

# Práctica 4

## CERTIFICADOS DIGITALES



Manuel Ruiz Maldonado

## Índice

Ejercicio 1.....	3
Ejercicio 2.....	6
Ejercicio 3.....	7
Ejercicio 4.....	8
Ejercicio 5.....	9
Ejercicio 6.....	10
Conclusión.....	12
Índice de imágenes.....	14
Bibliografía.....	15

En esta práctica voy a utilizar OpenSSL para aprender a gestionar Autoridades de Certificación y certificados X509.

## Ejercicio 1

**Cread una autoridad certificadora. En este caso se premiará el uso de openssl ca frente a CA.pl, aunque este último comando es admisible.**

Crear una autoridad certificadora (CA) permitirá emitir y revocar certificados a nivel personal, ya que no es una autoridad de confianza.

Los pasos seguidos para crear una CA han sido extraídos de [1] y [2].

Lo primero que voy a hacer es crear un directorio de trabajo que permita gestionar toda la información asociada a los certificados expedidos por la autoridad certificadora.

```
$$ cd ~/Documentos/SPSI/P4/  
$$ mkdir CA  
$$ cd CA/
```

A continuación, tengo que crear la configuración inicial que tendrá la CA. Para ello voy a copiar el fichero de configuración que tiene por defecto OpenSSL y voy a modificar algunos campos. Este fichero se encuentra en `/etc/ssl/openssl.cnf`.

```
$$ cp /etc/ssl/openssl.cnf
```

Ahora abro el fichero con algún editor de texto y cambio algunos de sus atributos.

```
$$ gedit openssl.cnf &
```

```
-----  
dir                                = /home/manolo/Documentos/SPSI/P4/CA  
countryName_default               = ES  
stateOrProvinceName_default       = Spain  
0.organizationName_default         = SPSI  
-----
```

Con esto he cambiado el directorio por defecto en el que se almacenan todos los datos de la CA. Además he cambiado algunos atributos relativos a la identificación de la CA.

Analizando en profundidad el archivo `openssl.cnf` se puede observar que se hace referencia a algunos directorios y algunos ficheros que no existen actualmente en la estructura de la CA que yo estoy creando. Por ello el siguiente paso es completar dicha estructura para que el funcionamiento de la CA sea el correcto.

Para completar la estructura de directorios es necesario incluir 4 directorios dentro del directorio raíz de la CA.

```
$$ mkdir CA/private  
$$ mkdir CA/certs  
$$ mkdir CA/newcerts  
$$ mkdir CA/crl
```

“certs” será el directorio donde se guardan los certificados emitidos. “newcerts” es el lugar predeterminado en el que se almacenarán los nuevos certificados. “crl” es el directorio en el que se guarda la lista de certificados revocados. En “private” se almacenan las claves privadas utilizadas por la CA.

Para acabar de completar la configuración inicial de la CA, los ficheros necesarios que hay que incluir en su directorio raíz son los siguientes:

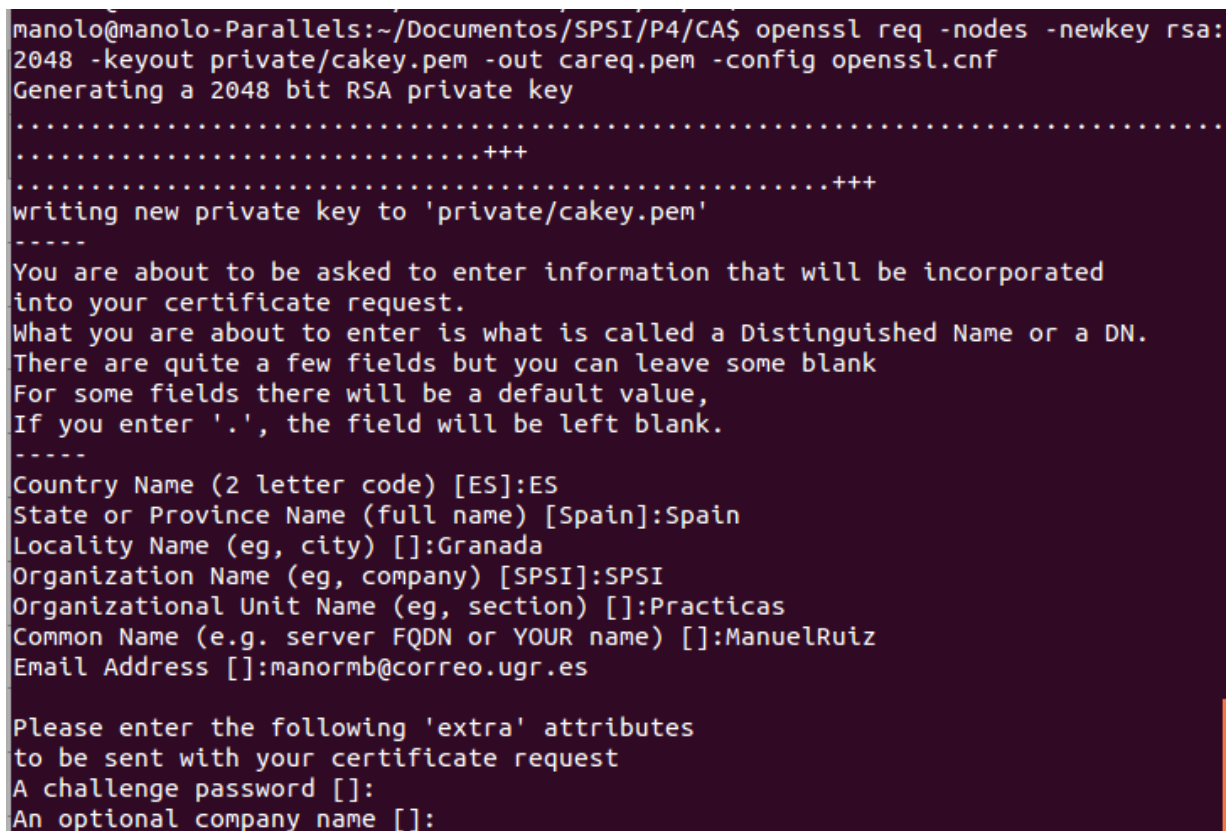
```
$$ touch index.txt
$$ echo '01' > serial
```

“index.txt” es el archivo de índice de la base de datos de los certificados de la CA. “serial” es el archivo que mantiene el número actual de la serie, es decir, el número del siguiente certificado firmado.

Finalmente, para acabar de crear la CA hay que realizar una CSR (Certificate Signing Request – petición de firma de certificado, o lo que es lo mismo, una solicitud de certificado) de la autoridad certificadora y crear el certificado autofirmado. Este proceso se divide en dos órdenes de OpenSSL.

En el primer paso voy a crear una solicitud de certificado de la CA con una clave privada generada en ese mismo momento (las claves recién generadas siempre son de RSA). La orden usada ha sido:

```
$$ openssl req -nodes -newkey rsa:2048 -keyout private/cakey.pem -out careq.pem -config openssl.cnf
```



```
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl req -nodes -newkey rsa:
2048 -keyout private/cakey.pem -out careq.pem -config openssl.cnf
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '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]:ES
State or Province Name (full name) [Spain]:Spain
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [SPSI]:SPSI
Organizational Unit Name (eg, section) []:Practicas
Common Name (e.g. server FQDN or YOUR name) []:ManuelRuiz
Email Address []:manormb@correo.ugr.es

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

*Imagen 1: Solicitud de certificado de la CA.*

El funcionamiento de esta orden se puede aprender de [3]. El comando *openssl req* permite realizar distintas tareas relacionadas con la generación y la solicitud de certificados. En este caso se utiliza para realizar una solicitud que será firmada por sí misma, dando así lugar a la creación de una CA. *-nodes* permite que si se crean claves privadas éstas no sean encriptadas. Simplemente la utilizo para ahorrar tiempo en no tener que escribir la contraseña cada vez que se utilice una clave privada. *-newkey* crea una nueva solicitud de certificado y hace que ésta se realice con las condiciones especificadas a continuación, en este caso generando una clave privada con *RSA* de 2048 bits. *-keyout* da el nombre del archivo que almacenará la clave privada recién creada. *-out* especifica el nombre del fichero de salida utilizado para escribir la solicitud realizada. *-config* permite utilizar un archivo de configuración alternativo, en este caso el que hemos creado para la CA.

Con esto tenemos como resultado una clave privada almacenada en el directorio *CA/private* y una solicitud de certificado.

En la Imagen 1 se puede ver que al realizar la solicitud se ha pedido que se rellenen algunos datos de identificación de la solicitud. En este caso los he rellenado simulando los datos de esta CA personal que estoy creando.

A continuación lo que hay que hacer es que la CA autofirme esa solicitud que acaba de emitir, creando así el certificado raíz que será el que le permitirá firmar otras solicitudes de certificados de forma que pueda tanto verificar la identidad de los solicitantes como emitir los certificados.

