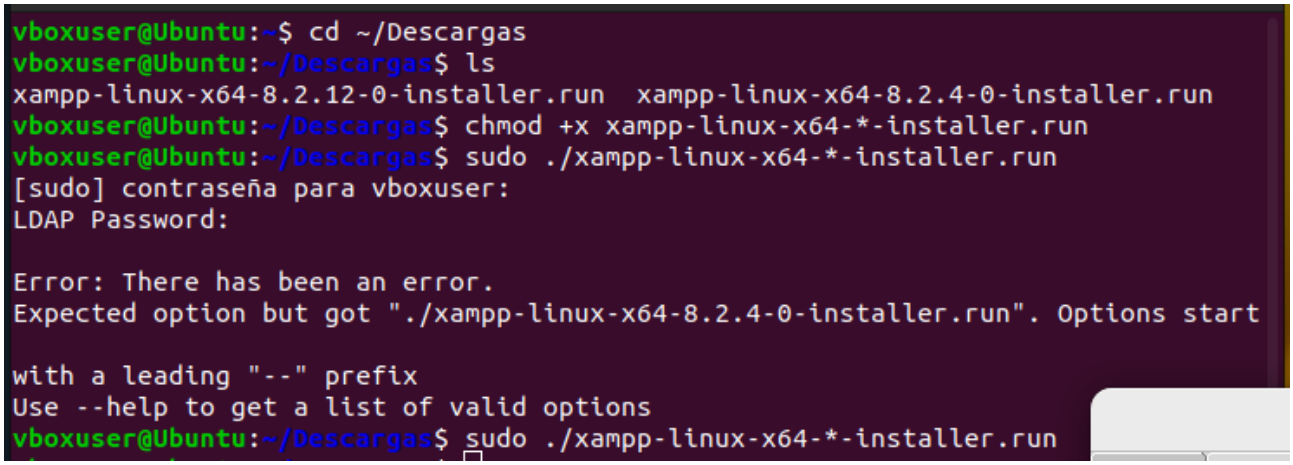
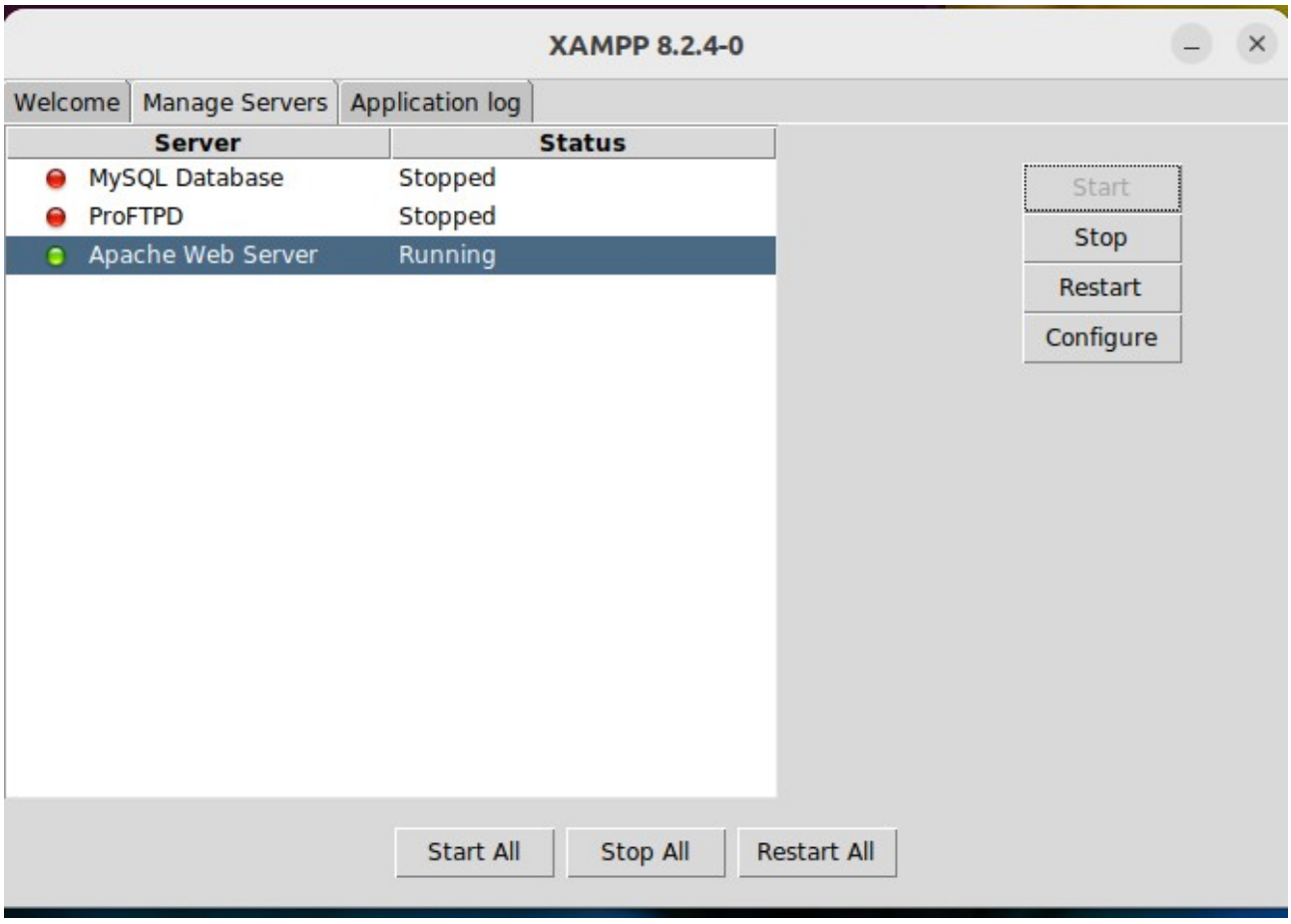
Práctica: Servidores de certificados en Apache

