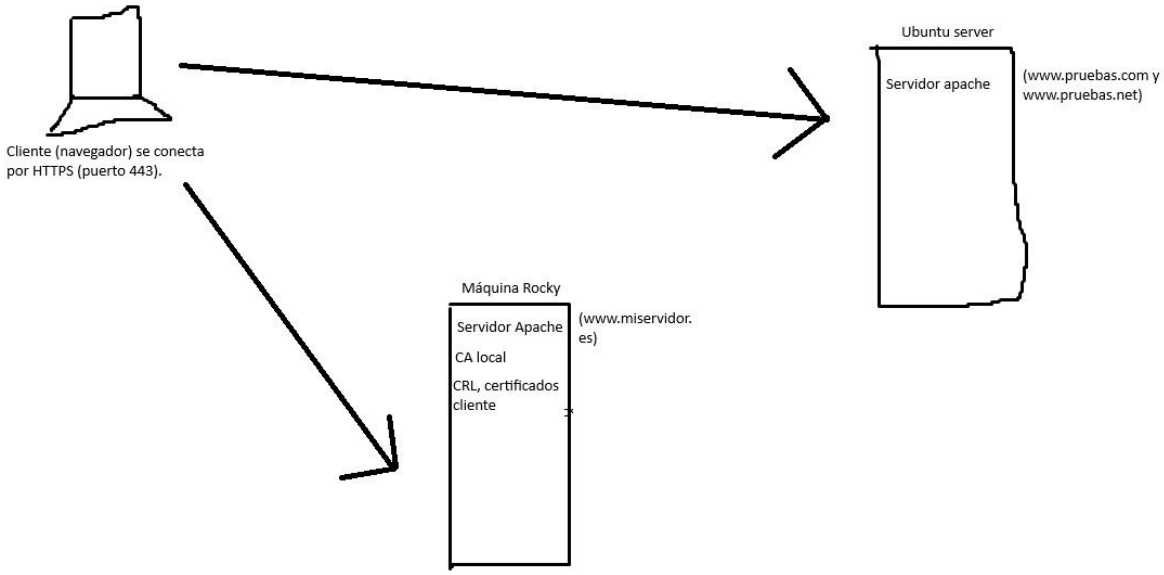
# PKI para HTTPS:

**Esquema visual práctica:**




1. **Comprobar fecha y hora de la máquina virtual. Si no está bien, actualizar la fecha y hora del sistema (importante porque si no fallará al emitir los certificados.**

```
[root@server ~]# date
jue 30 oct 2025 11:33:57 CET
[root@server ~]#
```

La fecha y la hora se muestran correctamente.

```
[root@server ~]# timedatectl
               Local time: jue 2025-10-30 11:36:14 CET
           Universal time: jue 2025-10-30 10:36:14 UTC
                 RTC time: jue 2025-10-30 10:36:14
                Time zone: Europe/Madrid (CET, +0100)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
[root@server ~]# _
```

Si la zona horaria no fuera correcta y además NTP no estuviera activo entonces deberíamos ejecutar la siguiente serie de comandos:

- **sudo timedatectl set-timezone Europe/Madrid** (poner la zona horaria)
- **timedatectl** (ver la información de la zona y fecha)
- **sudo dnf install –y chrony** (insalar el servicio chrony – cliente moderno de NTP)
- **sudo systemctl enable –now chronyd** (habilitar el servicio)
- **system status chronyd** (mirar el status del demonio)
- **chronyc tracking** (verificar sincronización)
- **sudo chronyc makestep** (Forzar la sincronización inmediata)
- **Timedatectl** (Verificar la información)

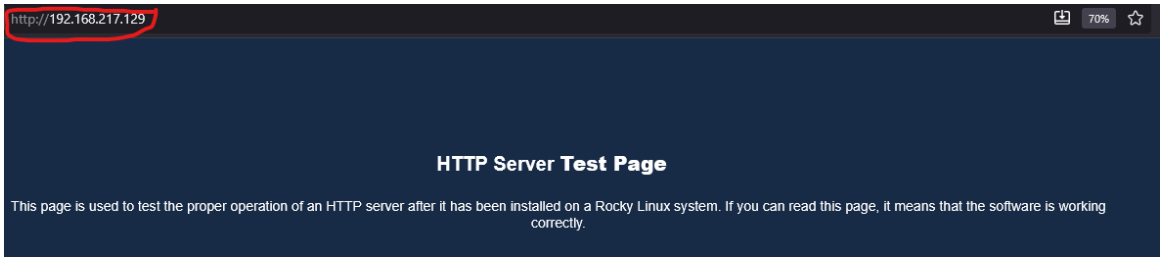
## 2. Instalar Apache y SSL

Rocky:

```
[root@server ~]# sudo dnf update
Extra Packages for Enterprise Linux 9 - x86_64
```

```
[root@server ~]# sudo dnf install httpd -y
Última comprobación de caducidad de metadatos hecha hace 0:
```

```
[root@server ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/s
rvice.
[root@server ~]# systemctl start\ httpd
Unknown command verb start httpd.
[root@server ~]# systemctl start httpd
[root@server ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-10-30 12:21:53 CET; 3s ago
```




```
¡Listo!
[root@server ~]# sudo dnf install mod_ssl
```

Se crea automaticamente un certificado autofirmado en /etc/pki/tls/certs/localhost.cert:

```
-----BEGIN CERTIFICATE-----
MIIEqDCCApCgAwIBAgIIMpldcTMwuREwDQYJKoZIhvcNAQELBQAwcDELMAkGA1UE
BhMCVVMxFDASBgNVBAoMC1Vuc3BlY2lmaWVkMR4wHAYDVQQLDBVjYS05MjI1OTMw
MDYxMzQyMTk0MjQxDzANBgNVBAMMBnNlcnZlcjEaMBgGCSqGSIb3DQEJARYLcm9v
dEBzZXJ2ZXIwHhcNMjUxMDMwMTEyOTMwWhcNMjYxMDMwMTEyOTMwWjBQMQswCQYD
VQQGEwJVUzEUMBIGA1UECgwLVW5zcGVjaWZpZWQxDzANBgNVBAMMBnNlcnZlcjEa
MBgGCSqGSIb3DQEJARYLcm9vdEBzZXJ2ZXIwggEiMA0GCSqGSIb3DQEBAQUAA4IB
```

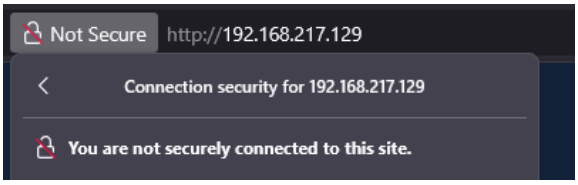
Y la clave en /etc/pki/tls/private/localhost.key:

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCmJRmkFCkR+4UA
EHHa7EPH2WG/FRMknVcxZaQ5HqLO4opSLUvDo+yjinhsRDD48jWmjKqQ331XEiPy
Ya4mWcx5PeOydIT0Nsst0dn6sZAU/4TgHa4KzGYvxT74j4Lq0mBAuDFfe6ObNmvR
7FLmrRDTUGrlNavrh/IozzHtnX+0pkMJA9WkspAG9CDShJNlwh9fSP9P7+Kn0ISu
TVhs+QhYM110WgBZRWv0InxvOEFqH01UWT98ofIq6NWKbmdbu6a+n7+JgRxDnThd
```

Verificamos que está levantado el puerto 443 (hay que reiniciar el servicio de httpd si nos se descargan a la vez):

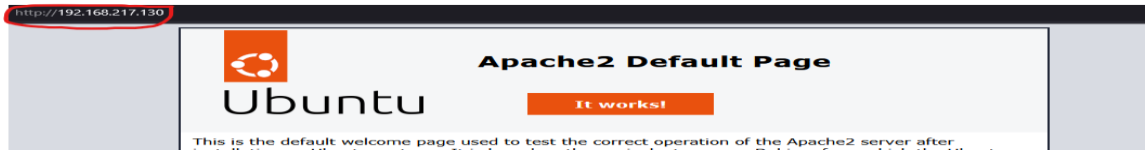
```
[root@server ~]# sudo ss -tlnp | grep httpd
LISTEN 0      511                *:80              *:*    users:(("httpd",pid=25
514,fd=4),("httpd",pid=25513,fd=4),("httpd",pid=25512,fd=4),("httpd",pid=25510,f
d=4))
[root@server ~]# sudo systemctl restart httpd
[root@server ~]# sudo ss -tlnp | grep httpd
LISTEN 0      511                *:443             *:*    users:(("httpd",pid=25
838,fd=6),("httpd",pid=25837,fd=6),("httpd",pid=25836,fd=6),("httpd",pid=25834,f
d=6))
LISTEN 0      511                *:80              *:*    users:(("httpd",pid=25
838,fd=4),("httpd",pid=25837,fd=4),("httpd",pid=25836,fd=4),("httpd",pid=25834,f
d=4))
[root@server ~]#
```

Aun con SSL puesto, nos da este aviso porque el certificado esta autofirmado, y no por una CA conocida y porque el CN (Common Name) no coincide con el nombre del dominio:




Ubuntu:

