

Abstract geometric lines in the top left corner, consisting of several overlapping, irregular polygons and lines in a light beige color.

# PRÁCTICA 2: PKI PARA HTTPS

Paloma Pérez de Madrid e Inés del Río

# OBJETIVO

En esta práctica se aborda el protocolo HTTPS (*Hypertext Transfer Protocol over Secure Socket Layer*), que es la base del servicio WWW seguro. También se trata la certificación X.509 y las infraestructuras PKI. Cubriremos los siguientes objetivos:

- Descripción de las conexiones HTTPS
- Presentación de la librería OpenSSL
- Configuración del modulo SSL/TLS de Apache. Servidores virtuales.
- Definición de una CA (Autoridad de Certificación) y gestión de certificados
- Autenticación de clientes Web mediante certificados

Usaremos una máquina virtual Linux con distribución Ubuntu para construir la PKI y configurar servidores virtuales HTTPS. Para realizar las pruebas usaremos nuestro propio equipo de usuario, o cualquier otro que cuente con interface gráfico y navegador web.

Nota: en el SO Windows, ubicar los certificados de la práctica de en el repositorio de **USUARIO**.

# ÍNDICE

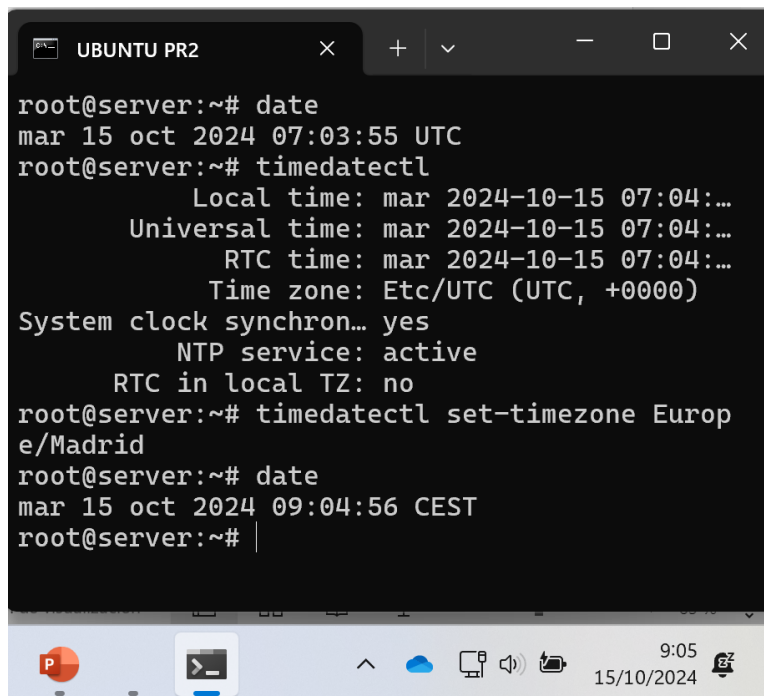
1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# ÍNDICE

- 1. Comprobación y actualización de la fecha y hora del sistema**
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPKI, Dogtag, etc.)

# COMPROBACIÓN Y ACTUALIZACIÓN DE LA FECHA Y HORA DEL SISTEMA

Comprobar que la máquina virtual Linux cuenta con fecha y hora válidas. Si no es así, actualizar la fecha/hora del sistema. Puede utilizarse el comando `date -s`, o NTP



```
UBUNTU PR2
root@server:~# date
mar 15 oct 2024 07:03:55 UTC
root@server:~# timedatectl
    Local time: mar 2024-10-15 07:04:...
    Universal time: mar 2024-10-15 07:04:...
    RTC time: mar 2024-10-15 07:04:...
    Time zone: Etc/UTC (UTC, +0000)
    System clock synchron... yes
    NTP service: active
    RTC in local TZ: no
root@server:~# timedatectl set-timezone Europe/Madrid
root@server:~# date
mar 15 oct 2024 09:04:56 CEST
root@server:~#
```

Máquina Ubuntu

Máquina Anfitriona  
(Windows)

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
- 2. Instalación y configuración de Apache con SSL**
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# COMPROBACIÓN Y ACTUALIZACIÓN DE LA FECHA Y HORA DEL SISTEMA

Instalar en la VM el servidor Apache y activar el módulo SSL. Iniciar el servidor y comprobar que acepta conexiones HTTPS. Describir las características del certificado que envía el servidor. Justificar las razones de la alerta que nos muestra nuestro navegador.

# apt install apache2

# a2enmod ssl → activa el módulo SSL

# a2ensite default-ssl.conf → activa la configuración por defecto de SSL

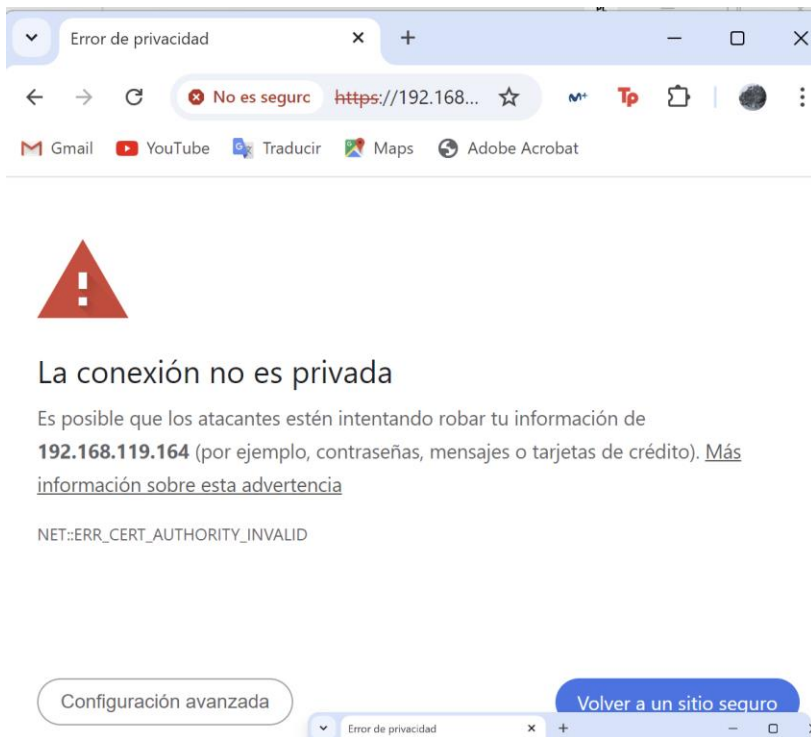
```
root@server:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz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

```
root@server:~# a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
root@server:~# systemctl restart apache2
root@server:~# curl https://localhost
curl: (60) SSL: no alternative certificate subject name matches target host name 'localhost'
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
root@server:~# |
```

# COMPROBACIÓN Y ACTUALIZACIÓN DE LA FECHA Y HORA DEL SISTEMA

Instalar en la VM el servidor Apache y activar el módulo SSL. Iniciar el servidor y comprobar que acepta conexiones HTTPS. Describir las características del certificado que envía el servidor. Justificar las razones de la alerta que nos muestra nuestro navegador.





# COMPROBACIÓN Y ACTUALIZACIÓN DE LA FECHA Y HORA DEL SISTEMA

Instalar en la VM el servidor Apache y activar el módulo SSL. Iniciar el servidor y comprobar que acepta conexiones HTTPS. Describir las características del certificado que envía el servidor. Justificar las razones de la alerta que nos muestra nuestro navegador.

1. **Certificado autofirmado:** El certificado ha sido emitido por el mismo servidor (server). Los navegadores marcan esto como inseguro porque no proviene de una Autoridad de Certificación (CA) de confianza reconocida internacionalmente.
2. **Falta de información:** El certificado no contiene datos sobre la organización o unidad organizativa, lo que también puede generar desconfianza en los navegadores, ya que no pueden verificar a quién pertenece.
3. **Nombre común (CN) genérico:** El CN es simplemente "server", que no coincide con un dominio público o un nombre específico que permita verificar la identidad del sitio. Esto puede provocar una advertencia de seguridad relacionada con la falta de coincidencia del nombre del certificado.

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. **Creación de una Autoridad de Certificación (CA)**
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

*Identificar la ubicación del archivo openssl.cnf.* Crear una Autoridad de Certificación propia, con los siguientes parámetros por defecto: CountryName (ES), StateOrProvinceName (Madrid), LocalityName (Alcorcon) y OrganizationName (CA de Pruebas). A continuación, obtendremos las claves pública y privada de la CA (archivo cakey.pem, protegido por frase de paso) y un certificado autofirmado con CN = “CA de Pruebas” (cacert.pem). Por último, obtenemos el listado de certificados revocados (crl.pem) que inicialmente estará vacío.

```
root@server:~# ls /etc/ssl
certs  openssl.cnf  private
```

# apt install openssl

Crear estructura de directorios:

```
mkdir -p /etc/pki/tls
cd /etc/pki/tls
mkdir certs
mkdir crl
mkdir newcerts
mkdir private
touch index.txt
echo 01 > serial
echo 01 > crlnumber
```

```
root@server:~# mkdir -p /etc/pki/tls
root@server:~# cd /etc/pki/tls
root@server:/etc/pki/tls# mkdir certs
root@server:/etc/pki/tls# mkdir newcerts
root@server:/etc/pki/tls# mkdir private
root@server:/etc/pki/tls# touch index.txt
root@server:/etc/pki/tls# echo 01 > serial
root@server:/etc/pki/tls# echo 01 > crlnumber
```

# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

Identificar la ubicación del archivo openssl.cnf. Crear una Autoridad de Certificación propia, con los siguientes parámetros por defecto: *CountryName* (ES), *StateOrProvinceName* (Madrid), *LocalityName* (Alcorcon) y *OrganizationName* (CA de Pruebas). A continuación, obtendremos las claves pública y privada de la CA (archivo cakey.pem, protegido por frase de paso) y un certificado autofirmado con CN = “CA de Pruebas” (cacert.pem). Por último, obtenemos el listado de certificados revocados (crl.pem) que inicialmente estará vacío.

Crear una Autoridad de Certificación:

```
# vim /etc/ssl/openssl.cnf
```

```
#####
[ CA_default ]

dir                = /etc/pki/tls
```

```
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
prompt             = yes
```

```
[ tsa_config1 ]

# These are used by the TSA repl
dir              = /etc/pki/tsl
```

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

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

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

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

organizationalUnitName     = Organizational Unit Name (eg, section)

commonName                 = Common Name (eg, YOUR name)
commonName_max             = 64
```

# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

Identificar la ubicación del archivo openssl.cnf. Crear una Autoridad de Certificación propia, con los siguientes parámetros por defecto: CountryName (ES), StateOrProvinceName (Madrid), LocalityName (Alcorcon) y OrganizationName (CA de Pruebas). A continuación, *obtendremos las claves pública y privada de la CA (archivo cakey.pem, protegido por frase de paso)* y un certificado autofirmado con CN = “CA de Pruebas” (cacert.pem). Por último, obtenemos el listado de certificados revocados (crl.pem) que inicialmente estará vacío.

## Crear clave pública y privada de la CA:

```
# openssl genpkey -algorithm RSA -out
/etc/pki/tls/private/cakey.key -pkeyopt
rsa_keygen_bits:2048 -aes256
```

Frase de Paso: palomaines

Nota:

- El comando RSA también es válido: `openssl genrsa -aes256 -out private/server.pem`
- Si usas otro `openssl.cnf` hay que indicarlo en el comando con `-config /etc/pki/tls/openssl.cnf` (por ejemplo)

[illegible]

# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

Identificar la ubicación del archivo openssl.cnf. Crear una Autoridad de Certificación propia, con los siguientes parámetros por defecto: CountryName (ES), StateOrProvinceName (Madrid), LocalityName (Alcorcon) y OrganizationName (CA de Pruebas). A continuación, obtendremos las claves pública y privada de la CA (archivo cakey.pem, protegido por frase de paso) y un *certificado autofirmado* con CN = “CA de Pruebas” (cacert.pem). Por último, obtenemos el listado de certificados revocados (crl.pem) que inicialmente estará vacío.

