

Tarea FTP - despliegue de aplicación en VPS

Tarea realizada por Carolina Medina Llorente

Tarea FTP - despliegue de aplicación en VPS

Url del repositorio Github

Objetivo

Software utilizado

Proceso del despliegue

Url del repositorio Github

https://github.com/ZPY12803/FTP_VPS

Objetivo

Desplegar una aplicación web Full Stack en un VPS, siguiendo las indicaciones de la playlist facilitada en el aula virtual. (<https://www.youtube.com/watch?v=ibln4qIniv8&list=PL4bT56Uw3S4zIHpaNjz4OVAmXGkYgLpjo>)

Como Servidor Privado Virtual (VPS) se utilizará el servidor remoto instalado en la nube privada del aula con la IP 172.16.21.111

Software utilizado

Putty - para la conexión en remoto

Filezilla - para la subida de archivos por fts

mysql - para la base de datos

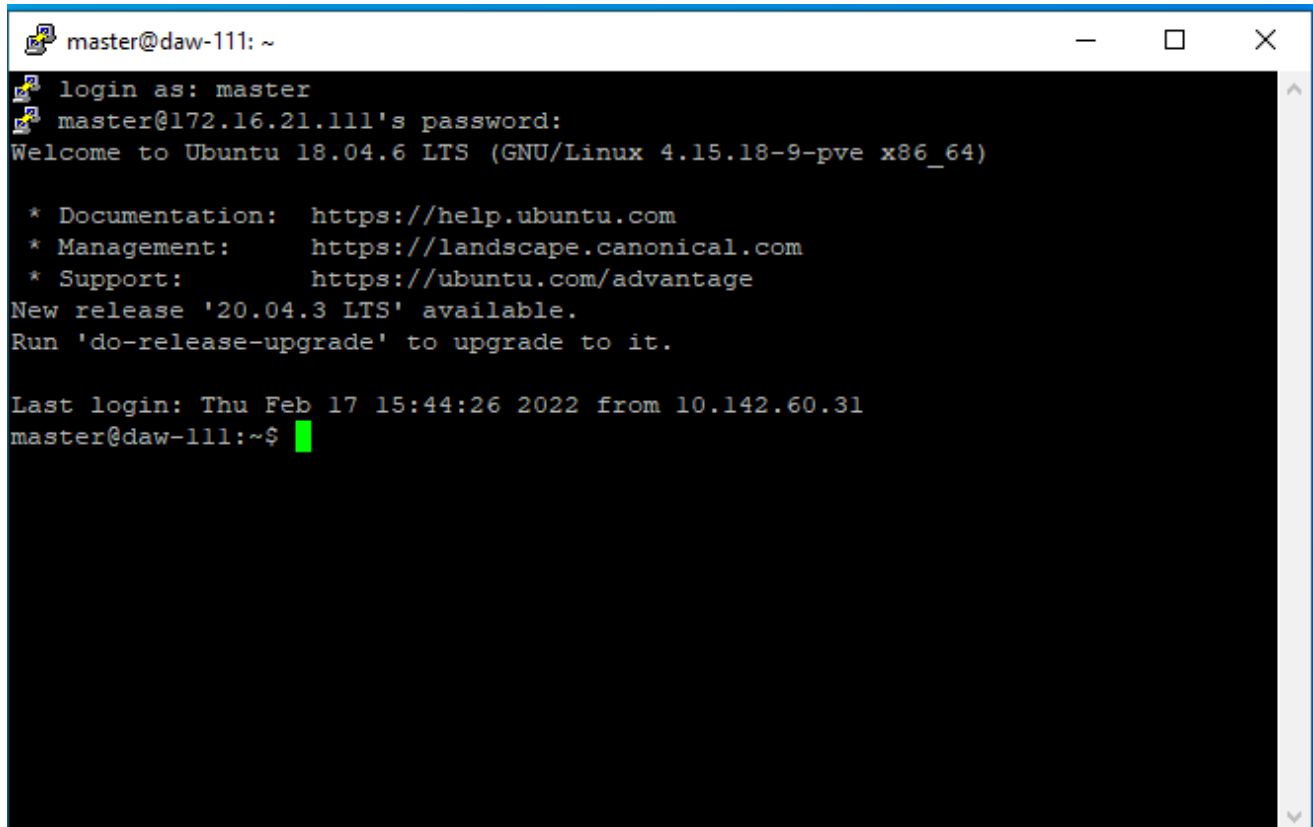
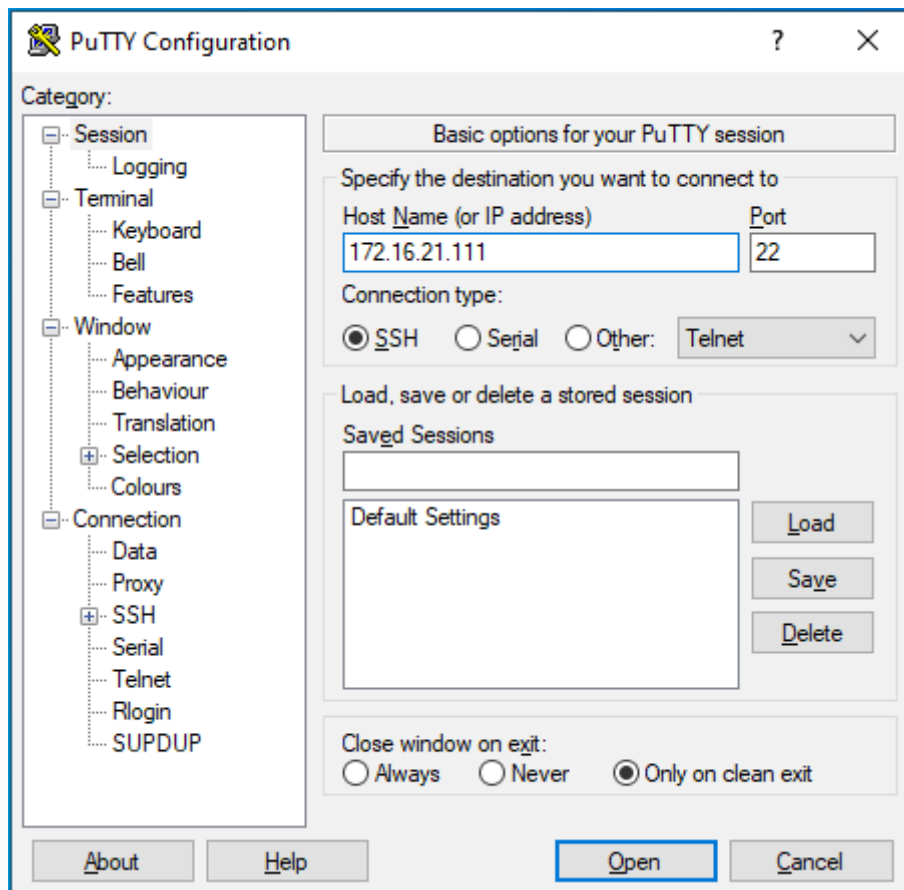
vsftp - para la conexión por ftp