```
root@server:~# sudo apt update
Des:1 http://security.ubuntu.com/ubuntu noble-security InRelease [1
root@server:~# sudo apt install apache2
Leyendo lista de paquetes... Hecho
root@server:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-10-30 12:13:56 CET; 11s ago
```



### 3. Ubicacion de openssl.cnf y creacion de la CA

Se encuentra bajo el siguiente directorio:

```
[root@server ~]# cd /etc/pki/tls/
[root@server tls]# ls
cert.pem  ct_log_list.cnf  misc         openssl.d
certs     fips_local.cnf   openssl.cnf  private
[root@server tls]#
```

Creamos la estructura necesaria:

```
[root@server tls]# ls
cert.pem  certs  ct_log_list.cnf  fips_local.cnf  misc  openssl.cnf  openssl.d  private
[root@server tls]# sudo mkdir crl
[root@server tls]# sudo mkdir newcerts
[root@server tls]# vim index.txt
[root@server tls]# echo 01 > serial
[root@server tls]# echo 01 > crlnumber
[root@server tls]# ls
cert.pem  crl        ct_log_list.cnf  index.txt  newcerts     openssl.d  serial
certs     crlnumber  fips_local.cnf   misc       openssl.cnf  private
[root@server tls]#
```

Editamos el archivo *openssl.cnf* para el dir:

```
[ CA_default ]

dir             = /etc/pki/tls          # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
```

```
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = ES
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Madrid

localityName                    = Locality Name (eg, city)
localityName_default            = Alcorcon

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = CA de Pruebas
```

Creamos la clave privada:

```
[root@server tls]# openssl genrsa -aes256 -out private/cakey.pem 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

1234

Y el certificado con los datos predeterminados:

```
[root@server tls]# openssl req -new -key private/cakey.pem -out ca-csr.pem
Enter pass phrase for private/cakey.pem:
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) [ES]:
State or Province Name (full name) [Madrid]:
Locality Name (eg, city) [Alcorcon]:
Organization Name (eg, company) [CA de Pruebas]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:CA del CEU para Pruebas
Email Address []:
```

defaults

```
[root@server tls]# openssl req -x509 -extensions v3_ca -in ca-csr.pem -out cacert.pem -key private/cakey.pem
 -days 3652
Enter pass phrase for private/cakey.pem:
Warning: Not placing -key in cert or request since request is used
Warning: No -copy_extensions given; ignoring any extensions in the request
```

??

Y creamos el listado de certificados revocados (crl.pem) que inicialmente estará vacío:

```
[root@server tls]# openssl ca -gencrl -out crl.pem
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
[root@server tls]# ls
cacert.pem  cert.pem  crl        crlnumber.old  ct_log_list.cnf  index.txt  newcerts     openssl.d  serial
ca-csr.pem  certs     crlnumber  crl.pem        fips_local.cnf   misc       openssl.cnf  private
```

VER SI METER CERT.PEM EN CERTS????

Estructura por ahora:

```
[root@server tls]# tree
.
├── cacert.pem
├── ca-csr.pem
├── cert.pem -> /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
├── certs
│   ├── ca-bundle.crt -> /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
│   ├── ca-bundle.trust.crt -> /etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt
│   └── localhost.crt
├── crl
├── crlnumber
├── crlnumber.old
├── crl.pem
├── ct_log_list.cnf
├── fips_local.cnf -> /etc/crypto-policies/back-ends/openssl_fips.config
├── index.txt
├── misc
├── newcerts
├── openssl.cnf
├── openssl.d
├── private
│   ├── cakey.pem
│   └── localhost.key
└── serial
```

**4. Crear claves público/privada para el servidor web (2048 bits) y un certificado firmado por nuestra CA. El CN debe corresponder con el nombre del servidor. Configurar apache para que atienda peticiones HTTP y HTTPS. Página de inicio se encontrará en (/projects/miservodir). Conseguir que el cliente pueda conectarse a la web sin presenciar el "Warning".**

Vamos al final de fichero **openssl.cnf** y configuramos esta nueva directiva para añadir el bloque server_SAN:

```
[ server_SAN ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = www.miservidor.es
DNS.2 = miservidor.es

"openssl.cnf" 414L, 12583B
```

Creamos la clave privada para el servidor de 2048 bits:

```
root@server:/etc/pki/tls
[root@server tls]# sudo openssl genrsa -out server-key.pem 2048
[root@server tls]# ls
ca-csr.pem  crlnumber      ct_log_list.cnf  misc         openssl.d  server-key.pem
certs       crlnumber.old  fips_local.cnf   newcerts     private
crl         crl.pem        index.txt        openssl.cnf  serial
[root@server tls]#
```

Creamos el CSR y ponemos que el common name = nombre dominio servidor web:

```
[root@server tls]# sudo openssl req -new -key server-key.pem -out server-csr.pem
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.
-----
CountryName [ES]:
Province name [Madrid]:
Locality name [Alcorcon]:
Organization Name [CA de Pruebas]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:www.miservidor.es
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

Vemos el resultado:

```
root@server:/etc/pki/tls
[root@server tls]# ls
ca-csr.pem  crlnumber      ct_log_list.cnf  misc         openssl.d  server-csr.pem
certs       crlnumber.old  fips_local.cnf   newcerts     private    server-key.pem
crl         crl.pem        index.txt        openssl.cnf  serial
[root@server tls]#
```

Y firmamos el CSR certificado con la CA:

```
[root@server tls]# openssl ca -extensions server_SAN -in miservidor-csr.pem -out miservidor-crt.pem -days 73
0
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Oct 30 14:54:07 2025 GMT
            Not After : Oct 30 14:54:07 2027 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Madrid
            organizationName          = CA de Pruebas
            commonName                = www.miservidor..es
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                DNS:www.miservidor.com, DNS:miservidor.com
Certificate is to be certified until Oct 30 14:54:07 2027 GMT (730 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
```

Configuramos el apache de la siguiente manera:

Creamos la carpeta projects en /var/www y metemos un fi

```
[root@server ~]# ls /var/www/projects/
miservidor
[root@server ~]#
```

Y dentro de /projects metemos una carpeta miservidor que es donde estará almacenado el HTML de la página:

```
root@server:/var/www/projec ×   root@server:/etc/pki/tls
[root@server ~]# ls /var/www/projects/
miservidor
[root@server ~]# cd /var/www/projects/miservidor/
[root@server miservidor]# ls
index.html
[root@server miservidor]#
```

Creamos un fichero de configuración para esa página web de la siguiente manera:

```
miservidor.conf  ssl.conf          welcome.conf
[root@server ~]# ls /etc/httpd/conf.d/
autoindex.conf  miservidor.conf  README  ssl.conf  userdir.conf  welcome.conf
[root@server ~]#
```

Creamos el fichero miservidor.conf y metemos el siguiente contenido:

```
root@server:~

# Servidor HTTPS principal - www.miservidor.es
Listen 443 https

<VirtualHost *:443>
    ServerName www.miservidor.es
    DocumentRoot /var/www/projects/miservidor

    SSLEngine on
    SSLCertificateFile /etc/pki/tls/server-crt.pem
    SSLCertificateKeyFile /etc/pki/tls/server-key.pem
    SSLCACertificateFile /etc/pki/tls/certs/cacert.pem

    <Directory /var/www/projects/miservidor>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog /var/log/httpd/miservidor_error.log
    CustomLog /var/log/httpd/miservidor_access.log combined
</VirtualHost>