## Crear Certificado autofirmado (cacert.pem):

```
# openssl req -x509 -new -key  
/etc/pki/tls/private/cakey.key -out  
/etc/pki/tls/cacert.pem -days 365
```

Frase de Paso: palomaines

```
root@server:/etc/pki/tls# ls | grep ".pem"  
cacert.pem  
root@server:/etc/pki/tls#
```

```
root@server:/etc/pki/tls# openssl req -x509 -new -key /etc/pki/tls/private/cakey.key -out /etc/pki  
/tls/cacert.pem -days 365  
Enter pass phrase for /etc/pki/tls/private/cakey.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) [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) []:CA de Pruebas
```

# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

Verificación:

```
# openssl x509 -in cacert.pem -noout -text
```

```
root@server:/etc/pki/tls# openssl x509 -in cacert.pem -noout -text
Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number:
      7f:e4:85:23:3e:a7:d0:e9:53:0c:da:8d:ee:40:ab:da:d4:b1:06:bd
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = ES, ST = Madrid, L = Alcorcon, O = CA de Pruebas, CN = CA de Pruebas
    Validity
      Not Before: Oct 19 09:17:15 2024 GMT
      Not After : Oct 19 09:17:15 2025 GMT
    Subject: C = ES, ST = Madrid, L = Alcorcon, O = CA de Pruebas, CN = CA de Pruebas
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:b3:d1:ce:4f:d5:13:74:06:ec:55:ae:28:5d:c7:
        c2:b0:97:b4:03:a5:e0:fe:1a:b1:85:b2:2e:12:bf:
        e5:91:82:fe:6a:3e:ec:d7:80:f7:8d:a5:67:36:06:
```

# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

Identificar la ubicación del archivo openssl.cnf. Crear una Autoridad de Certificación propia, con los siguientes parámetros por defecto: CountryName (ES), StateOrProvinceName (Madrid), LocalityName (Alcorcon) y OrganizationName (CA de Pruebas). A continuación, obtendremos las claves pública y privada de la CA (archivo cakey.pem, protegido por frase de paso) y un certificado autofirmado con CN = “CA de Pruebas” (cacert.pem). Por último, *obtenemos el listado de certificados revocados (crl.pem)* que inicialmente estará vacío.

## Crear la Lista de Certificados Revocados (CRL):

```
# openssl ca -gencrl -keyfile  
/etc/pki/tls/private/cakey.key -cert  
/etc/pki/tls/cacert.pem -out  
/etc/pki/tls/crl/crl.pem
```

El archivo no está vacío: Aunque la lista de certificados revocados esté vacía (no hay certificados revocados), la CRL aún contiene datos que describen el certificado de la CA, las fechas de emisión y expiración de la CRL, y otros metadatos.

```
root@server:/etc/pki/tls# openssl ca -gencrl -keyfile /etc/pki/tls/private/cakey.key -cert /etc/pki/tls/cacert.pem -out /etc/pki/tls/crl/crl.pem  
Using configuration from /usr/lib/ssl/openssl.cnf  
Enter pass phrase for /etc/pki/tls/private/cakey.key:  
root@server:/etc/pki/tls# cat crl/crl.pem  
-----BEGIN X509 CRL-----  
MIIBujCBowIBATANBgkqhkiG9w0BAQsFADBhMQswCQYDVQQGEwJFUzEPMA0GA1UE  
CAwGTWFKcmllkMREwDwYDVQQHDAhBbGNvcmNvbWJEWMBQGA1UECgwNQ0EgZGUgUHJ1  
ZWJhc2EwMBQGA1UEAwwNQ0EgZGUgUHJ1ZWJhc2EwMBQGA1UECgwNQ0EgZGUgUHJ1  
MTE4MDkxOTU5WqAOMAwWcG9YDVR0UBAMCAQYwDQYJKoZIhvcNAQELBQADggEBAHFj  
v2XG0AS2wi69KfLZH5imLEkBlai2rKf5HgJb3MztarGIniTNj/bDGp5GXAi7w+R7  
EpFsAjvEYdiUo655Uy30ps/3nomX1BHiP8Ra09TM8P0hEe0oMhemLeEs/wbtHEOH  
YTRKTRnqq5ovRXFcRr91UhDru7/A2Rc/1qQ0g2yv1sUT1t046wOpSCwLUq81Tz1  
m7Eigx9VScqx9n/x7IKQr0MwXaSOFXsFUVt2nfWD9MbznAw8TuCHhwWkTzm+2gsy  
DPAa2dtK0QyI5vUWnb4Fov02BaNzpwPCqyk/w3kv/V1Vr6pODrxkwC8ZJLK7QhbW  
74GmPoqptqykxf52Lt4=  
-----END X509 CRL-----
```



# CREACIÓN DE UNA AUTORIDAD DE CERTIFICACIÓN (CA)

Crear la Lista de Certificados Revocados (CRL):

```
root@server:/etc/pki/tls# openssl crl -in /etc/pki/tls/crl/crl.pem -noout -text
Certificate Revocation List (CRL):
  Version 2 (0x1)
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: C = ES, ST = Madrid, L = Alcorcon, O = CA de Pruebas, CN = CA de Pruebas
  Last Update: Oct 19 09:19:59 2024 GMT
  Next Update: Nov 18 09:19:59 2024 GMT
  CRL extensions:
    X509v3 CRL Number:
      6
No Revoked Certificates.
  Signature Algorithm: sha256WithRSAEncryption
  Signature Value:
    71:63:bf:65:c6:d0:04:b6:c2:2e:bd:29:f2:d9:1f:98:8c:94:
    49:01:d5:a8:b6:ac:a7:f9:1e:02:5b:dc:cc:ed:6a:b1:88:9e:
    24:cd:8f:f6:c3:1a:9e:46:5c:08:bb:c3:e4:7b:12:91:6c:02:
    3b:c4:61:d8:94:a3:ae:79:53:2d:ce:a6:cf:f7:9e:89:97:d4:
    11:e2:3f:c4:5a:d3:d4:cc:f0:fd:21:11:e3:a8:32:17:a6:2d:
    e1:2c:ff:06:ed:1c:43:87:61:34:4a:4d:34:67:aa:ae:68:bd:
    15:c5:71:1a:fd:d5:48:43:ae:ee:ff:03:64:5c:ff:5a:90:d2:
    0d:b2:bf:5b:14:4f:5b:74:e3:ac:0e:a5:20:b0:2d:4a:bc:d5:
    3c:f5:9b:b1:22:83:1f:55:49:ca:b1:f6:7f:f1:ec:82:90:af:
    43:30:5d:a4:8e:17:1b:05:51:54:f6:9d:f5:83:f4:c6:f3:34:
    0c:3c:4e:e0:87:87:05:a4:4f:39:be:da:0b:32:0c:f0:1a:d9:
    db:4a:39:0c:88:e6:f5:16:9d:be:05:a2:f3:b6:05:a3:73:a7:
    03:c2:ab:29:3f:c3:79:2f:fd:5d:55:af:aa:4e:0e:bc:64:c0:
    2f:19:24:b2:bb:42:16:d6:ef:81:a6:3e:8a:a9:b6:ac:a4:c5:
    fe:76:2e:de
root@server:/etc/pki/tls#
```

No hay certificados  
revocados

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
- 4. Generación de claves y certificado para el servidor web**
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Crear una pareja de claves pública y privada RSA de 2048 bits para nuestro servidor web y un certificado firmado por nuestra CA. El CN (y el nombre alternativo SAN) deberá corresponder con el nombre del servidor (www.seguro.com). Configurar Apache para que atienda las peticiones dirigidas al dominio por HTTPS (redirigir también las conexiones HTTP hacia HTTPS). La página de inicio estará en /proyectos/seguro. Llevar a cabo los pasos necesarios para que nuestro cliente pueda conectar con el servidor vía HTTPS sin que se visualice ningún warning (reconociendo la CA y el certificado del servidor).

Abre en openssl.cnf y añade:

```
[ v3_req ]
# Extensions to add to a certificate request

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

[ alt_names ]
DNS.1 = www.seguro.com
DNS.2 = seguro.com
```

```
[ req ]
default_bits      = 2048
distinguished_name = req_distinguished_name
prompt            = yes
req_extensions    = v3_req
```

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Crear una pareja de claves pública y privada RSA de 2048 bits para nuestro servidor web y un certificado firmado por nuestra CA. El CN (y el nombre alternativo SAN) deberá corresponder con el nombre del servidor (www.seguro.com). Configurar Apache para que atienda las peticiones dirigidas al dominio por HTTPS (redirigir también las conexiones HTTP hacia HTTPS). La página de inicio estará en /proyectos/seguro. Llevar a cabo los pasos necesarios para que nuestro cliente pueda conectar con el servidor vía HTTPS sin que se visualice ningún warning (reconociendo la CA y el certificado del servidor).

1. Crear pareja de claves para el servidor web
2. Crear una solicitud de firma del certificado (CSR)
3. Firmar el certificado con nuestra CA

Crear pareja de claves para el servidor web:

- Generar la clave privada

```
# openssl genpkey -algorithm RSA -out  
/etc/pki/tls/private/servidor.key -pkeyopt rsa_keygen_bits:2048  
(uso rutas absolutas)
```

- Generar la CSR (Solicitud de firma de certificado)

```
# openssl req -new -key private/servidor.key -out  
certs/servidor.csr -extensions server_SAN  
(uso rutas relativas)
```

## GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Crear una pareja de claves pública y privada RSA de 2048 bits para nuestro servidor web y un certificado firmado por nuestra CA. El CN (y el nombre alternativo SAN) deberá corresponder con el nombre del servidor (www.seguro.com). Configurar Apache para que atienda las peticiones dirigidas al dominio por HTTPS (redirigir también las conexiones HTTP hacia HTTPS). La página de inicio estará en /proyectos/seguro. Llevar a cabo los pasos necesarios para que nuestro cliente pueda conectar con el servidor vía HTTPS sin que se visualice ningún warning (reconociendo la CA y el certificado del servidor).

