Realización de Cursos y Talleres

Manual de despliegue

Versión: v01r00

Fecha: 26-12-2023

HOJA DE CONTROL

Título	Realización de Cursos y Talleres		
Entregable	Manual de despliegue		
Nombre del Fichero	UD02-Actividad Evaluable 01-Realización de Cursos y Talleres PMS.pdf		
Autor	Pablo Merinas Soto		
Versión/Edición	v01r00	Fecha Versión	26-12-2023
Aprobado por		Fecha Aprobación	DD/MM/AAAA
		Nº Total Páginas	023

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio

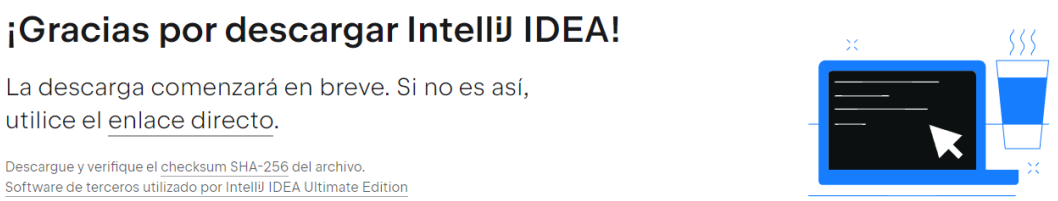
CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Área	Nº Copias

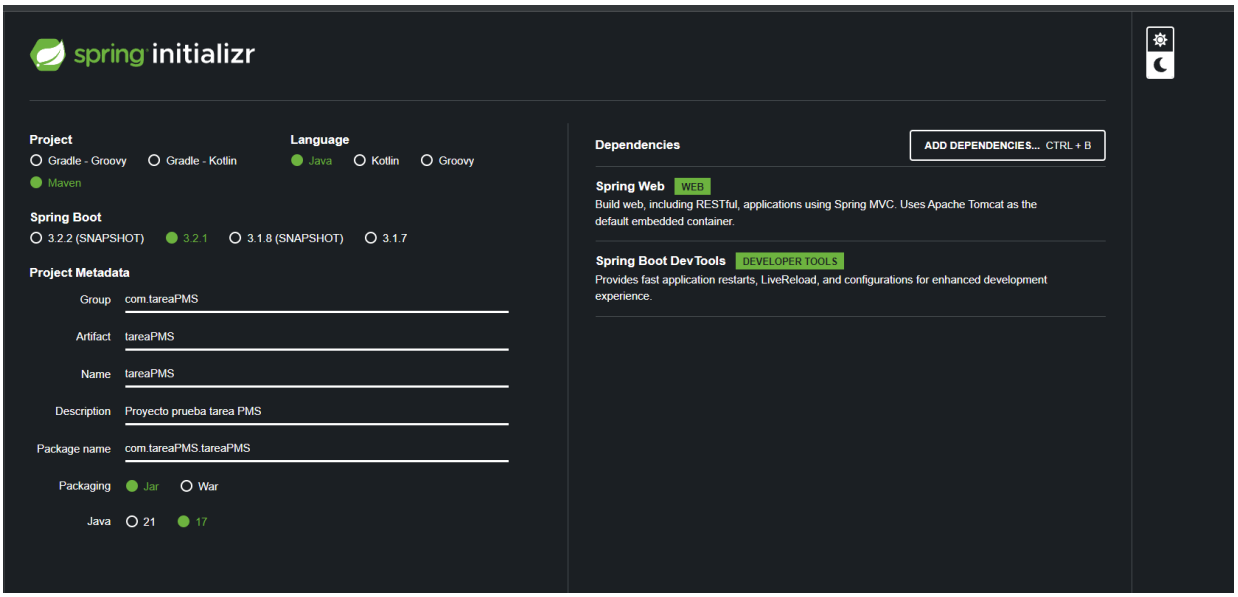
Indice

Desplegar app Spring Boot con Docker

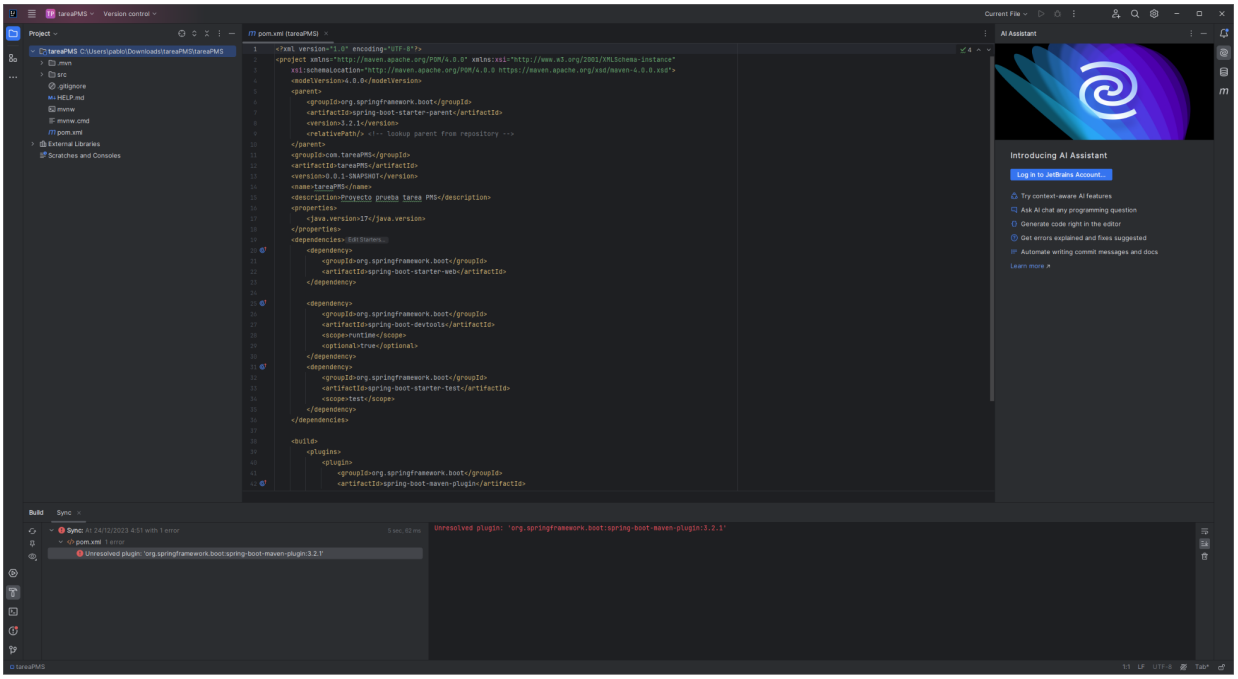
Primero instalamos el IDE intelliJ para trabajar siguiendo los pasos del curso



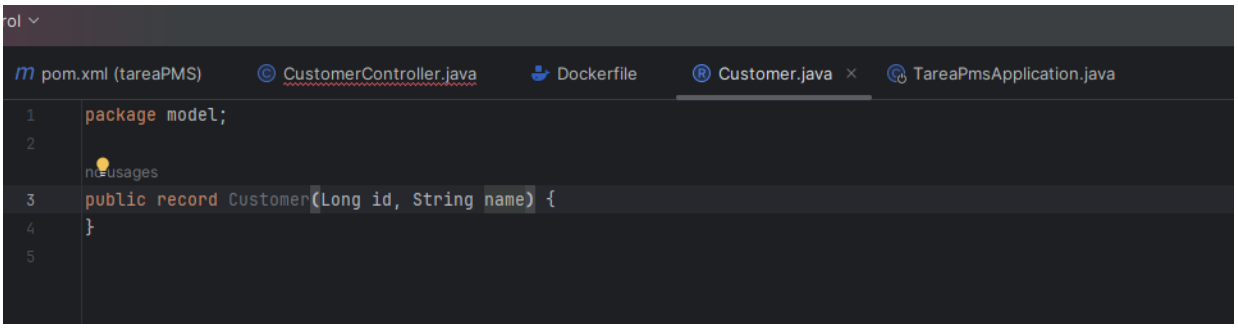
Creamos el proyecto con spring initializr



