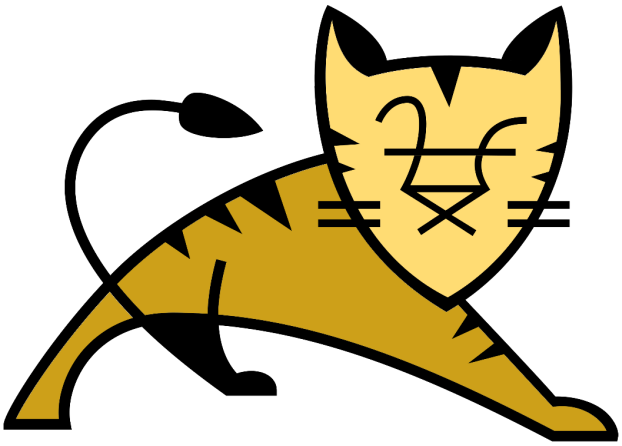
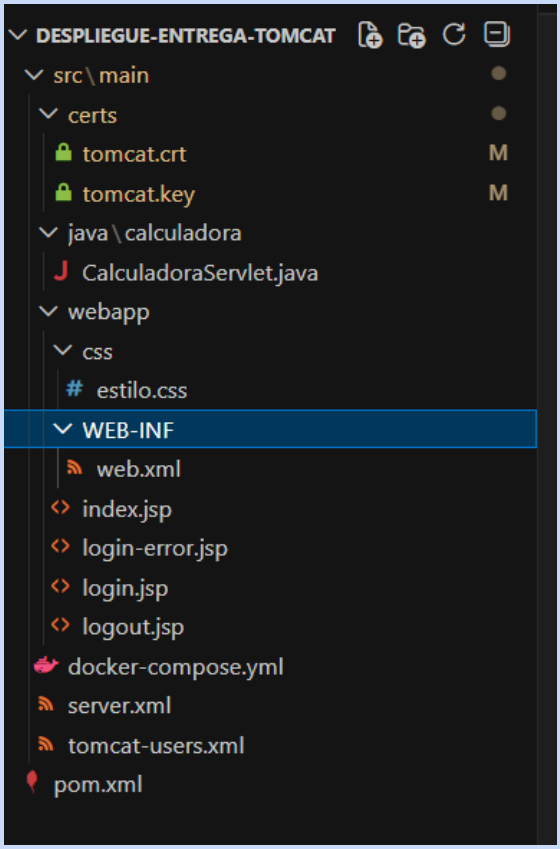
PRÁCTICA TEMA 3



Apache Tomcat

DANIEL AYBAR SOTO

PASO 0

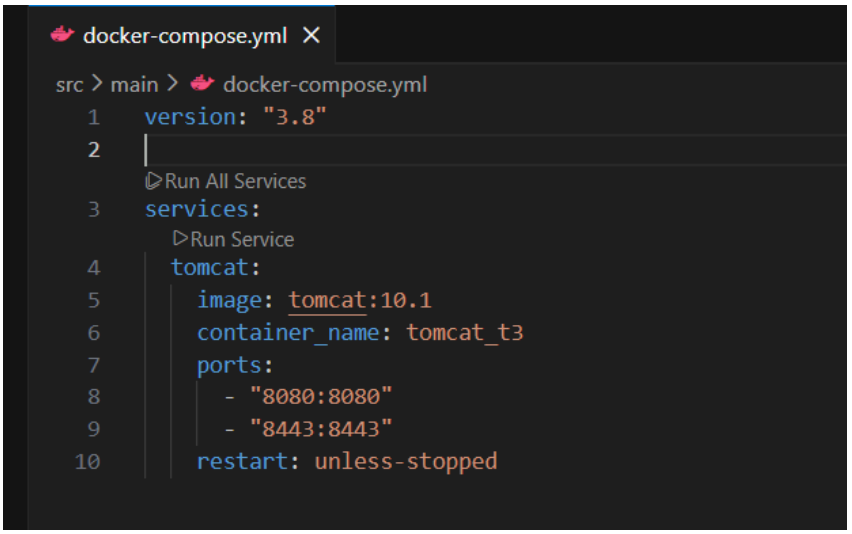


Para empezar, he creado los archivos según la estructura que aparece en la guía de la práctica.

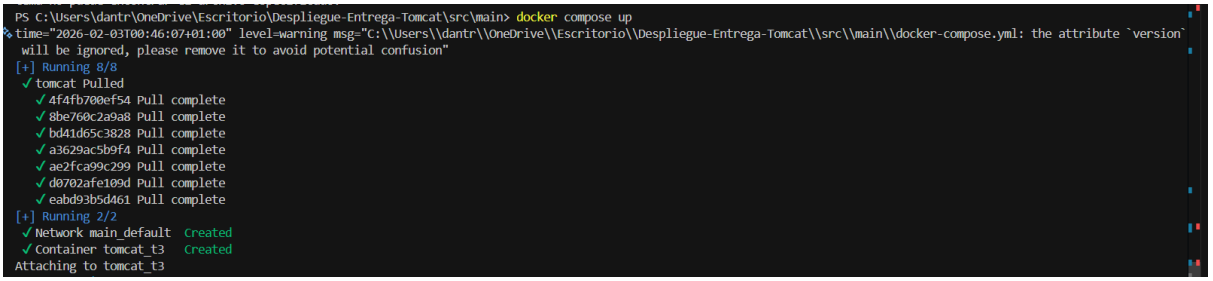
Esto es sólo orientativo, ya que por ejemplo, en el siguiente punto tendré que reemplazar los certificados con unos generados automáticamente.

PASO 1

Crea tu archivo docker-compose y comprueba que tu contenedor se levanta con normalidad



Este es el archivo de docker-compose que he creado por ahora. Compruebo que se levanta:

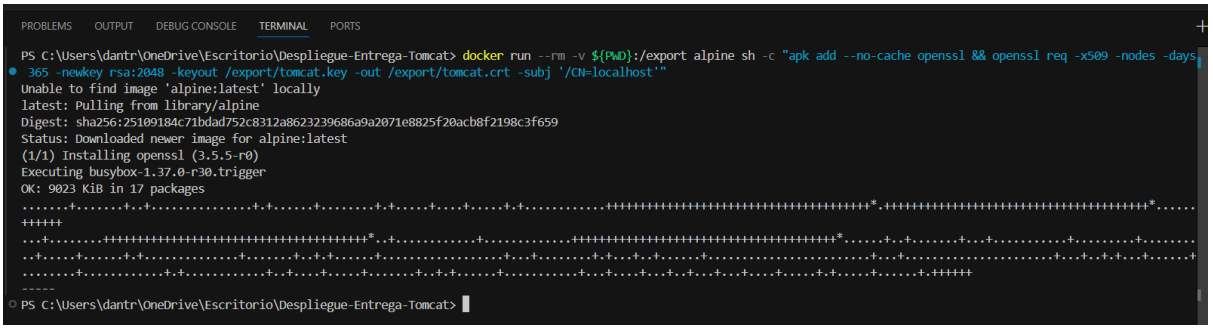


Trabajando en mi PC particular con SO Windows, al principio no me ha funcionado el comando “docker compose up -d”, pero eso se debía a que no tenía docker desktop abierto al mismo tiempo (jajan’t). Una vez lo he abierto, ya se ha levantado el contenedor sin problemas, mostrando la página por defecto de Apache.

PASO 2

Generar los certificados autofirmados SSL/TLS y modificar docker-compose en consecuencia. Comprueba que puedes entrar por “https”.

Con este comando, he creado los certificados SSL/TLS a través del contenedor de docker:



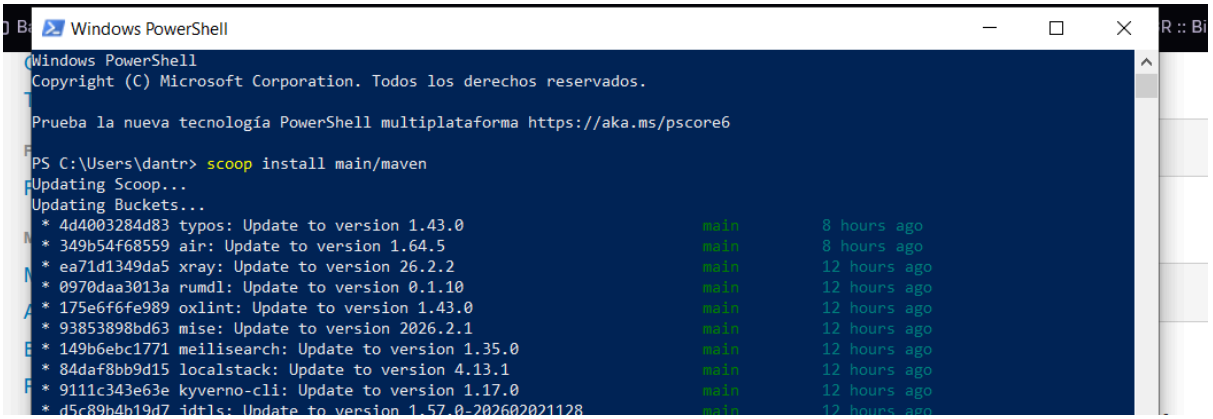
Se me han creado los certificados en la carpeta “src”, así que los he movido a la carpeta de “certs”.

Tal y como tengo los archivos, todavía no puedo acceder con el método https.

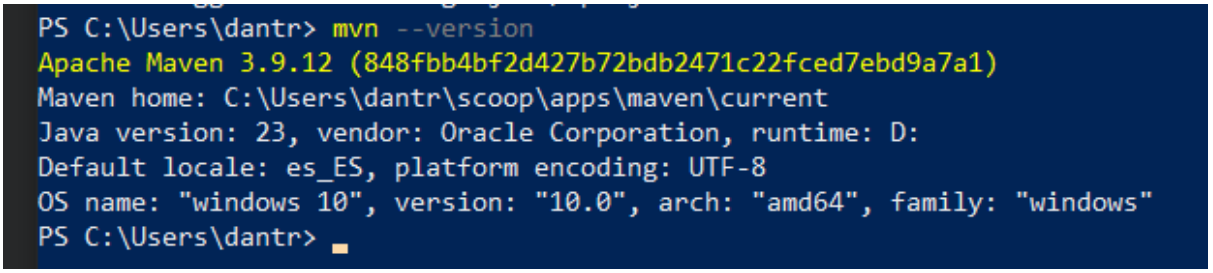
PASO 3

Instala Maven y comprueba que lo has hecho correctamente.

He visto que los desarrolladores de Maven permiten instalarlo usando scoop, así que he aprovechado que ya lo he usado antes y he probado en este caso. Por suerte, no ha habido ningún problema hehehehehehe >:)

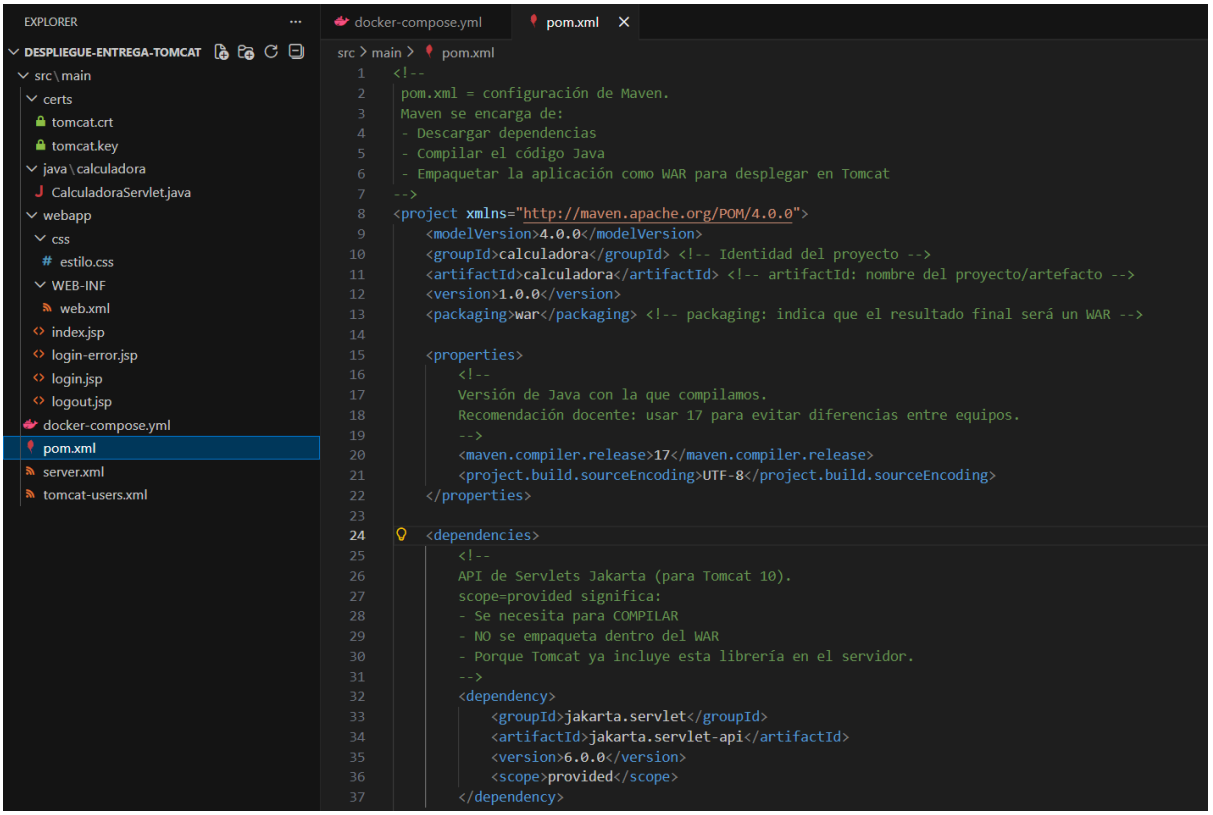


Compruebo que lo he instalado bien: todo perfe



PASO 4

Crea el archivo pom.xml y asegúrate de que concuerde con la versión de java que está usando maven.



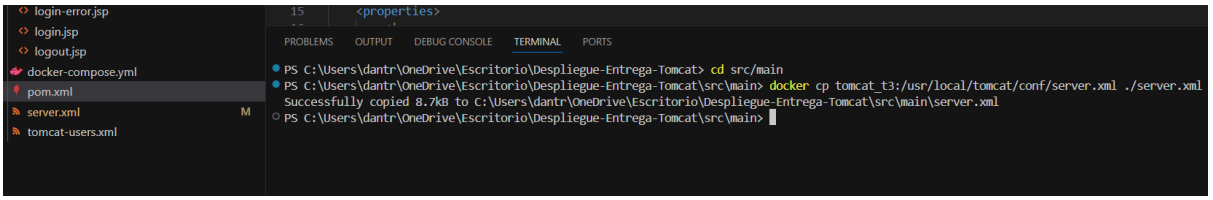
El uso de la etiqueta <scope>provided</scope> es una buena práctica. Significa que Maven usará la librería para compilar el código, pero no la meterá dentro del archivo final porque el servidor Tomcat ya la tiene instalada.

No debería darme ningún problema la compatibilidad de las versiones de Java que usan tanto Maven en mi equipo (Java 23) como el archivo pom.xml (Java 17), porque hay retrocompatibilidad y la etiqueta maven.compiler.release genera un archivo compatible con la versión 17.

PASO 5

Crea tu archivo server.xml, copiando el generado por Tomcat en tu proyecto.

En este paso no ha habido ningún problema.

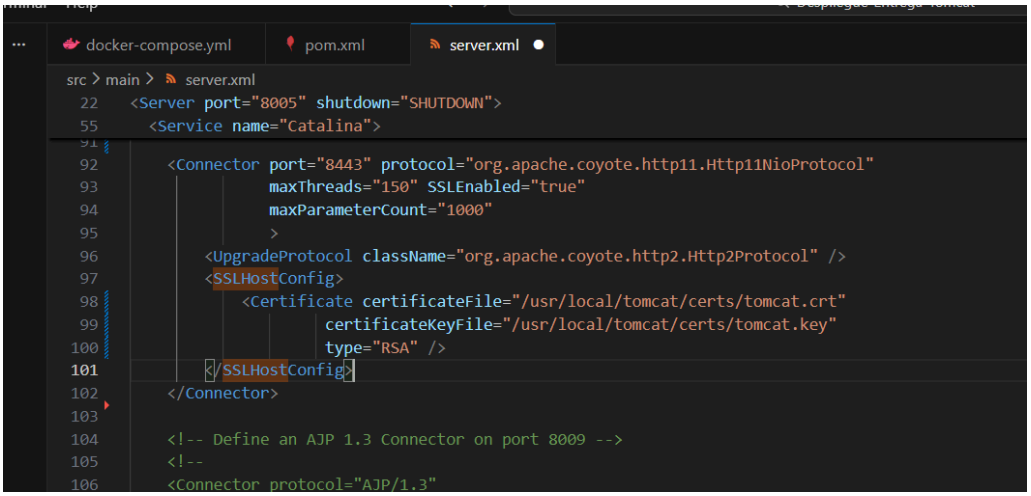


Para que Tomcat use este nuevo archivo, lo incluyo en el docker-compose.yml:

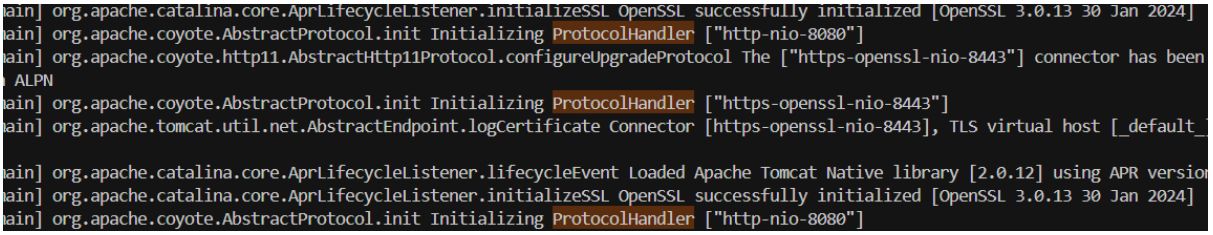
PASO 6

Modifica el server.xml para indicar donde encontrará el contenedor los certificados.

Descomento el bloque y cambio las rutas de los certificados designados para que Tomcat los busque:



Tumbo el contenedor y lo vuelvo a levantar para comprobar que los cambios sean correctos. Aquí he tenido problemas para comprobar que se estableciera conexión, pero después de tumbar otra vez el contenedor, parece que ha funcionado. Usando el comando “docker logs tomcat_t3”, veo que se ha inicializado el manejador de protocolos de comunicación segura:



PASO 7

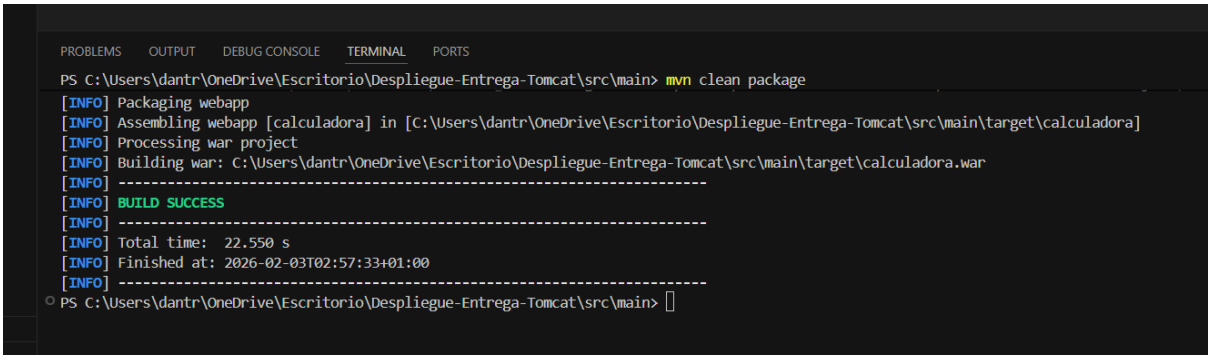
Crea tu aplicación calculadora en dos pasos, un archivo java que contendrá la lógica y un index jsp que será tu interfaz.

Voy a probar primero con los archivos orientativos que aparecen en la guía de la práctica para ver si funcionan.

PASO 8

Añade tu archivo web.xml e intenta probar a generar el archivo.war con maven.

Al parecer, en un momento había descargado el daemon de Maven y había referenciado su carpeta bin en las variables de entorno, así que he tenido que descargar el binario y cambiar el path. Después de esto y cerrar y volver a abrir todas las terminales, ya me funciona el comando mvn clean package:

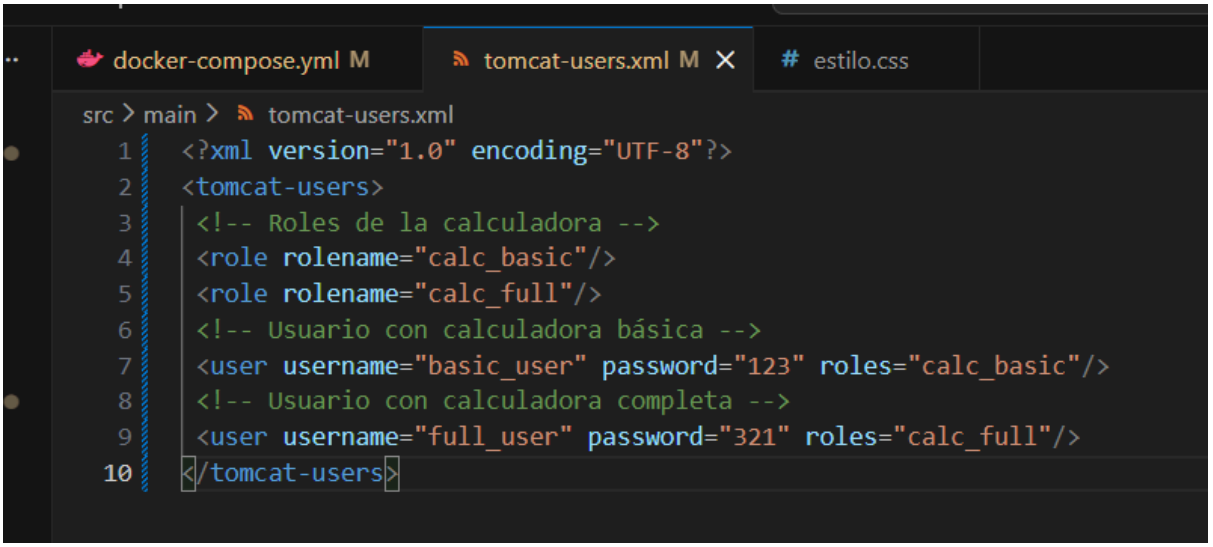


Con esto, se me ha creado una carpeta “target” con varios archivos dentro, “calculadora.war“ entre ellos. Ahora tengo que actualizar el docker-compose.yml, tirar el contenedor de docker y volver a levantarlo para poder ver el archivo generado en el navegador en la url “https://localhost:8443/calculadora/”.

PASO 9

Añade roles a tu app con el archivo tomcat-users.xml y comprobar su correcto funcionamiento, tanto de los usuarios con sus contraseñas como con las restricciones impuestas a los roles.

Edito tomcat-users.xml para que tenga los datos dados en la guía:



Y entonces, edito el docker-compose.yml para que tomcat busque a los usuarios en la dirección de tomcat-users.xml:

Sigo sin poder acceder a ninguna vista de los archivos .js que tengo. He intentado de todo: docker compose down/up -d, comprobar que existe el archivo calculadora.war en la carpeta target, revisar los volumes del archivo docker-compose.yml….y nada.