

Herramienta Rentabilidad y costes

Instalación local

Tabla de contenido

| | | |
|-----|--------------------------------------------------------------|----|
| 1. | Requisitos previos | 2 |
| 2. | Instalación del <i>back</i> | 3 |
| 2.1 | <i>Instalación del servidor para el back</i> | 6 |
| 3. | Instalación del front | 10 |
| 4. | Consideraciones para trabajar con el proyecto en local | 11 |

1. Requisitos previos

Antes de hacer funcionar las diferentes partes de la herramienta debemos de realizar una serie de pasos previos:

Mediante una herramienta de gestión de repositorios (*sourcetree*, *fork* o similar) debemos de descargarnos de la rama *master*, la última versión de cada proyecto y volver a situarnos, como mínimo en *developer*. Los proyectos implicados son:

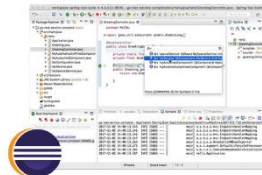
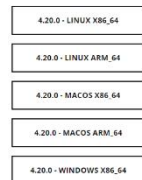
- **rentabilidadycostes** * **BACK EN JAVA**
- **rentabilidadycostesfront** * **FRONT EN VUE**

También debemos de instalar un conjunto de programas para poder modificar los proyectos. En el caso del back necesitaremos:

- **SpringToolSuite4c**. Es un *framework* de *Eclipse*, preparado con las herramientas de *SpringBoot*. Visitando <https://spring.io/tools> elegimos la opción de *Spring Tool 4 for eclipse*, para *Windows*. Si todo va bien descargarás el archivo **spring-tool-suite-4-4.20.0.RELEASE-e4.29.0-win32.win32.x86_64.self-extracting.jar**

Spring Tools 4 for Eclipse

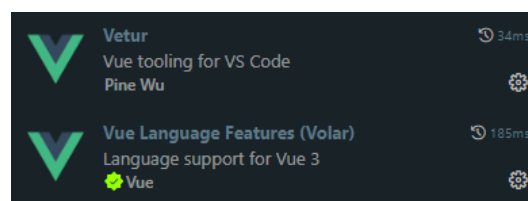
The all new Spring Tool Suite 4. Free. Open source.



- **Apache 9.0 y apache-tomcat-9.0.58**. Esta es la versión del servidor de *Tomcat* usada en el *back*.
- **ojdbc8.jar**. Librería de java necesaria para hacer funcionar el servidor. Se instala en la ruta *Tomcat9.0\apache-tomcat-9.0.58\lib*

| Documentos > Tomcat9.0 > apache-tomcat-9.0.58 > lib | | | | |
|-----------------------------------------------------|-----------------------|---------------------|----------|--|
| Nombre | Fecha de modificación | Tipo | Tamaño | |
| catalina-tribes.jar | 06/09/2023 14:07 | Executable Jar File | 331 KB | |
| ecj-4.20.jar | 06/09/2023 14:07 | Executable Jar File | 3.061 KB | |
| el-api.jar | 06/09/2023 14:07 | Executable Jar File | 88 KB | |
| jasper.jar | 06/09/2023 14:07 | Executable Jar File | 558 KB | |
| jasper-el.jar | 06/09/2023 14:07 | Executable Jar File | 169 KB | |
| jspic-api.jar | 06/09/2023 14:07 | Executable Jar File | 27 KB | |
| jsp-api.jar | 06/09/2023 14:07 | Executable Jar File | 63 KB | |
| ojdbc8.jar | 30/08/2023 9:59 | Executable Jar File | 4.923 KB | |

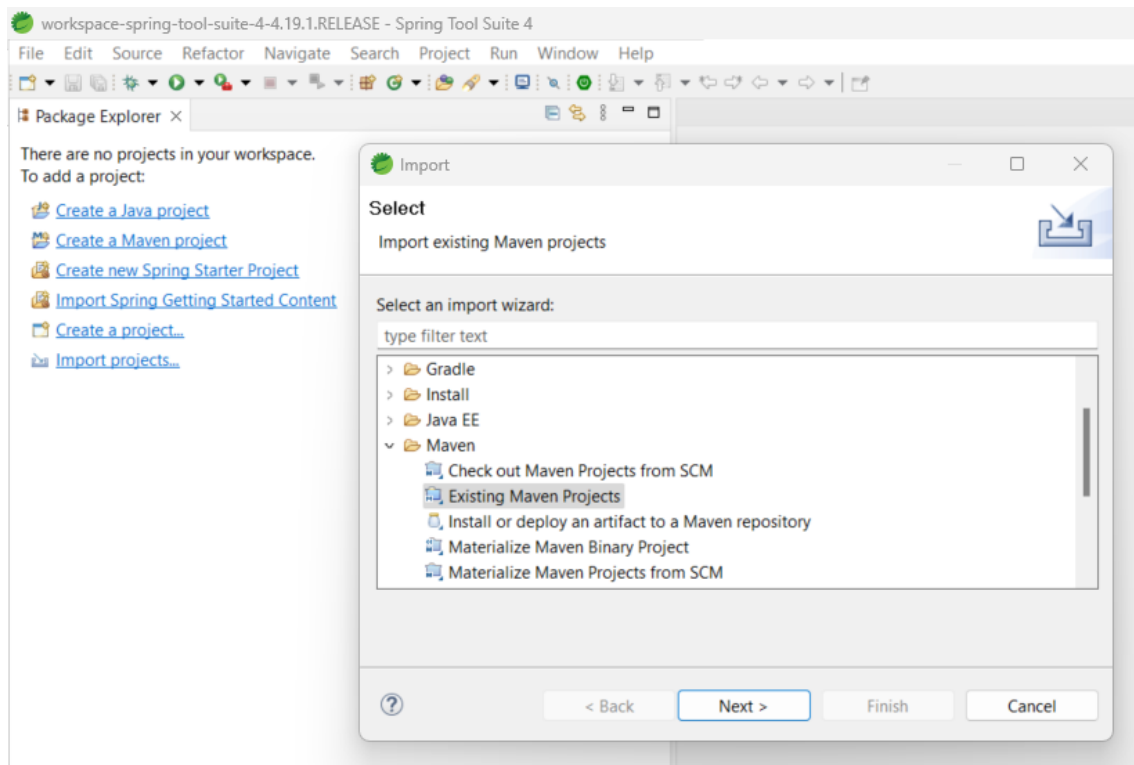
- **Visual studio code**. En este *framework* es donde trabajaremos con el *front*. Debes de instalar las extensiones de Vue que necesites. Como poco **Vetur**, **Vue Lenguaje Features (Volar)** y **npm**.