# Redirección HTTP → HTTPS
<VirtualHost *:80>
    ServerName www.miservidor.es
</VirtualHost>
```

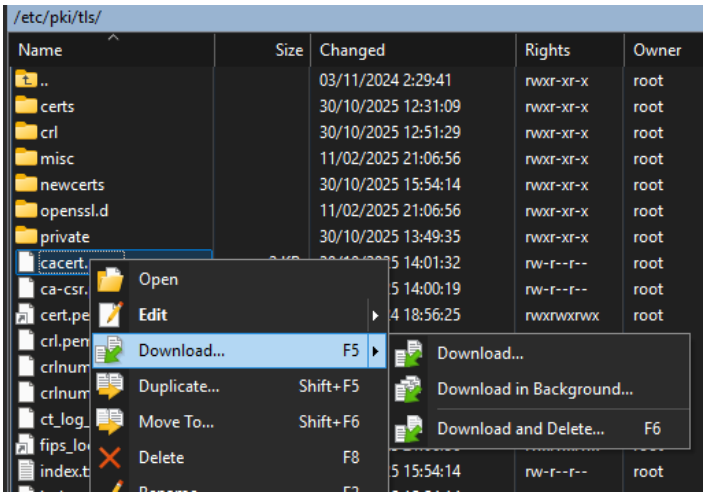
Verificamos que todo esté bien en la configuración usando el siguiente comando:

- **sudo apachectl configtest**

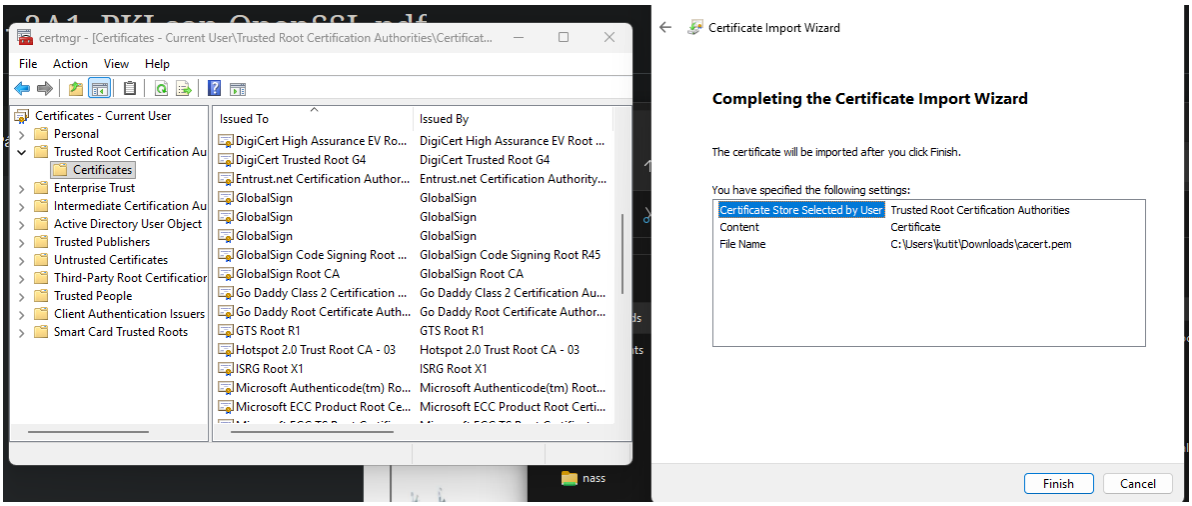
Reiniciamos el servicio apache:

- **sudo systemctl restart httpd**
- **sudo systemctl status httpd**

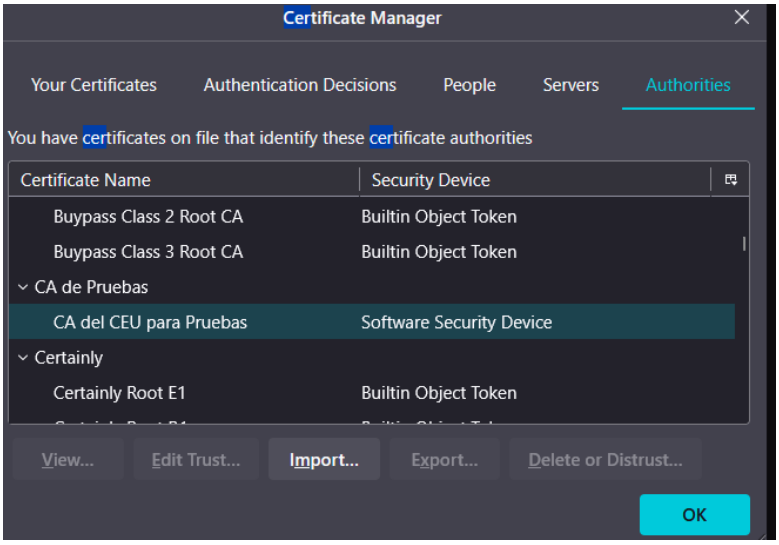
Con WinSCP descargo en certificado de la CA en mi máquina de windows:




Ponemos el certificado de la CA en Trusted Root Certification Authorities (windows -> manage user certificates):



Y lo metemos en Firefox:




Y modificamos el archivo hosts de windows (c:/windows/system32/drivers/etc/hosts) y metemos lo siguiente:

```
192.168.217.129 www.miservidor.com miservidor.com https://miservidor.com http://miservidor.com
```

Una vez hecho esto, ya podemos poner el dominio en el buscador y debería completar la conexión sin avisos ni problemas:


(yo me he empanado y en el certificado puse miservidor.com en vez de miservidor.es, pero como no es realmente relevante y no quiero volver a hacer el certificado, sigo pa'lante)

## 5. Creamos el un certificado de cliente para el acceso a la pagina web

Creamos el certificado de Carmen Cabrera firmado por la CA:

```
[root@server tls]# openssl req -new -keyout carmen-cabrera-key.pem -out carmen-cabrera-csr.pem -noenc
..+...+..................+.......+..+...+...+....+...+..+....+.....++++++++++++++++++++++++++++++++++++
....+.+........+.++++++++++++++++++++++++++++++++++++++*..+...+...+...+.........................+...+..
.........+........+....+..+.........+.+...........+....+...........+............+..+.......+...+..+.+.
....+....+......+.........+...+...+.....+.........+..+...........+.......+...+..+......+.......+.....
+............+..+..............+......+........+.+.....+.+......+.....+...+.+...+..+.+............+.
+.....+...+.................+...+..+....+.....+.+........+.+..+...........+...+.........+..........
..+..................+..............+...+..........+.+..............+.........+...........+.......+.
.........+...+.+........+.+......+.....+...+......+.+......+..+.+.....+.........+...+.......+..+.+.....
.+.........+...+......+..........+..+....+.....+....+......+...........+.........+..............+.+...
...................+...+......+......+.....+....+..........+.+.....+...........+......+.+...........
..+.......+.................................+..+.+..+.+......+........+...+........+........+....+...+...+.
.+..+....+.....++++++
...+..++++++++++++++++++++++++++++++++++++*....+.......+...........+......+...+...+....+.....+....+.
.+...+.+........+.+..............+.+..+....+......+.....+....+...++++++++++++++++++++++++++++++++++++++
.........+.+...........+......+...+....+......+..+.........+.+........+.........+......+....+.....+...
.........+......+...........+................+..+.+..+....+.....++++++
-----
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) [ES]:
State or Province Name (full name) [Madrid]:
Locality Name (eg, city) [Alcorcon]:
Organization Name (eg, company) [CA de Pruebas]:
Organizational Unit Name (eg, section) []:Contabilidad
Common Name (eg, your name or your server's hostname) []:Carmen Cabrera
```

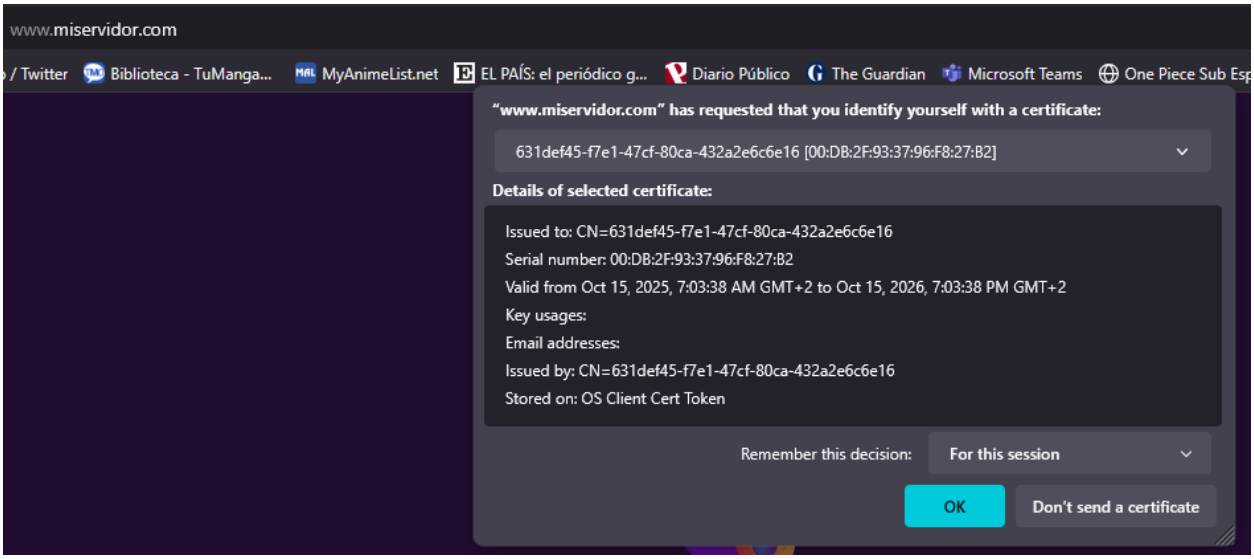
```
[root@server tls]# openssl ca -extensions user -in carmen-cabrera-csr.pem -out carmen-cabrera-crt.pem -days
730
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 2 (0x2)
        Validity
            Not Before: Oct 31 12:25:02 2025 GMT
            Not After : Oct 31 12:25:02 2027 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Madrid
            organizationName          = CA de Pruebas
            organizationalUnitName    = Contabilidad
            commonName                = Carmen Cabrera
```

