



SnowyDune

Manual de Instalación y Despliegue

Proyecto Desarrollo de Aplicaciones Web

Oria de Rueda Pérez, Pablo
14-12-2020



ÍNDICE

1. Requerimientos previos de Instalación.	2
2. Instalación del sistema.	3
2.1 Instalación de persistencia.	3
2.2 Instalación de contenedor de aplicaciones y servidor local Apache Tomcat.	6
3. Requerimientos previos al Despliegue.	9
4. Despliegue del sistema.	13
4.1 Contenedores y servicios.	13
4.2 Conexión con repositorio.	15
4.3 Logs del sistema.	16
4.4 Primera puesta en servicio.	16



1. Requerimientos previos de Instalación.

Para llevar a cabo la instalación de los distintos elementos que componen nuestros sistemas, debemos contar con una serie de requisitos previos básicos de los que poder comenzar sin problemas. En estas explicaciones se tiene en cuenta que el usuario que realiza la instalación tiene unos conocimientos básicos de informática, siendo importante tener cierta soltura con un entorno de Linux y sus comandos básicos.

Los requisitos son los siguientes:

- De Hardware.
 - o Contar con un equipo que al menos disponga de 20-25 Gbs de espacio de almacenamiento libre.
 - o Contar con al menos 2 Gbs de memoria RAM.
 - o Contar con un procesador de al menos 300 Mhz x 86.
 - o Disponer de conexión estable a Internet.
- De Software.
 - o Tener un sistema operativo instalado, en el presente documento se van a realizar las instalaciones sobre un sistema operativo Ubuntu 20.04, que se trata de una distribución de Linux basada en Debian.
Es conveniente que el sistema se encuentre actualizado para mayor seguridad.
 - o Disponer de un navegador web, preferiblemente Chrome o Firefox, en sus versiones más actualizadas.



2. Instalación del sistema.

Teniendo instalado previamente el sistema operativo Ubuntu 20.04, la instalación se centra en 3 puntos fundamentales.

- Instalación de persistencia.

Será el primer paso a realizar, la persistencia se va a llevar a cabo en nuestro proyecto empleando una base de datos MySQL y para su gestión se emplea una herramienta visual de diseño de bases de datos como MySQL Workbench que integra desarrollo de software, administración de bases de datos, diseño, gestión y mantenimiento para el sistema de base de datos MySQL.

- Instalación de contenedor de aplicaciones.

En nuestro caso, este paso viene embebido junto con el siguiente, ya que nuestro servidor local va a incluir un contener de Servlets/JSP llamado Catalina.

- Instalación de servidor local Apache Tomcat.

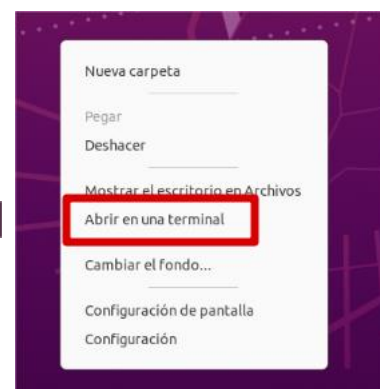
Se va a realizar la instalación de un contener web con soporte de servlets, que nos va a permitir manejar contenido web tanto estático como dinámico.

2.1 Instalación de persistencia.

Como hemos comentado anteriormente, la persistencia se va a conseguir mediante una base de datos MySQL, para realizar la instalación en Ubuntu 20.04 debemos seguir los siguientes pasos.

- Primero, accedemos a la consola de Ubuntu y actualizamos el índice de paquetes.

```
usuario@usuario-Standard-PC-i440FX-PIIX-1996:/var/lib/tomcat9$ sudo apt update
```





- A continuación, instalamos el paquete mysql-server.

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo apt install mysql-server
```

Este comando instalará MySQL, pero no le solicitará establecer una contraseña ni realizar otros cambios de configuración. Debido a que esto deja su instalación de MySQL sin proteger, debemos realizar el siguiente paso.

- Configuramos MySQL.

Ejecutamos el siguiente comando para llevar a cabo cambios en las opciones de seguridad de la instalación de MySQL.

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo mysql_secure_installation
```

Se nos solicitará el nivel de seguridad que deseamos en nuestra contraseña y también la contraseña que deseamos. Recuerda que esta contraseña debe cumplir el nivel de seguridad, de lo contrario, no será válida.

- Ajustar autenticación y privilegios de usuario (opcional).

Ejecutamos el siguiente comando y entramos en mysql para poder cambiar nuestros datos y privilegios.



```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 141
Server version: 8.0.22-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Mediante la siguiente modificación, cambiamos la contraseña de nuestro usuario y A continuación, indicamos al servidor que vuelva a cargar la tabla de permisos y aplique sus nuevos cambios.



```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '1Q2w3e4r';
```

```
mysql> FLUSH PRIVILEGES;
```

- Por último, vamos a comprobar que MySQL está en funcionamiento sin problemas.

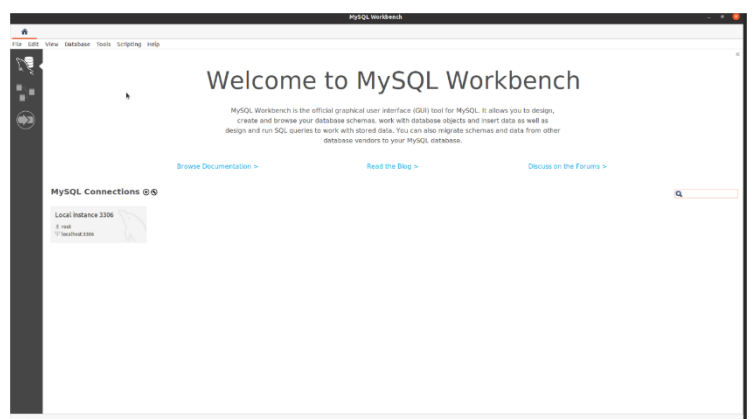
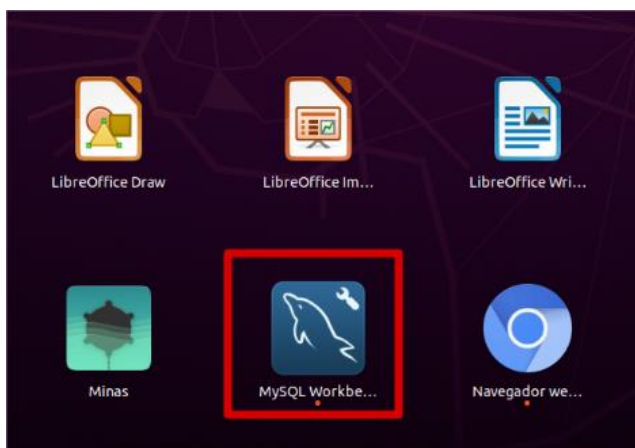
```
usuario@usuario-Standard-PC-i440FX-PIIX-1996:/var/lib/tomcat9$ sudo systemctl status mysql.service
[sudo] contraseña para usuario:
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-12-09 13:37:04 CET; 4h 4min ago
     Main PID: 789 (mysqld)
       Status: "Server is operational"
        Tasks: 50 (limit: 7068)
       Memory: 385.7M
        CGroup: /system.slice/mysql.service
               └─789 /usr/sbin/mysqld