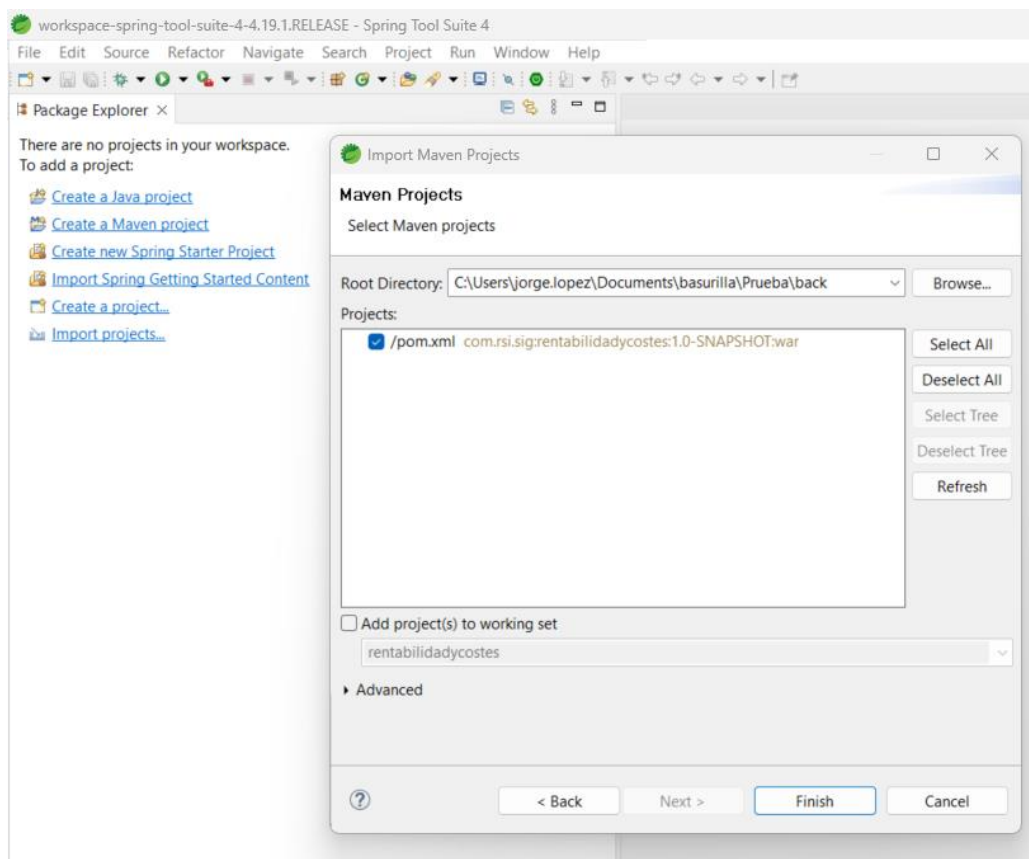
Con todo descargado he instalado, pasamos a importar y configurar cada parte.

2. Instalación del *back*

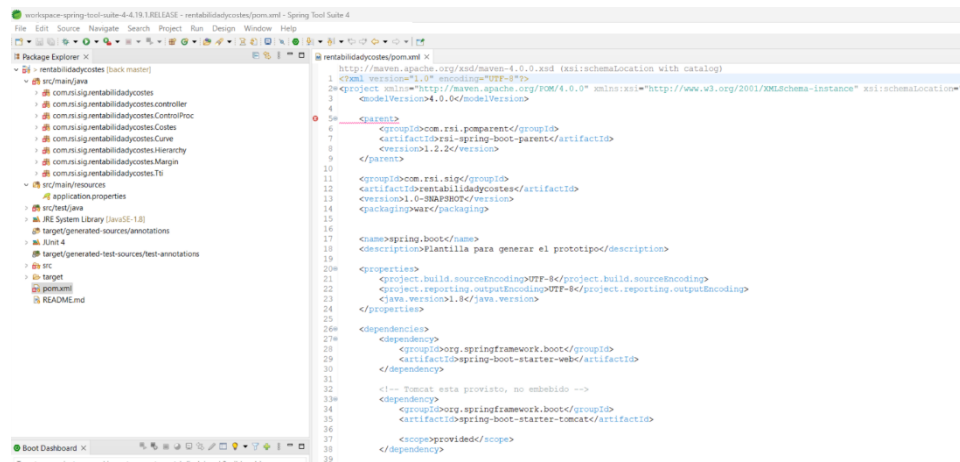
Tras los pasos previos, abrimos **SpringToolSuite4c**, pinchamos en **Import projects..** y seleccionamos **Maven**, dentro, **Existing Maven Projects**:



Tras pulsar *Next* seleccionamos el directorio donde se encuentra el proyecto del *back*, debería de seleccionar el archivo **pom.xml**, por defecto:



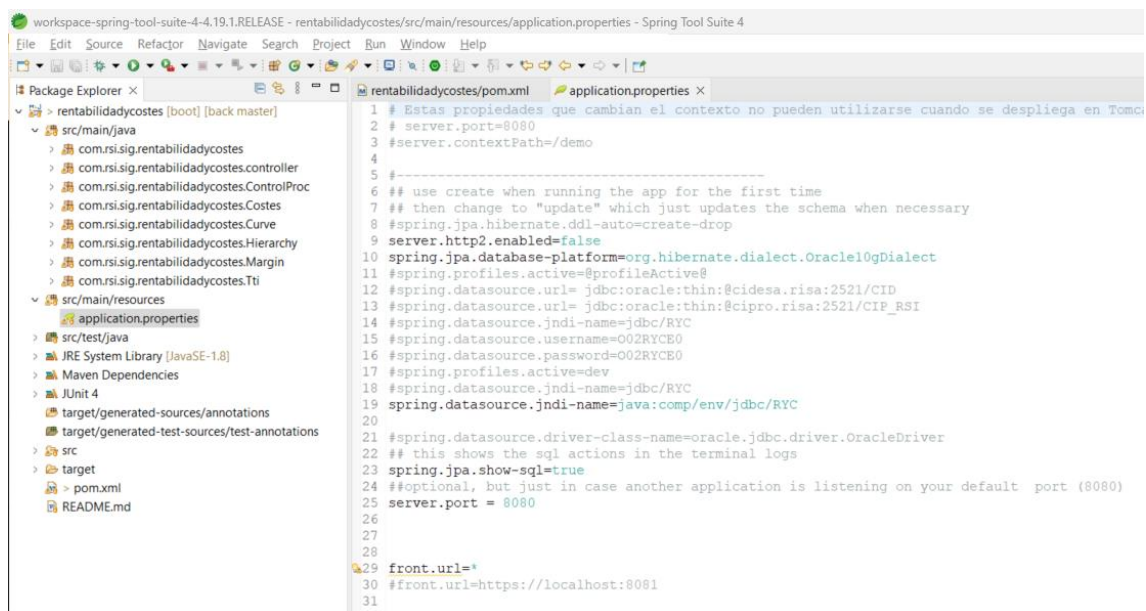
Al finalizar esta importación tendremos el proyecto importado, pero con multitud de errores:



Lo primero que vamos a hacer es abrir el archivo **pom.xml**, para modificarlo, ya que tiene un error a la hora de llamar al componente **parent**. Además, le falta una dependencia relacionada con los test. La parte de **parent** podemos cambiar toda la que hay, por la que vamos a incluir a continuación y la dependencia nueva se incluye dentro de la etiqueta **dependencies**, por norma general al final tras la última incluida:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.7.4</version>
  <relativePath />
.....
</parent>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
</dependency>
```

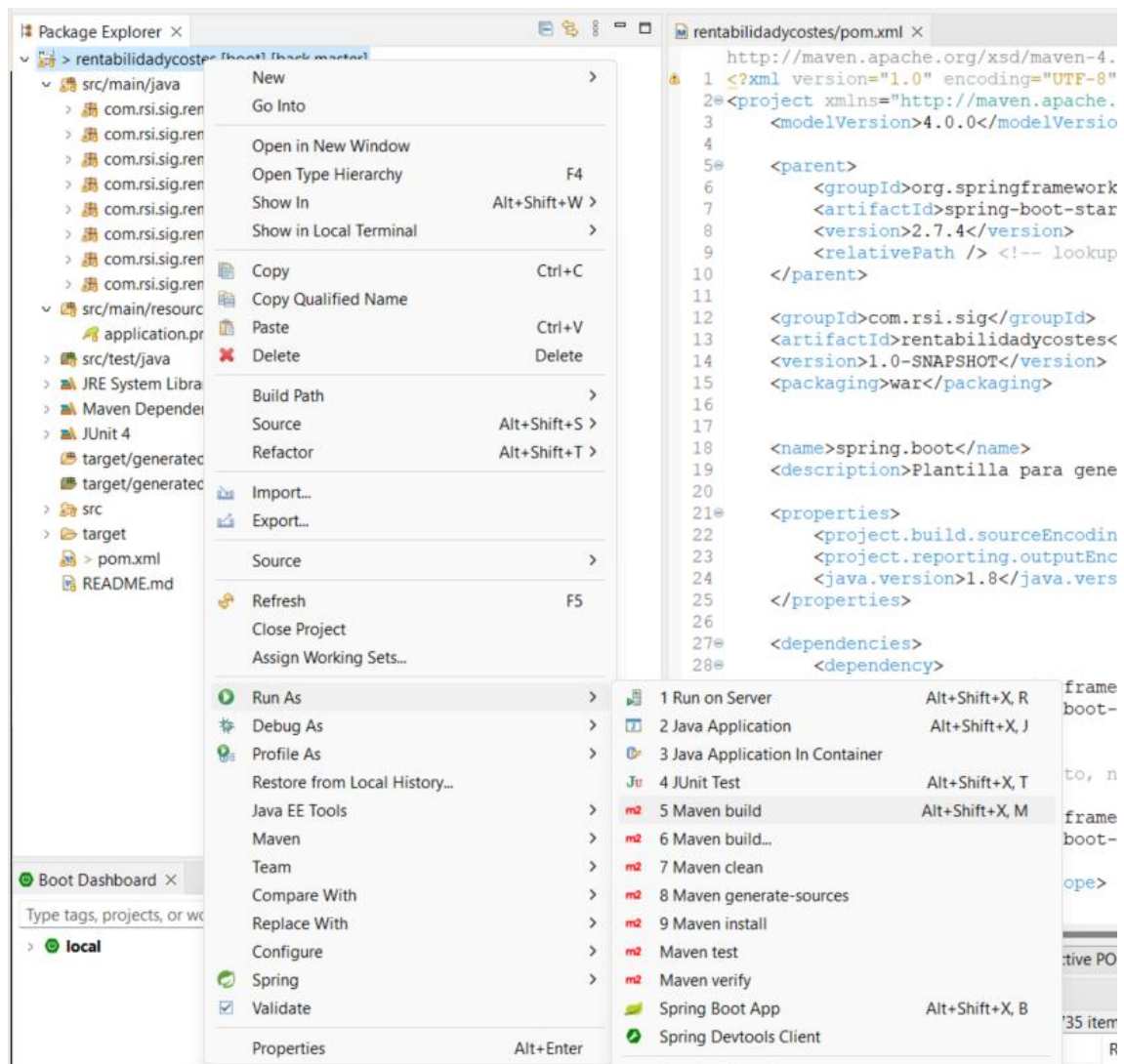
Otro archivo que debemos de modificar es **Application.properties**:



Podemos prescindir de la mayoría de las líneas existentes, si se quiere se pueden comentar. Las que necesitaremos son:

```
server.http2.enabled=false
spring.jpa.database-platform=org.hibernate.dialect.Oracle10gDialect
spring.jpa.show-sql=true
front.url=https://localhost:8080
spring.datasource.jndi-name=java:comp/env/jdbc/Ryc
```

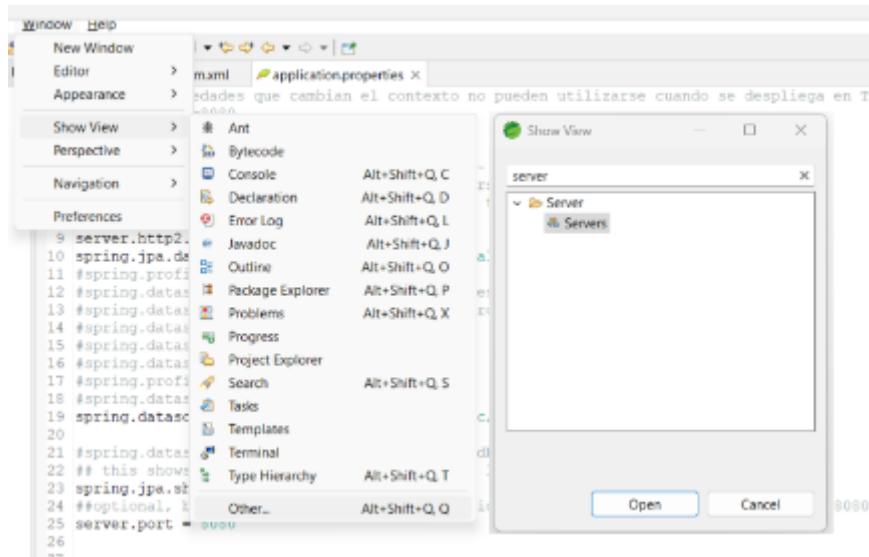
Al modificar el **pom.xml**, es posible que desaparezcan los errores, pero, de todas formas, vamos a ejecutar los siguientes comandos, por si se nos hubiese escapado algo o en el caso de seguir con errores. Nos situamos en la raíz del proyecto y con botón derecho, vamos a **Run as, Maven install**(opción 9) y al finalizar **Maven clean**.



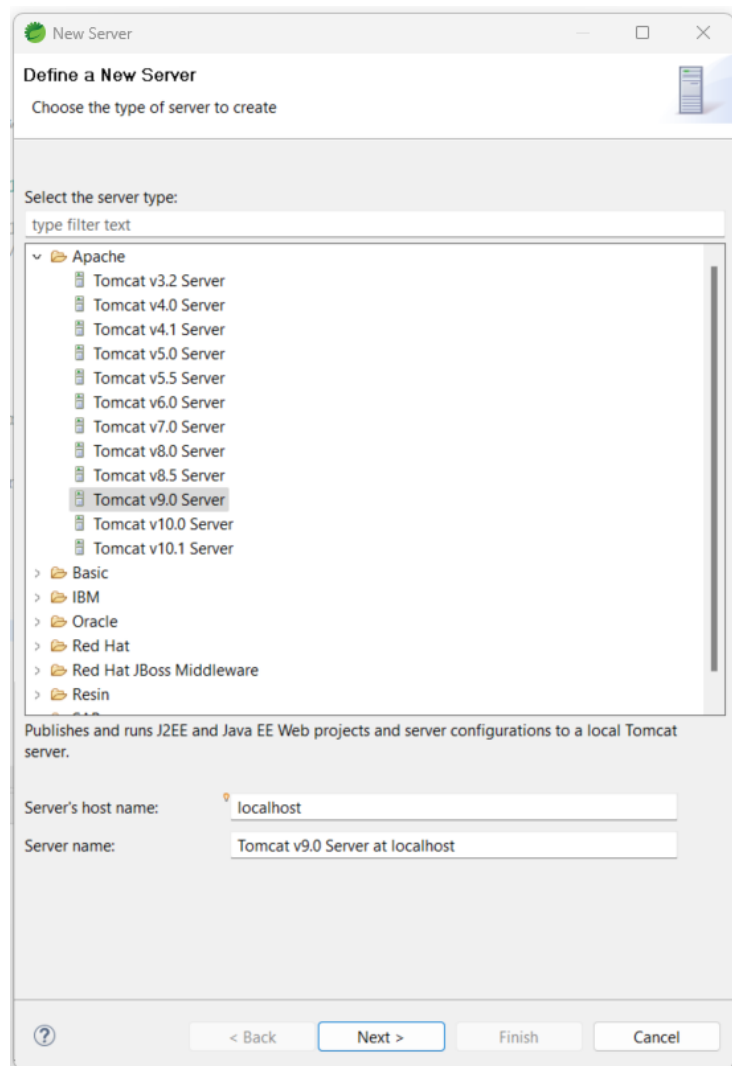
2.1 Instalación del servidor para el back

Una vez configurado el proyecto, tenemos que instalar un servidor que levante el servicio. En esta ocasión vamos a usar un servidor Tomcat. En concreto la versión **apache-tomcat-9.0.58**.

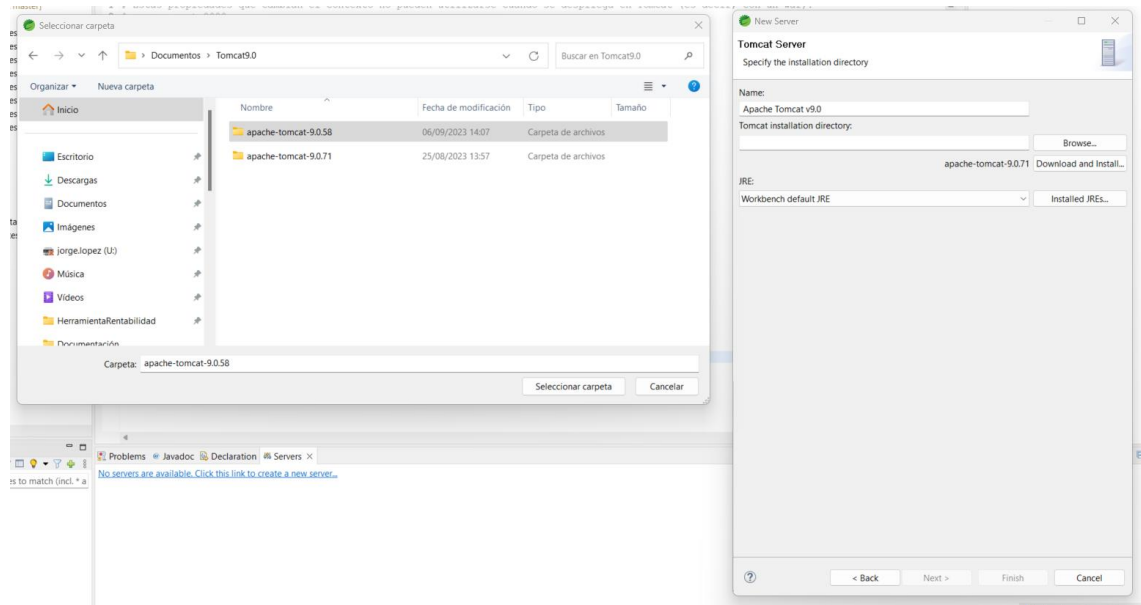
Para mostrar la pestaña de servidores en el programa, iremos a **window, show view, other** y en la ventana que nos abra escribimos **server**:



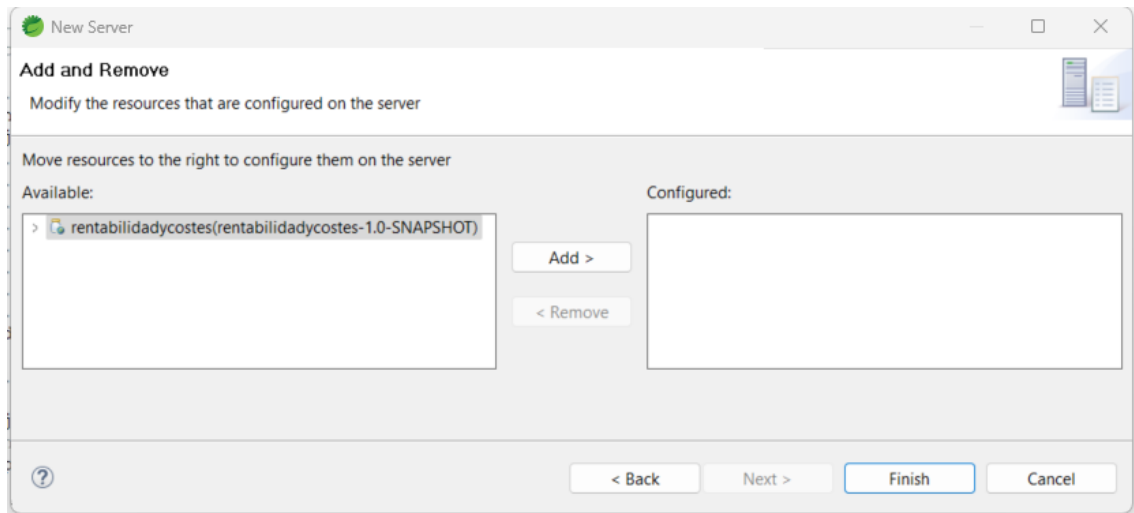
En la nueva pestaña que se abre, elegimos un nuevo **server, Apache, Tomcat v9.0 Server**



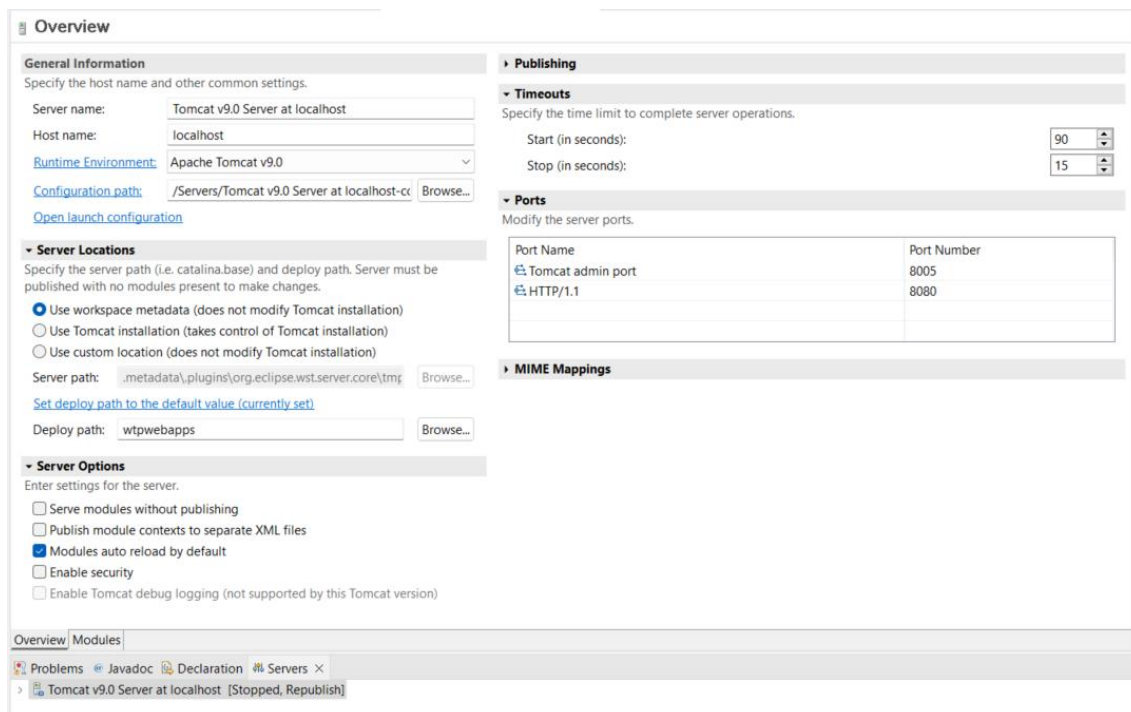