apache2 - como servidor web

Virtual Studio Code - para modificar los archivos de front

Proceso del despliegue

Conectamos al servidor a través de Putty con nuestro usuario y contraseña, e instalamos mysql.



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

Metemos una nueva contraseña, y en las opciones decimos no a usuarios anonimos, deshabilitamos el login remoto, borramos la base de datos del test y recargamos los privilegios.

Conectamos al mysql como root, comprobamos que podemos ver las bases de datos y creamos una bd y usuario para el ejercicio.

```
mysql -u root -p

show databases ;
create database test_virtual;
create user 'user'@'localhost' identified by 'user';
select user from mysql.user;
grant all privileges on test_virtual.* to 'user'@'localhost' with grant option;
flush privileges;
quit
```

Conectamos con el usuario recién creado para comprobar que puede ver la base de datos creada.

```
mysql -u user -p

show databases;
quit
```

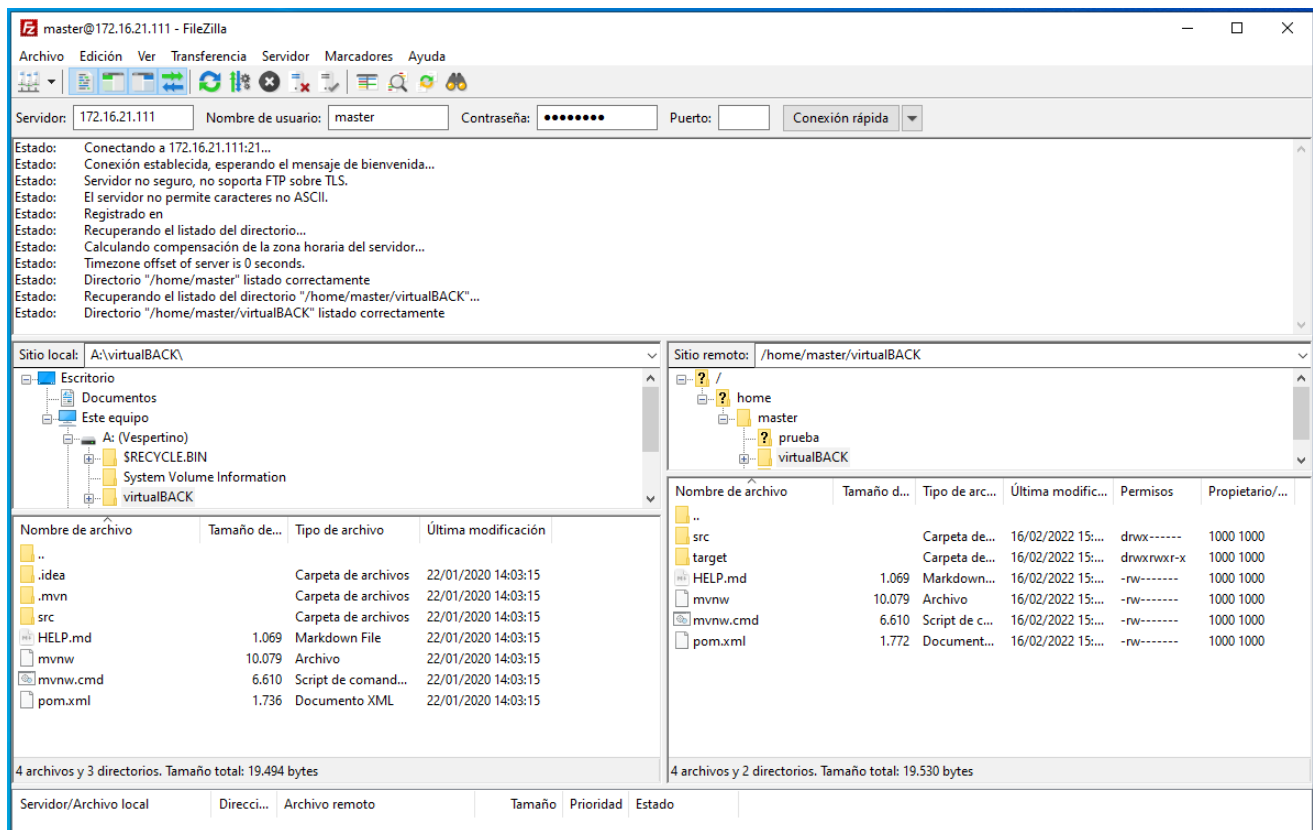
El siguiente paso es instalar maven y java, y comprobar que versión tenemos.

```
apt-get install maven
apt-get update
apt-get install openjdk-8-jdk
apt-get update
java -version
```

Ahora, instalamos vsftpd.

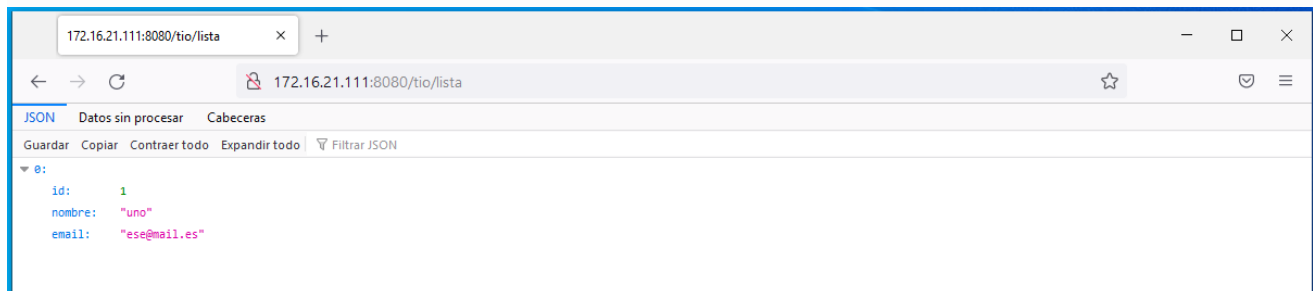
```
apt-get install vsftpd
apt-get update
nano /etc/vsftpd.conf
# Cambiamos la linea -- write enable
systemctl restart vsftpd.service
```