```
$$ openssl ca -out cacert.pem -days 3650 -keyfile private/cakey.pem -selfsign -extensions v3_ca  
-config openssl.cnf -infiles careq.pem
```

Para analizar la orden utilizada voy a hacer uso de [4]. Con el comando *openssl ca* se pueden firmar solicitudes de certificados y generar CRL. Además, este comando mantiene una base de datos de texto de los certificados emitidos y su estado. *-out* indica el archivo de salida en el que se almacenará el certificado, en formato PEM. *-days* indica el número de días que ese certificado es válido. En este caso como estoy creando la CA voy a ponerle una duración de 10 años. *-keyfile* indica el fichero que contiene la clave privada con la que firmar las solicitudes. *-selfsign* indica que los certificados emitidos se deben firmar con la clave con la que se firmaron las solicitudes de certificado. *-extensions* indica las extensiones de certificado que se agregarán al emitir el certificado. En este caso he usado *v3\_ca* porque es la misma que viene por defecto en el archivo de configuración *openssl.cnf*. *-config* especifica el archivo de configuración que hay que utilizar. *-infiles* debe ser la última opción que aparezca en la orden. Todos los argumentos que aparezcan a continuación se entenderán que son solicitudes de certificado y se procederá a su firma.

Como resultado de esta operación se puede ver que se han creado una serie de archivos en la estructura de directorios de la CA. Lo primero que se puede ver es que se ha creado un fichero *cacert.pem*, que es el certificado que sale como resultado de la petición *careq.pem*. Además, la CA almacena este mismo certificado en su directorio *newcerts* con el nombre *01.pem*, siendo 01 el valor que tenía el fichero *serial*. Además, el mismo *serial* ha cambiado su valor y ahora muestra un 02. Vemos también que *index.txt*, que antes estaba vacío, ahora contiene una línea con la información del certificado emitido recientemente.

En la Imagen 2 se puede observar el resultado de aplicar esa orden. Se muestra el resultado de la solicitud, en el que se indica la validez que tendrá el certificado que se acaba de generar. En el apartado “subject” se pueden observar los datos introducidos por el solicitante, que en este caso es el propio CA.

También se podría haber realizado esta operación en un único paso, utilizando con el comando *openssl req* la opción *-x509*, la cual genera un certificado autofirmado en lugar de una solicitud de certificado.

```
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl ca -out cacert.pem -days 3650 -keyfile private/cakey.pem -selfsign -extensions v3_ca -config openssl.cnf -infiles careq.pem
Using configuration from openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: Dec  9 10:09:23 2017 GMT
    Not After : Dec  7 10:09:23 2027 GMT
  Subject:
    countryName           = ES
    stateOrProvinceName   = Spain
    organizationName       = SPSI
    organizationalUnitName = Practicas
    commonName             = ManuelRuiz
    emailAddress           = manormb@correo.ugr.es
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      17:D8:AE:A9:BB:55:85:4D:6C:97:A7:ED:C1:FF:69:EB:7D:D3:AB:76
    X509v3 Authority Key Identifier:
      keyid:17:D8:AE:A9:BB:55:85:4D:6C:97:A7:ED:C1:FF:69:EB:7D:D3:AB:76
6
    X509v3 Basic Constraints:
      CA:TRUE
Certificate is to be certified until Dec  7 10:09:23 2027 GMT (3650 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
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$
```

Imagen 2: Creación de certificado para la solicitud de certificado de la CA.

Y así, con todo este proceso se completa la creación de la autoridad certificadora.

## Ejercicio 2

**Cread una solicitud de certificado que incluya la generación de claves en la misma.**

Para realizar este ejercicio me voy a basar en los conocimientos adquiridos durante el ejercicio anterior. Para realizar una solicitud de certificado hay que utilizar el comando *openssl req*. Si quiero que la generación de claves esté en la misma orden utilizo, como antes, la opción *-newkey*. La clave generada la guardo en el directorio *private* e indico que el fichero de salida que contiene la solicitud sea *ejercicio2req.pem*.