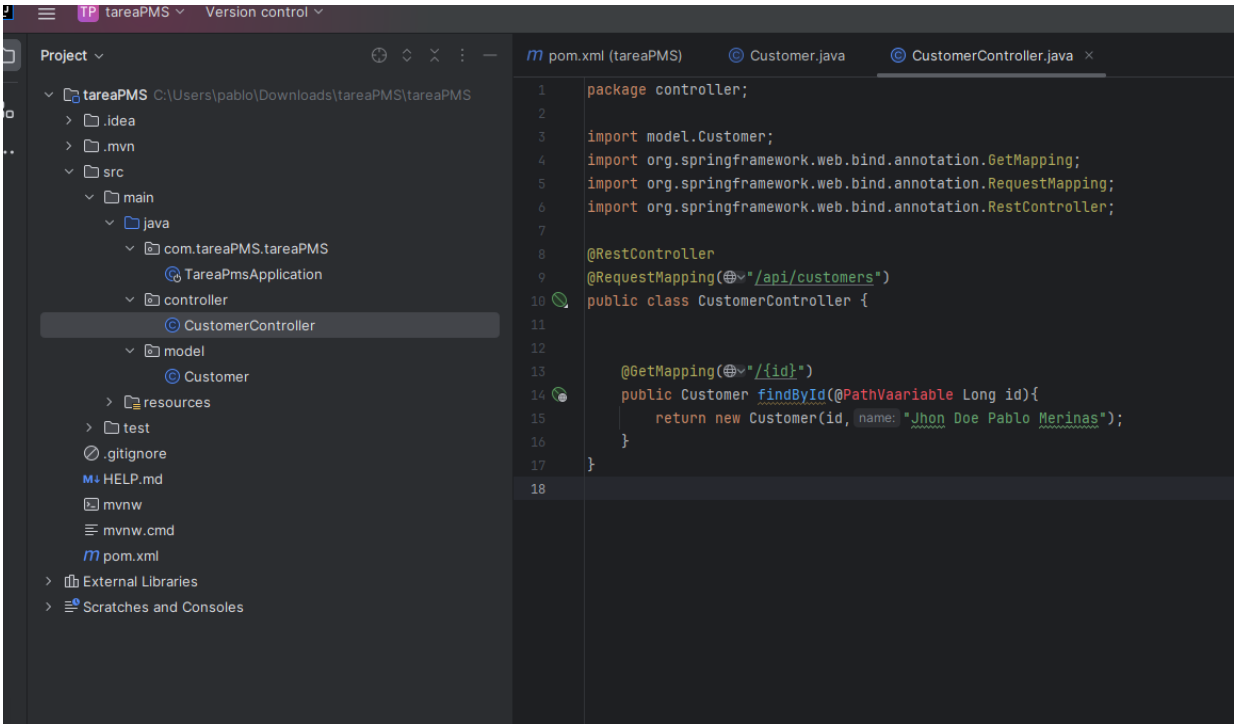
Importo el proyecto a IntelliJ



Creamos los controladores modelos al igual que el curso



Y el controlador REST

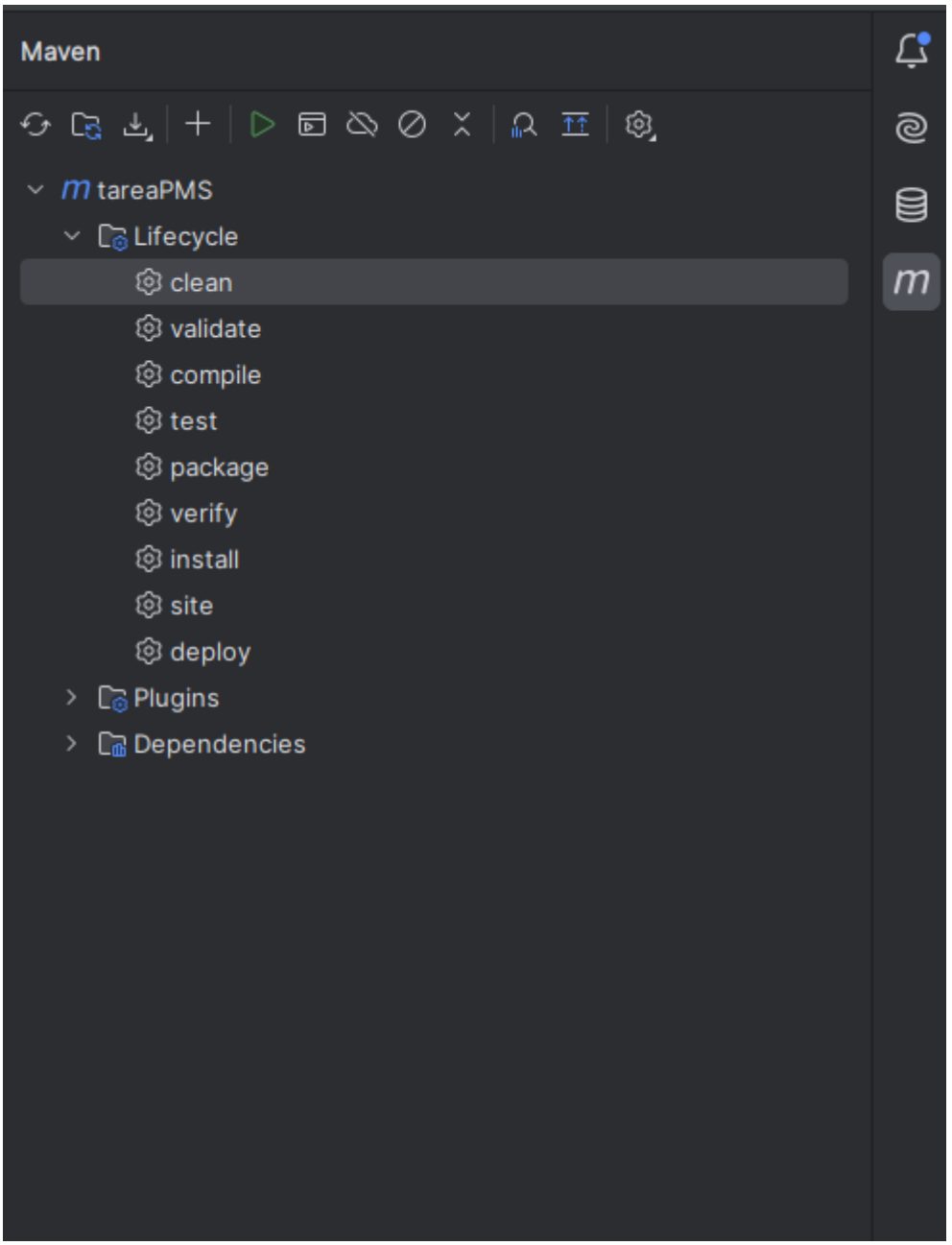


Cambiamos la versión del proyecto a la 1.0

pom.xml (tareaPMS) × TareaPmsApplication.java

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSche
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.2.1</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.tareaPMS</groupId>
    <artifactId>tareaPMS</artifactId>
    <version>1.0</version>
    <name>tareaPMS</name>
    <description>Proyecto prueba tarea PMS</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies> Edit Starters...
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
```

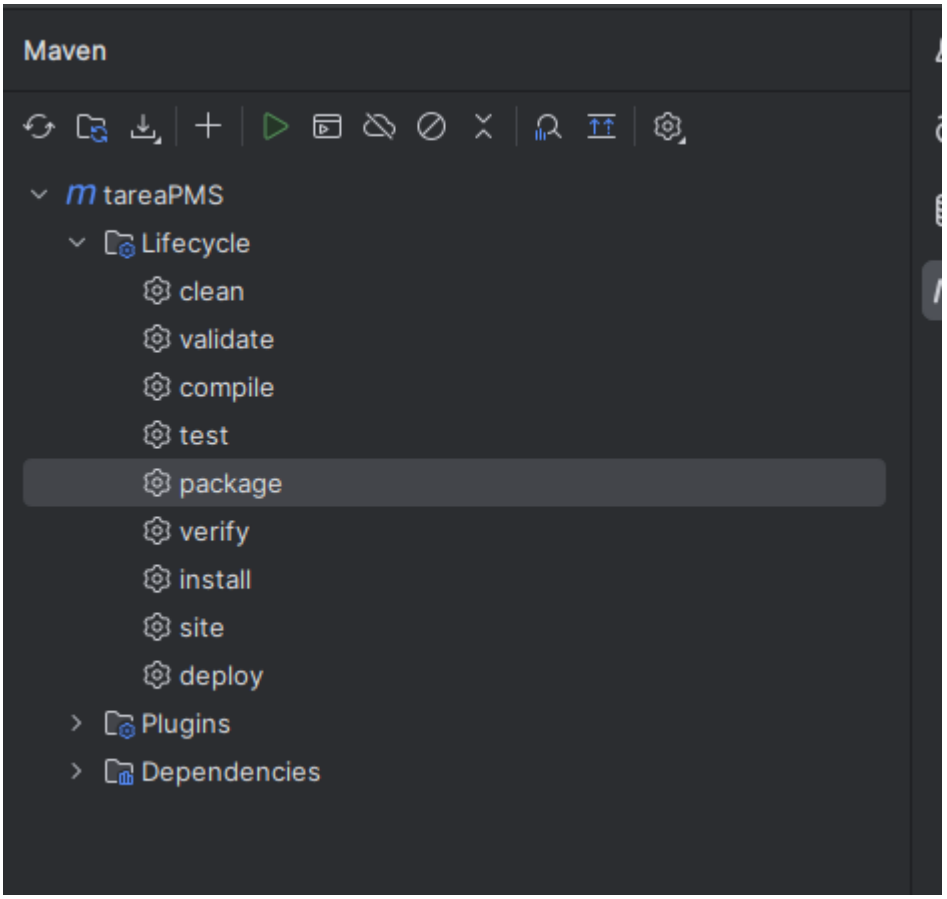
Hacemos un clean con maven


Maven
tareaPMS
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Dependencies

```
[INFO] --- clean:3.3.2:clean (default-clean) @ tareaPMS ---
[INFO] Deleting C:\Users\pablo\Downloads\tareaPMS\tareaPMS\target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.216 s
[INFO] Finished at: 2023-12-24T05:01:08+01:00
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