```
[root@server tls]# openssl pkcs12 -export -in carmen-cabrera-crt.pem -inkey carmen-cabrera-key.pem -out carm
en-cabrera-crt.p12
Enter Export Password:
Verifying - Enter Export Password:      1234
```

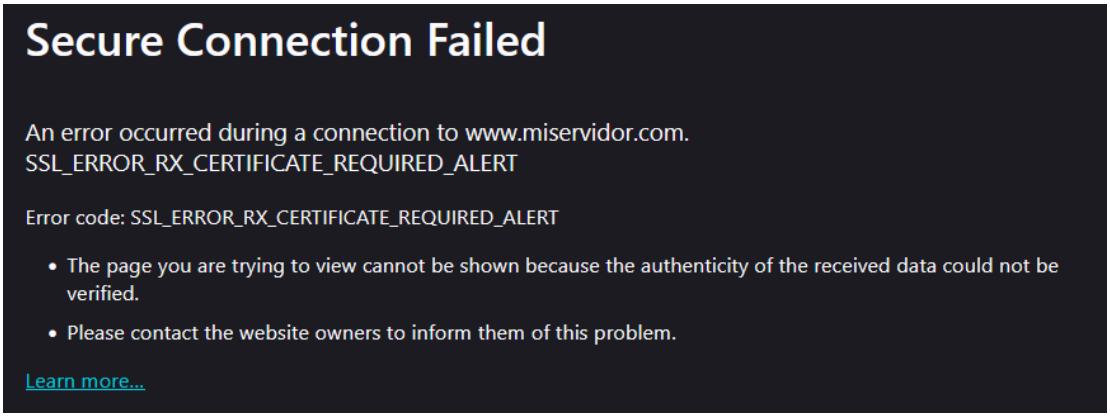
Metemos un ‘SSLVerifyClient require’ en nuestra configuración de la página web para que pida un certificado de user:

```
SSLCertificateChainFile /etc/pki/tls/miservidor-csi

SSLVerifyClient require

<Directory /projects/miservidor>
```

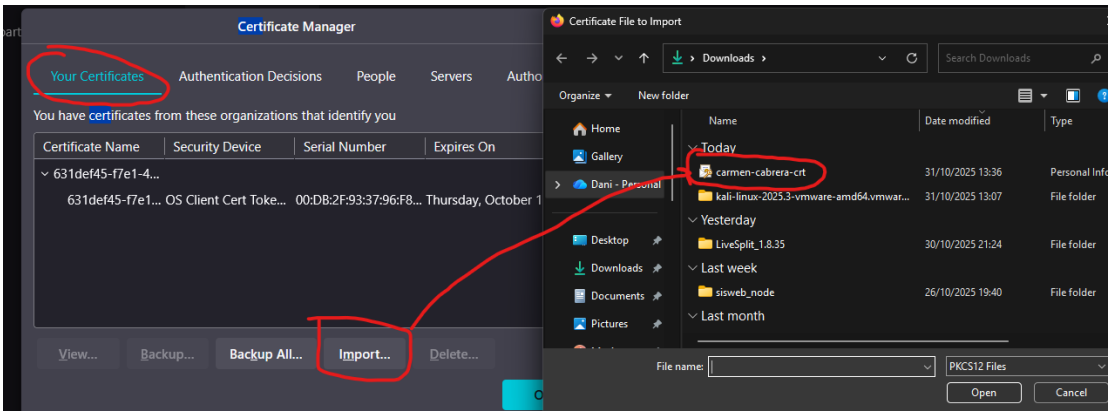
Y despues de reiniciar httpd, si intentamos entrar a la pgina, nos pide un certificado valido:

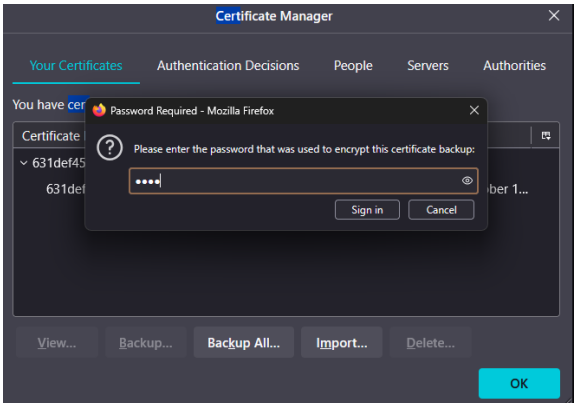



Si no damos uno valido nos sale lo siguiente:

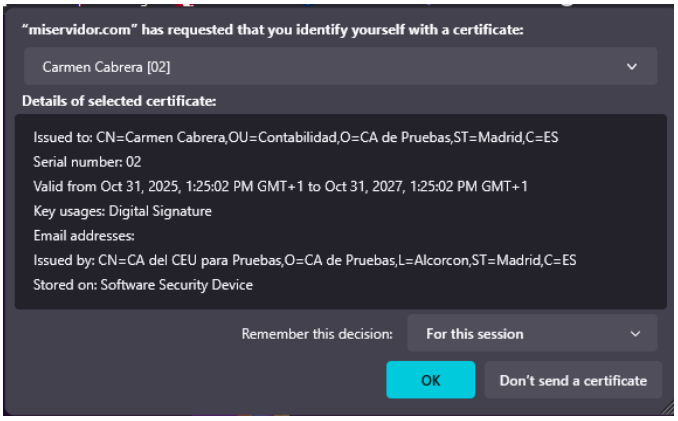


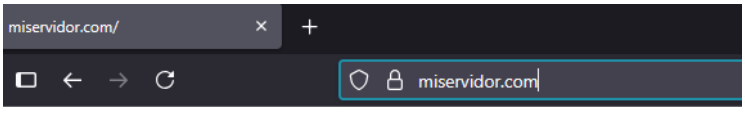
Importamos el certificado en el buscador:




Y cuando verificamos con el certificado de Carmen, nos deja entrar:




# Helloooo

## 6. Acceso a paginas por departamento

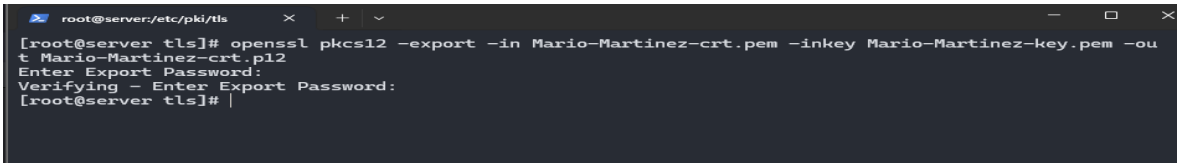
Creamos el certificado de Mario Martinez de Marketing:

```
[root@server tls]# openssl req -new -keyout mario-martinez-key.pem -out mario-martinez-csr.pem -noenc
....+.....+.......+..+....+.....................+........++++++++++++++++++++++++++++++++++++++++*.....
...+....++++++++++++++++++++++++++++++++++++++++*.....................+.........+..+.........+.+......+..
...+......+......+.............++++++
..+............+......+...........+...+....+........+...+....+...........+.+..................+.....+.
++++++++++++++++++++++++++++++++++++++*...+.....+.+.....+.......+.....+.......+..+...+.......+...+...+
.+.......++++++++++++++++++++++++++++++++++++++*.................+...+.........+..........+......+.....
...+.....+....+...+.....+.......+.....+.+.............................+...+.+..+...+....+...+..+.+.....
..+..+...........+...+.+...........+.+........+.+...+..+....+.....+..........+...........+...........
..+...+.......+......+......+......+.....+....+......+..................+.....+....+..+.++++++
-----
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) [ES]:
State or Province Name (full name) [Madrid]:
Locality Name (eg, city) [Alcorcon]:
Organization Name (eg, company) [CA de Pruebas]:
Organizational Unit Name (eg, section) []:Marketing
Common Name (eg, your name or your server's hostname) []:Mario Martinez
Email Address []:
```

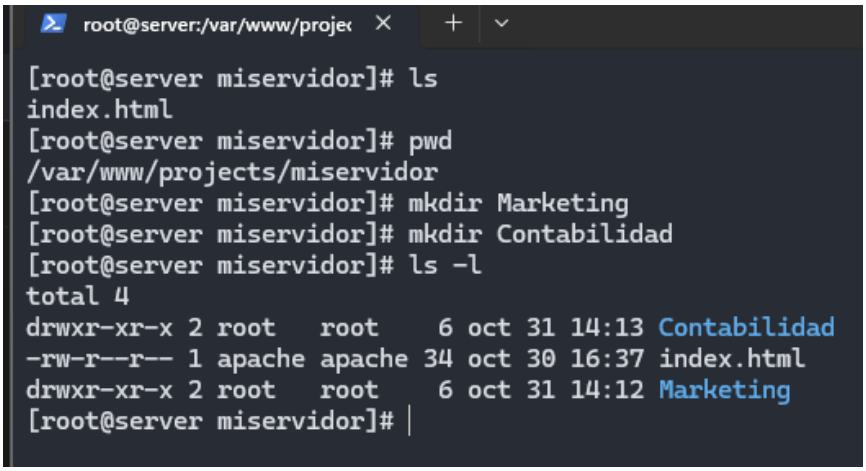
Firmamos la el certificado con la CA

```
[root@server tls]#
[root@server tls]# openssl ca -extensions user -in Mario-Martinez-csr.pem -out Mario-Martinez-crt.pem
-days 730
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 3 (0x3)
        Validity
            Not Before: Oct 31 13:08:03 2025 GMT
            Not After : Oct 31 13:08:03 2027 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Madrid
            organizationName          = CA de Pruebas
            organizationalUnitName    = Marketing
            commonName                = Mario Martinez
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                C4:FF:A6:DC:E3:EA:71:EE:CE:4E:4C:23:61:5A:AF:3C:55:2B:7D:73
            X509v3 Authority Key Identifier:
                keyid:F5:33:6E:DC:CF:D5:42:C9:16:BB:7D:EC:A9:4F:6D:B5:04:29:74:EA
                DirName:/C=ES/ST=Madrid/L=Alcorcon/O=CA de Pruebas/CN=CA del CEU de Pruebas
                serial:05:C4:D0:4F:BE:CF:C3:A4:0E:72:1A:44:0F:E3:F7:E2:6F:F9:73:55
            X509v3 Key Usage:
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
Certificate is to be certified until Oct 31 13:08:03 2027 GMT (730 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
[root@server tls]#
```

Lo empaquetamos en un contenedor PKCS12:




Creamos los directorios para los distintos departamentos:

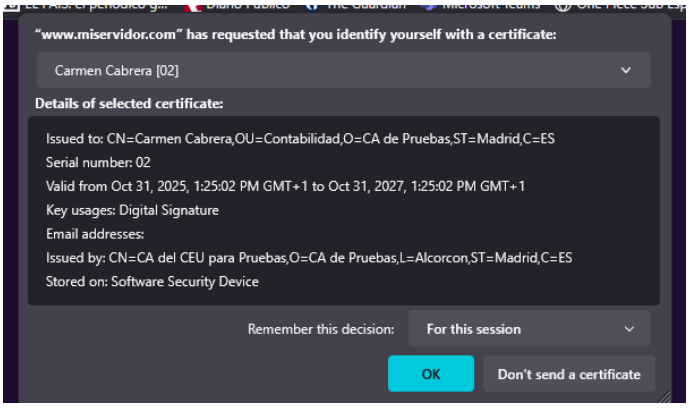



En nuestra configuración del servidor web, aseguramos que compruebe por el campo de OU para cada directorio:

```
[root@server ~]# sudo vi /etc/httpd/conf.d/miservidor.conf
```

```
SSLCACertificateFile /etc/pki/tls/cacert.pem
SSLVerifyClient require
<Location /Contabilidad/>
   SSLRequire ( %{SSL_CLIENT_S_DN_OU} eq "Contabilidad")
</Location>

<Location /Marketing/>
    SSLRequire ( %{SSL_CLIENT_S_DN_OU} eq "Marketing")
</Location>
```

Cuando queremos entrar a la página principal, nos pide que pongamos un certificado:




Si elegimos por ejemplo a Carmen, entramos y podemos ver Contabilidad, pero no Marketing:


# Contabilidad


# Forbidden

You don't have permission to access this resource.

### 7. Revocacion de certificados y del registro

Revocamos el certificado de Mario:

```
crl                     index.txt.attr   Mario-Martinez-key.pem  serial.old
[root@server tls]# openssl ca -revoke Mario-Martinez-crt.pem
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
Revoking Certificate 03.
Database updated
[root@server tls]#
```

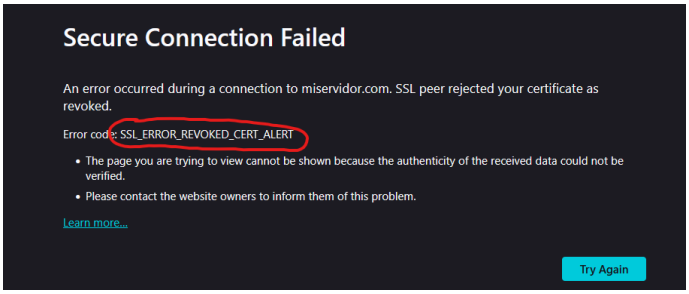
Y actualizamos el crl.pem:

```
[root@server tls]# openssl ca -gencrl -out crl.pem
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
```

**Importante** tener en cuenta que al añadir las directivas de la captura de abajo hay que ponerlo justo debajo de **SSLVerifyClient** para que se parsee del todo bien.

```
SSLCACertificateFile /etc/pki/tls/cacert.pem
SSLVerifyClient require
SSLCARevocationCheck chain
SSLCARevocationFile /etc/pki/tls/crl.pem
<Location /Contabilidad/>
   SSLRequire ( %{SSL_CLIENT_S_DN_OU} eq "Contabilidad")
</Location>

<Location /Marketing/>
    SSLRequire ( %{SSL_CLIENT_S_DN_OU} eq "Marketing")
</Location>
```

Y cuando intentamos acceder con el certificado de Mario, nos de error de REVOKED:




**8. Servidor Apache HTTPS en Ubuntu firmado por la CA y con varios alias:**

Generamos la clave y CSR de Ubuntu para www.pruebas.com, pruebas.com, www.pruebas.net y pruebas.net:

```
root@server:~# openssl genrsa -out pruebas.key 2048
root@server:~# cd
```

Creamos un archivo SAN pruebas.cnf:

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req

[req_distinguished_name]

[v3_req]
subjectAltName = @alt_names

[alt_names]
DNS.1 = pruebas.com
DNS.2 = www.pruebas.com
DNS.3 = pruebas.net
DNS.4 = www.pruebas.net
```

```
root@server:~# openssl req -new -key pruebas.key -out pruebas.csr -config pruebas.cnf -subj "/CN=pruebas.com"
root@server:~# ls
pruebas.cnf  pruebas.csr  pruebas.key  snap
```

Copiamos el archivo en la maquina Rocky con la CA:

```
root@server:~# scp pruebas.csr root@192.168.217.129:/etc/pki/tls
The authenticity of host '192.168.217.129 (192.168.217.129)' can't be established.
ED25519 key fingerprint is SHA256:68/moxxSXL4vdJXONJMpqONBGrEa5ll3xZvG6sve2OQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.217.129' (ED25519) to the list of known hosts.
root@192.168.217.129's password:
pruebas.csr                                                   100% 1013     2.1MB/s   00:00
```

También transferimos el fichero pruebas.cnf desde la Ubuntu a la Rocky.