En filezilla subimos a `/home` la carpeta con virtualBACK (subido a `/home/master`)



```
cd /home/master/virtualBack
mvn clean
nano pom.xml
# añadir <finalName>virtual</finalName> despues de executable en el build
mvn install
java -jar target/virtual.jar
```

Y comprobamos que funciona en el navegador conectando a 172.16.21.111:8080/tio/lista



Vamos a convertirlo en servicio. Creamos el siguiente archivo:

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

Y le ponemos dentro lo siguiente:

```
[Unit]
Description=Mi aplicacion Spring Boot
[Service]
Restart=Always
ExecStart=/home/master/virtualBACK/target/virtual.jar
SuccessExitStatus=143
[Install]
WantedBy=multi-user.target
```

```
master@daw-111: ~
GNU nano 2.9.3 /etc/systemd/system/spring.service

[Unit]
Description=Mí aplicación Spring Boot

[Service]
Restart=Always
ExecStart=/home/master/virtualBACK/target/virtual.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Guardamos y salimos, y lo habilitamos.

```
systemctl enable spring.service
systemctl is-enabled spring.service # ya aparece como habilitado
systemctl is-active spring.service # sale inactivo
systemctl start spring.service # lo iniciamos para que pase a activo
```

Comprobamos su funcionamiento de nuevo en el navegador introduciendo **172.16.21.111:8080/tio/detalle/7** y con esto estaría toda la parte de back.

Ahora pasamos a front, instalando el servidor apache.

```
apt-get update
apt-get install apache2
is-active apache2
```

Comprobamos que está funcionando viendo la página de apache en **172.16.21.111**

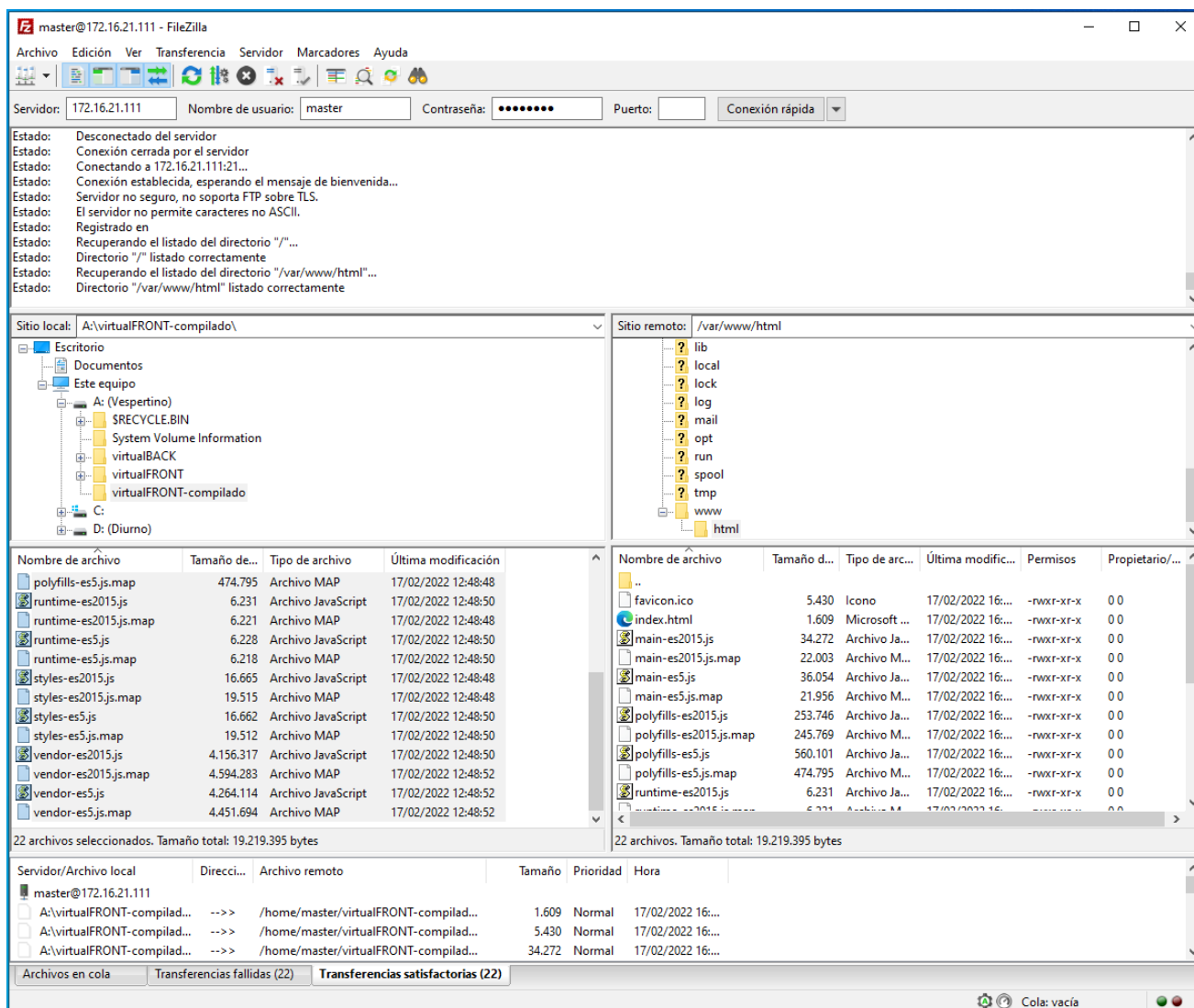
```
is-enabled apache2
# movemos la página de apache a nuestra carpeta en home
mv /var/www/html/index.html /home/master
# ahora 172.16.21.111 muestra Index of /
```

En Virtual Studio Code cambiamos la ip dentro del archivo **tio.service.ts** a la ip de nuestro VPS y desde la terminal, situándonos en la carpeta de virtualFRONT introducimos:

```
npm update
npm install -g @angular/cli@latest
ng build --prod
```

Con el front ya compilado, metemos los archivos generados en la carpeta **dist/virtualFRONT** en **/var/www/html** con el Filezilla y cambiamos los permisos de la carpeta.

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



Configuramos apache para poder editar.

```
a2enmod rewrite
systemctl restart apache2
# por seguridad hacemos una copia del archivo original
cp/etc/apache2/sites-available/000-default.conf/etc/apache2/sites-available/000-
default.conf.bak
nano /etc/apache2/sites-available/000-default.conf
```

En el archivo añadimos lo siguiente dentro de virtualhost.

```
<Directory "/var/www/html">
    AllowOverride All
</Directory>
```

Hacemos una copia del archivo de configuración de apache y lo editamos.

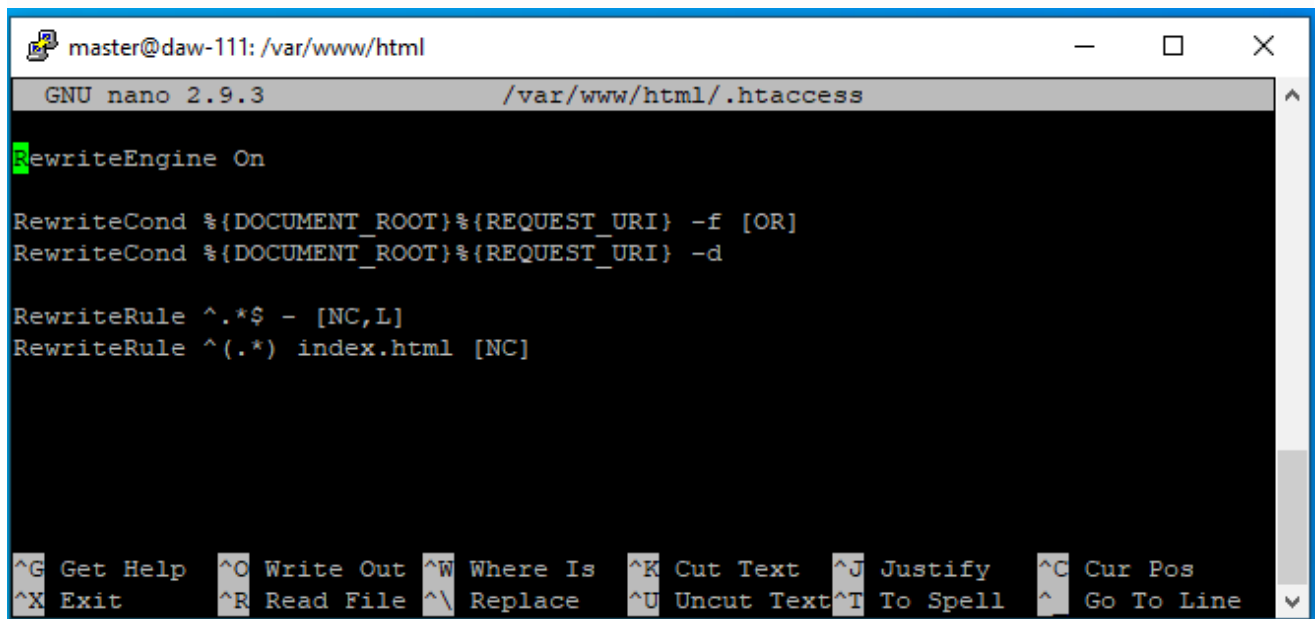
```
cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.bak
nano /etc/apache2/apache2.conf
```

Cambiamos dentro de los Directory el AllowOverride: None a All en los tres, y reiniciamos apache.