dic 09 13:36:59 usuario-Standard-PC-i440FX-PIIX-1996 systemd[1]: Starting MySQL Community Server...
dic 09 13:37:04 usuario-Standard-PC-i440FX-PIIX-1996 systemd[1]: Started MySQL Community Server.
```

- Con el servidor de persistencia instalado, vamos a instalar la herramienta visual MySQL Workbench, para ello simplemente introducimos el siguiente comando.

```
usuario@usuario-Standard-PC-i440FX-PIIX-1996:/var/lib/tomcat9$ sudo apt install mysql-workbench
```

Dejamos que se descargue e instale y ya podemos iniciarlo.





2.2 Instalación de contenedor de aplicaciones y servidor local Apache Tomcat.

Como se ha explicado anteriormente, la instalación del contenedor de aplicaciones viene de la mano de nuestro servidor local Tomcat, por tanto, se ha decidido unificar la instalación de los componentes.

Para realizar la instalación se requieren los siguientes pasos.

- Se actualizan nuevamente las listas de paquetes.

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo apt update
```

- Se instalan los paquetes principales, correspondientes al núcleo de Tomcat y a las aplicaciones administrativas.

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo apt install tomcat9 tomcat9-admin
```

- Si se desea instalar adicionalmente la aplicación de la documentación y algunos ejemplos de aplicaciones básicas funcionando con Tomcat 9 puede hacerlo instalando los siguientes paquetes.

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo apt install tomcat9-docs tomcat9-examples
```

- Después de realizar la instalación se comprueba el estado del servicio Tomcat 9 empleando el siguiente comando.

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo systemctl status tomcat9
[sudo] contraseña para usuario:
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-12-09 13:36:59 CET; 4h 25min ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Main PID: 667 (java)
    Tasks: 40 (limit: 7068)
   Memory: 929.3M
   CGroup: /system.slice/tomcat9.service
           └─667 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/conf/logging.properties -Djava.awt.headless=true -Dcom.sun.management.jmxremote.port=8082 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.local.only=false -Djdk.jmx.remote.enable=true -jar /usr/share/tomcat9/bin/bootstrap.jar -Dcatalina.base=/usr/share/tomcat9 -Dcatalina.home=/usr/share/tomcat9 -Djava.io.tmpdir=/usr/share/tomcat9/temp
```

- A continuación, en caso de tener activado el firewall en Ubuntu 20.04, se debe configurar y añadimos la siguiente regla (el puerto debe ser el que vayamos a emplear).

```
usuario@usuario-Standard-PC-l440FX-PIIX-1996:/var/lib/tomcat9$ sudo ufw allow 8082/tcp
```

- Ahora se debe configurar Tomcat 9 antes de realizar prueba alguna, ya que se necesitará realizarlas para trabajar con la aplicación porque no basta con la configuración por defecto.



```
usuario@usuario-Standard-PC-i440FX-PIIX-1996:/var/lib/tomcat9$ sudo nano /etc/tomcat9/server.xml
```

```
usuario@usuario-Standard-PC-i440FX-PIIX-1996:/var/lib/tomcat9$ sudo nano /etc/tomcat9/tomcat-users.xml
```

```
sudo nano /usr/share/tomcat9-admin/manager/META-INF/context.xml
```

```
-->
<Context antiResourceLocking="false" privileged="true" >
  <!--
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" />
  -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer
</Context>
```

```
1996:/var/lib/tomcat9$ sudo systemctl restart tomcat9
```




- Si la configuración ha sido satisfactoria, ya se puede ver en acción Tomcat 9, accediendo desde un navegador indicando la dirección IP y el puerto de conexión, en el caso del ejemplo es el 8082.



Obteniendo la siguiente pantalla.

It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/README.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.

Con todo esto, se ha finalizado la instalación del contenedor y servidor local Apache Tomcat.



3. Requerimientos previos al Despliegue.

Los principales requisitos previos al despliegue emanan del cumplimiento de las instalaciones explicadas anteriormente. Con las instalaciones realizadas, se necesitarán los archivos necesarios para realizar su despliegue, en este caso se van a necesitar un archivo .war, que va a permitir desplegar la parte de Spring destinada al Back-end y una carpeta con los archivos construidos para un entorno de producción que obtenemos de Angular.

Se va a explicar los pasos para construir el archivo war y la carpeta de Angular necesaria a partir del proyecto original, pero esos archivos se van a poder obtener directamente del repositorio de GitHub destinado a tal uso.

Para descargar los archivos directamente puede realizar en el siguiente enlace:

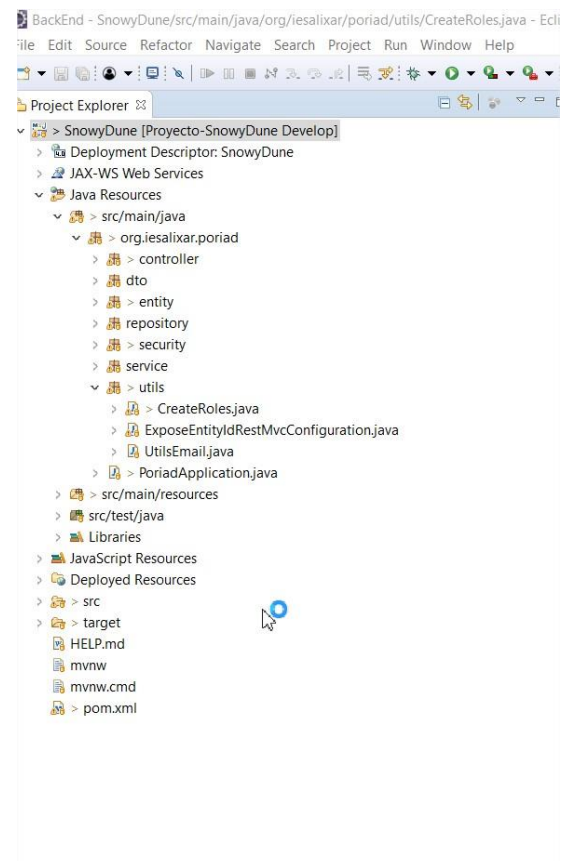
<https://github.com/poriad/Proyecto-SnowyDune/tree/master/docs>

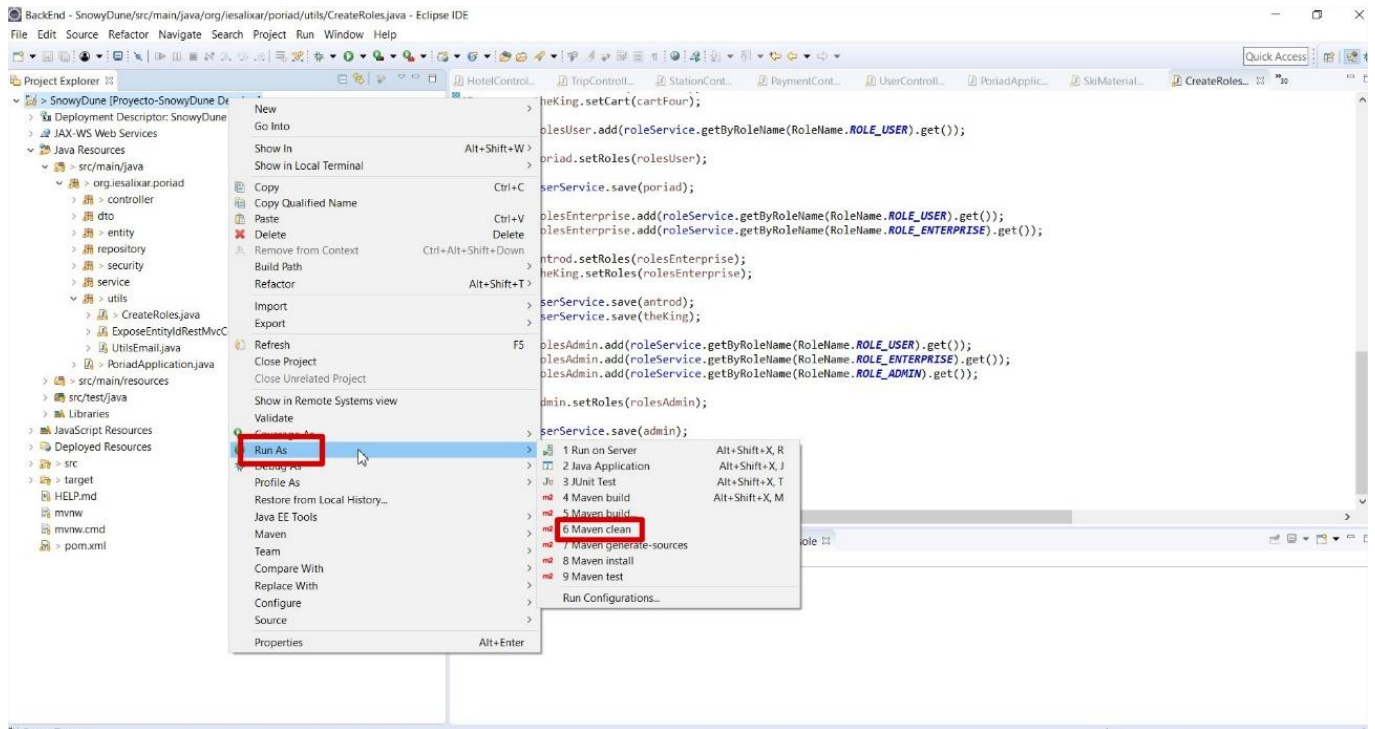
Ahora procedemos al proceso de generación de los documentos.

- Archivo War (Back-end).

Para generar el archivo necesario, primero debemos abrir un IDE en el que se importará el proyecto.

Con el proyecto abierto, se hace click derecho sobre la carpeta principal y se selecciona "Run as" y "Maven clean". De esta forma eliminamos los archivos que existan de forma que no podamos encontrarnos problemas inesperados.

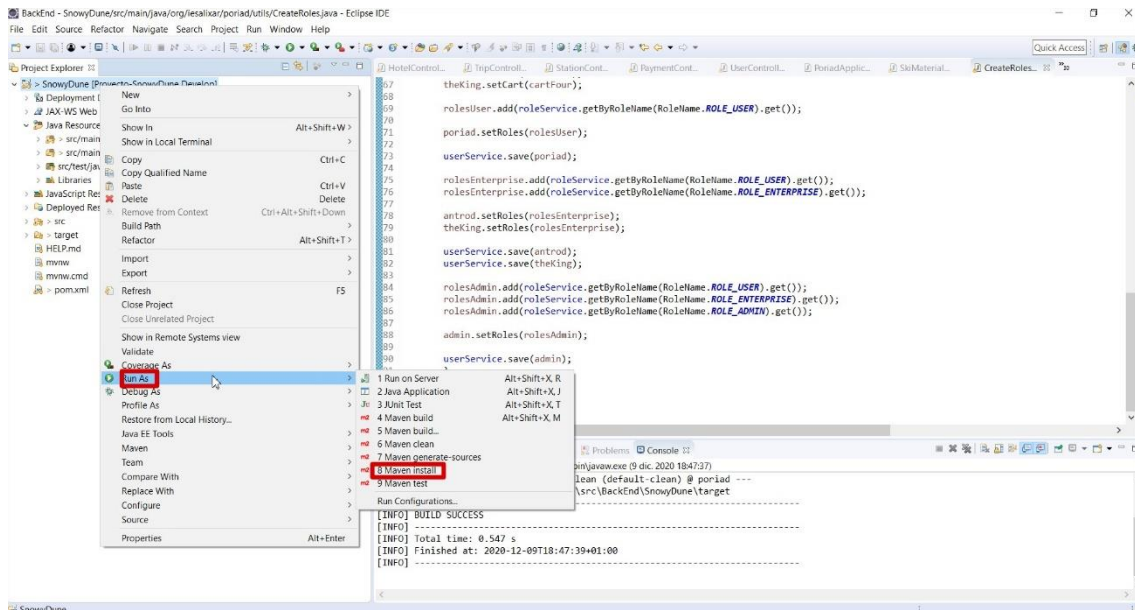




Realizado ese paso, se mostrará por consola el siguiente mensaje si todo ha ido correctamente.

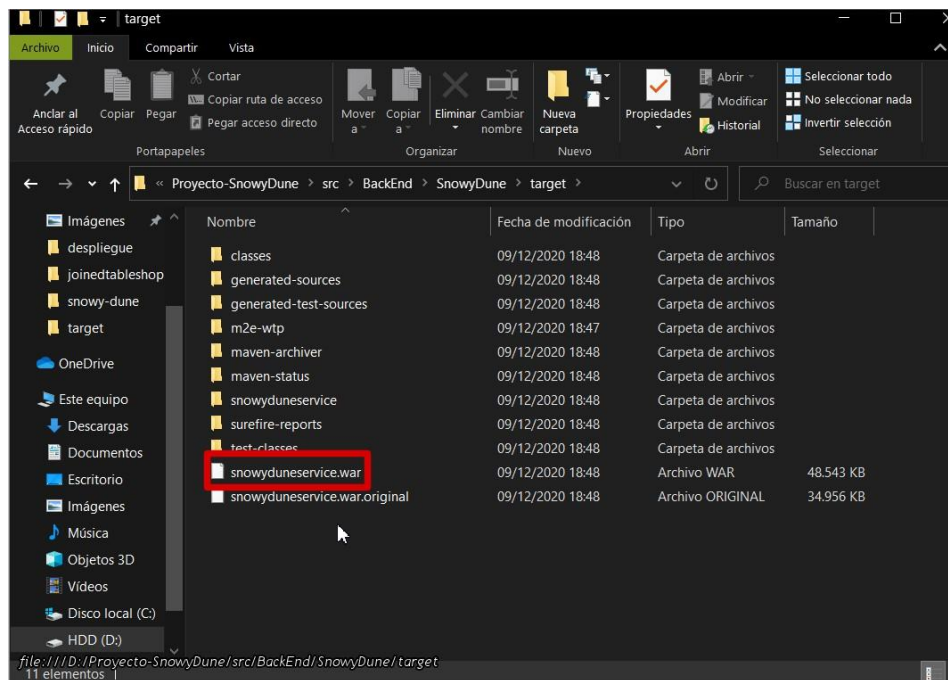
```
<terminated> C:\Program Files\Java\jdk1.8.0_271\bin\javaw.exe (9 dic. 2020 18:47:37)
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ poriad ---
[INFO] Deleting D:\Proyecto-SnowyDune\src\BackEnd\SnowyDune\target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.547 s
[INFO] Finished at: 2020-12-09T18:47:39+01:00
[INFO] -----
```

A continuación, se realiza un paso similar al anterior, click derecho sobre la carpeta principal del proyecto, "Run as" y "Maven install". Con esto se generará el archivo war y los distintos paquetes de Maven.



Una vez hecho, se muestra por consola lo siguiente en caso de no existir problemas y se crea el siguiente archivo en la carpeta "target" del proyecto.