Y dentro de la Rocky lo firmamos con la CA:

```
[root@server tls]# openssl x509 -req -in pruebas.csr -CA cacert.pem -CAkey private/cakey.pem -CAcreateserial -out pruebas.crt -days 365
-sha256 -extfile pruebas.cnf -extensions v3_req
Certificate request self-signature ok
subject=CN=pruebas.com
```

Y enviamos el certificado a la maquina Ubuntu:

```
[root@server tls]# scp pruebas.crt root@192.168.217.130:~
root@192.168.217.130's password:
pruebas.crt                                   100% 1310     2.5MB/s   00:00
```

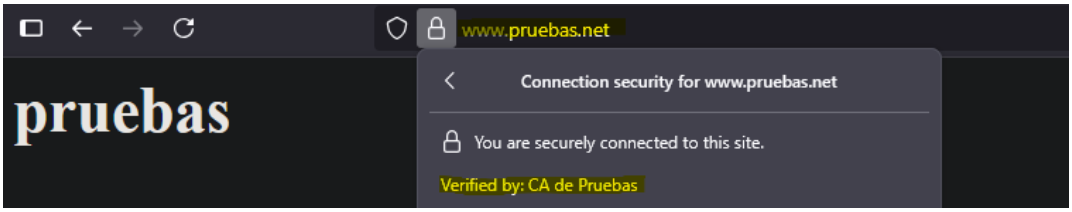
Y ya dentro de la Ubuntu activamos SSL y ponemos en /etc/apache2/sites-availible/000-default.conf lo siguiente:

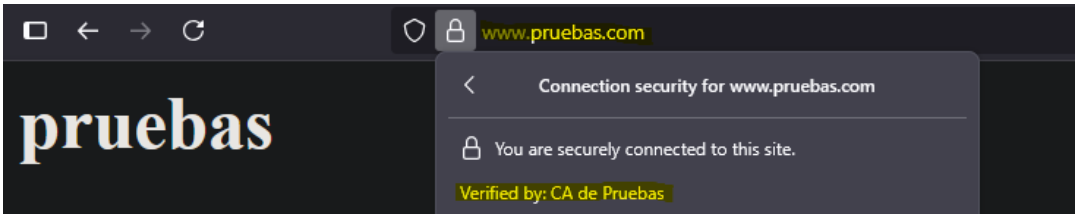
```
root@server:~# sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.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

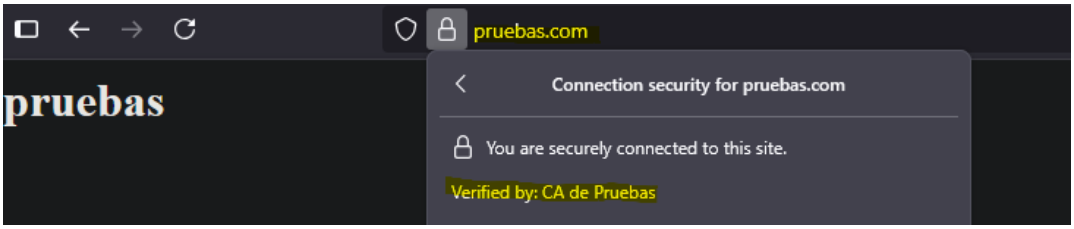
```
<VirtualHost *:80>
    ServerName www.pruebas.com
</VirtualHost>

<VirtualHost *:443>
    ServerName www.pruebas.com
    ServerAlias pruebas.com www.pruebas.net pruebas.net

    DocumentRoot /projects/pruebas

    SSLEngine on
    SSLCertificateFile /root/pruebas.crt
    SSLCertificateKeyFile /root/pruebas.key
    SSLCertificateChainFile /root/pruebas.csr

    <Directory "/projects/pruebas">
        DirectoryIndex index.html
        Require all granted
        AllowOverride All
    </Directory>

</VirtualHost>
```

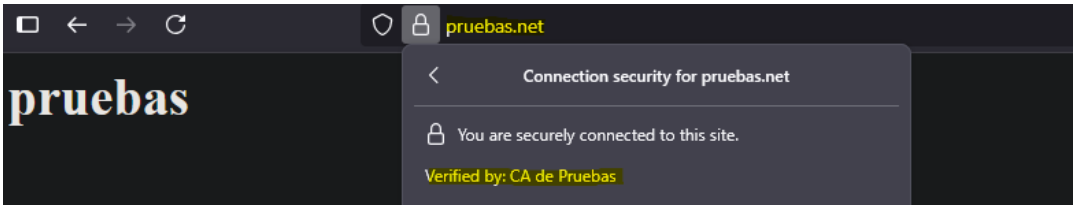
```
root@server:/etc/apache2/sites-available# systemctl restart apache2
```

Y probamos a entrar en las páginas www.pruebas.com, pruebas.com, www.pruebas.net y pruebas.net en el buscador (donde ya esta metida nuestra CA como de confianza):



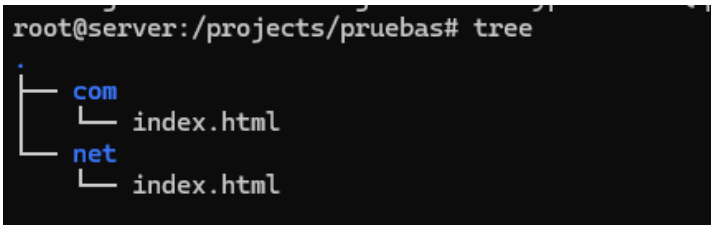



**9. Dividir dominios .net y .com y usar el mismo certificado:**

Creamos esta estructura para las dos paginas web (pruebas.com y pruebas.net):

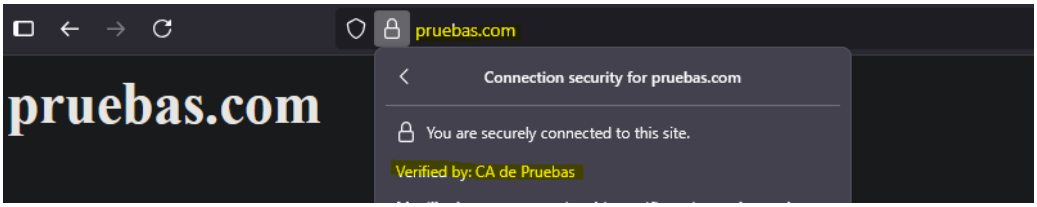


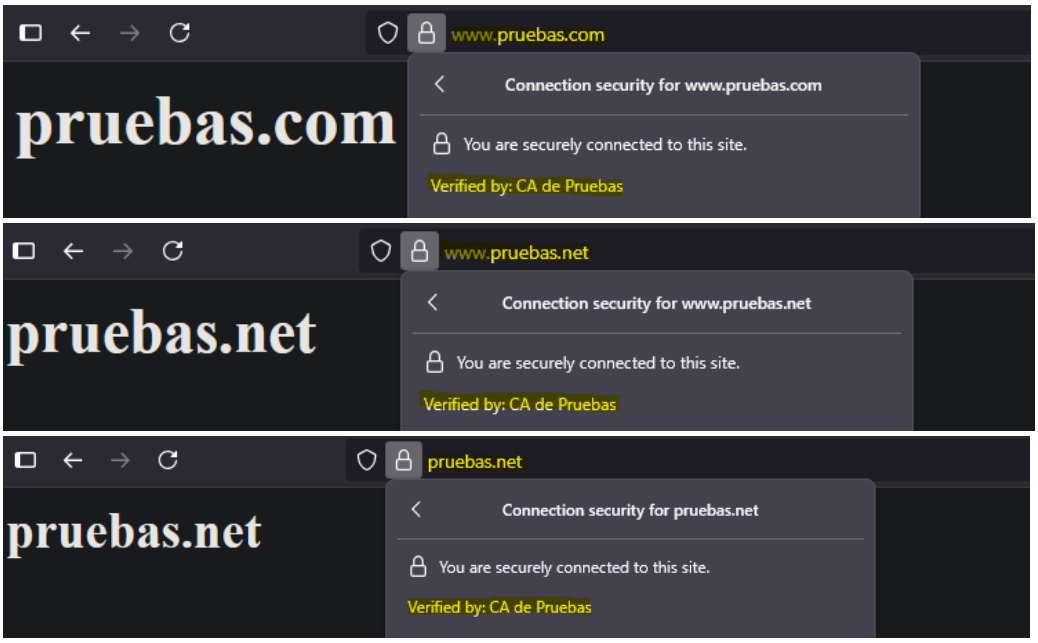

Y editamos el archivo /etc/apache2/sites-availible/000-default.conf para que tenga un virtualhost para cada página con sus propios DirectoryRoot y nombres (sigue usando los mismos certificados porque fueron emitidos para todos esos dominios):