```
$$ openssl req -nodes -newkey rsa:2048 -keyout private/ejercicio2key.pem -out ejercicio2req.pem -config openssl.cnf
```



```

manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl req -nodes -newkey rsa:
2048 -keyout private/ejercicio2key.pem -out ejercicio2req.pem -config openssl.cn
f
Generating a 2048 bit RSA private key
.+++
.....+++
writing new private key to 'private/ejercicio2key.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]:ES
State or Province Name (full name) [Spain]:Spain
Locality Name (eg, city) []:
Organization Name (eg, company) [SPSI]:SPSI
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Ejercicio 2
Email Address []:

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

```

*Imagen 3: Solicitud de certificado del Ejercicio 2.*

En la Imagen 3 se puede observar la solicitud del certificado emitida en este ejercicio. Para poder identificar esta solicitud al final de la práctica he indicado que el nombre del solicitante (Common Name) en este caso es “Ejercicio 2”.

## Ejercicio 3

**Cread un certificado para la solicitud anterior empleando la CA creada en el primer punto.**

Para crear el certificado a partir de una solicitud ahora hay que utilizar el comando *openssl ca*, y firmar esta vez teniendo en cuenta algunos cambios. Para empezar, la clave que hay que utilizar para firmar no es la clave del emisor de la solicitud, sino la de la CA. Así estaremos simulando que una autoridad certificadora es la encargada de emitir un certificado para un usuario solicitante. Además, la extensión en este caso cambia su valor a *v3\_req* porque para las solicitudes, en el archivo de configuración indica que ese debe ser el formato.

```

$$ openssl ca -config openssl.cnf -days 365 -keyfile private/cakey.pem -extensions v3_req -in
ejercicio2req.pem -out ejercicio3cert.pem

```

Así, para la solicitud de certificado almacenada en el fichero *ejercicio2req.pem* se ha generado el certificado asociado *ejercicio3cert.pem*, emitido y firmado por la CA creada en el ejercicio 1, cuya clave está en *private/cakey.pem*.

En la Imagen 4 se puede observar el resultado de la respuesta a la solicitud de certificado que se hizo en el ejercicio anterior. En este caso, a diferencia de la Imagen 2, el solicitante del certificado es “Ejercicio 2”.

```

manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl ca -config openssl.cnf
-days 365 -keyfile private/cakey.pem -extensions v3_req -in ejercicio2req.pem -
out ejercicio3cert.pem
Using configuration from openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 2 (0x2)
    Validity
        Not Before: Dec  9 10:43:00 2017 GMT
        Not After : Dec  9 10:43:00 2018 GMT
    Subject:
        countryName             = ES
        stateOrProvinceName     = Spain
        organizationName        = SPSI
        commonName              = Ejercicio 2
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Digital Signature, Non Repudiation, Key Encipherment
Certificate is to be certified until Dec  9 10:43:00 2018 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
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$

```

*Imagen 4: Creación de certificado para la solicitud de certificado del Ejercicio 2.*

## Ejercicio 4

**Cread una solicitud de certificado para cualquiera de las claves que habéis generado en las prácticas anteriores, excepto las RSA.**

En este ejercicio se repite el comando utilizado en el Ejercicio 2. La única diferencia es que no se genera la clave en la misma orden. Por ello se eliminan todas las opciones relativas a la generación de claves (*-nodes*, *-newkey* y *-keyout*) y se utiliza la opción *-key* cuyo argumento es un fichero con la clave que se quiere utilizar para generar la solicitud. Además, antes no se usaba la opción *-new* que indica que se crea una nueva solicitud ya que estaba implícito en la opción *-newkey*, pero ahora sí tengo que añadirla.

```

$$ openssl req -new -key private/ruizECpriv.pem -out ejercicio4req.pem -config openssl.cnf

```

El resultado de esta operación es igual que el del Ejercicio 2 y se puede observar en la Imagen 5. En este caso he vuelto a cambiar el nombre del solicitante para poder identificarlo más tarde.



```

manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl req -new -key private/r
uizECpriv.pem -out ejercicio4req.pem -config openssl.cnf
Enter pass phrase for private/ruizECpriv.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]:ES
State or Province Name (full name) [Spain]:Spain
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [SPSI]:SPSI
Organizational Unit Name (eg, section) []:Practica 4
Common Name (e.g. server FQDN or YOUR name) []:Ruiz-ECpriv
Email Address []:

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

```

*Imagen 5: Solicitud de certificado del Ejercicio 4.*

## Ejercicio 5

**Cread un certificado para la solicitud anterior utilizando la CA creada.**

Este ejercicio es calcado al Ejercicio 3. No existe ninguna diferencia a la hora de crear un certificado cuya solicitud esté generada por un tipo u otro de clave.

```

$$ openssl ca -config openssl.cnf -days 365 -keyfile private/cakey.pem -extensions v3_req -in
ejercicio4req.pem -out ejercicio5cert.pem

```

```

manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl ca -config openssl.cnf
-days 365 -keyfile private/cakey.pem -extensions v3_req -in ejercicio4req.pem -
out ejercicio5cert.pem
Using configuration from openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 3 (0x3)
    Validity
        Not Before: Dec 11 09:58:26 2017 GMT
        Not After : Dec 11 09:58:26 2018 GMT
    Subject:
        countryName           = ES
        stateOrProvinceName   = Spain
        organizationName      = SPSI
        organizationalUnitName = Practica 4
        commonName            = Ruiz-ECpriv
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Digital Signature, Non Repudiation, Key Encipherment
Certificate is to be certified until Dec 11 09:58:26 2018 GMT (365 days)

```

*Imagen 6: Creación de certificado para la solicitud de certificado del Ejercicio 4.*

## Ejercicio 6

Emplead las opciones `-text` y `-noout` para mostrar los valores de todos los certificados y solicitudes de los puntos anteriores, incluyendo el certificado raíz que habrá sido creado junto con la CA.

Para visualizar los certificados voy a utilizar el comando `openssl x509` cuya documentación está en [5]. La opción `-noout` evita la salida codificada de la solicitud. Con `-text` se muestra el certificado en formato de texto. Con esta opción se muestran todos los detalles que contiene el certificado, incluyendo la clave pública, el algoritmo de firma utilizado, los nombres tanto del solicitante como de la autoridad que ha generado el certificado y otros datos complementarios. Con `-in` se indica el nombre del certificado que se quiere mostrar.

```
$$ openssl x509 -noout -text -in cacert.pem
$$ openssl x509 -noout -text -in ejercicio3cert.pem
$$ openssl x509 -noout -text -in ejercicio5cert.pem
```

En la Imagen 7 se puede ver el certificado autofirmado por la CA. Tanto el Issuer como el Subject contienen los datos que le di a la CA, lo cual indica que el solicitante y el firmante son la misma entidad. Se puede ver que se ha utilizado un algoritmo RSA de encriptación con una clave de 2048 bits. En la Imagen 8 se puede ver que el algoritmo de firma usado ha sido SHA256 con encriptación en RSA.

```
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl x509 -noout -text -in cacert.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=ES, ST=Spain, O=SPSI, OU=Practicas, CN=ManuelRuiz/emailAddress=manormb@correo.ugr.es
    Validity
      Not Before: Dec  9 10:09:23 2017 GMT
      Not After : Dec  7 10:09:23 2027 GMT
    Subject: C=ES, ST=Spain, O=SPSI, OU=Practicas, CN=ManuelRuiz/emailAddress=manormb@correo.ugr.es
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:e8:b0:83:4d:ef:9f:ff:a5:b5:97:b3:ed:a7:be:
        1d:80:9a:54:26:93:9b:b1:64:3c:46:2c:a9:cf:56:
        d1:33:24:dc:59:58:01:43:e7:d4:1b:2c:66:8a:69:
        71:f8:a3:5e:f0:c6:89:51:30:de:29:35:5e:95:76:
        ea:10:91:21:39:da:10:7e:a4:b4:ca:14:cb:28:3c:
        00:03:0d:a7:6a:dd:bd:9c:c3:a3:0e:40:70:7e:27:
        14:81:50:b8:8d:bf:64:aa:47:56:b8:da:56:61:23:
        b9:1e:cf:65:91:d7:a3:05:ab:a6:5d:6f:43:d7:7d:
        5c:38:d8:3e:c5:da:a6:e1:c7:7e:98:88:a2:c5:72:
        eb:87:18:51:92:9d:4b:ae:9f:9d:04:10:d8:5f:27:
        0b:2d:0e:26:fc:4b:8b:d3:af:41:0e:ad:e9:e7:02:
        2c:e0:81:79:ed:74:ca:dd:7f:4f:57:72:c1:aa:53:
        b0:87:5b:aa:51:83:41:fa:03:08:ec:5e:f8:c7:a0:
        de:38:5d:c1:5c:09:18:9a:26:6d:bc:c2:4c:d3:39:
        54:66:dd:cf:9d:5d:48:2c:ad:e2:a7:b3:d0:0b:9d:
        82:1d:4d:08:fa:3c:bf:93:47:fb:46:72:a5:d4:46:
        c9:88:0e:6f:6d:6d:95:51:0e:e6:99:18:07:b4:b7:
        cf:e5
      Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      17:D8:AE:A9:BB:55:85:4D:6C:97:A7:ED:C1:FF:69:EB:7D:D3:AB:76
    X509v3 Authority Key Identifier:
      keyid:17:D8:AE:A9:BB:55:85:4D:6C:97:A7:ED:C1:FF:69:EB:7D:D3:AB:76
```

Imagen 7: Certificado de la CA, parte 1.

```

X509v3 Basic Constraints:
    CA:TRUE
Signature Algorithm: sha256WithRSAEncryption
40:50:56:98:34:53:50:5a:e2:af:24:be:7c:9a:ea:05:06:1b:
b3:a5:97:45:81:d2:6b:63:77:11:e0:43:f5:09:39:75:00:71:
79:62:0a:a2:8e:93:53:12:4a:2d:59:77:7d:ff:dd:c5:bb:c3:
f2:c9:c5:c9:ce:46:43:12:e3:3e:eb:50:00:9f:dc:84:b8:51:
5b:6e:f7:b6:b8:64:ec:73:32:13:53:5a:0b:e0:d5:8c:b5:85:
b8:a8:99:75:af:8a:ea:1c:58:36:58:03:5b:50:a5:73:aa:3e:
70:d7:8e:2f:c8:94:04:3e:81:b3:81:c7:3c:22:88:c7:99:dc:
d8:b8:a1:df:1e:60:33:11:e6:e0:77:81:67:a6:8a:a7:88:43:
a4:b6:72:b8:7f:32:be:e8:fe:df:b1:22:31:64:b7:ad:62:57:
51:15:3b:69:e5:f5:a6:af:14:88:a3:c1:a5:49:3c:65:af:c4:
6f:0b:a6:65:1f:75:20:f8:e1:6e:88:88:ca:18:79:18:ac:2d:
29:de:19:be:67:8e:9c:a2:f7:bc:64:6c:ba:fe:d7:b4:c4:db:
85:64:77:4f:70:34:a2:d5:55:bf:bd:9c:4e:20:29:ed:3e:e8:
d6:99:c1:4f:f1:f2:57:7e:d5:1a:95:66:91:ea:67:20:bb:02:
d8:9b:bb:13
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CAS$

```

*Imagen 8: Certificado de la CA, parte 2.*

```

manolo@manolo-Parallels:~/Documentos/SPSI/P4/CAS$ openssl x509 -noout -text -in ejercicio3cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=ES, ST=Spain, O=SPSI, OU=Practicas, CN=ManuelRuiz/emailAddress=manormb@correo.ugr.es
    Validity
        Not Before: Dec  9 10:43:00 2017 GMT
        Not After : Dec  9 10:43:00 2018 GMT
    Subject: C=ES, ST=Spain, O=SPSI, CN=Ejercicio 2
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:
            00:d4:cc:aa:fc:dd:75:8a:9e:be:37:cf:31:1e:f0:
            64:e1:56:9d:3c:8c:d6:4c:fd:20:22:7c:c4:b5:8b:
            39:0d:33:29:b7:14:58:83:9c:25:7e:70:e9:94:ec:
            bc:b1:83:24:86:5a:41:b6:3c:dd:12:9c:f5:6a:b8:
            43:71:19:d1:f8:24:63:8f:e2:05:6f:21:1a:7c:43:
            5e:5c:ae:ed:78:98:df:89:c9:ad:06:d0:4b:f9:5d:
            fc:b9:92:b4:60:af:1f:fb:f1:0a:e8:5d:7b:81:e5:
            01:16:38:97:1b:2c:01:99:7d:00:21:25:ba:87:69:
            f4:ed:75:27:c7:f5:56:b8:e3:9f:ff:2f:3e:8d:0b:
            8a:f2:33:21:13:e4:a5:32:86:a6:b3:f6:75:0a:3e:
            71:6d:4e:43:4c:bf:f2:b2:eb:ac:c2:00:54:21:af:
            5d:f5:94:40:e1:42:89:02:9f:87:e3:c8:17:fe:e6:
            3a:e3:25:58:88:73:d6:35:a5:74:7c:f1:d3:07:e0:
            de:94:ec:02:7e:1a:d9:54:f6:c1:94:06:0d:54:6d:
            54:c4:ea:94:4e:f6:f5:ef:ed:28:be:62:92:c3:0d:
            4b:5e:e3:68:0d:62:c2:7c:e5:fc:3a:e3:5f:97:8e:
            db:28:90:2b:69:e5:76:4c:0b:c8:00:66:ac:d1:17:
            13:39
        Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha256WithRSAEncryption
    e5:ba:76:76:ea:6e:c0:f4:04:11:8e:34:f9:ec:75:ca:67:f2:

```

*Imagen 9: Certificado del Ejercicio 3.*

En la Imagen 9 tenemos el certificado generado a la solicitud realizada en el Ejercicio 2. Se puede ver que el Issuer es justamente la autoridad creada en el Ejercicio 1, mientras que el solicitante, Subject, es justamente el “Ejercicio 2”. En este caso también tenemos un algoritmo RSA con clave

de 2048 bits ya que fue el generado directamente con una única orden. Además, se puede ver que en diferencia con la Imagen 8, aquí en el campo *Basic Constraints* pone *CA:FALSE*, lo cual indica que este certificado no corresponde a una autoridad certificadora. En cambio, en la Imagen 8 ponía *CA:TRUE* en ese mismo campo porque al estar autofirmado, se entiende que en ese caso sí corresponde al certificado de una CA.

```
manolo@manolo-Parallels:~/Documentos/SPSI/P4/CA$ openssl x509 -noout -text -in ejercicio5cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Spain, O=SPSI, OU=Practicas, CN=ManuelRuiz/emailAddress=manormb@correo.ugr.es
        Validity
            Not Before: Dec 11 09:58:26 2017 GMT
            Not After : Dec 11 09:58:26 2018 GMT
        Subject: C=ES, ST=Spain, O=SPSI, OU=Practica 4, CN=Ruiz-ECpriv
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
            Public-Key: (192 bit)
            pub:
                04:e8:50:9f:02:d6:8c:0f:56:fd:ae:17:dd:ac:14:
                c5:65:27:fd:9e:8a:c6:47:98:af:cf:4d:f4:1b:26:
                6a:f6:10:b6:b2:90:c4:7f:c2:74:dc:d2:b8:5d:6f:
                b0:bf:a4:83
            ASN1 OID: prime192v1
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
        Signature Algorithm: sha256WithRSAEncryption
        d3:1f:e2:b2:d3:b5:ce:ba:7c:a9:3f:9f:bb:ff:99:86:bf:04:
        ff:92:ec:d0:cb:52:1b:c8:d6:9d:bc:67:1e:ba:af:46:3a:e4:
        95:f1:c9:c5:37:da:f0:b9:a5:57:60:cd:d3:3a:2b:6e:75:4a:
        f7:98:e1:86:46:30:d4:cf:5a:ac:d7:5b:32:b5:2f:6b:22:ca:
        02:1a:60:60:4e:59:d4:02:60:2f:49:42:4c:ac:39:f6:5e:10:
        03:a3:5a:7a:cf:f0:c7:91:d1:26:d2:01:bd:29:c1:ca:b4:01:
        e9:7f:30:47:19:0d:f5:78:cc:c2:c3:93:a6:77:2a:32:ec:46:
        36:bb:84:85:82:e6:c4:90:46:9c:55:e3:3d:05:57:cb:37:6c:
        b0:56:68:b2:13:87:29:bd:17:71:69:78:20:16:41:57:47:b7:
        51:6a:80:fb:1c:7a:95:06:e8:bd:05:58:e5:1e:99:7f:25:28:
        db:6c:99:cd:39:9a:55:17:a0:12:08:a8:46:79:d7:d6:9d:de:
        95:7d:d2:3f:0c:3b:66:a2:c5:ed:36:4a:69:b9:4d:ef:aa:d0:
        53:a4:ac:33:6b:d6:88:43:b0:d8:19:4c:6f:d9:1a:b3:47:b1:
        a5:af:91:89:8e:0f:e0:78:68:18:c7:f7:4c:b8:d3:67:20:1f:
        68:ac:39:02
```

*Imagen 10: Certificado del Ejercicio 5.*

Finalmente, en esta Imagen 10 se puede ver que el solicitante fue el del Ejercicio 4, Ruiz-Ecpriv, que utilizaba una clave distinta. De nuevo la autoridad vuelve a ser la creada en el Ejercicio 1. Cabe destacar la diferencia de que en este caso no se ha utilizado un algoritmo RSA, sino un algoritmo de curvas elípticas, cuya clave tenía 192 bits (de la práctica anterior).

## Conclusión

Con esta práctica he aprendido el funcionamiento de solicitud y creación de certificados. Además, es importante tener en cuenta quién es la autoridad emisora del certificado así como el periodo de validez, ya que si se sobrepasa ese periodo, ese certificado no tendrá validez. En cambio, si el que sobrepasa el tiempo límite es el propio certificado de la CA, todos los certificados generados por ella mismo serán los que pierden la validez también.

También es importante saber crear una autoridad certificadora, ya que aunque no tenga

reconocimiento oficial ninguno, se puede utilizar para realizar comunicaciones de forma segura y dar validez a transacciones que realicemos en los que nosotros podamos actuar como autoridad.

## Índice de imágenes

Imagen 1: Solicitud de certificado de la CA.....	4
Imagen 2: Creación de certificado para la solicitud de certificado de la CA.....	6
Imagen 3: Solicitud de certificado del Ejercicio 2.....	7
Imagen 4: Creación de certificado para la solicitud de certificado del Ejercicio 2.....	8
Imagen 5: Solicitud de certificado del Ejercicio 4.....	9
Imagen 6: Creación de certificado para la solicitud de certificado del Ejercicio 4.....	9
Imagen 7: Certificado de la CA, parte 1.....	10
Imagen 8: Certificado de la CA, parte 2.....	11
Imagen 9: Certificado del Ejercicio 3.....	11
Imagen 10: Certificado del Ejercicio 5.....	12



## Bibliografía

[1] Tutorial para crear una CA.

<https://www.linuxito.com/gnu-linux/nivel-alto/26-como-crear-tu-propia-autoridad-certificante-ca>

[2] Otro tutorial para crear una CA.

<http://icewinddale.blogspot.com.es/2015/08/openssl-es-un-conjunto-de-utilidades-y.html>

[3] Documentación de OpenSSL para generación y solicitud de certificados.

<https://www.openssl.org/docs/man1.0.2/apps/req.html>

[4] Documentación de OpenSSL para gestionar CA.

<https://www.openssl.org/docs/man1.0.2/apps/ca.html>

[5] Documentación de OpenSSL para la visualización y firma de certificados.

<https://www.openssl.org/docs/manmaster/man1/x509.html>