Marta López de los Bueis 2ºDAW

1º Instalamos XAMPP en la maquina virtual



Una vez instalado, levantamos apache



2º Habilitamos el módulo SSL en Apache

$ sudo a2enmod ssl

```
vboxuser@Ubuntu:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-sig
ned certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
vboxuser@Ubuntu:~$
```

Después de habilitar el módulo, recargamos la configuración de Apache

$ sudo systemctl restart apache2

```
vboxuser@Ubuntu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-01-20 09:48:40 CET; 5s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2183 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 2187 (apache2)
      Tasks: 6 (limit: 7128)
     Memory: 13.1M
        CPU: 49ms
     CGroup: /system.slice/apache2.service
             ├─2187 /usr/sbin/apache2 -k start
             ├─2188 /usr/sbin/apache2 -k start
             ├─2189 /usr/sbin/apache2 -k start
             ├─2190 /usr/sbin/apache2 -k start
             ├─2191 /usr/sbin/apache2 -k start
             └─2192 /usr/sbin/apache2 -k start

ene 20 09:48:40 Ubuntu systemd[1]: Starting The Apache HTTP Server...
ene 20 09:48:40 Ubuntu systemd[1]: Started The Apache HTTP Server.
```

3º Creamos el directorio para almacenar certificados

$ sudo mkdir -p /etc/apache2/scrunkly

```
ubuntu@ubuntu:/$ sudo mkdir /etc/apache2/scrunkly
ubuntu@ubuntu:/$
```

3. Generamos el certificado SSL

3.1 Crear una clave privada

$ sudo openssl genrsa -des3 -out key 2048

```
ubuntu@ubuntu:/etc/apache2/scrunkly$ sudo openssl genrsa -des3 -out key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

3.2 Generamos el CSR

CSR (Certificate Signing Request)

$ sudo openssl req -new -key key -out scrunkly.csr