```
Listen 0.0.0.0:443

<VirtualHost *:80>
    ServerName www.pruebas.com
    ServerAlias pruebas.com
    Redirect / https://www.pruebas.com/
</VirtualHost>

<VirtualHost *:443>
    ServerName www.pruebas.com
    ServerAlias pruebas.com

    DocumentRoot /projects/pruebas/com

    SSLEngine on
    SSLCertificateFile /root/pruebas.crt
    SSLCertificateKeyFile /root/pruebas.key
    SSLCertificateChainFile /root/pruebas.csr

    <Directory "/projects/pruebas/com">
        DirectoryIndex index.html
        Require all granted
        AllowOverride All
    </Directory>

</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName www.pruebas.net
    ServerAlias pruebas.net
    Redirect / https://www.pruebas.net/
</VirtualHost>

<VirtualHost *:443>
    ServerName www.pruebas.net
    ServerAlias pruebas.net

    DocumentRoot /projects/pruebas/net

    SSLEngine on
    SSLCertificateFile /root/pruebas.crt
    SSLCertificateKeyFile /root/pruebas.key
    SSLCertificateChainFile /root/pruebas.csr

    <Directory "/projects/pruebas/net">
        DirectoryIndex index.html
        Require all granted
        AllowOverride All
    </Directory>

</VirtualHost>
```

Y ahora todas las paginas siguen siendo accedibles pero son proyectos separados:






**10. Crear certificado wildcard para *.miempresa.com. Añadir servidor virtual para proyecto1.miempresa.es y comprobar que tanto proyecto1.miempresa.es y www.miempresa.es funcionan con el mismo certificado y sin errores.**

Bajo el directorio /etc/pki/tls crear un archivo llamado san_miempresa.cnf y meter el siguiente contenido (en la rocky linux):

```
[ server_SAN ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = *.miempresa.es
DNS.2 = miempresa.es

~
~
~
```

Dentro dela ubuntu crear la clave y el CSR (dado que ahí es dónde estará el apache:

```
An optional company name []:
upache@upache:~$ openssl genrsa -out miempresa-key.pem 2048
```

```
root@server:/etc/pki/tls   upache@upache: ~
upache@upache:~$ openssl genrsa -out miempresa-key.pem 2048
upache@upache:~$ openssl req -new -key miempresa-key.pem -out miempresa-csr.pem
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]:Madrid
Locality Name (eg, city) []:Alcorcon
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CA de Pruebas
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:*.miempresa.es
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

Copiamos la CSR de la Ubuntu a la Rocky para firmarla por la CA:

```
upache@upache:~$ scp miempresa-csr.pem root@192.168.217.140:/etc/pki/tls
root@192.168.217.140's password:
miempresa-csr.pem                                     100%  997   621.7KB/s   00:00
upache@upache:~$
```

```
root@server:/etc/pki/tls   upache@upache: ~
root@server tls]# ls
cert.pem              crl              index.txt.attr.old     misc          san_miempresa.cnf
cert.srl              crlnumber        index.txt.old          newcerts      serial
-csr.pem              crlnumber.old    juan-csr.pem           openssl.cnf   serial.old
rmen-cabrera-crt.p12  crl.pem          Mario-Martinez-crt.p12 openssl.d     server-crt.pem
rmen-cabrera-crt.pem  ct_log_list.cnf  Mario-Martinez-crt.pem private       server-csr.pem
rmen-cabrera-csr.pem  fips_local.cnf   Mario-Martinez-csr.pem pruebas.cnf   server-key.pem
rmen-cabrera-key.pem  index.txt        Mario-Martinez-key.pem pruebas.crt
rts                   index.txt.attr   miempresa-csr.pem      pruebas.csr
root@server tls]#
```

Firmamos:

```
sudo openssl ca -extensions server_SAN -extfile /etc/pki/tls/san_miempresa.cnf \
-in miempresa-csr.pem -out miempresa-crt.pem -days 825
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4 (0x4)
        Validity
            Not Before: Nov  2 13:15:20 2025 GMT
            Not After : Feb  5 13:15:20 2028 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Madrid
            organizationName          = CA de Pruebas
            commonName                = *.miempresa.es
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                DNS:*.miempresa.es, DNS:miempresa.es
Certificate is to be certified until Feb  5 13:15:20 2028 GMT (825 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
[root@server tls]#
```

```
root@server:/etc/pki/tls   upache@upache: ~
[root@server tls]# ls
cacert.pem              crl              index.txt.attr.old  miempresa-csr.pem  pruebas.csr
cacert.srl              crlnumber        index.txt.old       misc               san_miempresa.cnf
ca-csr.pem              crlnumber.old    juan-csr.pem        newcerts           serial
carmen-cabrera-crt.p12  crl.pem          Mario-Martinez-crt.p12  openssl.cnf    serial.old
carmen-cabrera-crt.pem  ct_log_list.cnf  Mario-Martinez-crt.pem  openssl.d      server-crt.pem
carmen-cabrera-csr.pem  fips_local.cnf   Mario-Martinez-csr.pem  private        server-csr.pem
carmen-cabrera-key.pem  index.txt        Mario-Martinez-key.pem  pruebas.cnf    server-key.pem
certs                   index.txt.attr   miempresa-crt.pem      pruebas.crt
[root@server tls]#
```

Ahora pasamos el certificado firmado y la CA a la Ubuntu:

```
[root@server tls]# sudo scp /etc/pki/tls/miempresa-crt.pem upache@192.168.217.135:/home/upache
upache@192.168.217.135's password:
miempresa-crt.pem                                             100% 4707     1.2MB/s   00:00
[root@server tls]#
```

```
root@server:/etc/pki/tls   upache@upache: ~
[root@server tls]# sudo scp /etc/pki/tls/cacert.pem upache@192.168.217.135:/home/upache
upache@192.168.217.135's password:
cacert.pem                                                    100% 1119    61.7KB/s   00:00
[root@server tls]#
```

Creamos el contenido para la página de miempresa.es:

```
upache@upache:~$ sudo mkdir -p /projects/miempresa
[sudo] password for upache:
upache@upache:~$
```

```
upache@upache:/projects/miempresa$ sudo cat index.html
<html>

        <h1>Hola miempresa.es</h1>
</html>
upache@upache:/projects/miempresa$
```

Creamos el servidor virtual:

```
root@server:~   upache@upache: ~
upache@upache:~$ sudo vi /etc/apache2/sites-available/miempresa.conf
```