En el siguiente paso, en **Browse**, navegamos hasta la versión del servidor que queremos, en este caso la **9.0.58**:



En el nuevo paso seleccionamos el proyecto que vamos a incluir en el servidor:

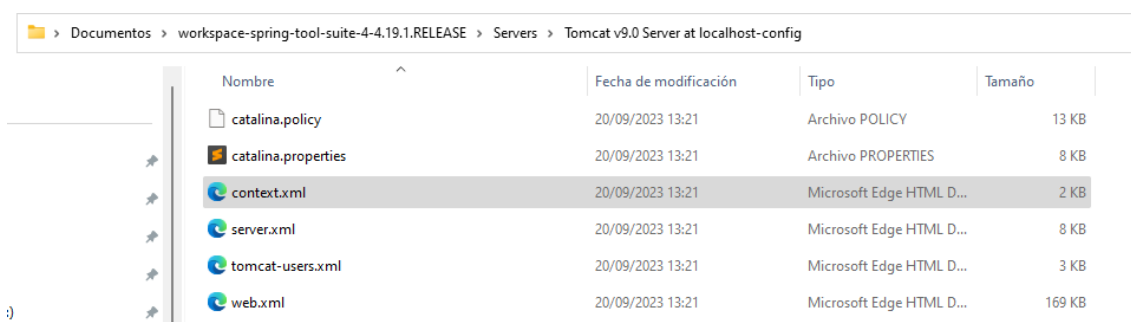


Tras finalizar, habremos completado la creación del servidor, ahora vamos a terminar de modificar algunos aspectos de este. Pulsado dos veces sobre el nombre del servidor abrimos su menú de opciones. Deberemos de cambiar, en la zona de **Timeouts**, el valor **start** a **90**, ya que el inicial es muy bajo. En esta zona de opciones generales es donde también podremos cambiar el puerto http del servidor. En la parte de **Ports**, si desplegamos tenemos indicado el puerto **8080** por defecto, lo vamos a modificar y se usará el puerto **8081**. El anterior lo reservaremos para el *front*.



Por último, se tiene que configurar un archivo **context.xml**, con los valores necesarios para conectarse a la base de datos en el que vaya a trabajar la herramienta. En este caso está configurado para trabajar en desarrollo. Es más útil tener tantos archivos **context.xml**, como entornos de trabajo se necesiten, que andar modificando un solo archivo **context.xml**

Para añadir este archivo tenemos que ir al **workspace** de **springbbot**, y en la carpeta de **server**, buscar el archivo **context.xml**. Podemos editarlo o pisarlo.



Estas son las líneas que se tienen que incluir en el caso de editar el archivo, para la configuración en **CID**:

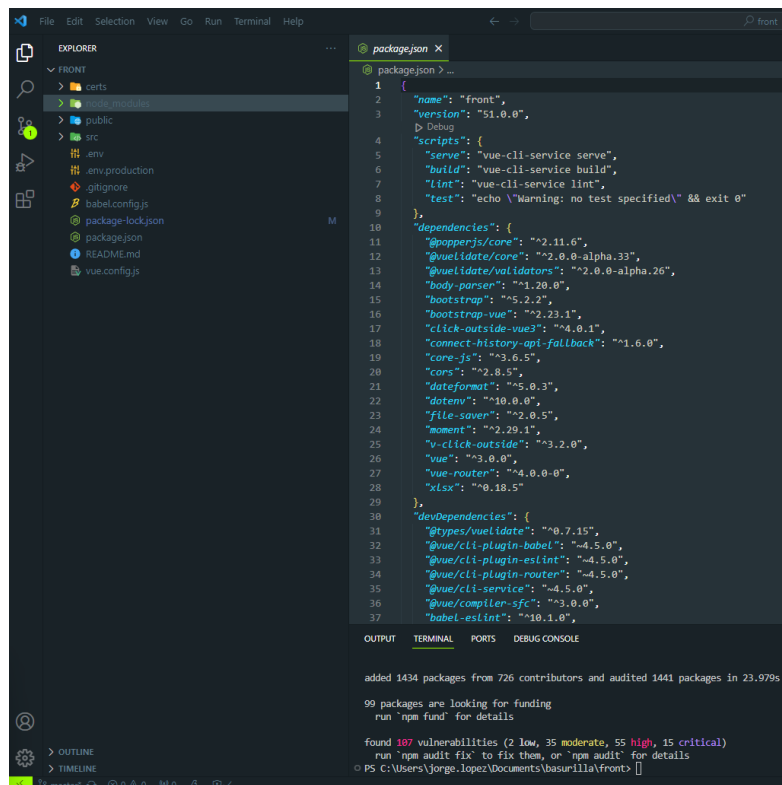
```
<Resource name="jdbc/RVC" auth="Container"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    type="javax.sql.DataSource"
    factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" minIdle="1"
    initialSize="1" maxActive="10" minEvictableIdleTimeMillis="900000"
    removeAbandoned="false" logAbandoned="true"
    abandonedUsageTracking="false" username="O02RYCE0" password="O02RYCE0"
    jmxEnabled="true" validationQuery="select 1 from DUAL"
    validationInterval="10000" testWhileIdle="true" testOnBorrow="true"
    testOnReturn="false"
    url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=cide
sa.risa)(PORT=2521)))(CONNECT_DATA=(SERVICE_NAME=CID_RSI)))" />
```

Tras este último paso, solo nos queda guardar todo, y probar a levantar el servicio. Botón derecho en el servidor, y la opción **start**.

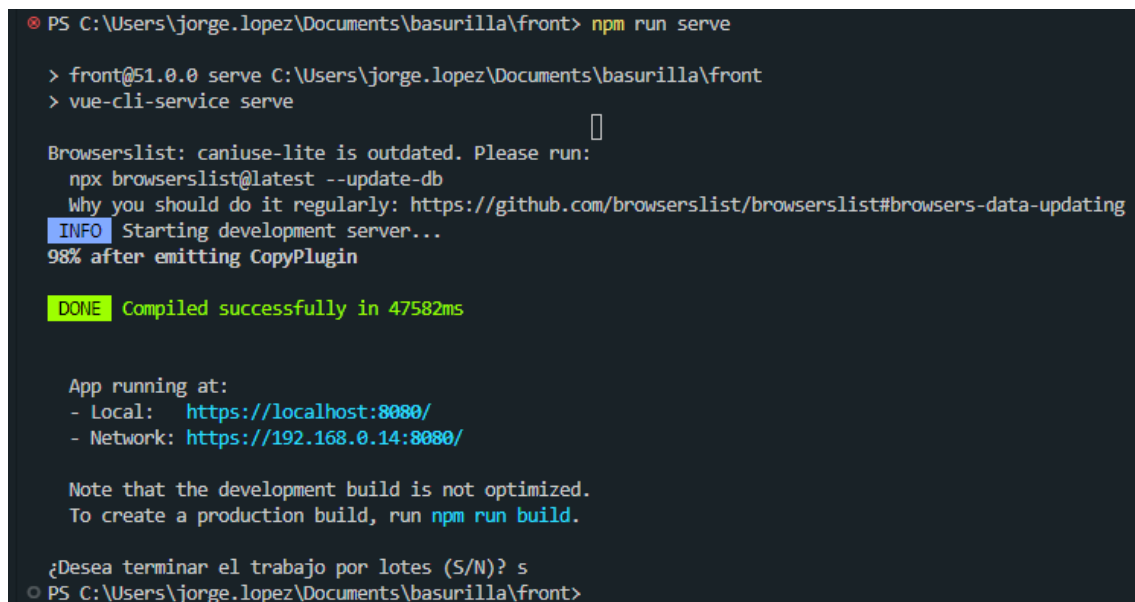
[illegible]

3. Instalación del front