[illegible]

```
root@server:/etc/pki/tls# openssl req -new -key private/servidor.key -out certs/servidor.csr -extensions server_SAN
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) []:www.seguro.com
```

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Firmar el CSR del servidor:

```
# openssl ca -in certs/servidor.csr -out  
certs/servidor.crt -cert cacert.pem -  
keyfile private/cakey.key -days 365 -  
extensions server_SAN
```

Nota: asegúrate que en openssl.cnf el archivo  
“cacert.pem” está ubicado en el sitio correcto  
[CA\_defaults]

Revocar certificado: `openssl ca -revoke  
certs/servidor.crt -keyfile private/cakey.key -  
cert cacert.pem`

```
root@server:/etc/pki/tls# openssl ca -in certs/servidor.csr -out certs/servidor.crt  
-cert cacert.pem -keyfile private/cakey.key -days 365 -extensions server_SAN  
Using configuration from /usr/lib/ssl/openssl.cnf  
Enter pass phrase for private/cakey.key:  
Check that the request matches the signature  
Signature ok  
Certificate Details:  
    Serial Number: 2 (0x2)  
    Validity  
        Not Before: Oct 24 10:01:22 2024 GMT  
        Not After : Oct 24 10:01:22 2025 GMT  
    Subject:  
        countryName           = ES  
        stateOrProvinceName    = Madrid  
        organizationName       = CA de Pruebas  
        commonName             = www.seguro.com  
    X509v3 extensions:  
        X509v3 Basic Constraints:  
            CA:FALSE  
        X509v3 Key Usage:  
            Digital Signature, Non Repudiation, Key Encipherment  
        X509v3 Subject Alternative Name:  
            DNS:www.seguro.com, DNS:seguro.com  
Certificate is to be certified until Oct 24 10:01:22 2025 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  
Data Base Updated
```

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Configurar Apache para HTTPS:

Activar el módulo write (para poder redirigir las peticiones HTTP a HTTPS):

```
# a2enmod rewrite
```

Crear un archivo de configuración del servidor:

```
# vim /etc/apache2/sites-available/seguro.conf
```

```
# mkdir /proyectos
```

```
# mkdir /proyectos/seguro
```

Habilitar y ejecutar apache:

```
# a2ensite seguro.conf
```

```
# systemctl restart apache2
```

```
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerName www.seguro.com
    ServerAlias seguro.com

    DocumentRoot /proyectos/seguro

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on

    SSLCertificateFile      /etc/pki/tls/certs/servidor.crt
    SSLCertificateKeyFile    /etc/pki/tls/private/servidor.key

    SSLCARevocationFile /etc/pki/tls/crl/crl.pem

    # <FilesMatch "\.(cgi|shtml|phtml|php)$">
    #     SSLOptions +StdEnvVars
    # </FilesMatch>
    # <Directory /usr/lib/cgi-bin>
    #     SSLOptions +StdEnvVars
    # </Directory>

    #SSLVerifyClient require
    #SSLVerifyDepth 1

    <Directory /proyectos/seguro>
        AllowOverride All
        Require all granted
        DirectoryIndex index.html
    </Directory>

  </VirtualHost>
  <VirtualHost *:80>
    ServerName www.seguro.com
    DocumentRoot /proyectos/seguro

    # Redirigir todo el tráfico HTTP a HTTPS
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^(.*)$ https://%{HTTP_HOST}%1 [R=301,L]
  </VirtualHost>
</IfModule>
```

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Crear una pareja de claves pública y privada RSA de 2048 bits para nuestro servidor web y un certificado firmado por nuestra CA. El CN (y el nombre alternativo SAN) deberá corresponder con el nombre del servidor (www.seguro.com). Configurar Apache para que atienda las peticiones dirigidas al dominio por HTTPS (redirigir también las conexiones HTTP hacia HTTPS). La página de inicio estará en /proyectos/seguro. Llevar a cabo los pasos necesarios para que nuestro cliente pueda conectar con el servidor vía HTTPS sin que se visualice ningún warning (reconociendo la CA y el certificado del servidor).

Si creas tu propia html para el servidor seguro:

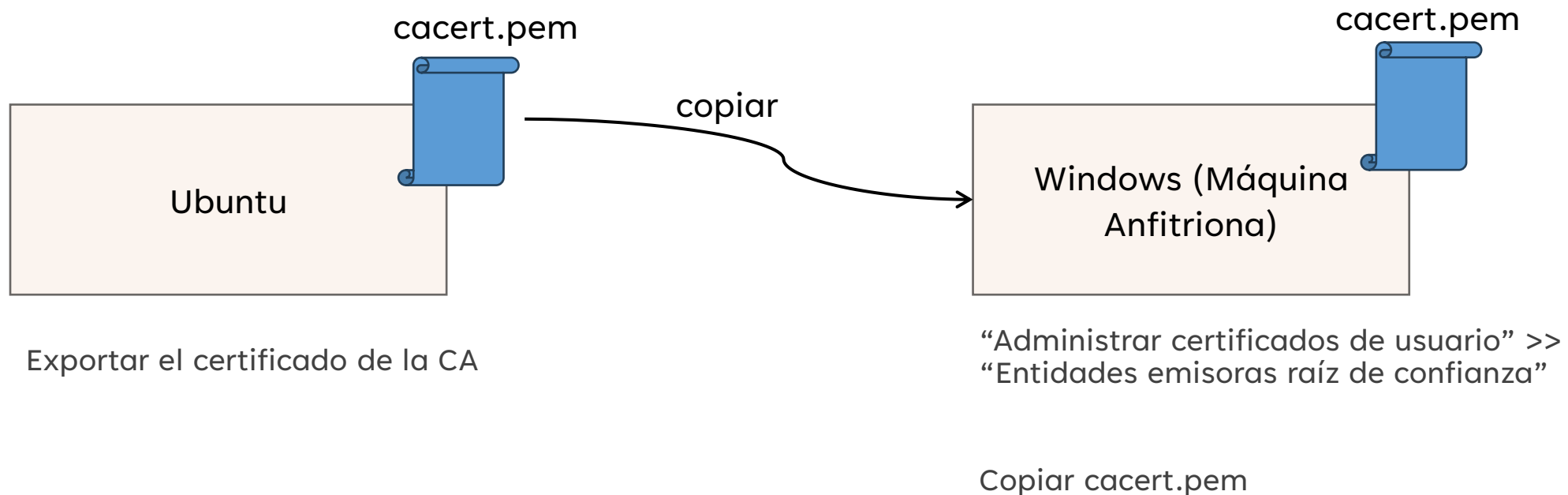
```
# a2dissite 000-default.conf
# a2dissite default-ssl.conf
# vim /proyectos/seguro/index.html
```





# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Crear una pareja de claves pública y privada RSA de 2048 bits para nuestro servidor web y un certificado firmado por nuestra CA. El CN (y el nombre alternativo SAN) deberá corresponder con el nombre del servidor (www.seguro.com). Configurar Apache para que atienda las peticiones dirigidas al dominio por HTTPS (redirigir también las conexiones HTTP hacia HTTPS). La página de inicio estará en /proyectos/seguro. Llevar a cabo los pasos necesarios para que nuestro cliente pueda conectar con el servidor vía HTTPS sin que se visualice ningún warning (reconociendo la CA y el certificado del servidor).



# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Seguridad – root@192.168.119.164 – WinSCP

Local Marcar Archivos Comandos Pestañas Opciones Remoto Ayuda

Sincronizar Cola ▾ Preajustes Predeterminado

root@192.168.119.164 X Nueva pestaña ▾


Escrito ▾ Subir ▾ Editar ▾ Propiedades ▾ Nuevo ▾

C:\Users\ppere\Desktop\Seguridad\

Nombre	Tamaño	Tipo	Modificado
Directorio superior			15/10/2024 8:34:30
PR1-VPN		Carpeta de archivos	15/10/2024 8:34:30
SPD - P02 - PKI para H...	166 KB	Microsoft Edge PDF...	15/10/2024 8:33:31

/etc/pki/tls/certs/

Nombre	Tamaño	Modificado	Permisos	Propietar...
..		16/10/2024 8:45:35	rwxr-xr-x	root
cacert.pem	2 KB	16/10/2024 8:40:46	rw-r--r--	root
servidor.crt	5 KB	16/10/2024 8:45:35	rw-r--r--	root
servidor.csr	1 KB	16/10/2024 8:44:46	rw-r--r--	root

 **Administrar certificados de usuario**  
Panel de control

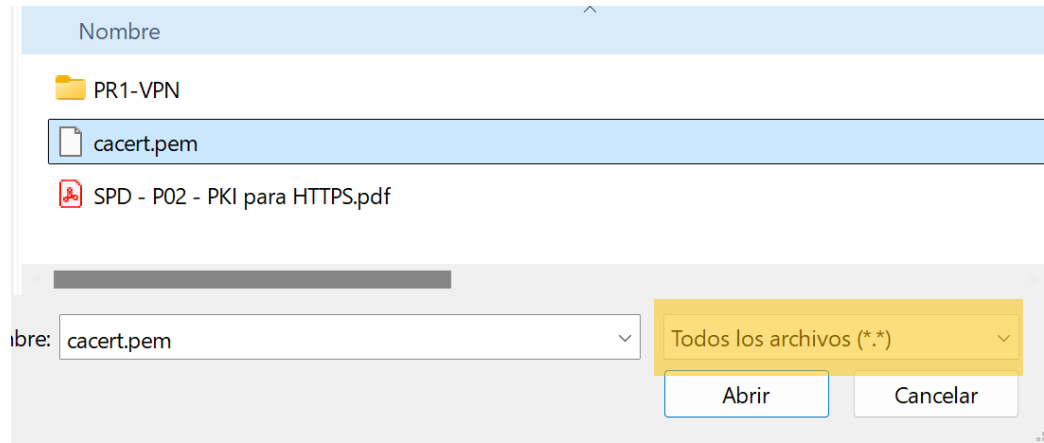
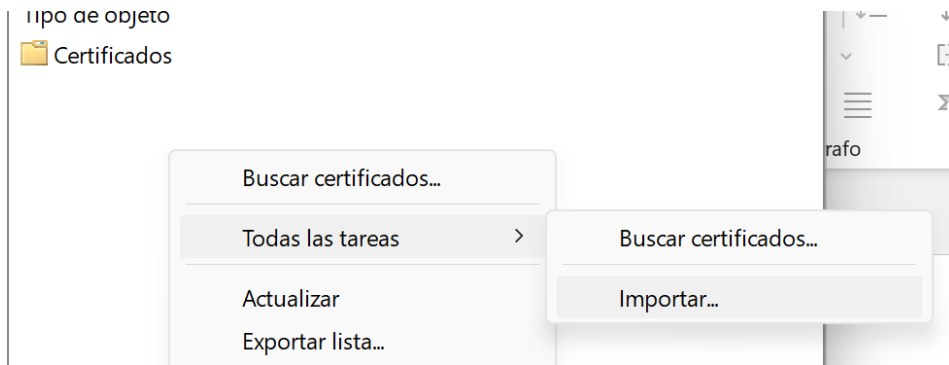
Certificados - Usuario actual

- Personal
- Entidades de certificación raíz de confianza
- Confianza empresarial
- Entidades de certificación intermedias
- Objeto de usuario de Active Directory

Nombre de almacén lógico

- Personal
- Entidades de certificación raíz de confianza
- Confianza empresarial
- Entidades de certificación intermedias

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB



## La conexión no es privada

Es posible que los atacantes estén intentando robar tu información de **192.168.119.164** (por ejemplo, contraseñas, mensajes o tarjetas de crédito). [Más información sobre esta advertencia](#)

NET::ERR\_CERT\_COMMON\_NAME\_INVALID

Configuración avanzada

Volver a un sitio seguro

Ahora sale un nuevo error, que el Common Name es inválido. Eso es porque estamos accediendo con la IP en vez de con “www.seguro.es”

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Para ello, hay que añadir en hosts.txt nuestro  
“www.seguro.com”

C:\Windows\System32\drivers\etc\hosts

Nota: abrirl cmd con permisos de administrador

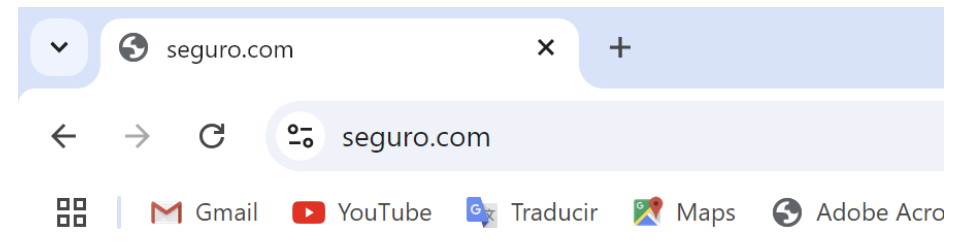
```
PS C:\Users\ppere> cd C:\Windows\System32\drivers\etc
PS C:\Windows\System32\drivers\etc> ls

Directorio: C:\Windows\System32\drivers\etc

Mode                LastWriteTime         Length Name
----                -
-a-----         14/03/2024    18:20         1017 hosts
-a-----         07/05/2022     7:22         3683 lmhosts.sam
-a-----         05/06/2021    14:08          407 networks
-a-----         05/06/2021    14:08         1358 protocol
-a-----         05/06/2021    14:08        17635 services

PS C:\Windows\System32\drivers\etc> notepad .\hosts
```

```
192.168.119.157 www.otraempresa.com otraempres
192.168.119.164 www.seguro.com seguro.com|
```



## Servidor Web Seguro

Servidor web seguro de la PR2 de Ciberseguridad

Nota: no salía error por no tener configurado el [server\_SAN] bien (diapositiva 19) y por no añadir el “-extensions” a la hora de generar el CSR (diapo. 21) y firmarlo (diapo. 23)

# GENERACIÓN DE CLAVES Y CERTIFICADO PARA EL SERVIDOR WEB

Visor de certificados: www.seguro.com

General

Detalles

Enviado a

Nombre común (CN)

www.seguro.com

Organización (O)

CA de Pruebas

Unidad organizativa (OU)

<No incluido en el certificado>

Emitido por

Nombre común (CN)

CA de Pruebas

Organización (O)

CA de Pruebas

Unidad organizativa (OU)

<No incluido en el certificado>

Período de validez

Emitido el

jueves, 24 de octubre de 2024, 12:01:22

Vencimiento el

viernes, 24 de octubre de 2025, 12:01:22

Huellas digitales SHA-256

Certificado

5b61ca3486f0e34fe4269c7280b270a8b3f0a2767aa7a290dcf73fe2aef05984

Clave pública

c1df9f3fbafd5b8ef701a0946d15e6759e22568291ffac2dc48ea77d755548c

Visor de certificados: www.seguro.com

General

Detalles

Jerarquía de certificados

CA de Pruebas

www.seguro.com

Campos de certificado

Uso de claves de certificado

Nombre alternativo de la entidad receptora del certificado

ID de clave de la entidad receptora del certificado

ID de clave de la entidad emisora de certificados

Algoritmo de firma de certificado

Valor de firma de certificados

Huellas digitales SHA-256

Certificado

Valor de campo

No crítica

Nombre de DNS: www.seguro.com

Nombre de DNS: seguro.com

Exportar...