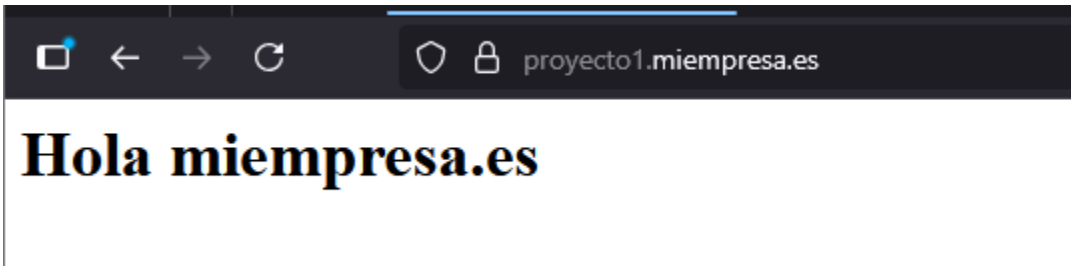
```
root@server:~   upache@upache: ~
<VirtualHost *:443>
    ServerName proyecto1.miempresa.es
    ServerAlias www.miempresa.es
    DocumentRoot /projects/miempresa

    SSLEngine on
    SSLCertificateFile miempresa-crt.pem
    SSLCertificateKeyFile miempresa-key.pem
    SSLCACertificateFile cacert.pem

    <Directory /projects/miempresa>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/miempresa_error.log
    CustomLog ${APACHE_LOG_DIR}/miempresa_access.log combined
</VirtualHost>

<VirtualHost *:80>
    ServerName proyecto1.miempresa.es
    ServerAlias www.miempresa.es
    Redirect permanent / https://proyecto1.miempresa.es/
</VirtualHost>

~
```

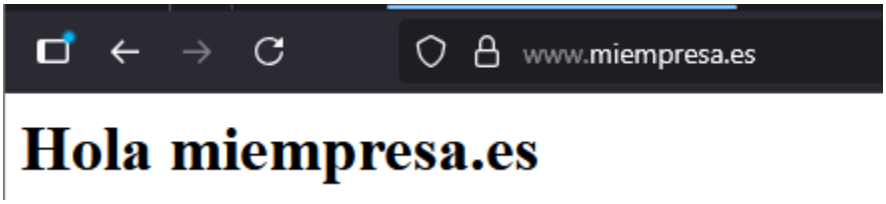
Activamos la configuración y reiniciamos el servicio de apache2:

```
upache@upache:~$ sudo a2ensite miempresa.conf
Site miempresa already enabled
upache@upache:~$ sudo systemctl restart apache2
upache@upache:~$
```

En /etc/hosts del cliente (windows en nuestro caso)

```
192.168.217.135 proyecto1.miempresa.es www.miempresa.es
```




**11. Firmar un documento PDF (el de esta memoria). Como ninguno del grupo dispone de un lector de DNIs ni de un certificado ofocial gubernamental, firmaremos todo con nuestra propia CA:**

Creamos claves, CSR y lo firmamos con la CA:

```
[root@server ~]# openssl genrsa -out dani_giulio_adolfo.key 2048
[root@server ~]# openssl req -new -key dani_giulio_adolfo.key -out dani_giulio_adolfo.csr
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.
-----
CountryName [ES]:ES
Province name [Madrid]:
Locality name [Alcorcon]:
Organization Name [CA de Pruebas]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:Dani_Giulio_Adolfo
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

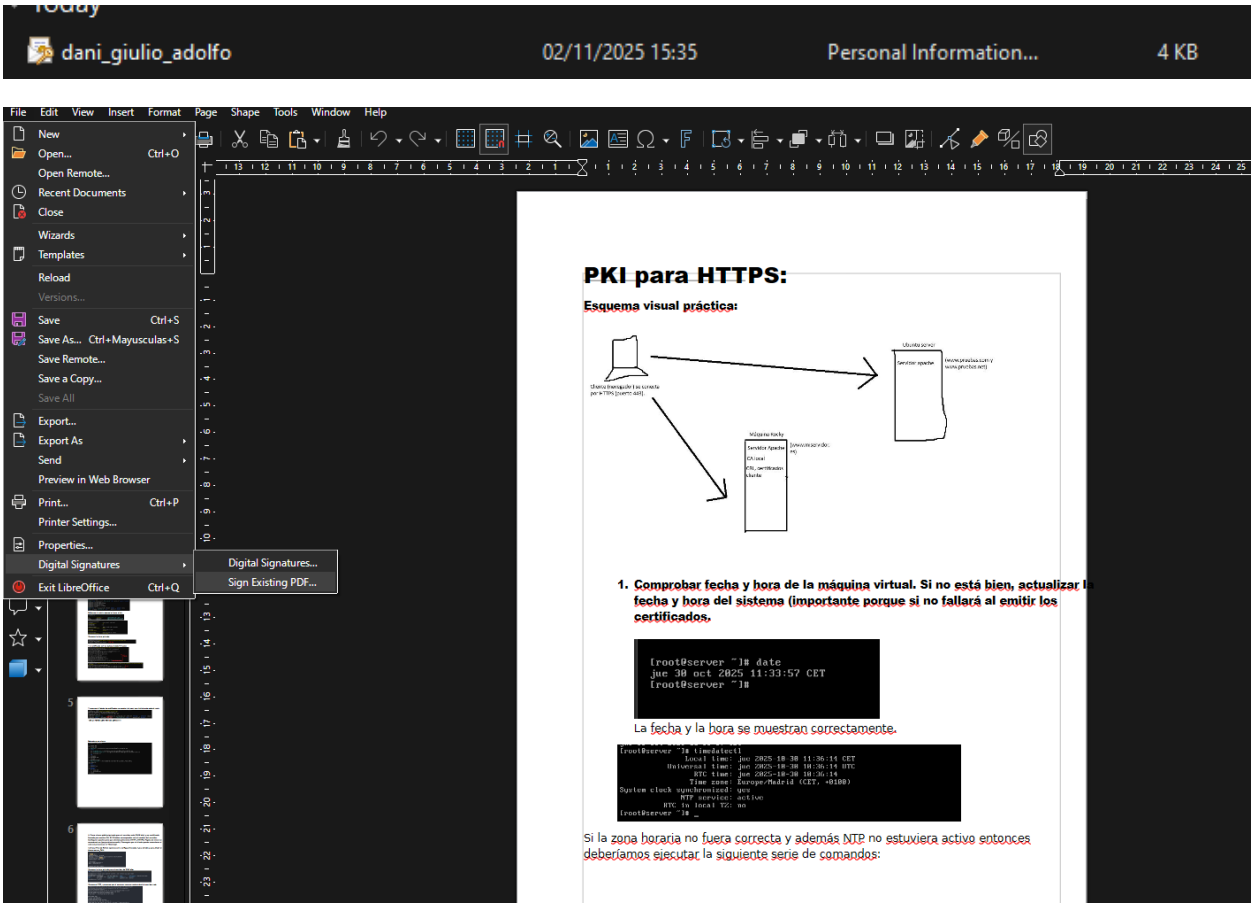
```
[root@server ~]# sudo openssl ca -in dani_giulio_adolfo.csr -out dani_giulio_adolfo.crt -days 365
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/tls/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 5 (0x5)
        Validity
            Not Before: Nov  2 14:33:02 2025 GMT
            Not After : Nov  2 14:33:02 2026 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Madrid
            organizationName          = CA de Pruebas
            commonName                = Dani_Giulio_Adolfo
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                80:62:E6:F1:46:B4:F7:02:C0:33:07:AC:D8:CE:82:02:37:5B:F2:EB
            X509v3 Authority Key Identifier:
                F5:33:6E:DC:CF:D5:42:C9:16:BB:7D:EC:A9:4F:6D:B5:04:29:74:EA
Certificate is to be certified until Nov  2 14:33:02 2026 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
[root@server ~]#
```

```
[root@server ~]# ls
anaconda-ks.cfg  dani_giulio_adolfo.crt  dani_giulio_adolfo.csr  dani_giulio_adolfo.key
[root@server ~]#
```

Exportamos el certificado a formato PKCS12 para firmar fácilmente con algún software:

```
[root@server ~]# ls
anaconda-ks.cfg  dani_giulio_adolfo.crt  dani_giulio_adolfo.csr  dani_giulio_adolfo.key
[root@server ~]# sudo openssl pkcs12 -export \
-inkey dani_giulio_adolfo.key \
-in dani_giulio_adolfo.crt \
-certfile /etc/pki/tls/cacert.pem \
-out /etc/pki/tls/dani_giulio_adolfo.p12 \
-name "Dani_Giulio_Adolfo"
Enter Export Password:
Verifying - Enter Export Password:
[root@server ~]#
```

Pasamos el certiticado en formato PKCS12 de la máquina a nuestro host porque firmaremos el PDF utilizando las herramientas de **LibreOffice Draw**:




Firmamos con nuestro certificado: