

## Planificación

Se manda la tarea el 15/02

- Creación del Repositorio: 15/02, hecho por Diego García. Se creó en el momento, al igual que la inclusión del resto de miembros del grupo como colaboradores.
- Instalación inicial: se empezará el 15/02 y se prevé que se acabe a primera hora del 17/02. Esto será llevado a cabo por Sergio Rodríguez y Sergio Frisuelos.
- Instalación final con el video/los videos explicativos correspondientes: al tener los pasos ya trabajados con anterioridad, supondremos que se podrá empezar y acabar esta parte el mismo 17/02, siendo Diego García el que realice el video solicitado con una guía facilitada por Sergio Rodríguez para que sea fácil y rápida la implantación definitiva y, así, el video salga lo más corto y conciso posible
- Documentación final: realizada por Sergio Frisuelos. Se prevé que, a última hora del propio 17/02, se puede presentar y subir ya la documentación final con todo lo que se exige en la entrega. Por problemas con la red, se tuvo que posponer este apartado hasta el 23/02, aprovechando el 22/02 para comprobar y confirmar la correcta instalación y funcionamiento de nuestro sitio web

## Pasos a seguir para la Instalación Local

Instalamos lo necesario para poder trabajar con **MariaDB** y hacemos la configuración necesaria para poder trabajar con la base de datos que creemos

```
sudo apt-get update
sudo apt-get install mysql-server mysql-client
```

```
sudo mysql

sudo mysql_secure_installation

sudo mysql

mysql -u root -p

mysql -u user -p
```

Instalamos **Maven** y la versión necesaria del **jkd**

```
sudo apt-get update

sudo apt-get install maven

sudo apt-get install openjdk-8-jdk
```

Comprobamos la correcta instalación

```
java -version
```

```
javac -version
```

Instalamos el servicio para poder usar `ftp` y comprobamos que este habilitado y activo

```
sudo apt-get install vsftpd
```

```
sudo nano /etc/vsftpd.conf y habilitamos el write_enable=YES
```

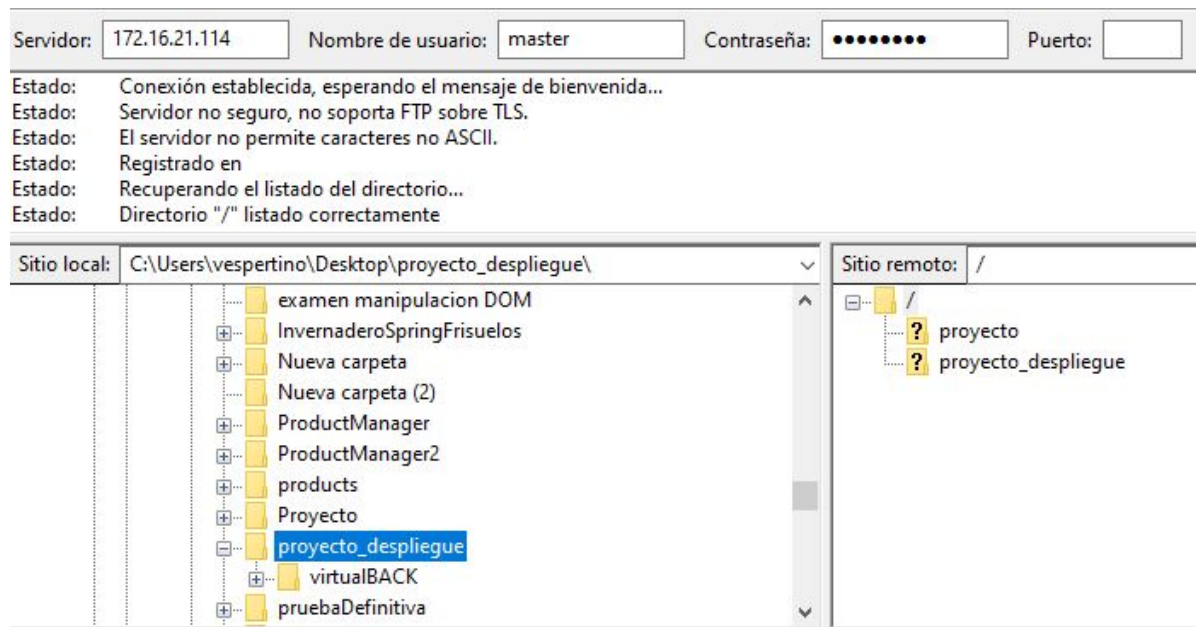
```
sudo systemctl restart vsftpd.service
```

```
sudo systemctl is-enabled vsftpd.service
```