Tras los pasos iniciales, abrimos el proyecto y seguidamente un terminal nuevo. Si todo va bien, el terminal se abrirá en la raíz del proyecto. Ahora ejecutamos **npm i** para instalar todas las dependencias necesarias.

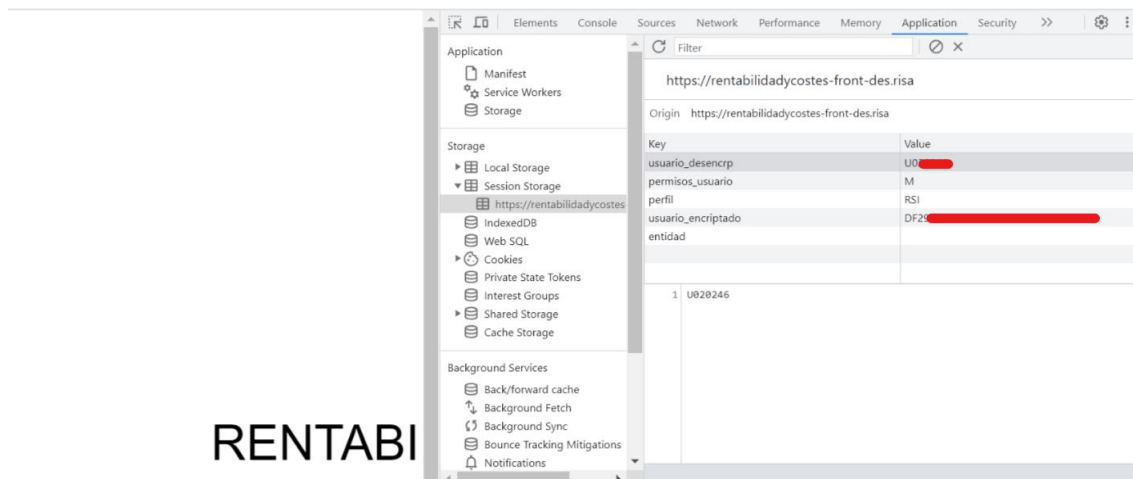


Finalizaremos, levantando el servicio con el comando **npm run serve** y si todo es correcto nos indicará ruta y puerto a en el que entrar.

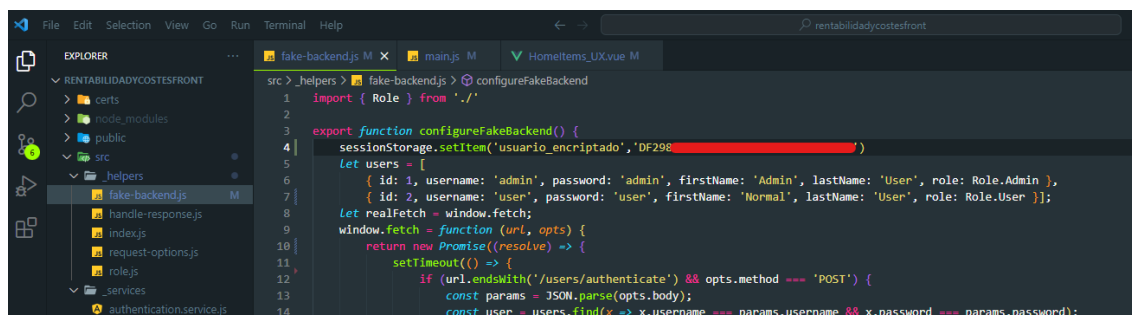


4. Consideraciones para trabajar con el proyecto en local

Aunque todo está preparado y funcionando, existe un problema. Por como está planteada la herramienta, para poder movernos por las distintas vistas, incluida la inicial, el programa recupera el usuario encriptado que deja Pórtico. Al levantar el servicio en local, este paso se omite, por lo que hay que falsear este paso y dar un usuario encriptado. La manera más sencilla de obtener este dato es entrar desde Pórtico a la herramienta y recuperar el dato del usuario encriptado del apartado *sesión storage*.



Una vez se recupere el dato, deberemos de modificar el primer método que es llamado en el *front*, para desencriptar este usuario. El método se llama ***configureFakeBackend()*** y se encuentra en el archivo ***fake-backend.js***.



De esta manera podremos movernos con la herramienta sin problema alguno.