```
Terminated C:\Program Files\Java\jdk-10.0.2\bin\java.exe (9 dic. 2020 18:48:10)
[INFO] Installing D:\Proyecto-SnowyDune\src\BackEnd\SnowyDune\target\snowyduneservice.war to C:\Users\johny\
[INFO] Installing D:\Proyecto-SnowyDune\src\BackEnd\SnowyDune\pom.xml to C:\Users\johny\.m2\repository\org\ie
[INFO] BUILD SUCCESS
[INFO] Total time: 17.945 s
[INFO] Finished at: 2020-12-09T18:48:37+01:00
[INFO]
```



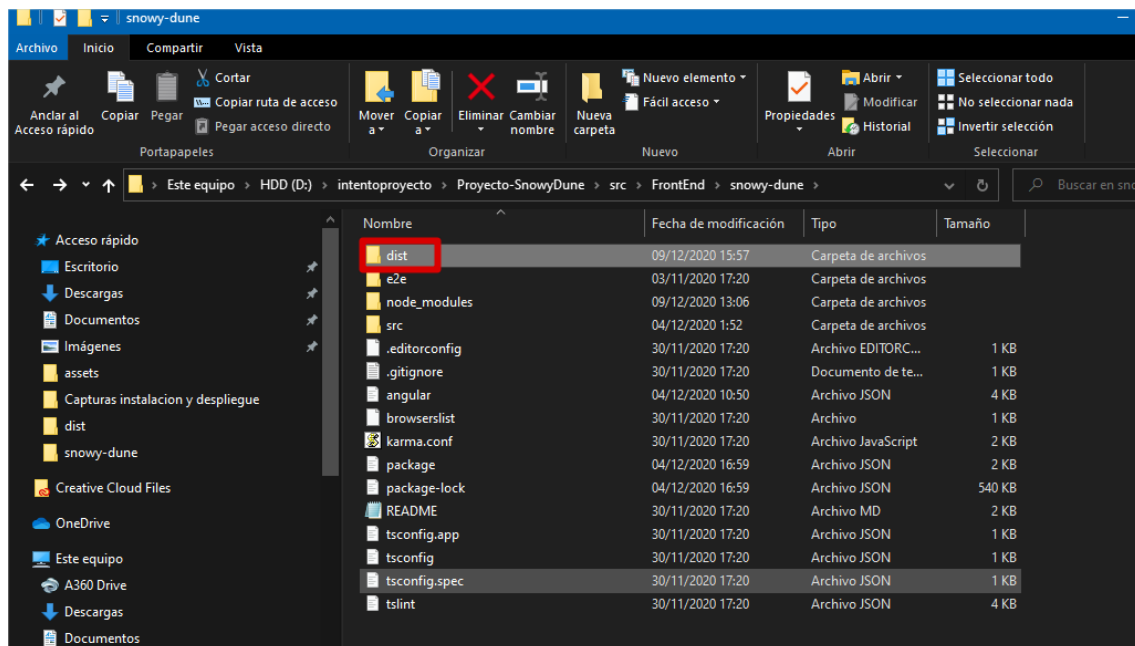


En el caso de la carpeta de Angular, simplemente ejecutamos el comando “ng build --prod” y creará una carpeta llamada “dist” que contiene la información necesaria para llevar a cabo el despliegue.

```
Date: 2020-12-09T14:53:25.138Z - Hash: c8d9a1986ed855b034ab - Time: 34957ms
PS D:\intentoproyecto\Proyecto-SnowyDune\src\FrontEnd\snowy-dune> ng build --prod
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {2} polyfills-es2015.2320f4dbebd0efa7fb56.js (polyfills) 36.8 kB [initial] [rendered]
chunk {3} polyfills-es5.c036c089e5e061e8f4ce.js (polyfills-es5) 130 kB [initial] [rendered]
chunk {0} runtime-es2015.0dae8cbc97194c7caed4.js (runtime) 1.45 kB [entry] [rendered]
chunk {0} runtime-es5.0dae8cbc97194c7caed4.js (runtime) 1.45 kB [entry] [rendered]
chunk {1} main-es2015.ecc8232a5a28cfd254e7.js (main) 1.28 MB [initial] [rendered]
chunk {1} main-es5.ecc8232a5a28cfd254e7.js (main) 1.44 MB [initial] [rendered]
chunk {4} styles.4479ce214ce9b9fc0a12.css (styles) 214 kB [initial] [rendered]
chunk {scripts} scripts.3cf90f8fc92130239739.js (scripts) 169 kB [entry] [rendered]
Date: 2020-12-09T14:56:31.842Z - Hash: d94715735b3c711090d4 - Time: 76265ms

PS D:\intentoproyecto\Proyecto-SnowyDune\src\FrontEnd\snowy-dune>
```





4. Despliegue del sistema.

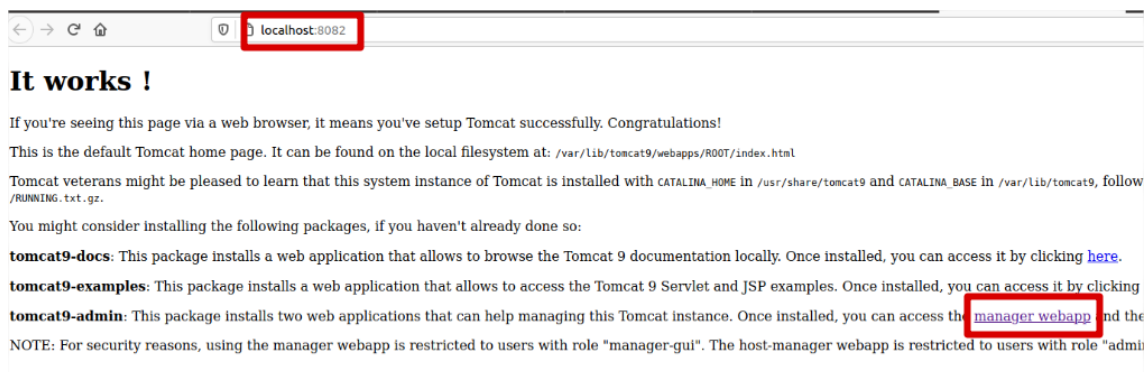
Con los archivos obtenidos anteriormente se procede al despliegue se procede del Back-end y del Front-end, empleando Apache Tomcat como servidor de aplicaciones.

4.1 Contenedores y servicios.

El despliegue se va a realizar en una máquina virtual PROXMOX, por tanto, no se contempla el uso de contenedores con servicios asociados.

Se va a realizar la instalación de Apache Tomcat y MySQL como servicios necesarios para poder realizar el despliegue.

Para gestionar Apache Tomcat, se accede a la url indicada y se pulsa en "manager webapp", solicitando claves de autorización para ingresar en la gestión (el nombre de usuario y la contraseña se corresponden con las introducidas en las modificación de la configuración de los puntos de instalación de Tomcat 9).

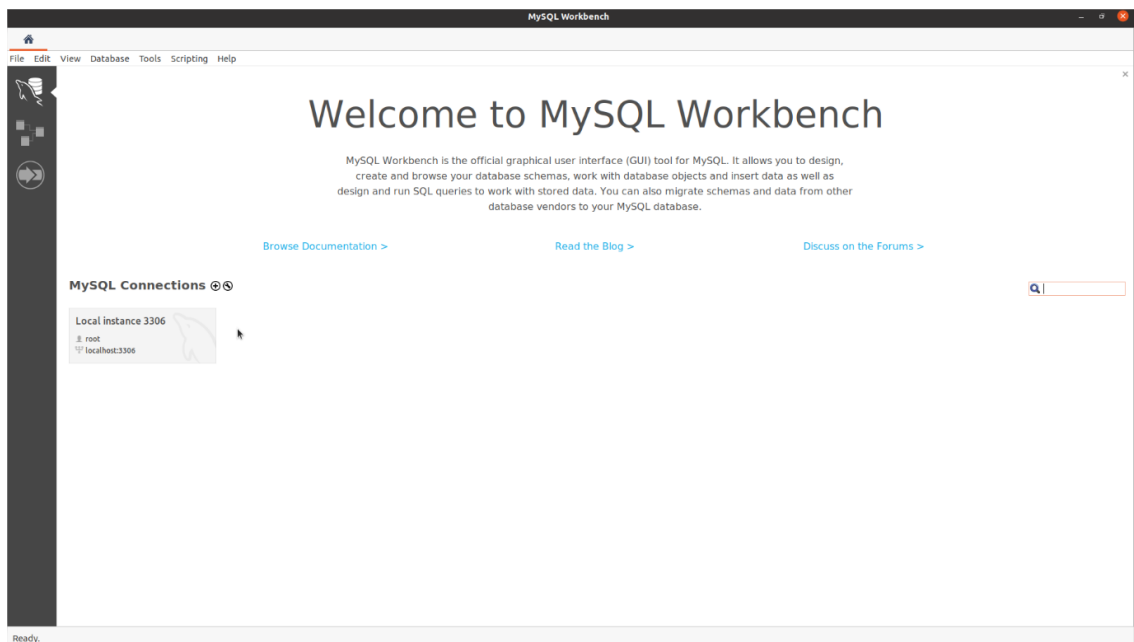


Dentro tenemos el primer servicio necesario, el gestor de aplicaciones web de Tomcat.

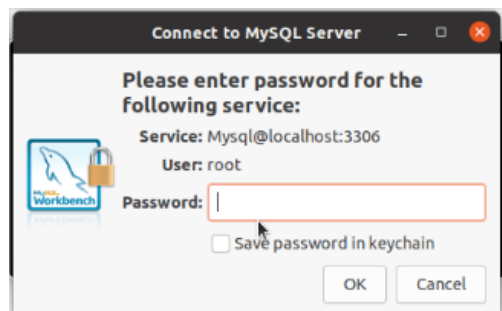


Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar a: 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar a: 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar a: 30 minutos
/snowy-dune	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar a: 30 minutos
/snowydunehservice	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar a: 30 minutos

El segundo servicio necesario, es el de MySQL, el cual se gestiona a través de MySQL Workbench.

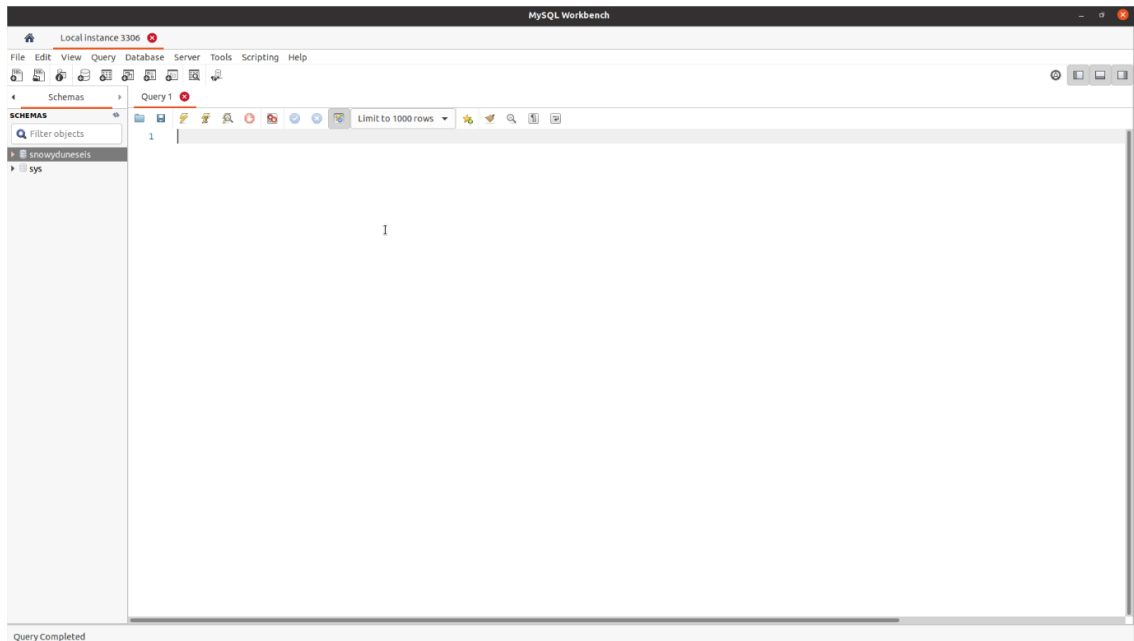


Se accede a la conexión con los datos procedentes de la instalación.



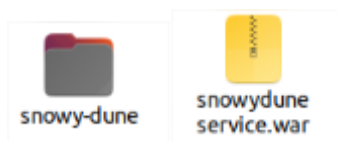


Con esto accedemos a la base de datos, y ya están listos los servicios que se van a emplear en el despliegue



4.2 Conexión con repositorio.

El despliegue no se encuentra íntimamente ligado con un repositorio, sin embargo, los documentos necesarios para el despliegue se van a poder encontrar en GitHub en el repositorio indicado en la sección “Requerimientos previos al Despliegue”. En el se puede encontrar tanto el archivo war, la carpeta “dist” creada en Angular y un archivo SQL el cual contiene datos para el correcto funcionamiento de la aplicación.

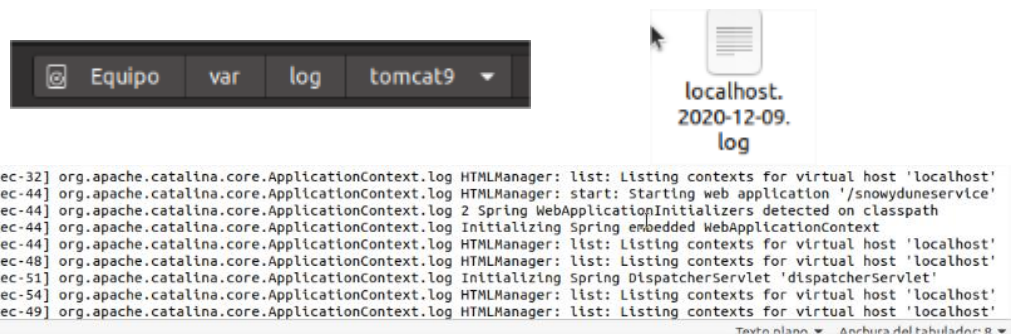




4.3 Logs del sistema.

En lo que se refiere a almacenamiento de información relativa al funcionamiento de la aplicación, se han implementado logs en el servidor empleando la librería log4j de Java, de forma que van a ser visibles y registradas las acciones realizadas por los usuarios a lo largo de la vida de uso del aplicativo, estos archivos de logs se guardan automática en la carpeta indicada para dicha acción, en un futuro, esos archivos se borrarán o se gestionarán de forma que no acumule demasiado espacio de almacenamiento en el servidor.

Además, Tomcat implementa logs por defecto, los cuales se pueden observar en la carpeta de Tomcat 9 en la ruta que se muestra a continuación.



4.4 Primera puesta en servicio

A continuación se va a realizar la primera puesta en servicio del aplicativo.

Para ello, se debe cumplir con todo lo expuesto tanto en los apartados de instalación como posteriormente en los apartados de despliegue.

Lo primero que se va a realizar, es la puesta en servicio de nuestro Back-end, para ello se necesita el archivo war que nos va a suministrar los servicios al Front-end.

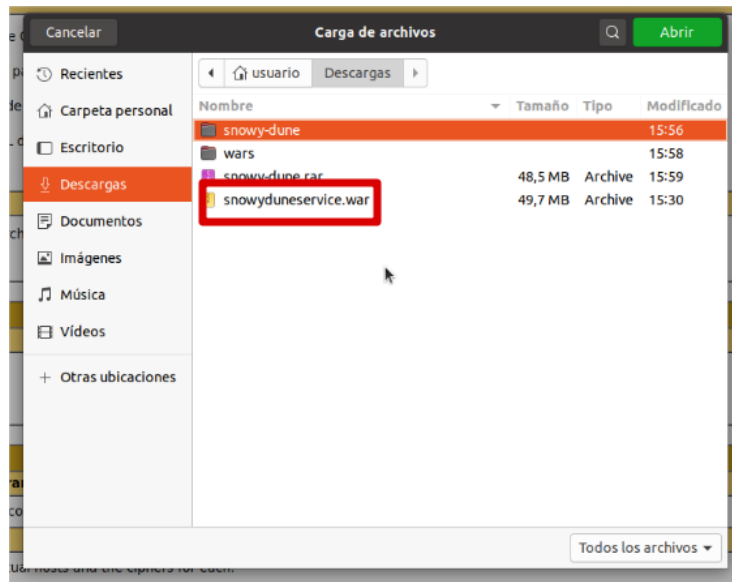
Para ello, en la página de gestión de aplicaciones web de Tomcat, se pulsa en el botón “Examinar” en la sección “Archivos WAR a Desplegar”.





Seleccione archivo WAR a cargar No se ha seleccionado ningún archivo.

Y escogemos el archivo WAR de nuestra aplicación.



Una vez realizado, se pulsa en "Desplegar". Se procederá al despliegue, esto puede tardar unos segundos y se mostrará un mensaje de "Ok".

Seleccione archivo WAR a cargar snowyduneservice.war

Mensaje:	OK
-----------------	----

Ahora ya dentro de la sección "Aplicaciones", nos aparece la ruta añadida al servidor.

Aplicaciones		
Ruta	Versión	
/	Ninguno especificado	
/host-manager	Ninguno especificado	Tomcat H
/manager	Ninguno especificado	Tomcat M
/snowy-dune	Ninguno especificado	
/snowyduneservice	Ninguno especificado	

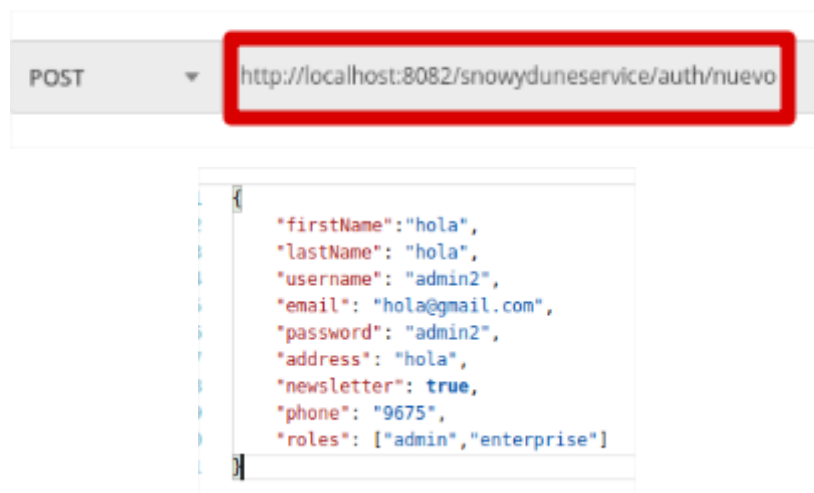


El siguiente paso que se va a realizar es la pruebas de que los servicios se encuentran en funcionamiento. Para ello se han realizado peticiones desde un software diseñado para ello, en concreto Postman.

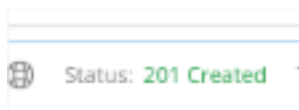


Se va a realizar el registro de un usuario y realizar el login, todo a nivel de Back-end. Si el funcionamiento es correcto, se procederá a desplegar el Front-end.

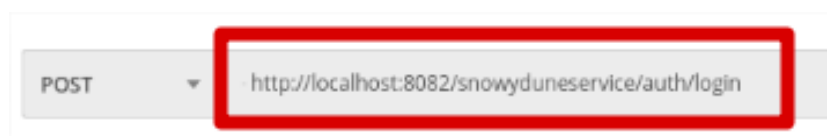
Primero se realiza una petición POST para crear un usuario, indicando como respuesta un body que se corresponde con un JSON.



Obtenemos una respuesta favorable.



Y ahora se va a realizar el Login en la aplicación, pasando como parámetros el nombre de usuario y la contraseña. La respuesta debe ser un token de autenticación, que será el que empleará el Front-end para obtener autorización en cada petición.





El body de respuesta:

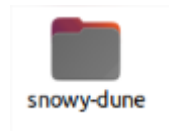
```
1 {  
2   "username": "admin",  
3   "password": "admin1"  
4 }
```

Y la respuesta del servidor devuelve lo que esperábamos, por tanto, se va a continuar con el despliegue del Front-end.

Status: 200 OK

```
"token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZGpbiIsImhhdCI6MTYwNzUyNDcyMiwiZXhwIjoxNjA3NTM0NzIyTQ..0Ltox3estnxPpYirheU8W848nbz--VjtIGP9SIUz4SUFcx13Gmupx_9mofCn0-lWV3cGt_uPz6Gak5PopKCQ",  
"bearer": "Bearer",  
"userName": "admin",
```

Para el despliegue del Front-end, necesitamos la carpeta que se encuentra dentro de "dist", se encuentra disponible en el repositorio y la necesitamos para los siguientes pasos.



Con la carpeta, simplemente la copiamos en la siguiente ruta de Tomcat.



Realizado este paso, se nos va a mostrar la ruta en el gestor de aplicaciones web de Tomcat.

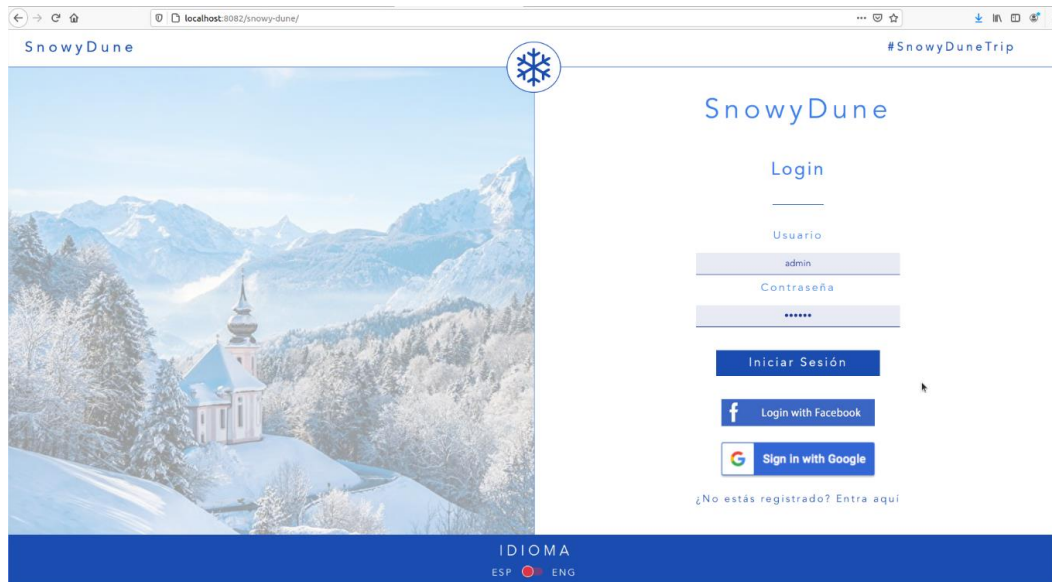
Aplicaciones	
Ruta	Versión
/	Ninguno especificado
/host-manager	Ninguno especificado
/manager	Ninguno especificado
<u>/snowy-dune</u>	Ninguno especificado
/snowyduneservice	Ninguno especificado



Ahora vamos a verificar que el Front-end se encuentra correctamente desplegado, abrimos la ruta.

`localhost:8082/snowy-dune/`

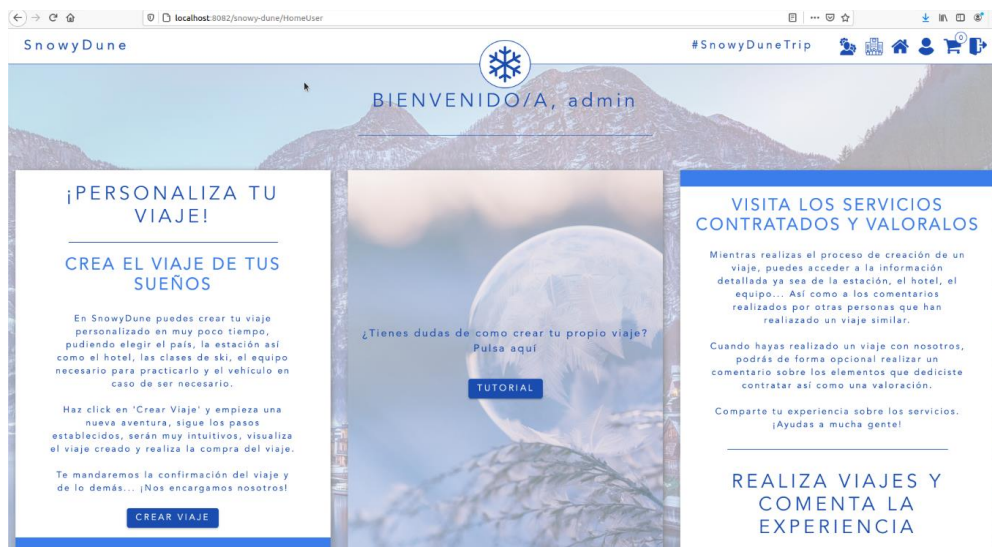
Y se muestra correctamente desplegado.



Vamos a verificar el funcionamiento de los servicios desde el Front-end, realizando el Inicio de Sesión.



Observamos que funciona correctamente.



Con esto se concluye la primera puesta en servicio de la aplicación.