```
systemctl restart apache2
```

Añadimos el archivo `.htaccess` editado a `/var/www/html` y reeditamos los permisos de la carpeta.

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



```
GNU nano 2.9.3 /var/www/html/.htaccess

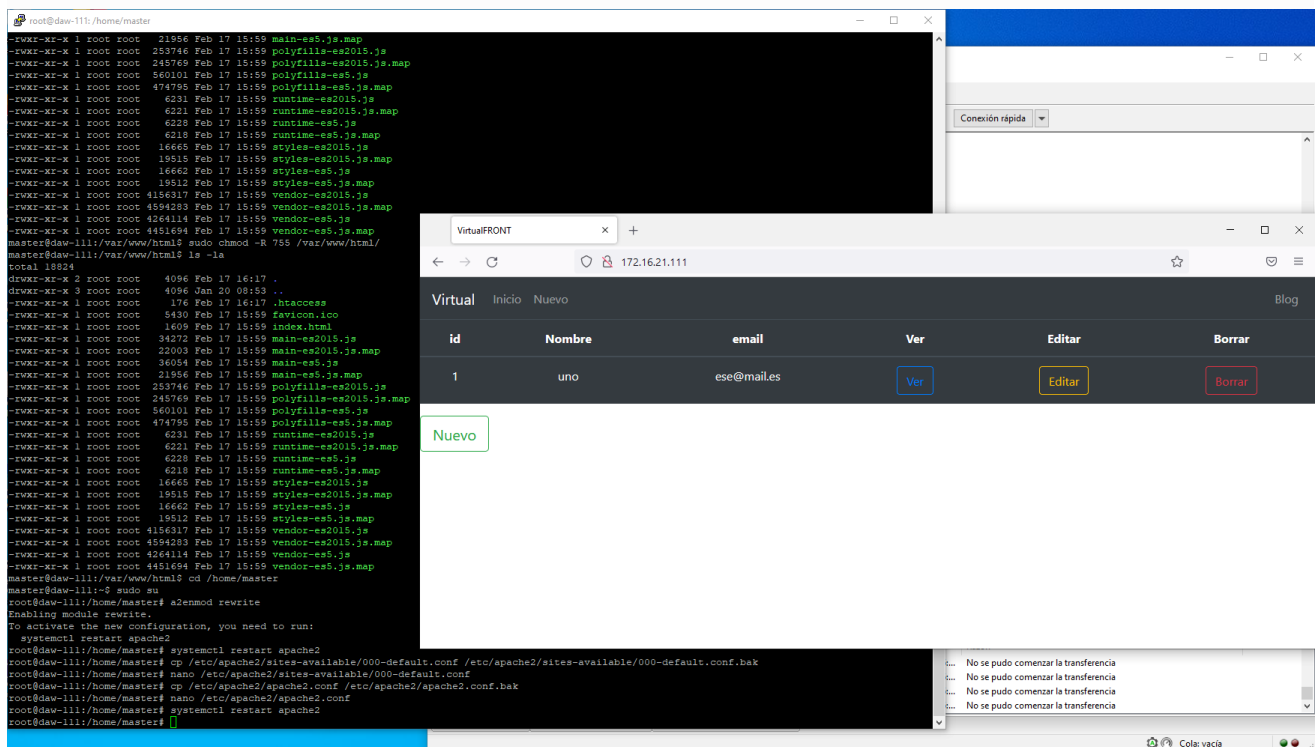
RewriteEngine On

RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d

RewriteRule ^.*$ - [NC,L]
RewriteRule ^(.*) index.html [NC]

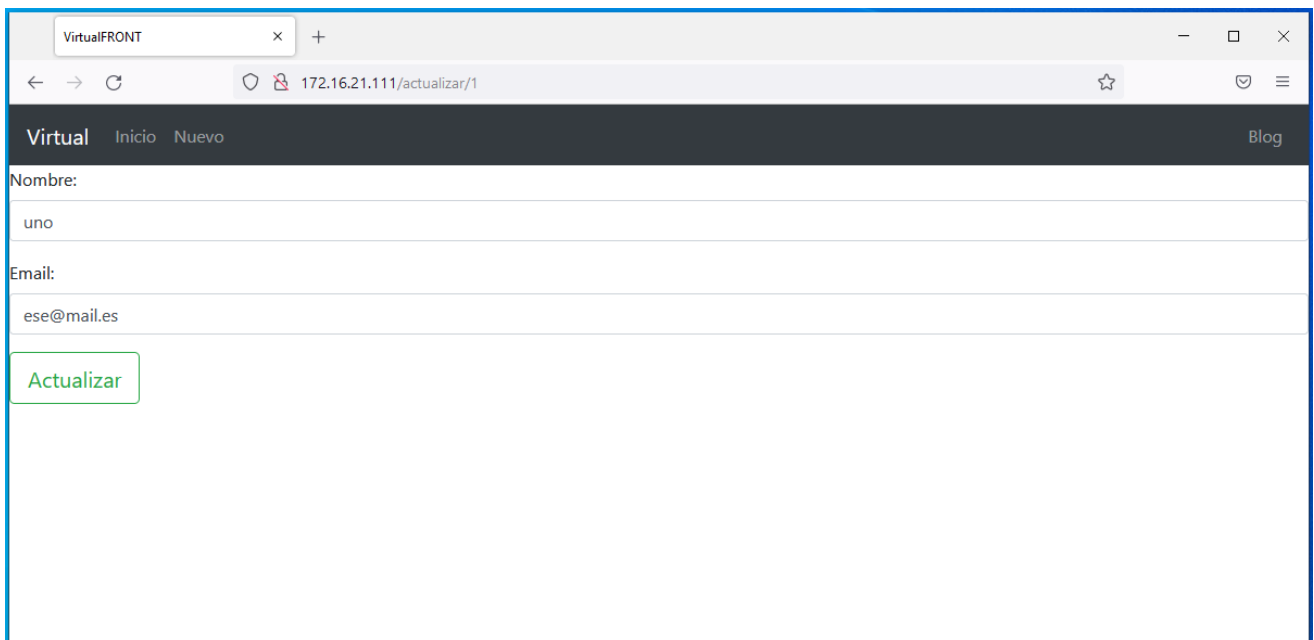
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Con esto, debería funcionar editar y recargar la página.



The terminal window shows the command `chmod -R 755 /var/www/html/` and the output of `ls -la` showing file permissions. The web browser shows a virtual page with a table of users. The file manager shows the contents of the `/etc/apache2/sites-available` directory.

| id | Nombre | email | Ver | Editar | Borrar |
|----|--------|-------------|-----|--------|--------|
| 1 | uno | ese@mail.es | Ver | Editar | Borrar |



Los siguientes pasos serían para crear un certificado para utilizar https.

```
cd /etc/apache2
mkdir ssl
cd ssl
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout test.key -out test.crt
```

Ahora meter las respuestas: country name: ES, provincia: asturias, localidad: gijon, company: DAW, section: vespertino, common name: carol, email: c@daw.ftp

Vamos a configurar apache para que soporte ssl:

```
apt-get update
a2enmod ssl
systemctl restart apache2
# Hacemos una copia de seguridad del archivo actual
cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.bak
# y lo editamos
nano /etc/apache2/sites-available/default-ssl.conf
```

Cambiamos lo siguiente dentro del archivo:

```
SSLCertificateFile /etc/apache2/ssl/test.crt
SSLCertificateKeyFile /etc/apache2/ssl/test.key
```

y lo habilitamos.

```
a2ensite default-ssl.conf
# editamos el archivo .htaccess del front
nano /var/www/html/.htaccess
```


y descomentamos las siguientes lineas:

```
RewriteCond %{HTTPS} !on  
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

Reiniciamos apache.

```
systemctl restart apache2
```

Ahora hay que poner en modo seguro spring para que también entre por https.

Creamos el certificado:

```
openssl pkcs12 -export -in test.crt -inkey test.key -name test -out  
/home/master/virtualBACK/src/main/resources/test.p12
```

Metemos como password test.

```
cd /home/master/virtualBACK  
# comprobamos que esté la key en resources  
ls src/main/resources  
# paramos el servicio de back  
systemctl stop spring.service  
# limpiamos las clases compiladas del proyecto  
mvn clean  
# y editamos el archivo application.properties  
nano src/main/resources/application.properties
```

Tenemos que añadir al final:

```
# SSL  
server.port=8443  
server.ssl.enabled=true  
server.ssl.key-store: /home/master/virtualBACK/src/main/resources/test.p12  
server.ssl.key-store-password: test  
server.ssl.key-store-type: PKCS12  
server.ssl.key-alias: test
```

Ahora volvemos a compilar el proyecto con maven y reiniciamos el back:

```
mvn install  
systemctl restart spring.service
```

Si entramos a **172.16.21.111:8443/tio/lista** vemos que ya funciona

Ahora quedaría cambiar en `.htaccess` la tioURI a **https://172.16.21.111:8443/tio/** y reconstruir los archivos de angular.

```
ng build --prod
```

Copiamos el `.htaccess` a la carpeta nueva generada en dist y movemos los archivos a `/var/www/html` una vez mas con el Filezilla por ftp. Cambiamos de nuevo los permisos de la carpeta.

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

```
master@daw-111: /var/www/html

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Feb 17 15:44:26 2022 from 10.142.60.31
master@daw-111:~$ sudo nano /etc/systemd/system/spring.service
[sudo] password for master:
master@daw-111:~$ sudo nano /var/www/html/.access
master@daw-111:~$ cd /var/www/html
master@daw-111:/var/www/html$ ls -la
total 18824
drwxr-xr-x 2 root root    4096 Feb 17 17:17 .
drwxr-xr-x 3 root root    4096 Jan 20 08:53 ..
-rwxr-xr-x 1 root root    176 Feb 17 16:17 .htaccess
-rwxr-xr-x 1 root root   5430 Feb 17 15:59 favicon.ico
-rwxr-xr-x 1 root root   1609 Feb 17 15:59 index.html
-rwxr-xr-x 1 root root  34272 Feb 17 15:59 main-es2015.js
-rwxr-xr-x 1 root root  22003 Feb 17 15:59 main-es2015.js.map
-rwxr-xr-x 1 root root  36054 Feb 17 15:59 main-es5.js
-rwxr-xr-x 1 root root   21956 Feb 17 15:59 main-es5.js.map
-rwxr-xr-x 1 root root 253746 Feb 17 15:59 polyfills-es2015.js
-rwxr-xr-x 1 root root 245769 Feb 17 15:59 polyfills-es2015.js.map
-rwxr-xr-x 1 root root  560101 Feb 17 15:59 polyfills-es5.js
-rwxr-xr-x 1 root root  474795 Feb 17 15:59 polyfills-es5.js.map
-rwxr-xr-x 1 root root    6231 Feb 17 15:59 runtime-es2015.js
-rwxr-xr-x 1 root root    6221 Feb 17 15:59 runtime-es2015.js.map
-rwxr-xr-x 1 root root    6228 Feb 17 15:59 runtime-es5.js
-rwxr-xr-x 1 root root    6218 Feb 17 15:59 runtime-es5.js.map
-rwxr-xr-x 1 root root   16665 Feb 17 15:59 styles-es2015.js
-rwxr-xr-x 1 root root   19515 Feb 17 15:59 styles-es2015.js.map
-rwxr-xr-x 1 root root   16662 Feb 17 15:59 styles-es5.js
-rwxr-xr-x 1 root root   19512 Feb 17 15:59 styles-es5.js.map
-rwxr-xr-x 1 root root 4156317 Feb 17 15:59 vendor-es2015.js
-rwxr-xr-x 1 root root 4594283 Feb 17 15:59 vendor-es2015.js.map
-rwxr-xr-x 1 root root 4264114 Feb 17 15:59 vendor-es5.js
-rwxr-xr-x 1 root root 4451694 Feb 17 15:59 vendor-es5.js.map
master@daw-111:/var/www/html$ sudo nano /var/www/html/.htaccess
master@daw-111:/var/www/html$
```

y ya funcionaría todo con https con nuestros certificados. El navegador no estará muy feliz con que sean autogenerados por nosotros pero funciona todo.