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
- 5. Certificado de cliente y autenticación en Apache**
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# CERTIFICADO DE CLIENTE Y AUTENTICACIÓN EN APACHE

Crear un certificado de cliente a nombre de Víctor Vera (CN), del departamento de Ventas (OU), firmado por nuestra CA. A continuación, configurar el servidor Apache para que requiera certificados cliente en la raíz del proyecto a todas las conexiones HTTPS. Establecer una conexión HTTPS con el navegador y verificar que se requiere el certificado de cliente para el acceso. Instalar el certificado en el navegador y verificar que es posible visualizar la página principal del servidor web.

1. Crear pareja de claves para el servidor web
2. Crear una solicitud de firma del certificado (CSR)
3. Firmar el certificado con nuestra CA

**Nota:** para que en el CSR te pregunte sobre el OrganizationUnitName(OU) añadir la siguiente línea a /etc/ssl/openssl.cnf [req\_distinguished\_name]:

organizationalUnitName = Organizational Unit Name  
(eg, section)

## Crear pareja de claves para el servidor web:

```
# openssl genrsa -out private/victorvera.key 2048
```

(También se puede usar “genrsa” en vez de “genpkey”)

## Generar la CSR (Solicitud de firma de certificado)

```
# openssl req -new -key private/victorvera.key -out  
certs/victorvera.csr -extensions server_SAN
```

## Firmar el certificado:

```
# openssl ca -in certs/victorvera.csr -out certs/victorvera.crt -  
cert cacert.pem -keyfile private/cakey.key -days 365 -extensions  
server_SAN
```

```
root@server:/etc/pki/tls# openssl req -new -key private/victorvera.key -out certs/vic
torvera.csr -extensions server_SAN
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) []:Ventas
Common Name (eg, YOUR name) []:Victor Vega
```

```
root@server:/etc/pki/tls# openssl ca -in certs/victorvera.csr -out certs/victorvera.c
rt -cert cacert.pem -keyfile private/cakey.key -days 365 -extensions server_SAN
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for private/cakey.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 4 (0x4)
  Validity
    Not Before: Oct 24 10:28:15 2024 GMT
    Not After : Oct 24 10:28:15 2025 GMT
  Subject:
    countryName           = ES
    stateOrProvinceName   = Madrid
    organizationName      = CA de Pruebas
    organizationalUnitName = Ventas
    commonName            = Victor Vera
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Key Usage:
      Digital Signature, Non Repudiation, Key Encipherment
    X509v3 Subject Alternative Name:
      DNS:www.seguro.com, DNS:seguro.com
Certificate is to be certified until Oct 24 10:28:15 2025 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
Data Base Updated
```



# CERTIFICADO DE CLIENTE Y AUTENTICACIÓN EN APACHE

Crear un certificado de cliente a nombre de Víctor Vera (CN), del departamento de Ventas (OU), firmado por nuestra CA. A continuación, configurar el servidor Apache para que requiera certificados cliente en la raíz del proyecto a todas las conexiones HTTPS. Establecer una conexión HTTPS con el navegador y verificar que se requiere el certificado de cliente para el acceso. Instalar el certificado en el navegador y verificar que es posible visualizar la página principal del servidor web.

Configurar el servidor Apache para que requiera certificados cliente en la raíz del proyecto a todas las conexiones HTTPS

Añadir en /etc/apache2/sites-available/seguro.conf:

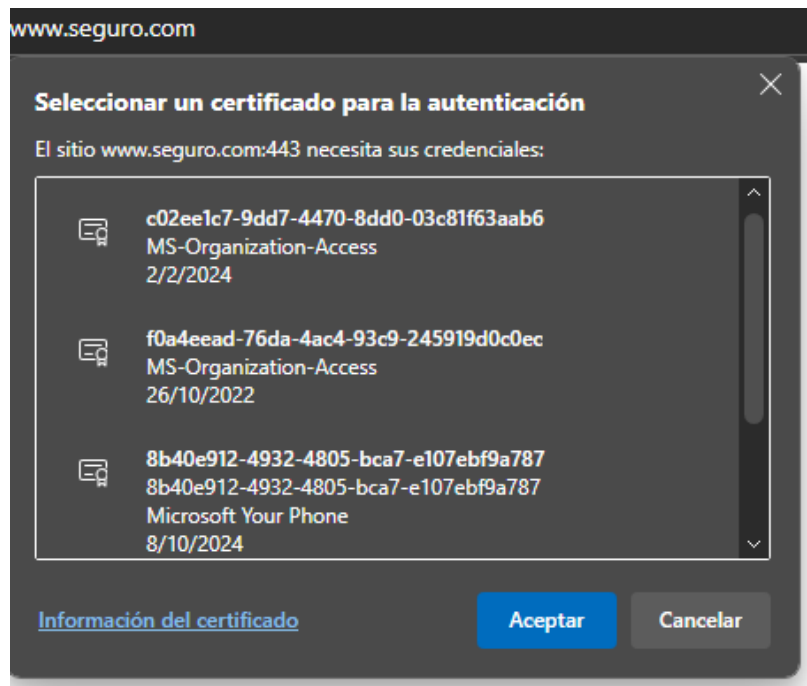
```
SSLVerifyClient require
SSLVerifyDepth 1
SSLCACertificateFile /etc/pki/tls/cacert.pem

# systemctl restart apache2
```

```
SSLVerifyClient require
SSLVerifyDepth 1
SSLCACertificateFile /etc/pki/tls/cacert.pem
```

# CERTIFICADO DE CLIENTE Y AUTENTICACIÓN EN APACHE

Crear un certificado de cliente a nombre de Víctor Vera (CN), del departamento de Ventas (OU), firmado por nuestra CA. A continuación, configurar el servidor Apache para que requiera certificados cliente en la raíz del proyecto a todas las conexiones HTTPS. Establecer una conexión HTTPS con el navegador y verificar que se requiere el certificado de cliente para el acceso. Instalar el certificado en el navegador y verificar que es posible visualizar la página principal del servidor web.



## Instalar certificado en el navegador

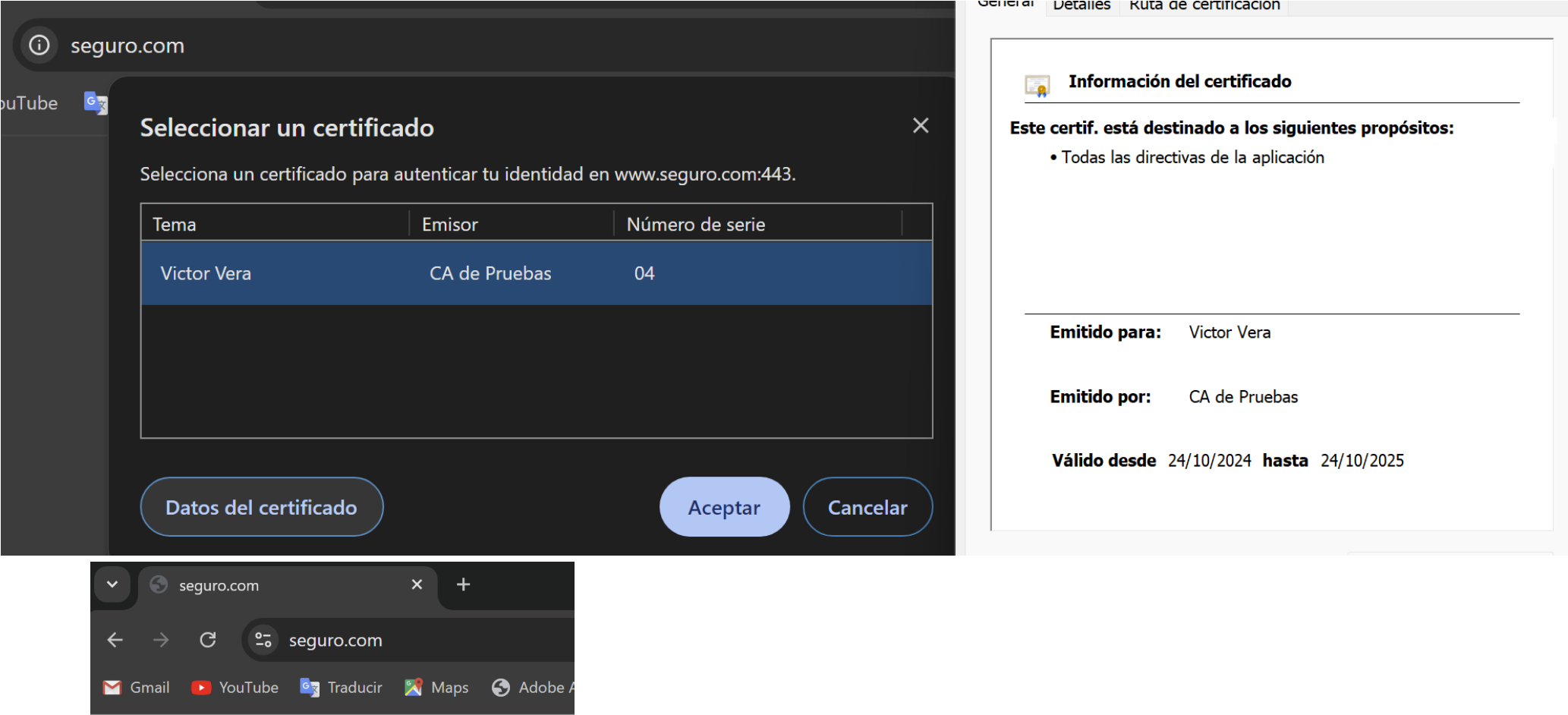
1. Convertimos el certificado de Víctor Vega en formato PKCS#12 (más amigable para los navegadores)

```
# openssl pkcs12 -export -in certs/victorvera.crt -inkey  
private/victorvera.key -out victorvera.p12 -name "Victor Vera"
```

Contraseña: victor (te pedirá una contraseña)

2. Lo subimos al navegador (en Chrome: chrome://certificate-manager/ )

# CERTIFICADO DE CLIENTE Y AUTENTICACIÓN EN APACHE



## Servidor Web Seguro

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
- 6. Acceso controlado a directorios mediante certificados de cliente**
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado Wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# ACCESO CONTROLADO A DIRECTORIOS MEDIANTE CERTIFICADOS DE CLIENTE

Crear un nuevo certificado a nombre de Pilar Perez (CN), del departamento de Personal (OU), e instalarlo en el navegador. En el servidor web, crear dos directorios, Ventas y Personal, con una página de inicio en cada uno. Configurar el servidor Apache para que sólo permita el acceso a estos directorios a los miembros de los respectivos departamentos (campo OU), tras la autenticación mediante certificado al entrar en el sitio web.

Crear certificado para Pilar Perez (Personal)

```
# openssl genrsa -out private/pilarperez.key 2048
# openssl req -new -key private/pilarperez.key -out
certs/pilarperez.csr -extensions server_SAN
# openssl ca -in certs/pilarperez.csr -out
certs/pilarperez.crt -cert cacert.pem -keyfile
private/cakey.key -days 365 -extensions server_SAN
# openssl pkcs12 -export -in certs/pilarperez.crt -inkey
private/pilarperez.key -out pilarperez.p12 -name "Pilar
Perez"
```

(contraseña: pilar)

```
root@server:/etc/pki/tls# openssl req -new -key private/pilarperez.key -out certs/pilarperez.
csr -extensions server_SAN
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) []:Personal
Common Name (eg, YOUR name) []:Pilar Perez
```

# ACCESO CONTROLADO A DIRECTORIOS MEDIANTE CERTIFICADOS DE CLIENTE

Crear un nuevo certificado a nombre de Pilar Perez (CN), del departamento de Personal (OU), e instalarlo en el navegador. En el servidor web, crear dos directorios, Ventas y Personal, con una página de inicio en cada uno. Configurar el servidor Apache para que sólo permita el acceso a estos directorios a los miembros de los respectivos departamentos (campo OU), tras la autenticación mediante certificado al entrar en el sitio web.

Crear dos directorios para cada departamento

```
root@server:/proyectos/seguro# mkdir ventas
root@server:/proyectos/seguro# mkdir personal
root@server:/proyectos/seguro# echo "<h1> Ventas </h1>" > ventas/index.html
root@server:/proyectos/seguro# echo "<h1> Personal </h1>" > personal/index.html
```

# ACCESO CONTROLADO A DIRECTORIOS MEDIANTE CERTIFICADOS DE CLIENTE

Crear un nuevo certificado a nombre de Pilar Perez (CN), del departamento de Personal (OU), e instalarlo en el navegador. En el servidor web, crear dos directorios, Ventas y Personal, con una página de inicio en cada uno. Configurar el servidor Apache para que sólo permita el acceso a estos directorios a los miembros de los respectivos departamentos (campo OU), tras la autenticación mediante certificado al entrar en el sitio web.

## Configurar Apache

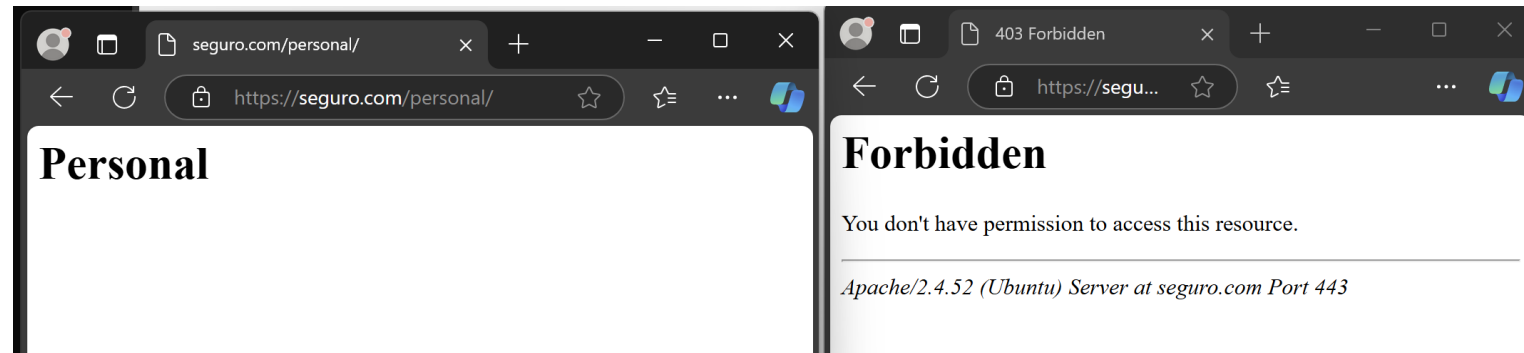
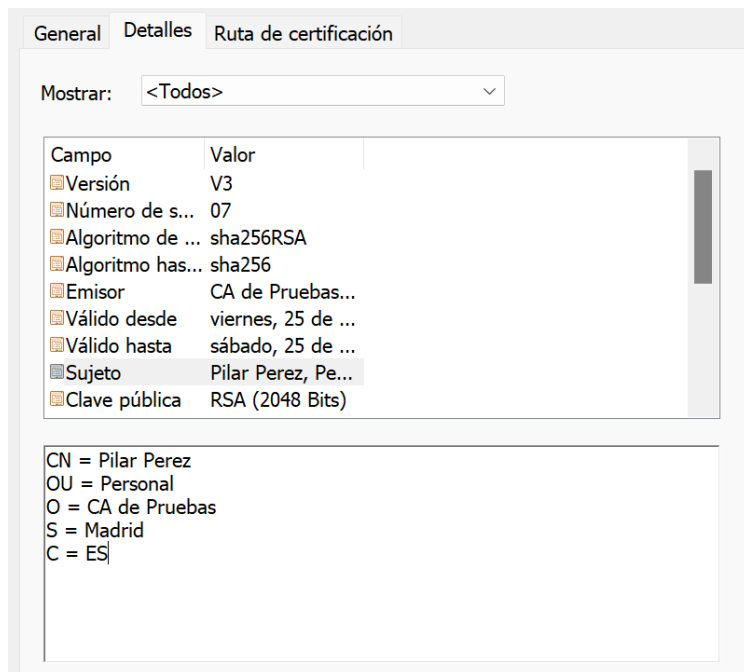
El bloque `SSLRequire %{SSL_CLIENT_S_DN_OU} eq "Ventas"` o `SSLRequire %{SSL_CLIENT_S_DN_OU} eq "Personal"` se encarga de validar que el campo OU (Organizational Unit) en el certificado del cliente coincide con el departamento correcto.

```
<Directory /proyectos/seguro/ventas>
    Options Indexes FollowSymLinks
    DirectoryIndex index.htm
    AllowOverride None
    SSLRequire %{SSL_CLIENT_S_DN_OU} eq "Ventas"
</Directory>

<Directory /proyectos/seguro/personal>
    Options Indexes FollowSymLinks
    DirectoryIndex index.html
    AllowOverride None
    SSLRequire %{SSL_CLIENT_S_DN_OU} eq "Personal"
</Directory>
```

# ACCESO CONTROLADO A DIRECTORIOS MEDIANTE CERTIFICADOS DE CLIENTE

Crear un nuevo certificado a nombre de Pilar Perez (CN), del departamento de Personal (OU), e instalarlo en el navegador. En el servidor web, crear dos directorios, Ventas y Personal, con una página de inicio en cada uno. Configurar el servidor Apache para que sólo permita el acceso a estos directorios a los miembros de los respectivos departamentos (campo OU), tras la autenticación mediante certificado al entrar en el sitio web.

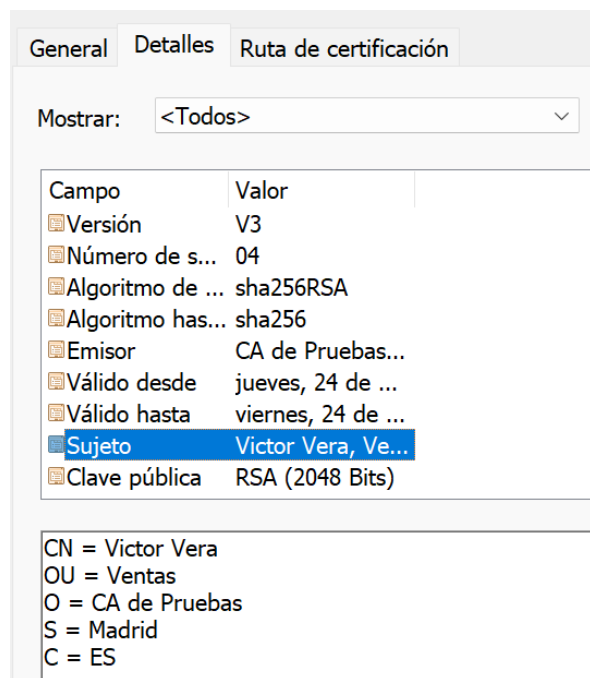


Pilar no puede entrar en ventas pero sí en personal

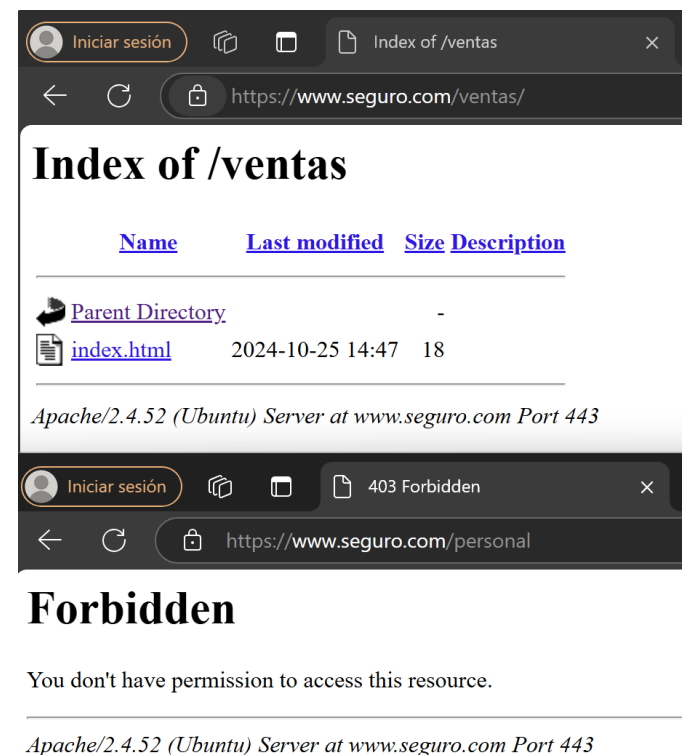


# ACCESO CONTROLADO A DIRECTORIOS MEDIANTE CERTIFICADOS DE CLIENTE

Crear un nuevo certificado a nombre de Pilar Perez (CN), del departamento de Personal (OU), e instalarlo en el navegador. En el servidor web, crear dos directorios, Ventas y Personal, con una página de inicio en cada uno. Configurar el servidor Apache para que sólo permita el acceso a estos directorios a los miembros de los respectivos departamentos (campo OU), tras la autenticación mediante certificado al entrar en el sitio web.



Víctor no puede entrar en Personal pero sí en ventas



# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
- 7. Revocación de certificados**
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado Wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# REVOCACIÓN DE CERTIFICADOS

Revocar el certificado del cliente Víctor Vera, y actualizar el archivo de certificados revocados de la CA. Configurar Apache para leer el archivo de revocaciones y comprobar a continuación que el usuario revocado ya no puede acceder vía HTTPS a los contenidos del servidor web.

1) Guardar el número de serie de victorvera.crt

```
# openssl x509 -in certs/victorvera.crt -noout -serial
```

2) Revoca el Certificado

```
# openssl ca -revoke certs/victorvera.crt -keyfile private/cakey.key -cert cacert.pem
```

3) Generar la lista de revocación de certificados (CRL)

```
# openssl ca -gencrl -keyfile private/cakey.key -cert cacert.pem -out crl/crl.pem
```

Actualizar /etc/apache2/sites-available/seguro.conf de apache:

```
SSLCARevocationFile /etc/pki/tls/crl/crl.pem
SSLCARevocationCheck chain
```

# REVOCACIÓN DE CERTIFICADOS

Revocar el certificado del cliente Víctor Vera, y actualizar el archivo de certificados revocados de la CA. Configurar Apache para leer el archivo de revocaciones y comprobar a continuación que el usuario revocado ya no puede acceder vía HTTPS a los contenidos del servidor web.

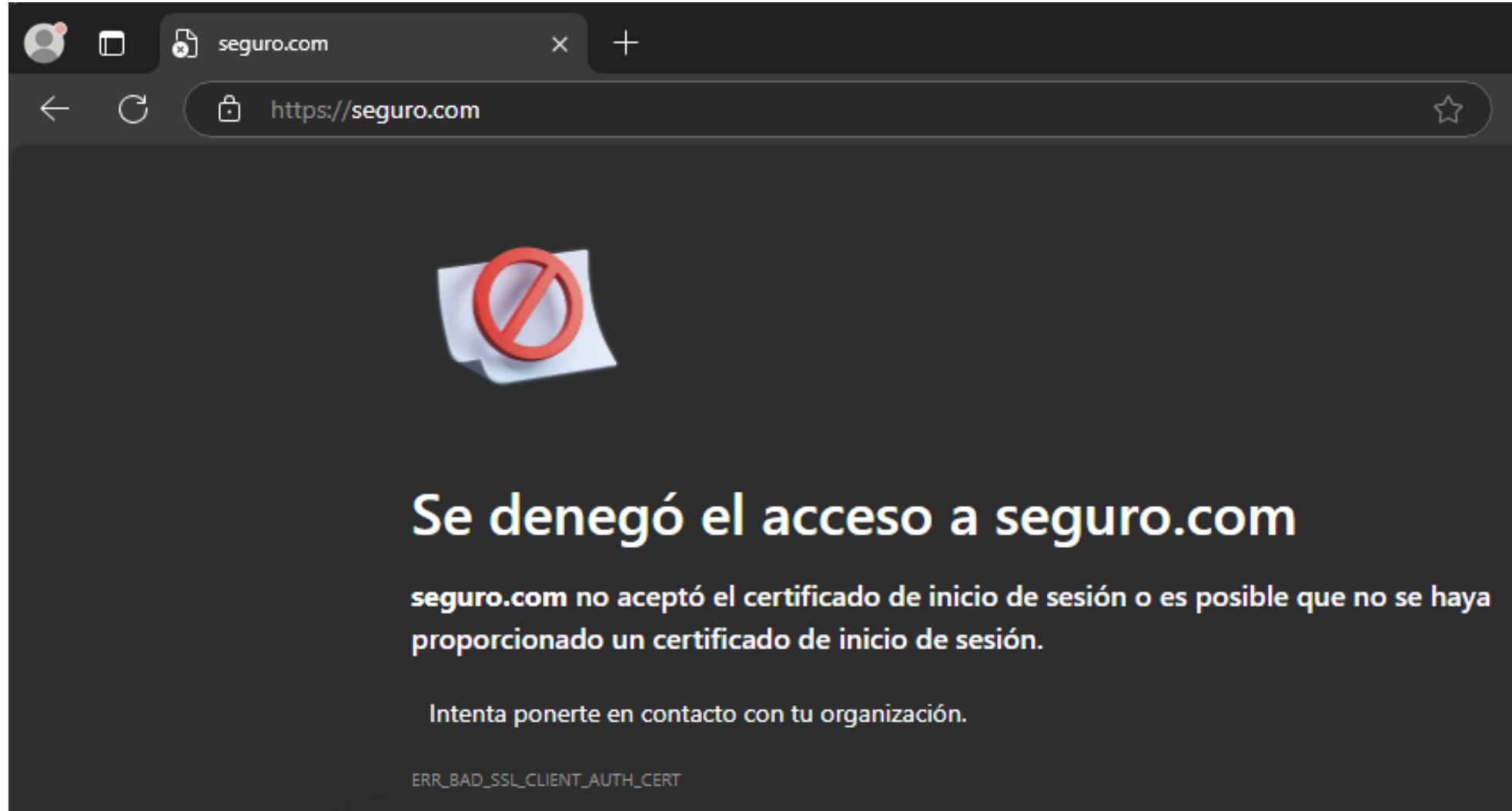
```
root@server:/etc/pki/tls# openssl x509 -in certs/victorvera.crt -noout -serial  
serial=08
```

```
root@server:/etc/pki/tls# openssl ca -revoke certs/victorvera.crt -keyfile private/cakey.key -cert cacert.pem  
Using configuration from /usr/lib/ssl/openssl.cnf  
Enter pass phrase for private/cakey.key:  
Revoking Certificate 08.  
Data Base Updated
```

```
root@server:/etc/pki/tls# openssl ca -gencrl -keyfile /etc/pki/tls/private/cakey.key -cert /etc/pki/tls/cacert.pem -out /etc/pki/tls/crl/crl.pem  
Using configuration from /usr/lib/ssl/openssl.cnf  
Enter pass phrase for /etc/pki/tls/private/cakey.key:  
root@server:/etc/pki/tls# |
```

```
root@server:/etc/pki/tls# openssl x509 -in certs/victorvera.crt -noout -serial  
serial=08  
root@server:/etc/pki/tls# openssl crl -in /etc/pki/tls/crl/crl.pem -noout -text | grep  
-A 1 "Serial Number: 08"  
    Serial Number: 08  
        Revocation Date: Oct 24 17:44:34 2024 GMT  
root@server:/etc/pki/tls# |
```

# REVOCACIÓN DE CERTIFICADOS



# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. **Configuración de un servidor virtual para el dominio `www.midominio.com`**
9. Configuración de proyectos independientes para dos dominios
10. Certificado Wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

Vamos a configurar un nuevo servidor virtual HTTPS para el dominio `www.midominio.com` (con su propio DocumentRoot, en `/proyectos/midominio`). Las peticiones podrán hacerse sin incluir el prefijo `www`, y también tendremos en cuenta que algunos usuarios prefieren usar el nombre de dominio `www.midominio.org` (nuevamente no será necesario incluir el prefijo `www`). Llevar a cabo la configuración del servidor, que contará con su página de inicio. Comprobar que accedemos al contenido del servidor sin errores ni alertas usando cualquiera de los nombres propuestos.

1) Creamos directorios y archivo de index:

```
root@server:/etc/pki/tls# mkdir -p /proyectos/midominio
root@server:/etc/pki/tls# echo "<h1> Mi dominio!! </h1>" > /proyectos/midominio/index.html
echo "<h1> Mi dominio" > /proyectos/midominio/index.html
root@server:/etc/pki/tls#
```

2) Añadimos los `alt_names` de `midominio` a `/etc/ssl/openssl.cnf`

```
[ alt_names ]
DNS.1 = www.seguro.com
DNS.2 = seguro.com
DNS.3 = www.midominio.com
DNS.4 = midominio.com
DNS.5 = www.midominio.org
DNS.6 = midominio.org
```

# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

Vamos a configurar un nuevo servidor virtual HTTPS para el dominio `www.midominio.com` (con su propio DocumentRoot, en `/proyectos/midominio`). Las peticiones podrán hacerse sin incluir el prefijo `www`, y también tendremos en cuenta que algunos usuarios prefieren usar el nombre de dominio `www.midominio.org` (nuevamente no será necesario incluir el prefijo `www`). Llevar a cabo la configuración del servidor, que contará con su página de inicio. Comprobar que accedemos al contenido del servidor sin errores ni alertas usando cualquiera de los nombres propuestos.

## 3) Crear archivos para OpenSSL:

- Generar la clave privada

```
# openssl genpkey -algorithm RSA -out /etc/pki/tls/private/midominio.key -pkeyopt rsa_keygen_bits:2048
```

- Generar la CSR (Solicitud de firma de certificado)

```
# openssl req -new -key private/midominio.key -out certs/midominio.csr -extensions server_SAN
```

- Firmar el certificado:

```
# openssl ca -in certs/midominio.csr -out certs/midominio.crt -cert cacert.pem -keyfile private/cakey.key -days 365 -extensions server_SAN
```



# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

[illegible]

# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

```
root@server:/etc/pki/tls# openssl ca -in certs/midominio.csr -out certs/midominio.crt -cert cacert.p
em -keyfile private/cakey.key -days 365 -extensions server_SAN
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for private/cakey.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 9 (0x9)
  Validity
    Not Before: Oct 24 18:48:41 2024 GMT
    Not After : Oct 24 18:48:41 2025 GMT
  Subject:
    countryName           = ES
    stateOrProvinceName   = Madrid
    organizationName      = CA de Pruebas
    commonName             = www.midominio.com
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Key Usage:
      Digital Signature, Non Repudiation, Key Encipherment
    X509v3 Subject Alternative Name:
      DNS:www.seguro.com, DNS:seguro.com, DNS:www.midominio.com, DNS:midominio.com, DNS:ww
w.midominio.org, DNS:midominio.org
Certificate is to be certified until Oct 24 18:48:41 2025 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
Data Base Updated
root@server:/etc/pki/tls#
```

# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

Vamos a configurar un nuevo servidor virtual HTTPS para el dominio `www.midominio.com` (con su propio `DocumentRoot`, en `/proyectos/midominio`). Las peticiones podrán hacerse sin incluir el prefijo `www`, y también tendremos en cuenta que algunos usuarios prefieren usar el nombre de dominio `www.midominio.org` (nuevamente no será necesario incluir el prefijo `www`). Llevar a cabo la configuración del servidor, que contará con su página de inicio. Comprobar que accedemos al contenido del servidor sin errores ni alertas usando cualquiera de los nombres propuestos.

## 4) Añadir la configuración de Apache:

```
<VirtualHost *:443>
    ServerName www.midominio.com
    ServerAlias midominio.com
    ServerAlias www.midominio.org
    ServerAlias midominio.org

    DocumentRoot /proyectos/midominio
    <Directory /proyectos/midominio>
        AllowOverride All
        Require all granted
    </Directory>
    SSLEngine on

    SSLCertificateFile /etc/pki/tls/certs/midominio.crt
    SSLCertificateKeyFile /etc/pki/tls/private/midominio.key
    SSLCertificateChainFile /etc/pki/tls/cacert.pem

    ErrorLog ${APACHE_LOG_DIR}/midominio-error.log
    CustomLog ${APACHE_LOG_DIR}/midominio-access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName www.midominio.com
    ServerAlias midominio.com
    ServerAlias www.midominio.org
    ServerAlias midominio.org
    DocumentRoot /proyectos/midominio

    # Redirigir todo el tráfico HTTP a HTTPS
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
</VirtualHost>
```

Nota: ponerlo debajo de los `<VirtualHost>` dentro de `<IfModule>`

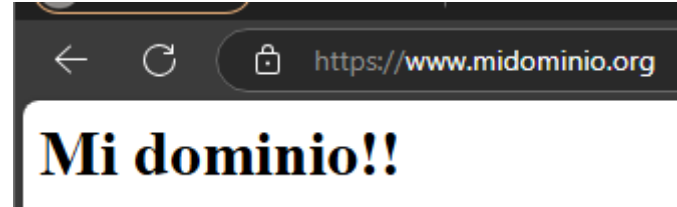
# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

Vamos a configurar un nuevo servidor virtual HTTPS para el dominio `www.midominio.com` (con su propio DocumentRoot, en `/proyectos/midominio`). Las peticiones podrán hacerse sin incluir el prefijo `www`, y también tendremos en cuenta que algunos usuarios prefieren usar el nombre de dominio `www.midominio.org` (nuevamente no será necesario incluir el prefijo `www`). Llevar a cabo la configuración del servidor, que contará con su página de inicio. Comprobar que accedemos al contenido del servidor sin errores ni alertas usando cualquiera de los nombres propuestos.

5) Para ello, hay que añadir en `hosts.txt` nuestro “`www.midominio.com`”

`C:\Windows\System32\drivers\etc\hosts`

```
192.168.119.157 www.otraempresa.com otraempresa.com
192.168.119.164 www.seguro.com seguro.com www.midominio.com midominio.com midominio.org www.midominio.org
```



# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
- 9. Configuración de proyectos independientes para dos dominios**
10. Certificado Wildcard para subdominios
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM

Suponga que ahora se decide usar ambos dominios (www.midominio.com y www.midominio.org) como dos proyectos independientes (con sus respectivos DocumentRoot y página de inicio). Configurar Apache para atender a ambos dominios. Nota: aunque es posible, en este apartado no usaremos nuevos certificados emitidos por nuestra CA; vamos a reutilizar el certificado creado para el apartado anterior.

/etc/apache2/sites-available/seguro.conf

```
<VirtualHost *:443>
    ServerName www.midominio.com
    ServerAlias midominio.com
    # ServerAlias www.midominio.org
    # ServerAlias midominio.org
    # DocumentRoot /proyectos/midominio
    DocumentRoot /proyectos/midominio/com

    <Directory /proyectos/midominio/com>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
    SSLEngine on

    SSLCertificateFile /etc/pki/tls/certs/midominio.crt
    SSLCertificateKeyFile /etc/pki/tls/private/midominio.key
    SSLCertificateChainFile /etc/pki/tls/cacert.pem

    ErrorLog ${APACHE_LOG_DIR}/midominiocom-error.log
    CustomLog ${APACHE_LOG_DIR}/midominiocom-access.log combined
</VirtualHost>
```

```
<VirtualHost *:443>
    ServerName www.midominio.org
    ServerAlias midominio.org
    DocumentRoot /proyectos/midominio/org

    <Directory /proyectos/midominio/org>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    SSLEngine on

    SSLCertificateFile /etc/pki/tls/certs/midominio.crt
    SSLCertificateKeyFile /etc/pki/tls/private/midominio.key
    SSLCertificateChainFile /etc/pki/tls/cacert.pem

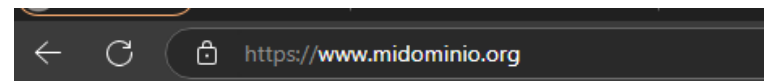
    ErrorLog ${APACHE_LOG_DIR}/midominioorg-error.log
    CustomLog ${APACHE_LOG_DIR}/midominioorg-access.log combined
</VirtualHost>
```

# CONFIGURACIÓN DE UN SERVIDOR VIRTUAL PARA EL DOMINIO WWW.MIDOMINIO.COM


Suponga que ahora se decide usar ambos dominios (www.midominio.com y www.midominio.org) como dos proyectos independientes (con sus respectivos DocumentRoot y página de inicio). Configurar Apache para atender a ambos dominios. Nota: aunque es posible, en este apartado no usaremos nuevos certificados emitidos por nuestra CA; vamos a reutilizar el certificado creado para el apartado anterior.

```
root@server:~# ls /proyectos/midominio/  
index.html  
root@server:~# mkdir /proyectos/midominio/com  
root@server:~# mkdir /proyectos/midominio/org
```

```
root@server:/proyectos/midominio/com# echo "midominio punto com" > midominiocom.txt  
root@server:/proyectos/midominio/com# cd ../org  
root@server:/proyectos/midominio/org# echo "midominio punto org" > midominioorg.txt
```




## Index of /

	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">midominioorg.txt</a>	2024-10-27 12:13	20	

Apache/2.4.52 (Ubuntu) Server at www.midominio.org Port 443



## Index of /

	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">midominiocom.txt</a>	2024-10-27 12:12	20	

Apache/2.4.52 (Ubuntu) Server at www.midominio.com Port 443

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
- 10. Certificado Wildcard para subdominios**
11. Firma de un documento PDF
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)



# CERTIFICADO WILDCARD PARA SUBDOMINIOS

Crear un certificado wildcard para **\*.gameover.com**. Añadir un nuevo servidor virtual para **pinball.gameover.com**, usando el nuevo certificado. Verificar que es posible acceder a los servicios de **pinball.gameover.com** y **www.gameover.com** usando el mismo certificado digital y sin errores.

Editar /etc/ssl/openssl.cnf

```
[ alt_names ]
DNS.1 = www.seguro.com
DNS.2 = seguro.com
DNS.3 = www.midominio.com
DNS.4 = midominio.com
DNS.5 = www.midominio.org
DNS.6 = midominio.org
DNS.7 = *.gameover.com
DNS.8 = www.gameover.com
```

la gracia es q no tengas un pinball en el DNS

Crear certificado para gameover

```
# openssl genrsa -out private/gameover.key 2048

# openssl req -new -key private/gameover.key -out certs/gameover.csr -extensions
server_SAN

# openssl ca -in certs/gameover.csr -out certs/gameover.crt -cert cacert.pem -keyfile
private/cakey.key -days 365 -extensions server_SAN


# openssl req -in certs/gameover.csr -noout -text (verificar)

# openssl ca -revoke certs/gameover.crt -keyfile private/cakey.key -cert cacert.pem
(revocar certificado)
```

# CERTIFICADO WILDCARD PARA SUBDOMINIOS

Crear un certificado wildcard para **\*.gameover.com**. Añadir un nuevo servidor virtual para **pinball.gameover.com**, usando el nuevo certificado. Verificar que es posible acceder a los servicios de **pinball.gameover.com** y **www.gameover.com** usando el mismo certificado digital y sin errores.

```
root@server:/etc/pki/tls# openssl genrsa -out private/gameover.key 2048
root@server:/etc/pki/tls# openssl req -new -key private/gameover.key -out certs/gameover.csr -extensions server_SAN
req: Use -help for summary.
root@server:/etc/pki/tls# openssl genrsa -out private/gameover.key 2048
root@server:/etc/pki/tls# openssl req -new -key private/gameover.key -out certs/gameover.csr -extensions server_SAN
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) []: *.gameover.com

root@server:/etc/pki/tls# openssl x509 -req -in certs/gameover.csr -CA cacert.pem -CAkey private/cakey
.key -CAcreateserial -out certs/gameover_wildcard.crt -days 365 -sha256
Certificate request self-signature ok
subject=C = ES, ST = Madrid, L = Alcorcon, O = CA de Pruebas, CN = *.gameover.com
Enter pass phrase for private/cakey.key:
root@server:/etc/pki/tls# |
```

# CERTIFICADO WILDCARD PARA SUBDOMINIOS

Crear un certificado wildcard para \*.gameover.com. Añadir un nuevo servidor virtual para pinball.gameover.com, usando el nuevo certificado. Verificar que es posible acceder a los servicios de pinball.gameover.com y www.gameover.com usando el mismo certificado digital y sin errores.

Configura Apache para Usar el Certificado Wildcard

```
# vim /etc/apache2/sites-available/gameover.conf
```

```
# a2ensite gameover.conf
```

```
<VirtualHost *:443>
    ServerName www.pinball.gameover.com
    ServerAlias pinball.gameover.com
    DocumentRoot /proyectos/gameover/pinball

    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/gameover.crt
    SSLCertificateKeyFile /etc/pki/tls/private/gameover.key

    <Directory /proyectos/gameover/pinball>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

```
<VirtualHost *:443>
    ServerName www.gameover.com
    ServerAlias *.gameover.com
    DocumentRoot /proyectos/gameover

    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/gameover.crt
    SSLCertificateKeyFile /etc/pki/tls/private/gameover.key

    <Directory /proyectos/gameover>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

# CERTIFICADO WILDCARD PARA SUBDOMINIOS

Crear un certificado wildcard para \*.gameover.com. Añadir un nuevo servidor virtual para pinball.gameover.com, usando el nuevo certificado. Verificar que es posible acceder a los servicios de pinball.gameover.com y www.gameover.com usando el mismo certificado digital y sin errores.

## Configura Apache para Usar el Certificado Wildcard

```
<VirtualHost *:80>
    ServerName www.gameover.com
    ServerAlias *.gameover.com

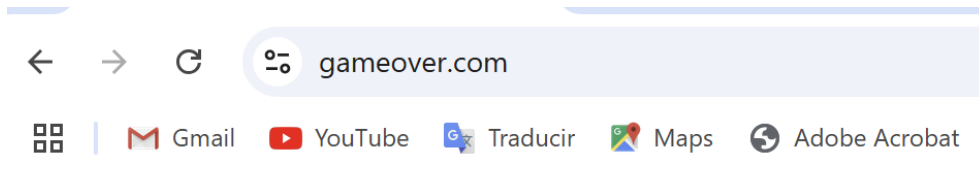
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</VirtualHost>
```

## Configurar hosts en la máquina anfitriona



```
PS C:\Windows\System32> notepad C:\Windows\System32\drivers\etc\hosts
PS C:\Windows\System32>
192.168.119.164 www.seguro.com seguro.com www.midominio.com midominio.com midominio.org www.m...
192.168.119.164 www.gameover.com gameover.com pinball.gameover.com www.pinball.gameover.com
```

# CERTIFICADO WILDCARD PARA SUBDOMINIOS

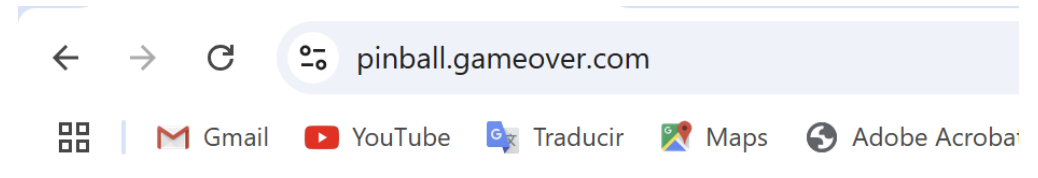
Crear un certificado wildcard para **\*.gameover.com**. Añadir un nuevo servidor virtual para **pinball.gameover.com**, usando el nuevo certificado. Verificar que es posible acceder a los servicios de **pinball.gameover.com** y **www.gameover.com** usando el mismo certificado digital y sin errores.




## Index of /

	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">gameover.txt</a>	2024-10-27 13:16	9	
	<a href="#">pinball/</a>	2024-10-27 13:16	-	

*Apache/2.4.52 (Ubuntu) Server at www.gameover.com Port 443*



## Index of /

	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">pinball.txt</a>	2024-10-27 13:16	8	

*Apache/2.4.52 (Ubuntu) Server at pinball.gameover.com Port 443*

# ÍNDICE

1. Comprobación y actualización de la fecha y hora del sistema
2. Instalación y configuración de Apache con SSL
3. Creación de una Autoridad de Certificación (CA)
4. Generación de claves y certificado para el servidor web
5. Certificado de cliente y autenticación en Apache
6. Acceso controlado a directorios mediante certificados de cliente
7. Revocación de certificados
8. Configuración de un servidor virtual para el dominio `www.midominio.com`
9. Configuración de proyectos independientes para dos dominios
10. Certificado Wildcard para subdominios
- 11. Firma de un documento PDF**
12. Partes opcionales
  - I. Creación de una PKI con autoridad raíz e intermedia
  - II. Estudio del proceso de obtención de certificados con Let's Encrypt
  - III. Instalación y pruebas con herramientas opensource para PKI (OpenXPki, Dogtag, etc.)

# FIRMA DE UN DOCUMENTO PDF

Firmar un documento PDF. Para realizar este apartado será necesario contar con una pareja de claves pública /privada y un certificado personal a nombre del alumno. Si es posible, se realizará con el DNI electrónico (se requiere un lector de SmartCard) o con un certificado personal emitido por alguna CA reconocida (por ejemplo, la FNMT). Si no se cuenta con ninguno de estos certificados, también es posible usar un certificado a nuestro nombre emitido por nuestra propia CA (en este último caso, se deberá adjuntar en el envío el certificado raíz de esta CA). El documento PDF puede ser la propia memoria de la práctica, y tendrá la firma visible en la última página del documento.

## [Autoridades de Certificación y Autoridades de Validación del DNle](#)

### **Autoridades de Certificación y Autoridades de Validación del DNle**

#### **Autoridad de Certificación**

La Dirección General de la Policía (Ministerio del Interior) actúa como Autoridad de Certificación (AC), relacionando dos pares de claves con un ciudadano concreto a través de la emisión de sendos Certificados de conformidad con los términos de esta DPC.

Las Autoridades de Certificación que componen la PKI del DNle son:

- **"AC Raíz"**: Autoridad de Certificación de primer nivel. Esta AC sólo emite certificados para sí misma y sus AC Subordinadas. Únicamente estará en funcionamiento durante la realización de las operaciones para las que se establece. Sus datos más relevantes son:
- **"AC Subordinadas"**: Autoridades de Certificación subordinadas de "AC Raíz". Su función es la emisión de certificados para los titulares de DNle.

## [Certificado con DNle - Sede](#)

## [Configuración Previa - Sede](#)

# FIRMA DE UN DOCUMENTO PDF

Firmar un documento PDF. Para realizar este apartado será necesario contar con una pareja de claves pública /privada y un certificado personal a nombre del alumno. Si es posible, se realizará con el DNI electrónico (se requiere un lector de SmartCard) o con un certificado personal emitido por alguna CA reconocida (por ejemplo, la FNMT). Si no se cuenta con ninguno de estos certificados, también es posible usar un certificado a nuestro nombre emitido por nuestra propia CA (en este último caso, se deberá adjuntar en el envío el certificado raíz de esta CA). El documento PDF puede ser la propia memoria de la práctica, y tendrá la firma visible en la última página del documento.

Aplicación: Autofirma (Gobierno de España)

Lector de tarjetas: Zoweetek

Nota: Hay que poner el pin del dni





# FIRMA DE UN DOCUMENTO PDF

Firmar un documento PDF. Para realizar este apartado será necesario contar con una pareja de claves pública /privada y un certificado personal a nombre del alumno. Si es posible, se realizará con el DNI electrónico (se requiere un lector de SmartCard) o con un certificado personal emitido por alguna CA reconocida (por ejemplo, la FNMT). Si no se cuenta con ninguno de estos certificados, también es posible usar un certificado a nuestro nombre emitido por nuestra propia CA (en este último caso, se deberá adjuntar en el envío el certificado raíz de esta CA). El documento PDF puede ser la propia memoria de la práctica, y tendrá la firma visible en la última página del documento.

## Solicitar Certificado - Sede

### 2. Solicitar Certificado

Antes de realizar este paso es necesario instalar el software del paso 1 Configuración previa.

Asegúrate que en esta solicitud te solicita establecer una contraseña nueva para solicitar el código y que será también requerida en el paso 4 de la Descarga.

#### SOLICITUD DE CERTIFICADO FNMT DE PERSONA FÍSICA

Para tramitar la solicitud de su Certificado FNMT de Persona Física, por favor introduzca la información requerida:

#### SOLICITUD DE CERTIFICADO FNMT DE PERSONA FÍSICA

Su solicitud ha sido procesada correctamente.

Por favor compruebe la exactitud de los datos introducidos:

Nº DEL DOCUMENTO DE IDENTIFICACIÓN

PRIMER APELLIDO

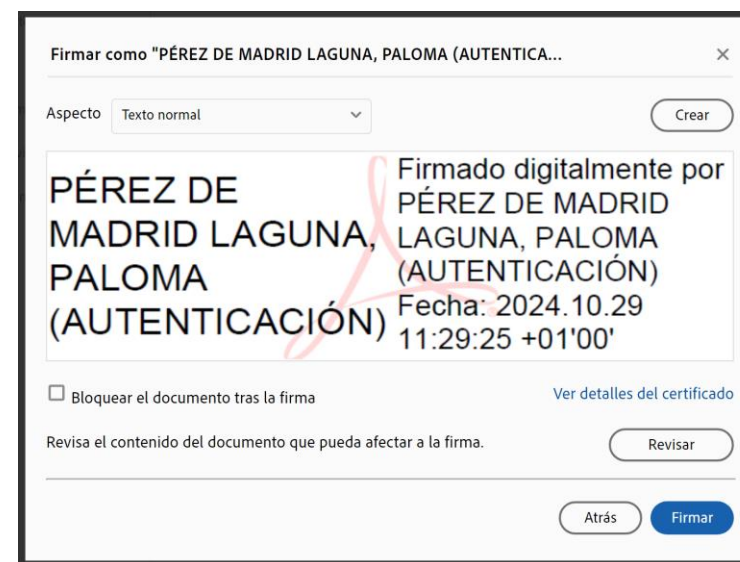
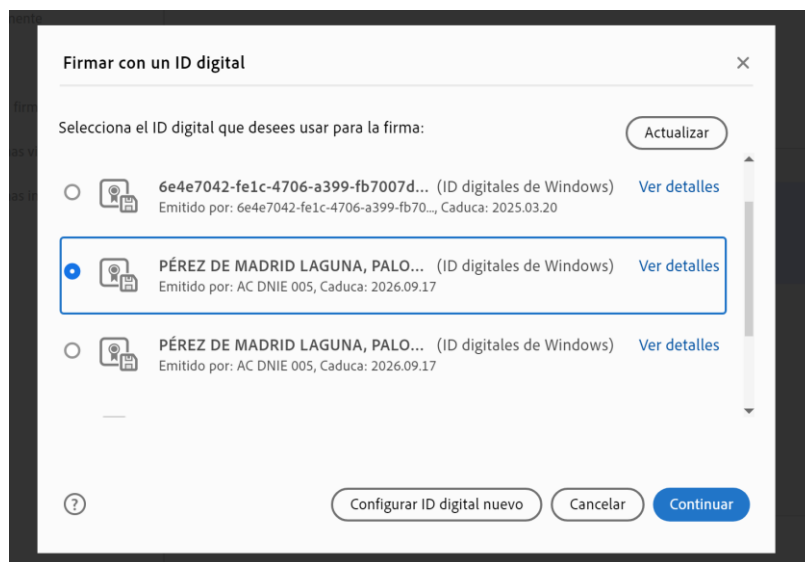
PÉREZ DE MADRID

En breve recibirá en su cuenta de correo electrónico **ppere** su CÓDIGO DE SOLICITUD. Este código y la documentación sobre su identidad le serán requeridos por la Oficina de Registro a la que se dirija para [acreditar su identidad](#) así como para la descarga de su certificado una vez que haya sido generado.

Asegúrese de que el correo electrónico asociado a su certificado es correcto, ya que a través de éste se enviarán todas las notificaciones sobre el ciclo de vida de su certificado.

# FIRMA DE UN DOCUMENTO PDF

Firmar un documento PDF. Para realizar este apartado será necesario contar con una pareja de claves pública /privada y un certificado personal a nombre del alumno. Si es posible, se realizará con el DNI electrónico (se requiere un lector de SmartCard) o con un certificado personal emitido por alguna CA reconocida (por ejemplo, la FNMT). Si no se cuenta con ninguno de estos certificados, también es posible usar un certificado a nuestro nombre emitido por nuestra propia CA (en este último caso, se deberá adjuntar en el envío el certificado raíz de esta CA). El documento PDF puede ser la propia memoria de la práctica, y tendrá la firma visible en la última página del documento.



► [Cómo firmar un Documento PDF con Certificado Digital](#)

# FIRMA DE UN DOCUMENTO PDF

Firmar un documento PDF. Para realizar este apartado será necesario contar con una pareja de claves pública /privada y un certificado personal a nombre del alumno. Si es posible, se realizará con el DNI electrónico (se requiere un lector de SmartCard) o con un certificado personal emitido por alguna CA reconocida (por ejemplo, la FNMT). Si no se cuenta con ninguno de estos certificados, también es posible usar un certificado a nuestro nombre emitido por nuestra propia CA (en este último caso, se deberá adjuntar en el envío el certificado raíz de esta CA). El documento PDF puede ser la propia memoria de la práctica, y tendrá la firma visible en la última página del documento.

Firmado con CA personal:

```
# openssl genrsa -out private/palomaines.key 2048
```

```
# openssl req -new -key private/palomaines.key -out  
certs/palomaines.csr
```

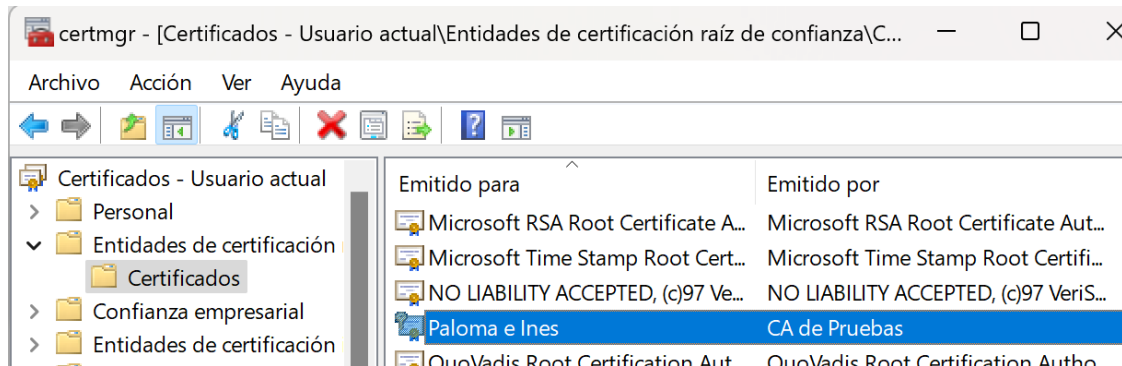
```
# openssl ca -in certs/palomaines.csr -out certs/palomaines.crt -cert  
cacert.pem -keyfile private/cakey.key -days 365
```

```
# openssl pkcs12 -export -out palomaines.p12 -inkey  
private/palomaines.key -in certs/palomaines.crt
```

```
root@server:/etc/pki/tls# openssl genrsa -out private/palomaines.key 2048
root@server:/etc/pki/tls# openssl req -new -key private/palomaines.key -out cert
s/palomaines.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.
-----
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) []:Paloma e Ines
root@server:/etc/pki/tls# openssl ca -in certs/palomaines.csr -out certs/palomai
nes.crt -cert cacert.pem -keyfile private/cakey.key -days 365
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for private/cakey.key:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 11 (0xb)
    Validity
        Not Before: Oct 28 09:41:51 2024 GMT
        Not After : Oct 28 09:41:51 2025 GMT
    Subject:
        countryName           = ES
        stateOrProvinceName   = Madrid
        organizationName      = CA de Pruebas
        commonName            = Paloma e Ines
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Subject Key Identifier:
            1E:BD:0A:BD:07:93:F8:62:8C:04:B9:BB:D7:46:B9:94:1B:7F:FA:A8
        X509v3 Authority Key Identifier:
            DirName:/C=ES/ST=Madrid/L=Alcorcon/O=CA de Pruebas/CN=CA de Prue
bas
                                serial:27:E9:6C:CB:D6:09:CE:4A:80:CC:40:C7:F3:70:8A:E6:57:99:FA:
CE
Certificate is to be certified until Oct 28 09:41:51 2025 GMT (365 days)
Sign the certificate? [y/n]:y
```

# FIRMA DE UN DOCUMENTO PDF

Firmar un documento PDF. Para realizar este apartado será necesario contar con una pareja de claves pública /privada y un certificado personal a nombre del alumno. Si es posible, se realizará con el DNI electrónico (se requiere un lector de SmartCard) o con un certificado personal emitido por alguna CA reconocida (por ejemplo, la FNMT). Si no se cuenta con ninguno de estos certificados, también es posible usar un certificado a nuestro nombre emitido por nuestra propia CA (en este último caso, se deberá adjuntar en el envío el certificado raíz de esta CA). El documento PDF puede ser la propia memoria de la práctica, y tendrá la firma visible en la última página del documento.



Autofirma:

A series of thin, light-brown lines forming an abstract geometric pattern in the top-left corner of the slide. The lines intersect to create various polygonal shapes, some of which are filled with a very light beige color.

# GRACIAS

Paloma Pérez de Madrid e Inés del Río