Creamos la carpeta en nuestra máquina cliente, donde descargaremos la parte `BACK`, desde Git

```
mkdir proyecto
```

Nos metemos en la carpeta y, usando `Filezilla`, metemos la aplicación en la carpeta que creamos anteriormente



Entramos en nuestra carpeta donde esta el proyecto y, dentro de este, en `virtualBACK`

```
cd /proyecto_despliegue/virtualBACK
```

Borramos el proyecto que pudiese haber sido creado y lo dejamos preparado para crear el nuevo

```
mvn clean
```

Modificamos el `pom.xml` y creamos el ejecutable

```
nano pom.xml
```

```
mvn install
```

Nos metemos en la carpeta target y ejecutamos el ejecutable de la aplicación

```
cd /target  
  
java -jar target/virtaul.jar
```

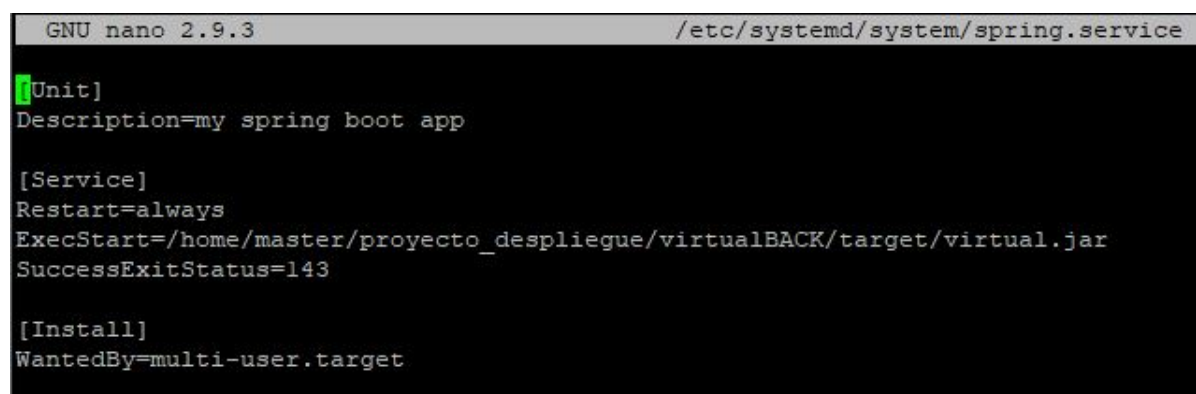
Habilitamos el puerto 8080 para poder acceder y ver que nuestra aplicación funciona

```
sudo ufw allow 8080/tcp
```



Creamos el archivo de configuración de nuestra aplicación para que se ejecute siempre que iniciemos nuestra máquina

```
sudo nano /etc/systemd/system/spring.service
```



Habilitamos el servicio

```
sudo systemctl enable spring.service
```

Y lo activamos

```
sudo systemctl start spring.service
```

AHORA PASAMOS AL FRONT

Instalamos `apache`

```
sudo apt-get install apache2
```

Movemos la página principal de `apache` para poder poner más adelante la nuestra

```
sudo mv /var/www/html/index.html /home/master/
```

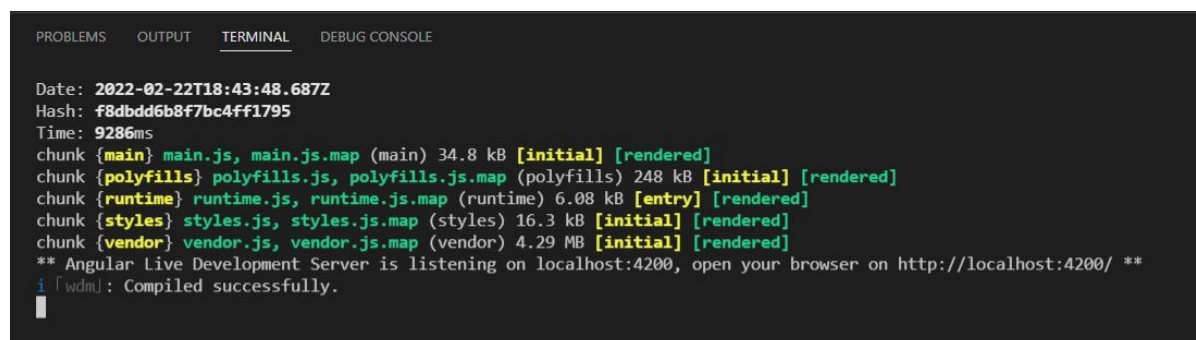


Descargamos de Git la parte del `FRONT`

```
git clone "aqui va la url correspondiente del repositorio de git"
```

Abrimos la carpeta la con `visual studio code` y, en esa carpeta, abrimos una nueva terminal y ejecutamos los siguientes comandos

```
-npm update  
-ng serve -o
```



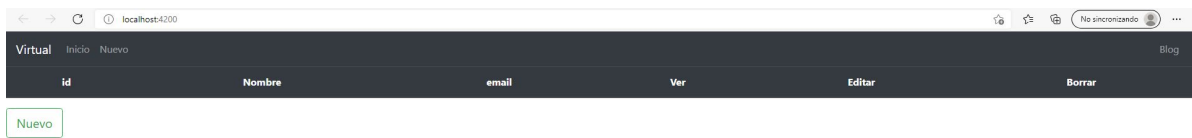
Si no detectase alguno de los comandos, se solucionaría de la siguiente manera:

- Añadimos en las variables de entorno, dentro de path, las siguientes rutas...

```
C:\Program Files\nodejs  
C:\Users\vespertino\AppData\Roaming\npm  
C:\Users\vespertino\AppData\Roaming\npm\node_modules\@angular\cli\bin
```

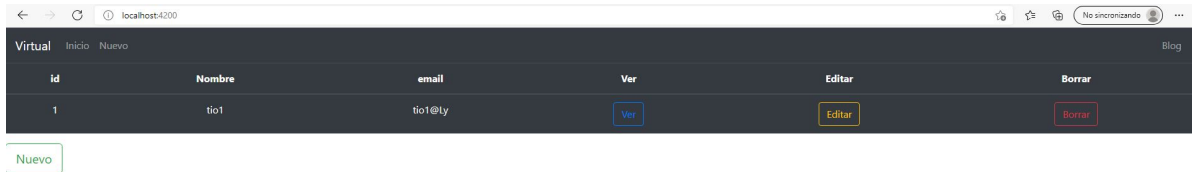
...y reiniciamos.

Una vez solucionados los inconvenientes si existiesen, así se vería la página



En el fichero `tio.services.ts` cambiamos el localhost por la ip de nuestra maquina virtual

Creamos en la pagina una nueva tupla para comprobar que funciona correctamente



Subimos el `FRONT` a producción

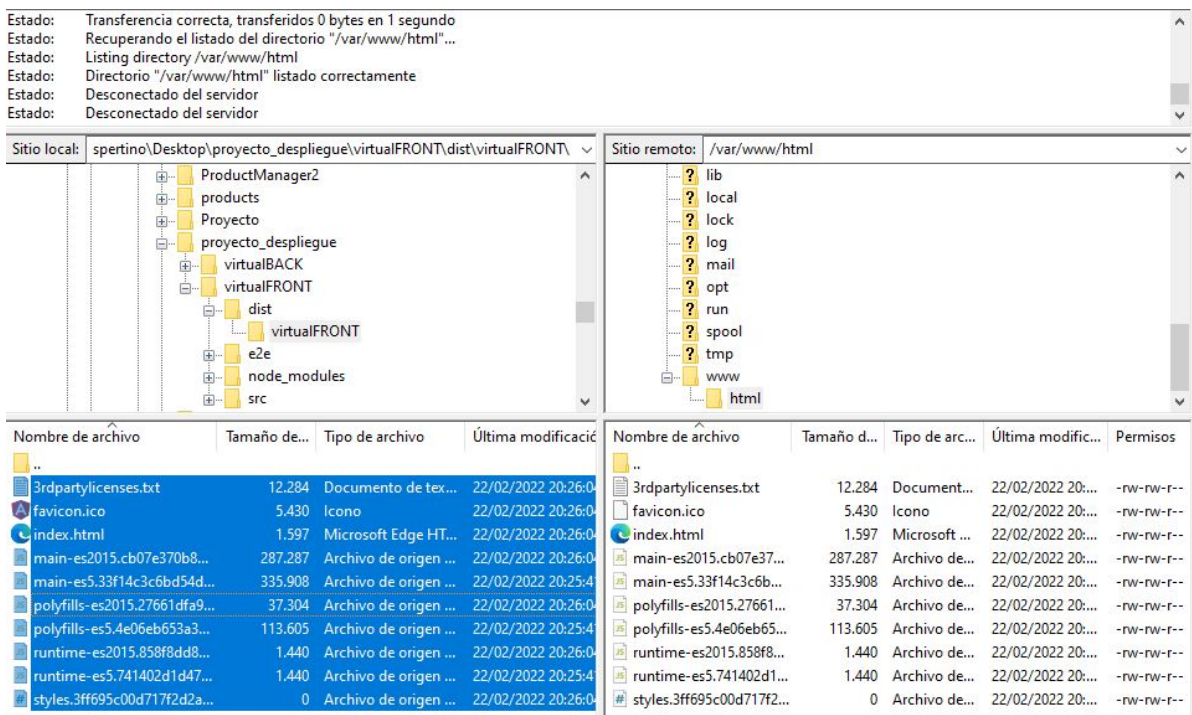
```
ng build --prod
```

Otorgamos propietario y permisos al directorio

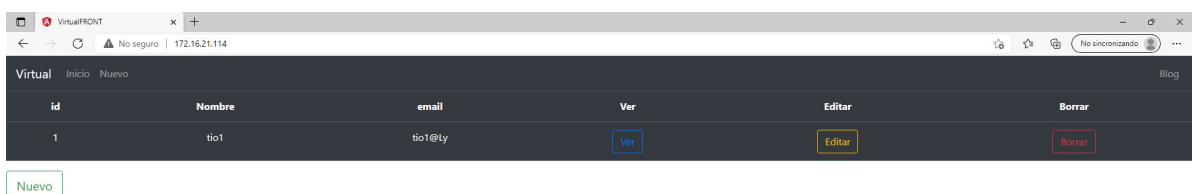
```
sudo chown -R master /var/www/html/

sudo chmod -R 755 /var/www/html/
```

Y subimos con `Filezilla` los archivos al directorio `/var/www/html`



Comprobamos que funciona accediendo a nuestro sitio a traves de la ip de nuestro contenedor



Añadimos, dentro de el archivo `app-routing.module.ts`, en `@NgModule`, donde viene la variable `routes`, `{useHash: true}`, para evitar que nos de errores si actualizásemos la pagina desde una página que se crease solo una vez, al tratarse de una aplicación que trabaja con una unica página creada

Ahora tendremos q recrear los archivos necesarios para q se apliquen dicho cambio y volver a subirlo a nuestro contenedor

```
sudo rm -R /var/www/html/  
  
sudo mkdir /var/www/html/  
  
sudo chown -R master /var/www/html/  
  
sudo chmod -R 755 /var/www/html/
```

```
master@daw-114:/var/www/html$ sudo rm -R /var/www/html/  
master@daw-114:/var/www/html$ sudo mkdir /var/www/html/  
master@daw-114:/var/www/html$ sudo chown -R master /var/www/html/  
master@daw-114:/var/www/html$ sudo chmod 755 -R /var/www/html/  
master@daw-114:/var/www/html$
```

Y con eso habríamos terminado el despliegue de nuestra miniaplicación en Spring