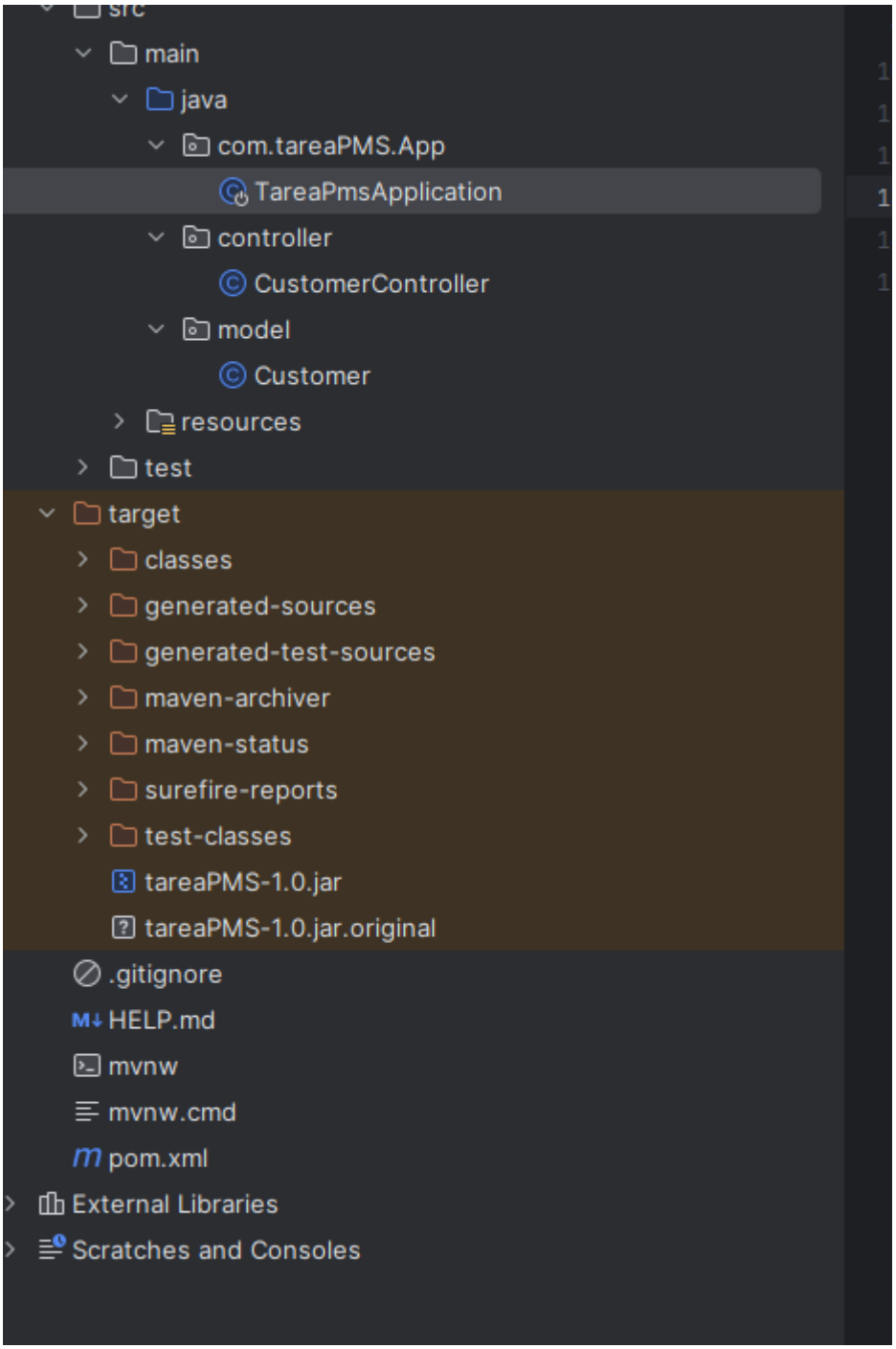
Con maven empaquetamos



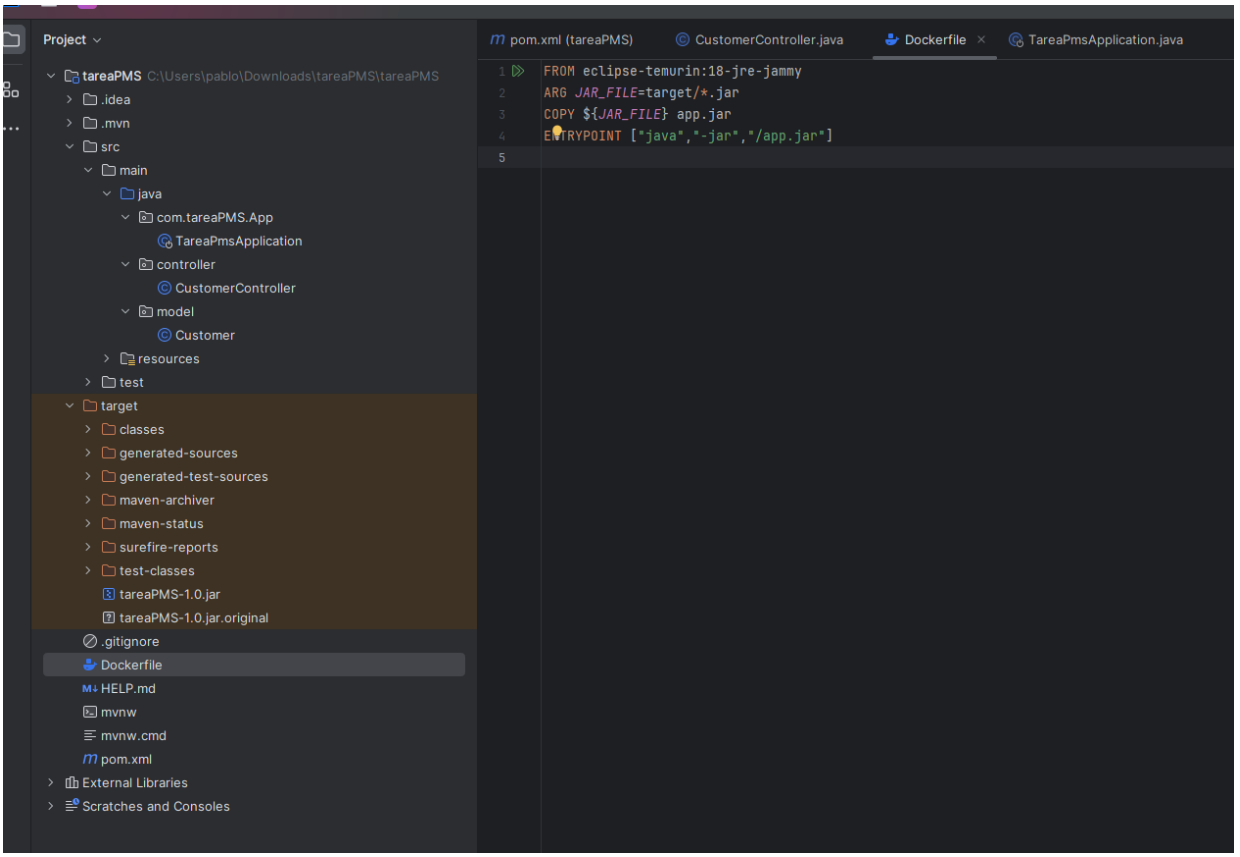
```
[INFO] Replacing main artifact C:\Users\pablo\Downloads\tareaPMS\tareaPMS\target\tareaPMS-1.0.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to C:\Users\pablo\Downloads\tareaPMS\tareaPMS\target\tareaPMS-1.0.jar.original
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.201 s
[INFO] Finished at: 2023-12-24T05:01:38+01:00
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

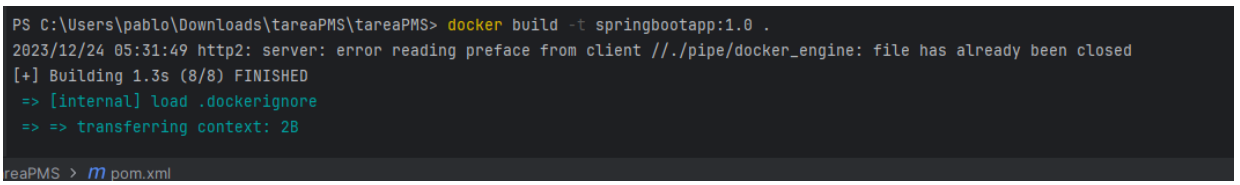
Para comprobar vemos que tenemos la carpeta target



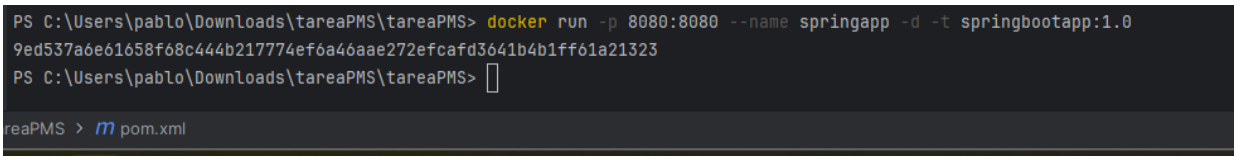
Creamos el archivo Dockerfile con la configuración



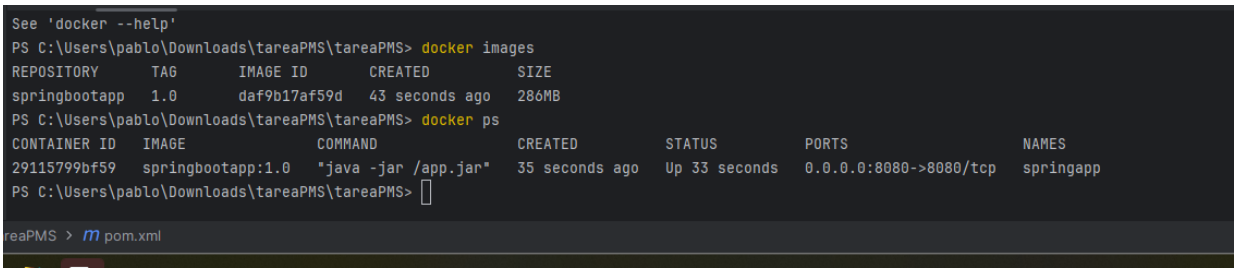
Creamos la imagen de docker



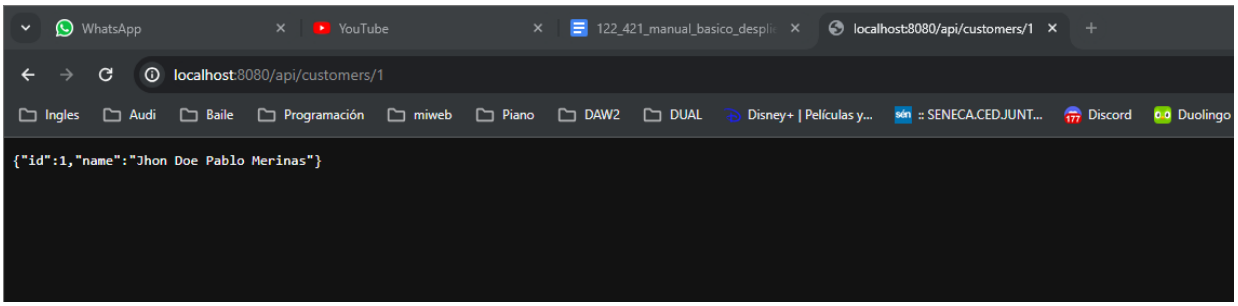
Creo el contenedor usando esta imagen



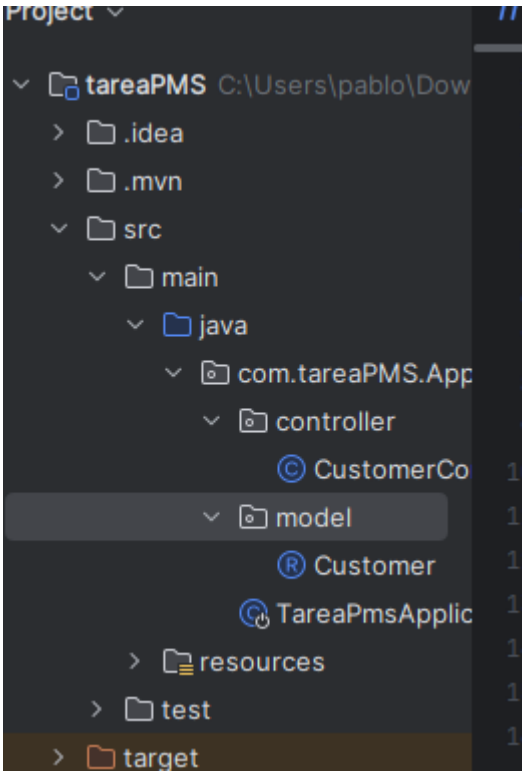
Comprobamos que se haya creado bien



Comprobamos que se ejecute desde el navegador



Aquí tuve varios problemas que no me salia eso, al final tras investigar me di cuenta que había montado mal la estructura del proyecto, la corregí:

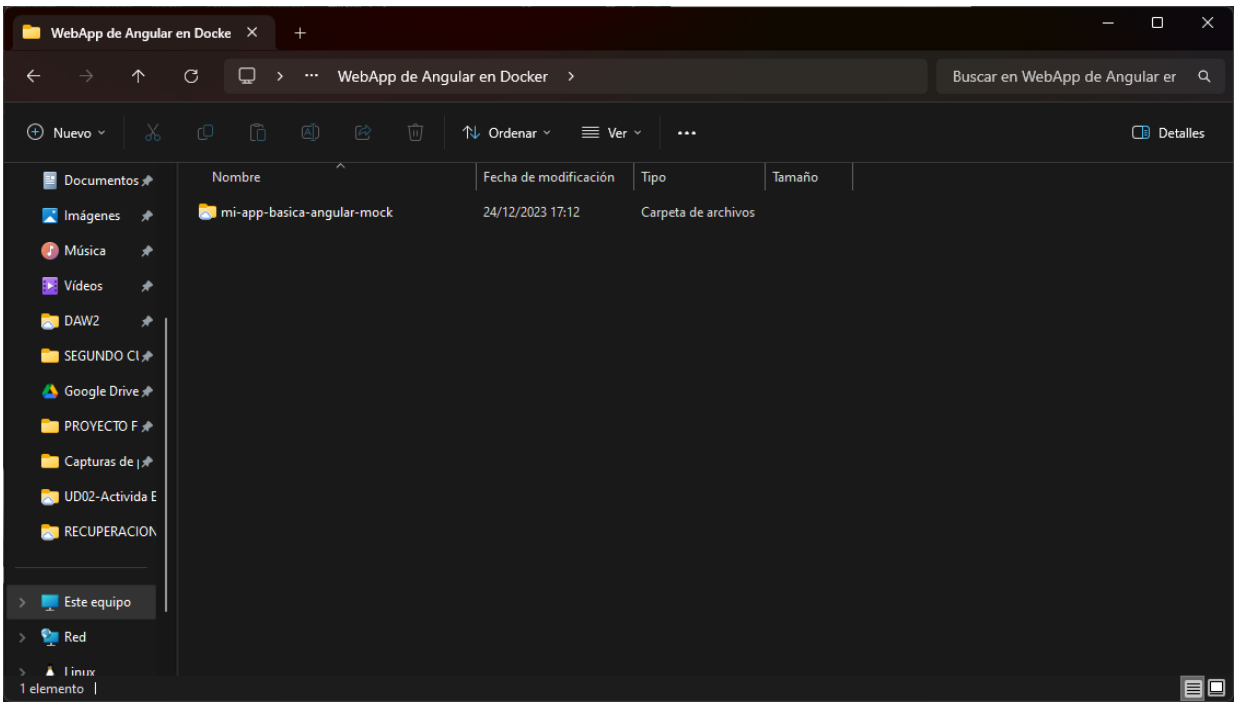


Volví a empaquetar y repetir todo el proceso y ya iba bien.

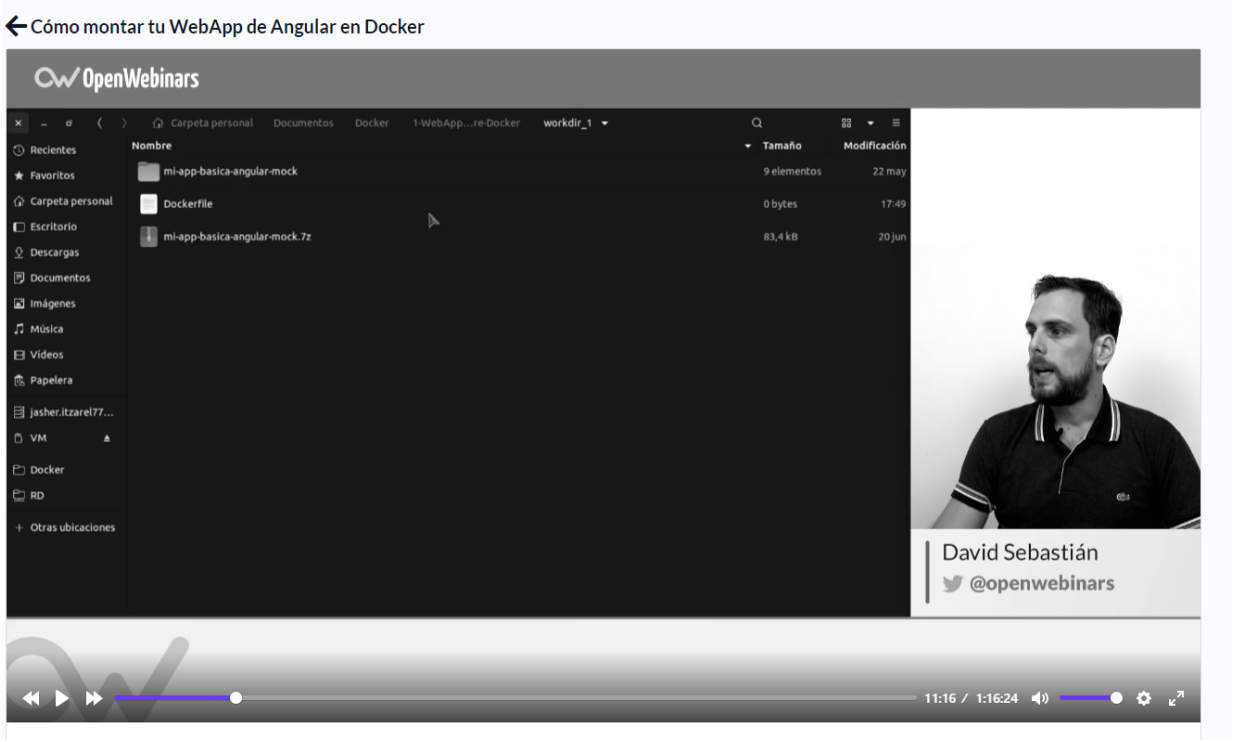
Y con esto ya queda desplegada la aplicación

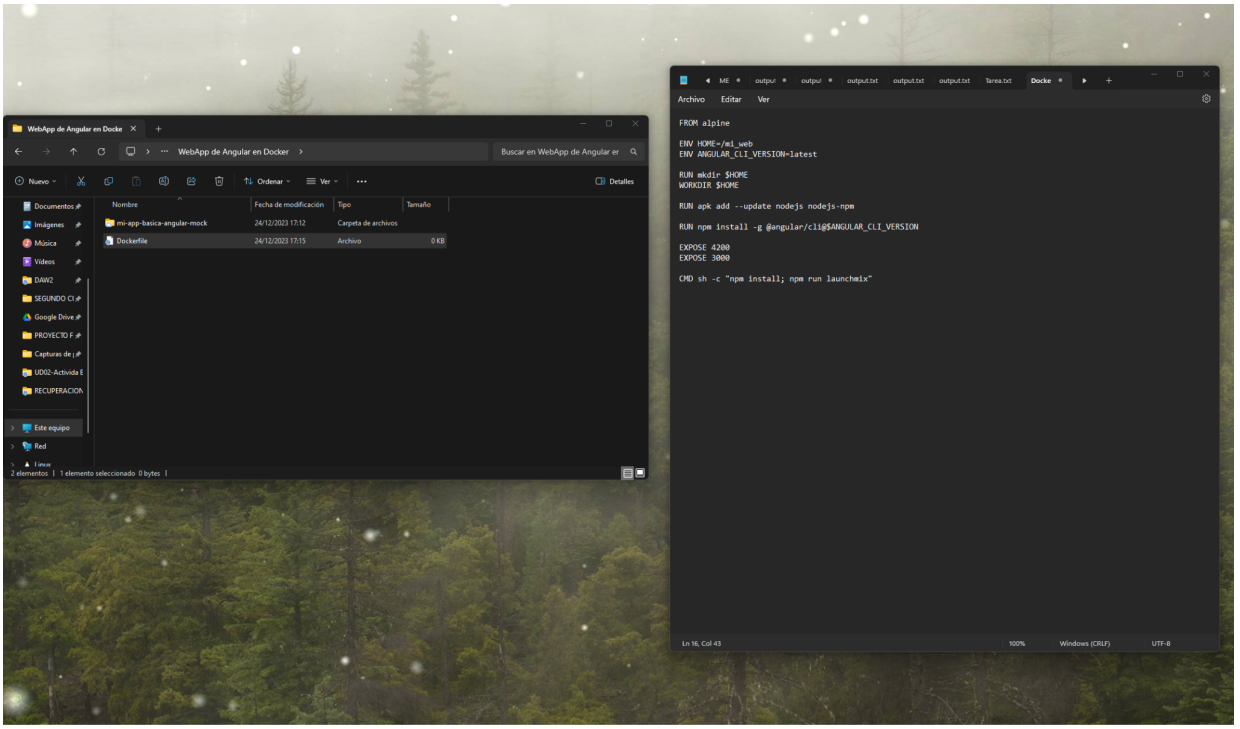
Montar tu WebApp de Angular en Docker

Primero descargamos el proyecto que facilita el curso a IntelliJ

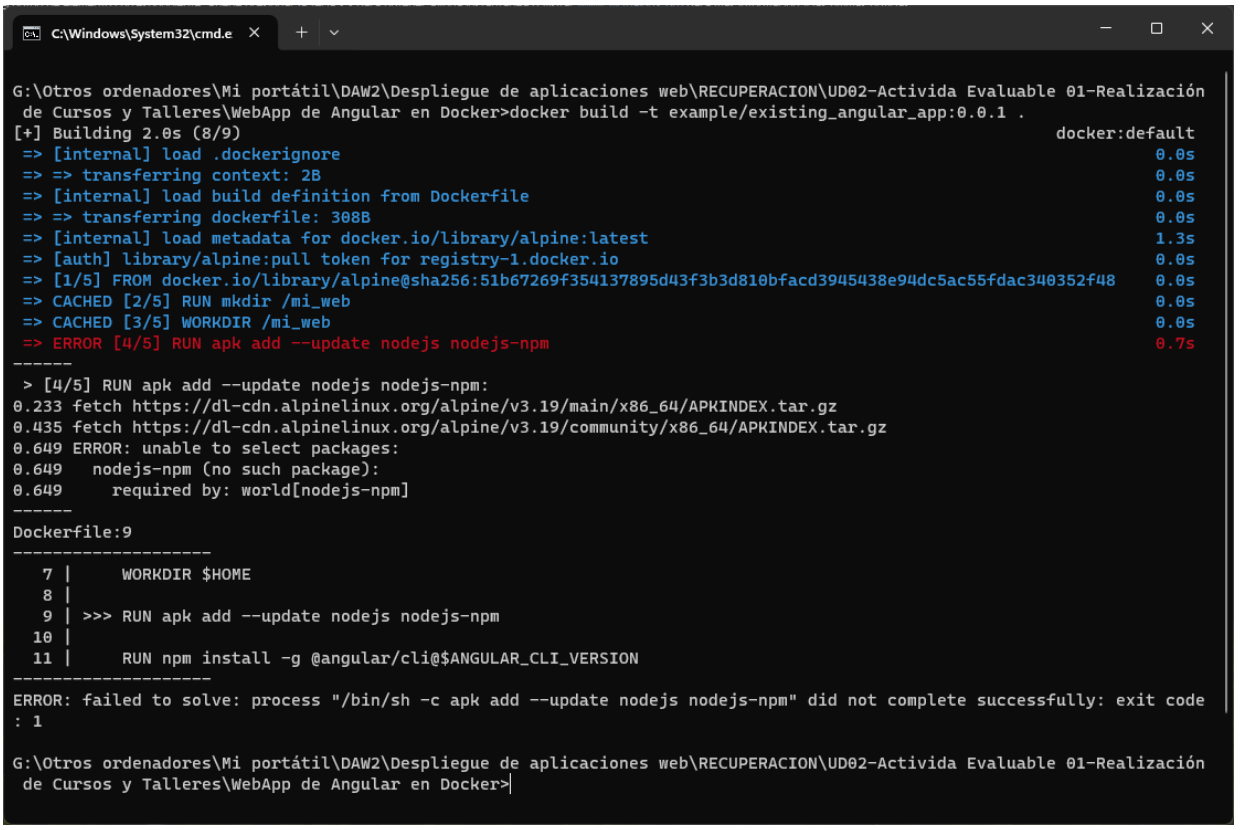


Creamos el archivo dockerfile

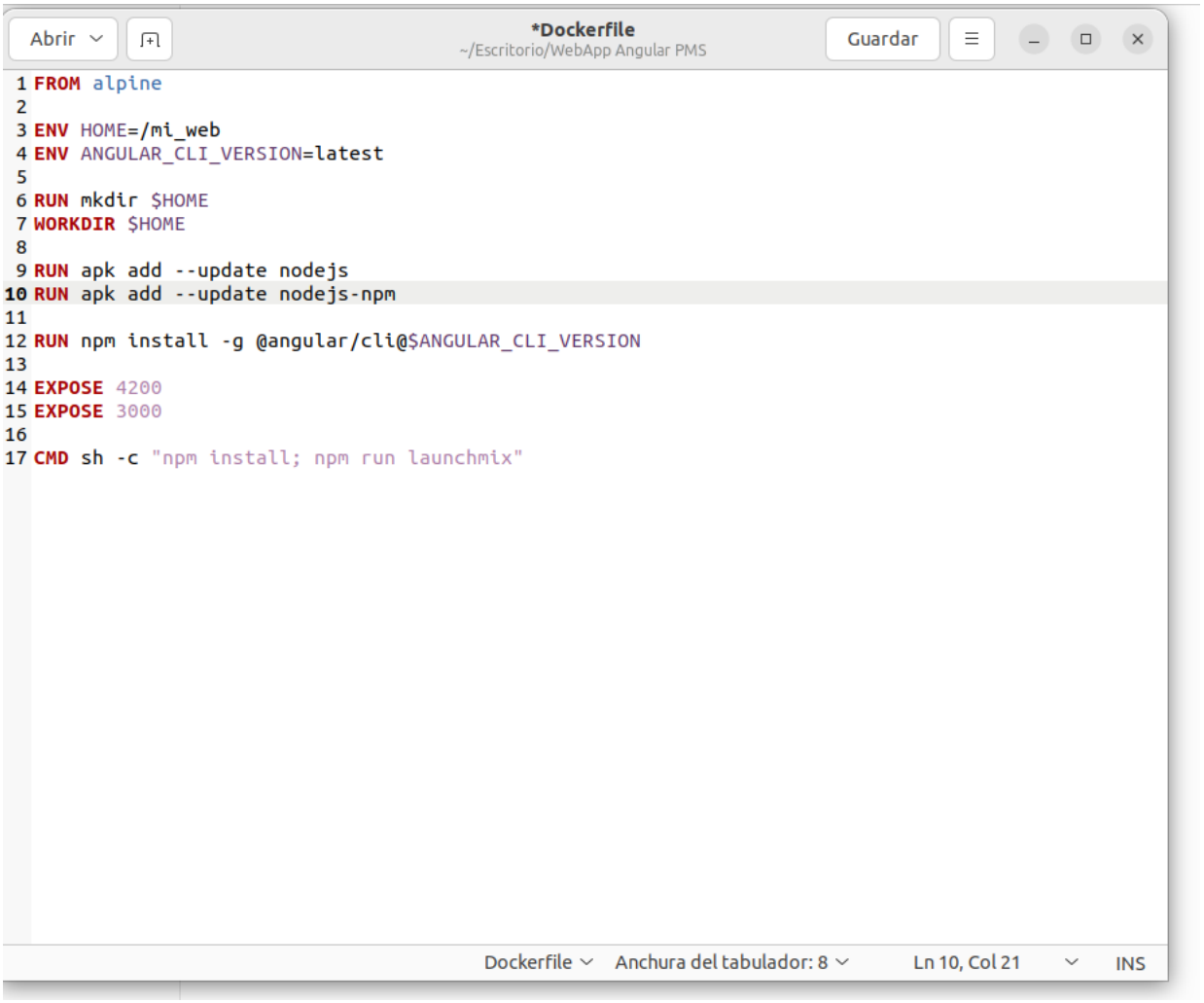




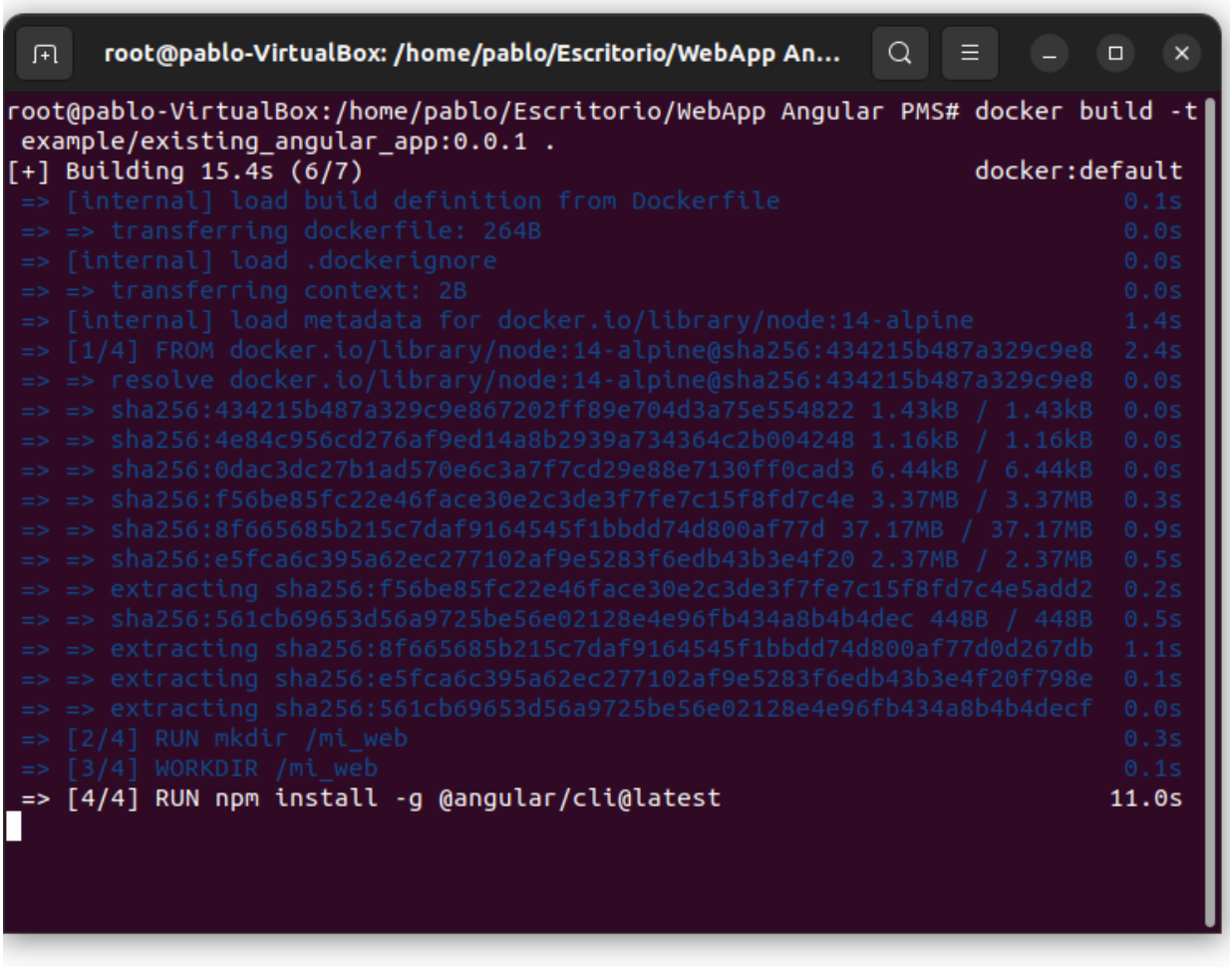
Creamos la imagen de docker



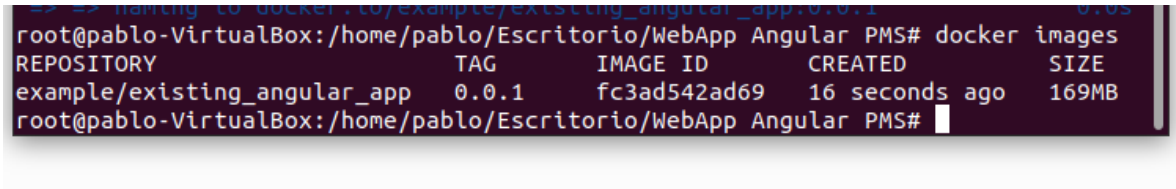
Aquí tuve el problema de que estaba usando windows, y el curso y comandos estaban hecho con linux, intente cambiar el Dockerfile para hacerlo compatible pero no fui capaz, así que instale una MV de linux para seguir con la tarea



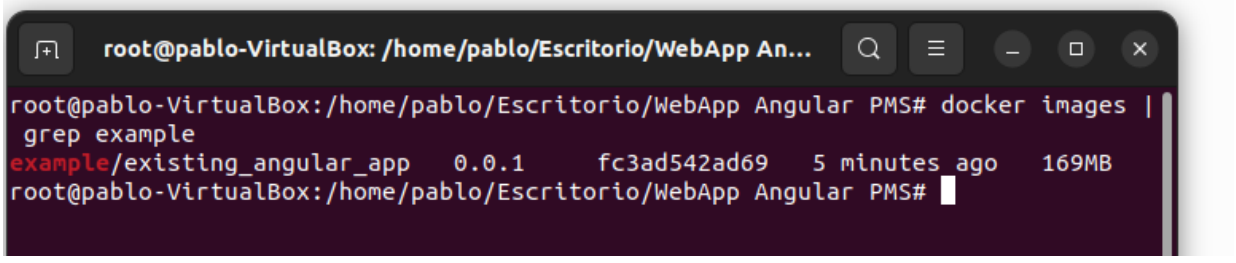
Ejecutamos el comando para crear el docker