```
ubuntu@ubuntu:/etc/apache2/scrunkly$ sudo openssl req -new -key key -out scrunkly.csr
Enter pass phrase for key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Albacete
Locality Name (eg, city) []:Caga y vete
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Scrunkly S.L
Organizational Unit Name (eg, section) []:no se
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:matriculadehonor@porfi.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234
An optional company name []:1234
ubuntu@ubuntu:/etc/apache2/scrunkly$
```

3.3 Generar el certificado autofirmado

Usando el comando sudo openssl x509 -req -sha256 -days 365 -in scrunkly.csr -signkeykey -out scrunkly.crt

```
ubuntu@ubuntu:/etc/apache2/scrunkly$ sudo openssl x509 -req -sha256 -day
s 365 -in scrunkly.csr -signkey key -out scrunkly.crt
Enter pass phrase for key:
Certificate request self-signature ok
subject=C = ES, ST = Albacete, L = Caga y vete, O = Scrunkly S.L, OU = n
o se, emailAddress = matriculadehonor@porfi.com
ubuntu@ubuntu:/etc/apache2/scrunkly$
```

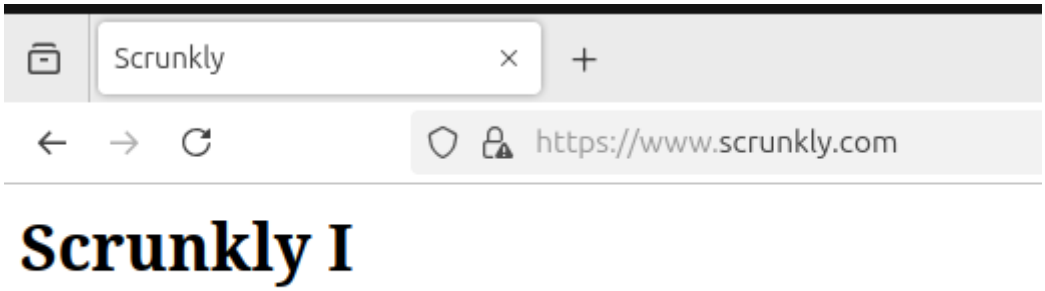
Agregamos nuevos parametros al archivo creado en la practica anterior

```
 GNU nano 7.2                    scrunkly.conf
<VirtualHost *:443>
        ServerAdmin webmaster@scrunkly.com
        ServerName scrunkly.com
        Serveralias www.scrunkly.com
        DocumentRoot /var/www/scrunkly.com/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on
        SSLCertificateFile /etc/apache2/scrunkly/scrunkly.crt
        SSLCertificateKeyFile /etc/apache2/scrunkly/key
        ErrorLog ${APACHE_LOG_DIR}scrunkly-access-ssl-log
        LogFormat "%h %l %u %t \ %r\" %>s" combined
</VirtualHost>
```

Reiniciamos apache2 y nos pedira la contraseña que usamos para crear el certificado

```
ubuntu@ubuntu:/$ sudo service apache2 restart
🔐 Enter passphrase for SSL/TLS keys for scrunkly.com:443 (RSA): (press
••••
ubuntu@ubuntu:/$
```



Podemos ver que aun que nos informe de que no es segura, ya que no esta firmado por una entidad valida, nos aparece cvomo https

Aqui podemos ver el certificado que hemos creado

Certificate

matriculadehonor@porfi.com

Subject Name

Country	ES
State/Province	Albacete
Locality	Caga y vete
Organization	Scrunkly S.L
Organizational Unit	no se
Email Address	matriculadehonor@porfi.com

Issuer Name

Country	ES
State/Province	Albacete
Locality	Caga y vete
Organization	Scrunkly S.L
Organizational Unit	no se
Email Address	matriculadehonor@porfi.com

Validity