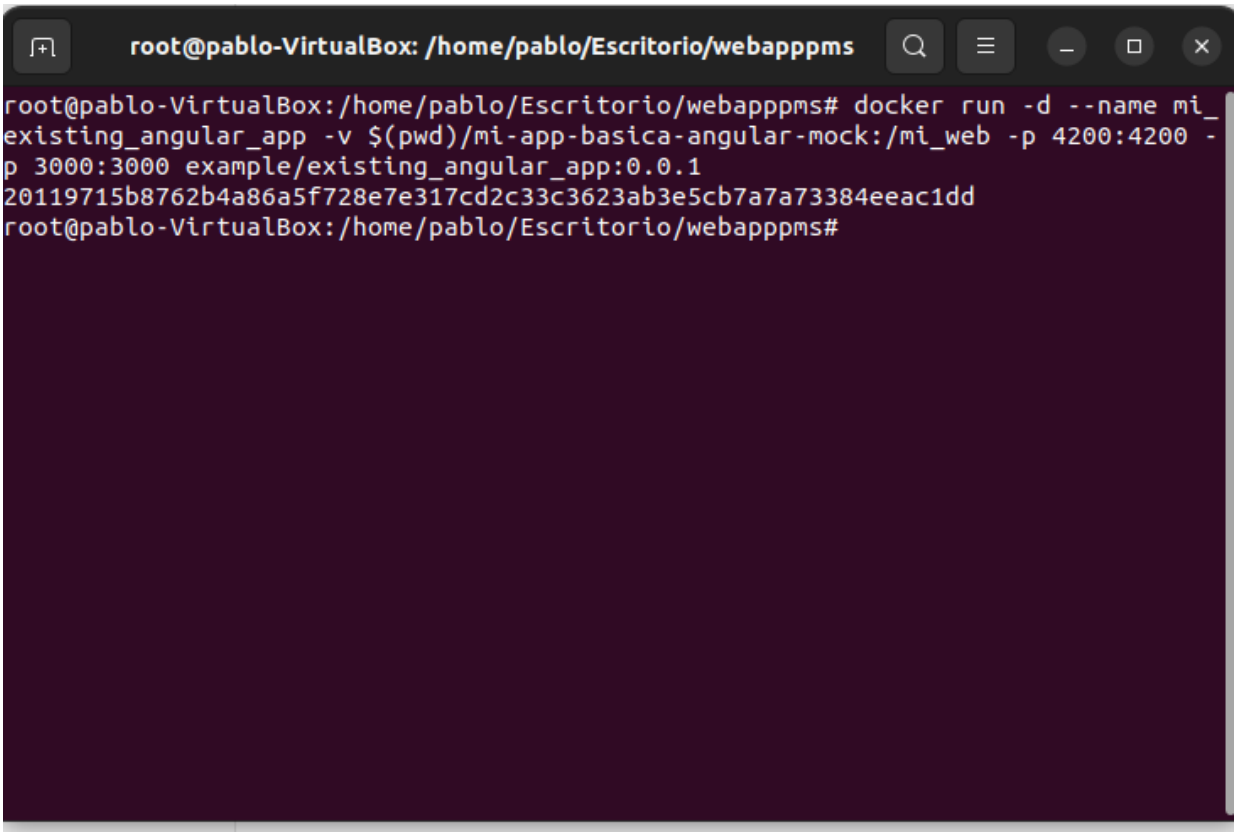


Comprobamos



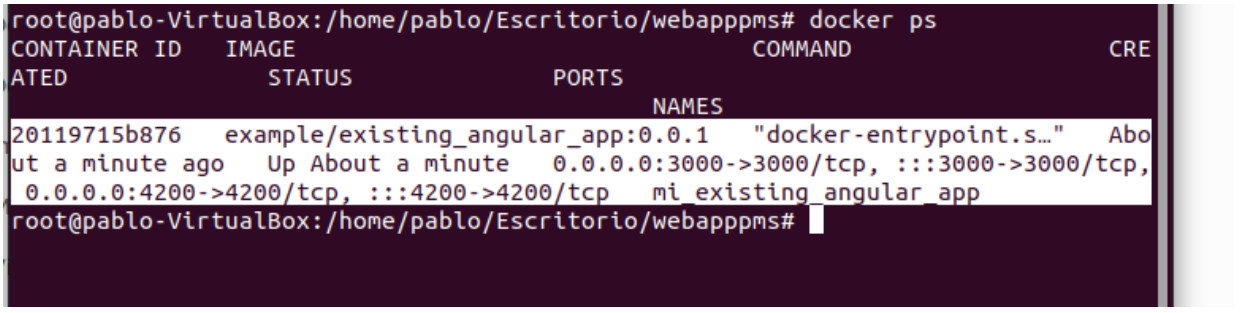
Creamos la imagen basada en este contenedor



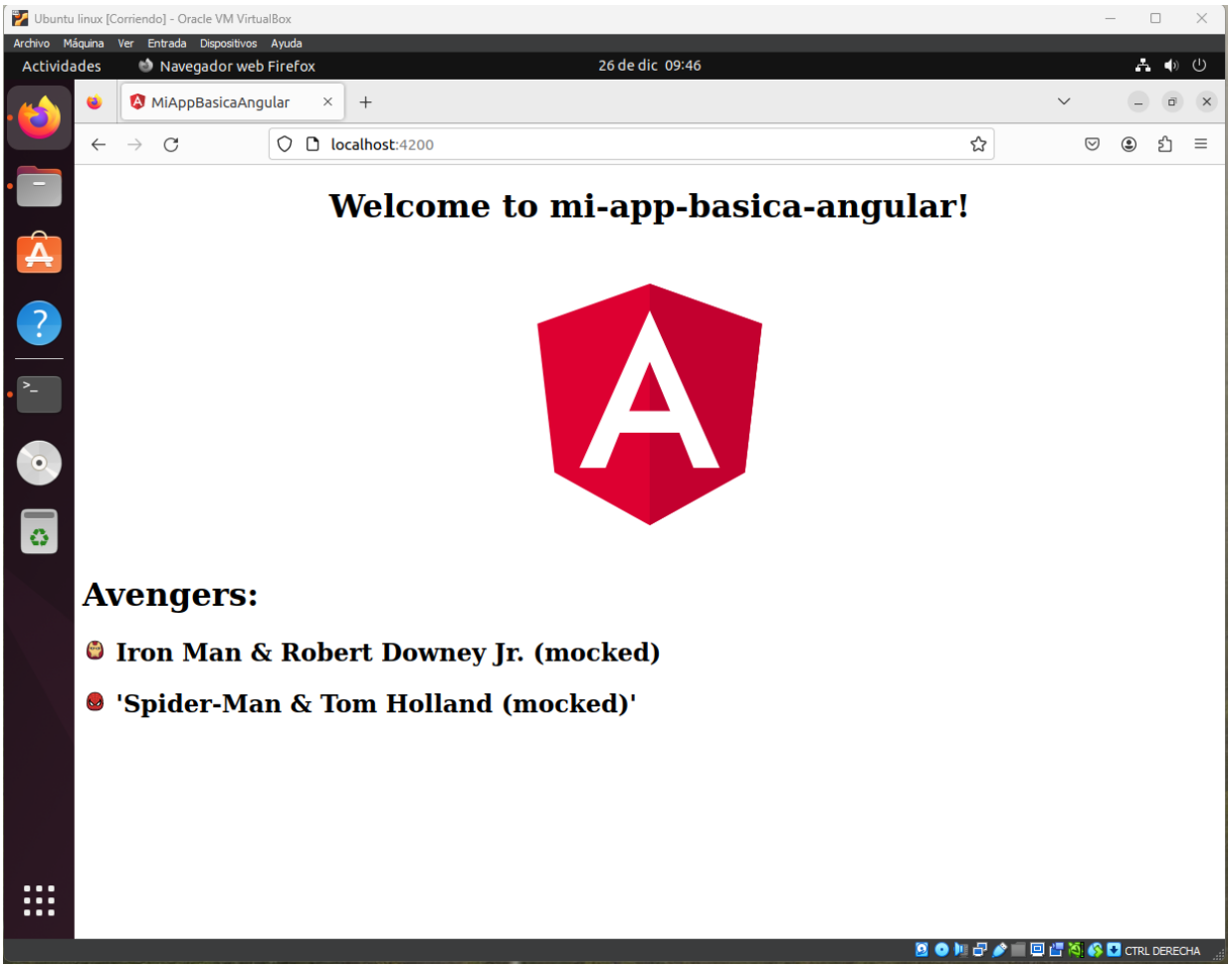


Aquí tuve otro problema, el nombre de mi carpeta anterior era WebApp Angular PMS, y al poner $(pwd) para que pusiera la ruta, no me lo generaba por los espacios del nombre de la carpeta, al darme cuenta cambie el nombre a webapppms sin espacios y ya funcionó sin problema.

Y con un docker ps comprobamos su estado

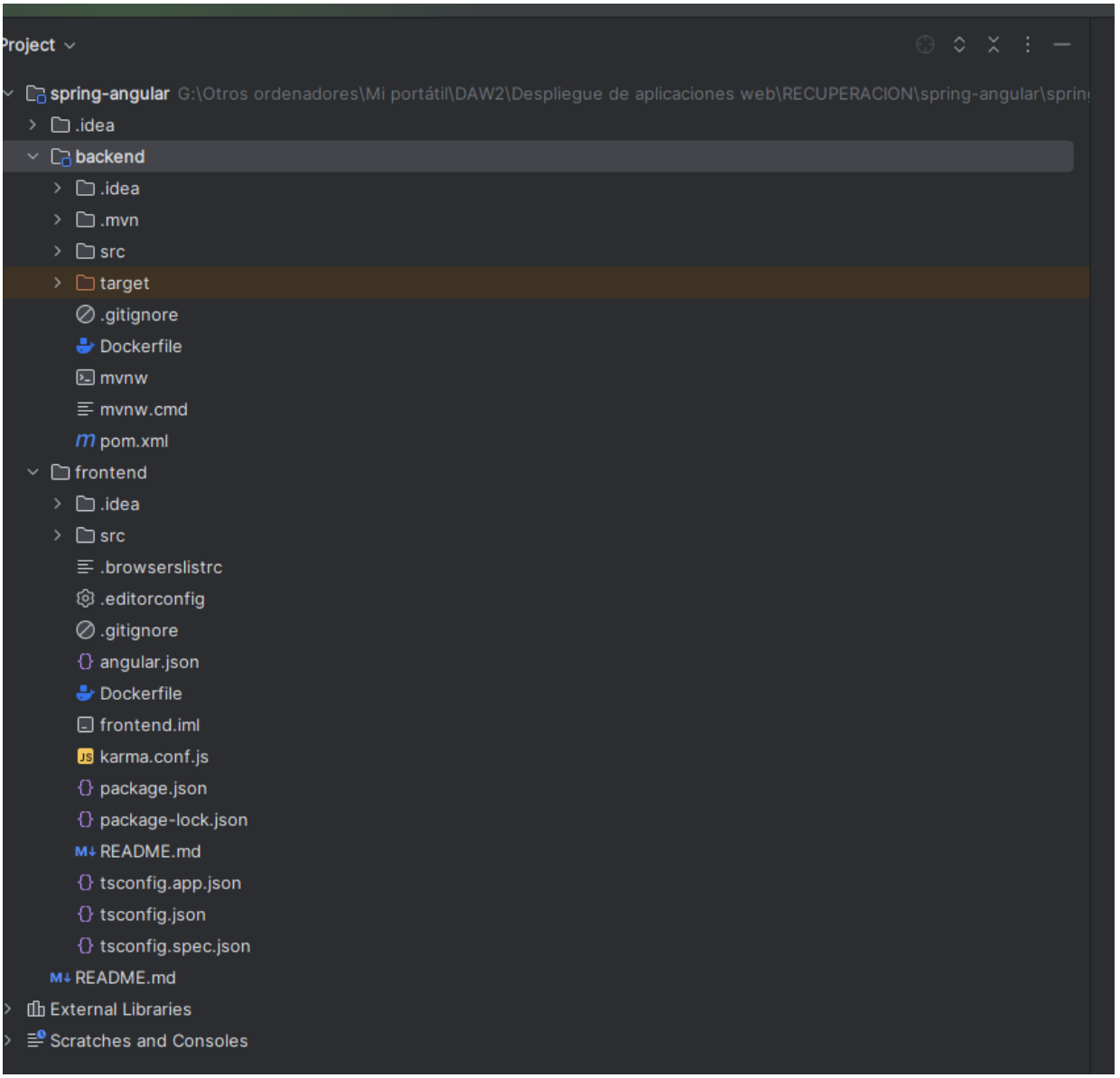


Por último comprobamos que la aplicación se haya desplegado correctamente.


Ubuntu linux [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades
Navegador web Firefox
26 de dic 09:46
MiAppBasicaAngular
localhost:4200
Welcome to mi-app-basica-angular!
Avengers:
Iron Man & Robert Downey Jr. (mocked)
'Spider-Man & Tom Holland (mocked)'
CTRL DERECHA

Despliega con Docker Compose tu app de Spring Boot, Angular y PostgreSQL

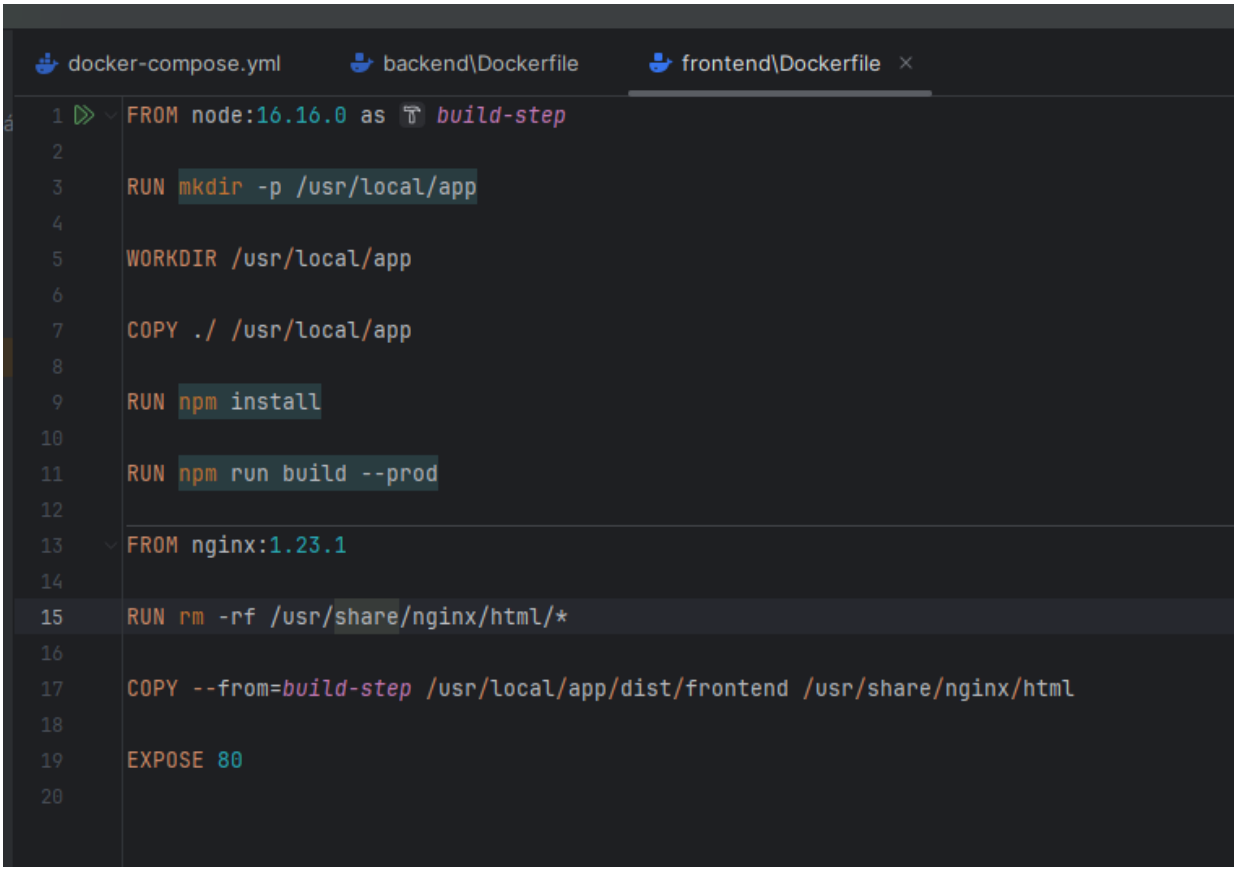
Descargo el proyecto del curso y lo importo a IntelliJ y hacemos un clean y package del proyecto



Creo los Dockerfile del frontend y del backend (Cada uno en su correspondiente carpeta de proyecto)

Backend

Frontend



Creo las imágenes del frontend y del backend (Cada uno en su correspondiente carpeta de proyecto)

Backend

```
PS G:\Otros ordenadores\Mi portátil\DAW2\Despliegue de aplicaciones web\RECUPERACION\spring-angular\spring-angular\backend> docker build -t backend:1.0 .
2023/12/26 11:03:08 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 1.0s (8/8) FINISHED
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 171B
 => [internal] load metadata for docker.io/library/eclipse-temurin:18-jre-jammy
 => [auth] library/eclipse-temurin:pull token for registry-1.docker.io
 => [internal] load build context
 => => transferring context: 3.32kB
 => [1/2] FROM docker.io/library/eclipse-temurin:18-jre-jammy@sha256:55373f54aacc86568b8427e7c1d5719e2ad08c2f66b66c2b86071cd82efd1632
 => CACHED [2/2] COPY  app.jar
 => exporting to image
 => => exporting layers
 => => writing image sha256:1b2103d219772b6958b27b23e6f56bac1f76099b44ec9fd4bc11f1796b1a25d8
 => => naming to docker.io/library/backend:1.0

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS G:\Otros ordenadores\Mi portátil\DAW2\Despliegue de aplicaciones web\RECUPERACION\spring-angular\spring-angular\backend>
```

```
pring-angular > docker-compose.yml
```

Frontend

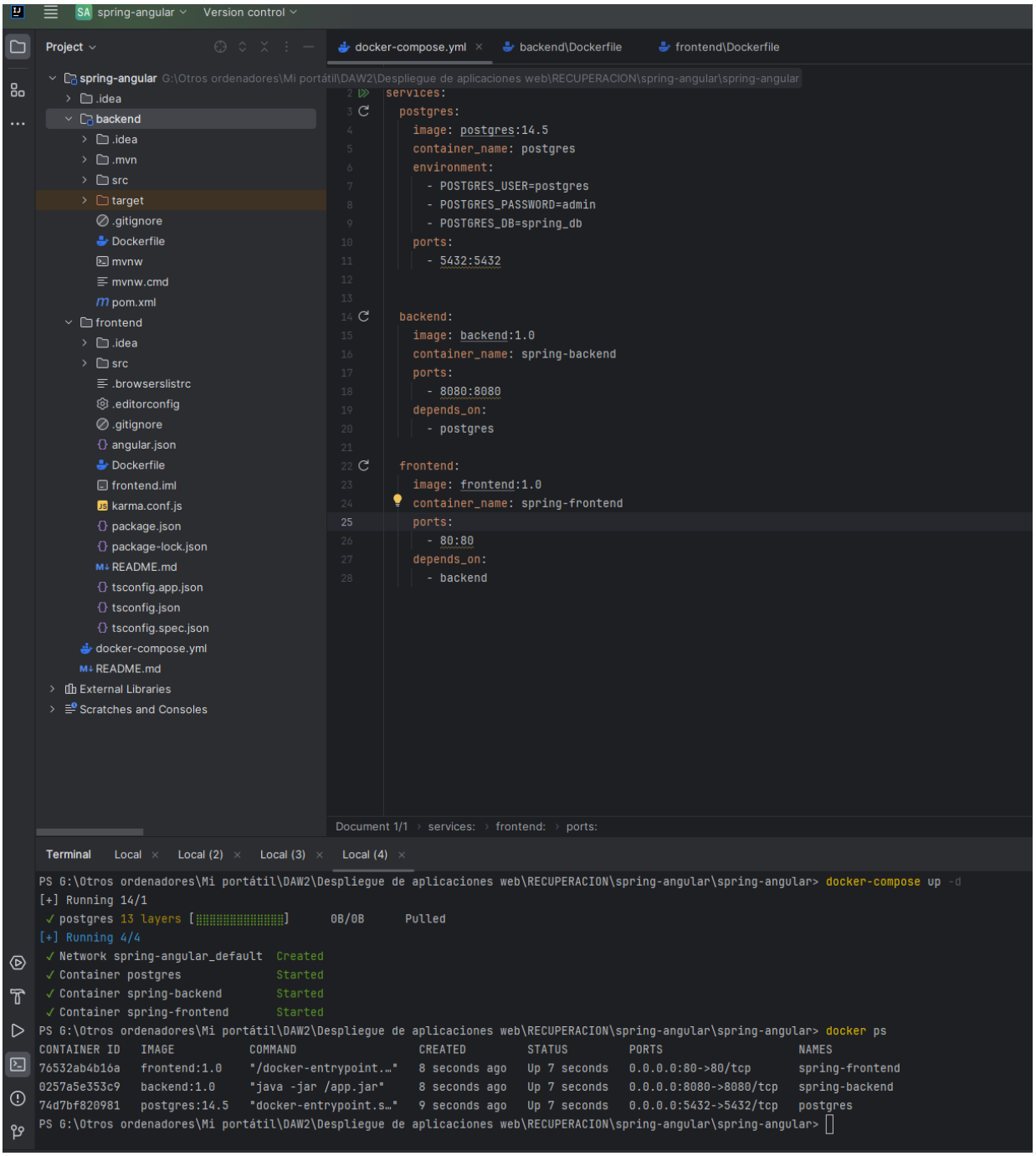
```
PS G:\Otros ordenadores\Mi portátil\DAW2\Despliegue de aplicaciones web\RECUPERACION\spring-angular\spring-angular\frontend> docker build -t frontend:1.0 .
2023/12/26 11:11:47 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 24.7s (15/15) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 352B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/nginx:1.23.1
 => [internal] load metadata for docker.io/library/node:16.16.0
 => [build-step 1/6] FROM docker.io/library/node:16.16.0@sha256:1ed1e17ccabb09038cfb8a965337ebcda51ef9e9d32082164c502d44d9731a02
 => [internal] load build context
 => => transferring context: 2.20kB
 => [stage-1 1/3] FROM docker.io/library/nginx:1.23.1@sha256:2f770d2fe27bc85f68fd7fe6a63900ef7076bc703022fe81b980377fe3d27b70
 => CACHED [stage-1 2/3] RUN rm -rf /usr/share//nginx/html/*
 => CACHED [build-step 2/6] RUN mkdir -p /usr/local/app
 => CACHED [build-step 3/6] WORKDIR /usr/local/app
 => [build-step 4/6] COPY ./ /usr/local/app
 => [build-step 5/6] RUN npm install
 => [build-step 6/6] RUN npm run build --prod
 => [stage-1 3/3] COPY --from=build-step /usr/local/app/dist/frontend /usr/share/nginx/html
 => exporting to image
 => => exporting layers
 => => writing image sha256:8ae3e89f2d7272d20cf382dd1fe343475749b8bda9efae28aa09d5e98274dbce
 => => naming to docker.io/library/frontend:1.0

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS G:\Otros ordenadores\Mi portátil\DAW2\Despliegue de aplicaciones web\RECUPERACION\spring-angular\spring-angular\frontend>
```

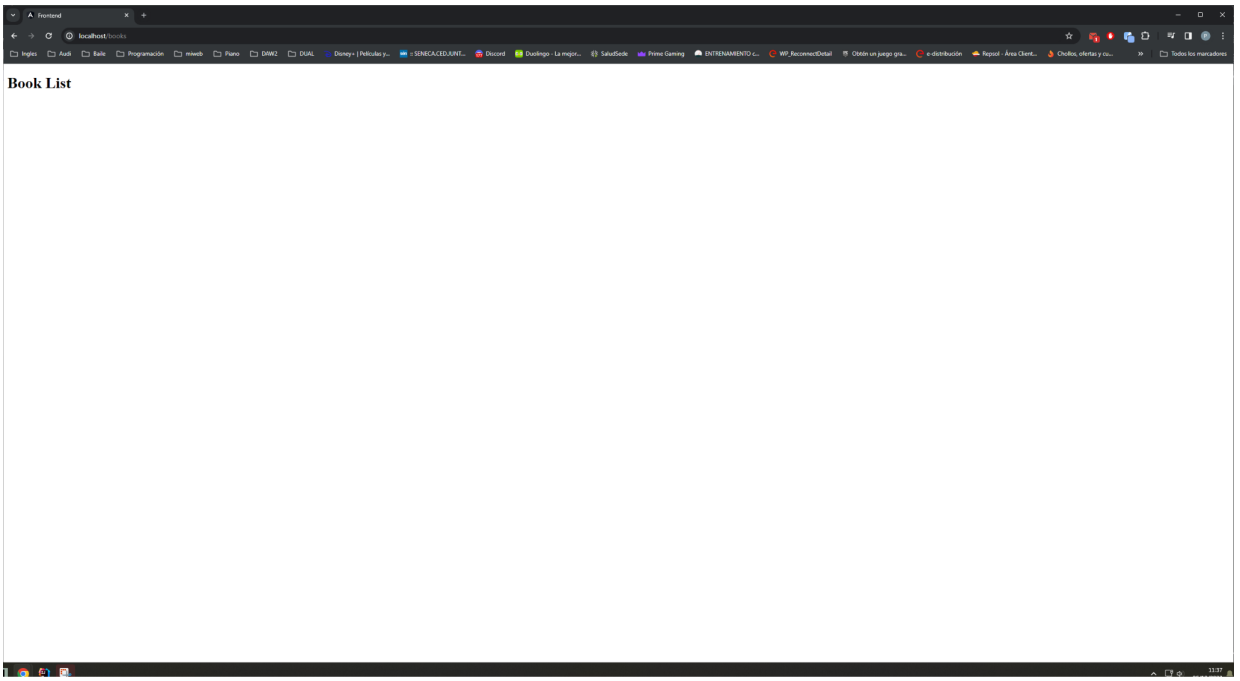
Comprobamos

```
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS G:\Otros ordenadores\Mi portátil\DAW2\Despliegue de aplicaciones web\RECUPERACION\spring-angular\spring-angular\backend> docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
backend      1.0       1b2103d21977   9 minutes ago   284MB
frontend     1.0       1b2103d21977   9 minutes ago   284MB
PS G:\Otros ordenadores\Mi portátil\DAW2\Despliegue de aplicaciones web\RECUPERACION\spring-angular\spring-angular\backend>
```

Creo el docker-compose y ejecuto el comando para levantarlo



Comprobamos que se haya desplegado:


Frontend
localhost/books
Book List