

Universidad Nacional Autónoma de México
Facultad de Ciencias

Asignatura: Redes de computadoras
Semestre: 2024-1

Profesor: Javier León Cotonieto

Ayudantes: Magdalena Reyes Granados
Itzel Gómez Muñoz
Sandra Plata Velázquez

Práctica 12. Certificados Digitales

Equipo 5

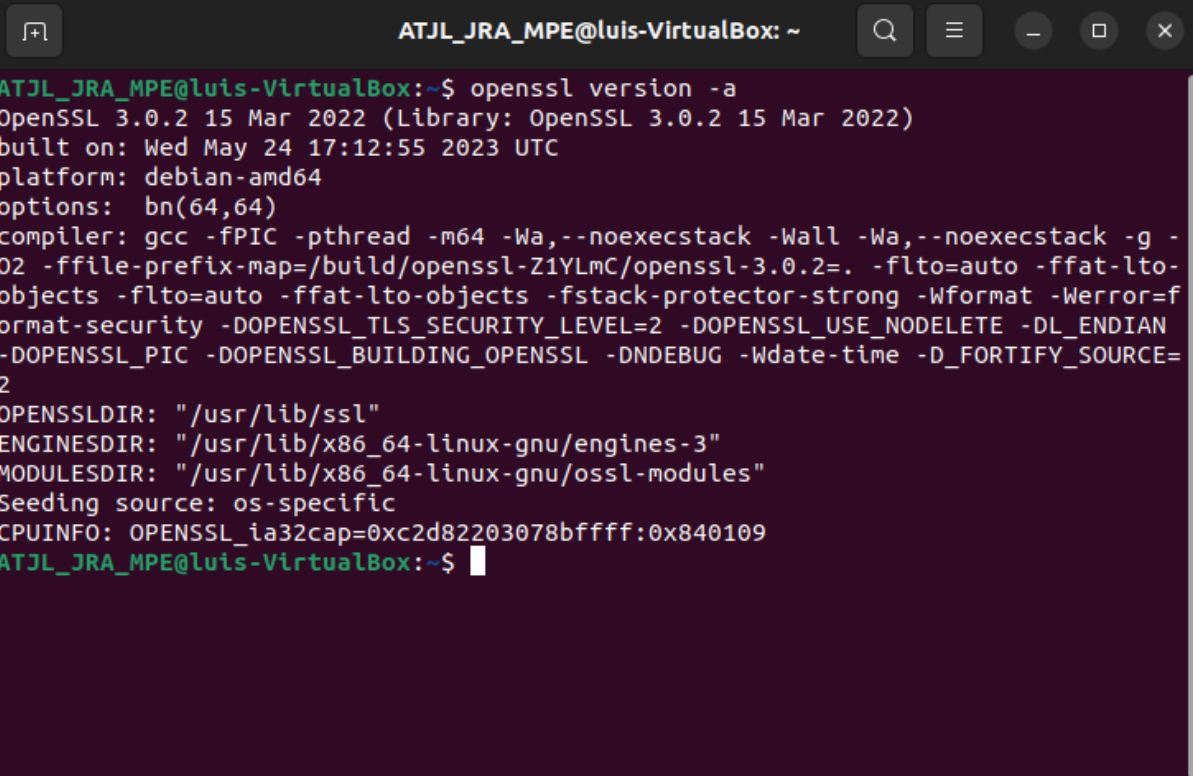
Integrantes:

- **Almanza Torres José Luis**
- **Jimenez Reyes Abraham**
- **Martínez Pardo Esaú**



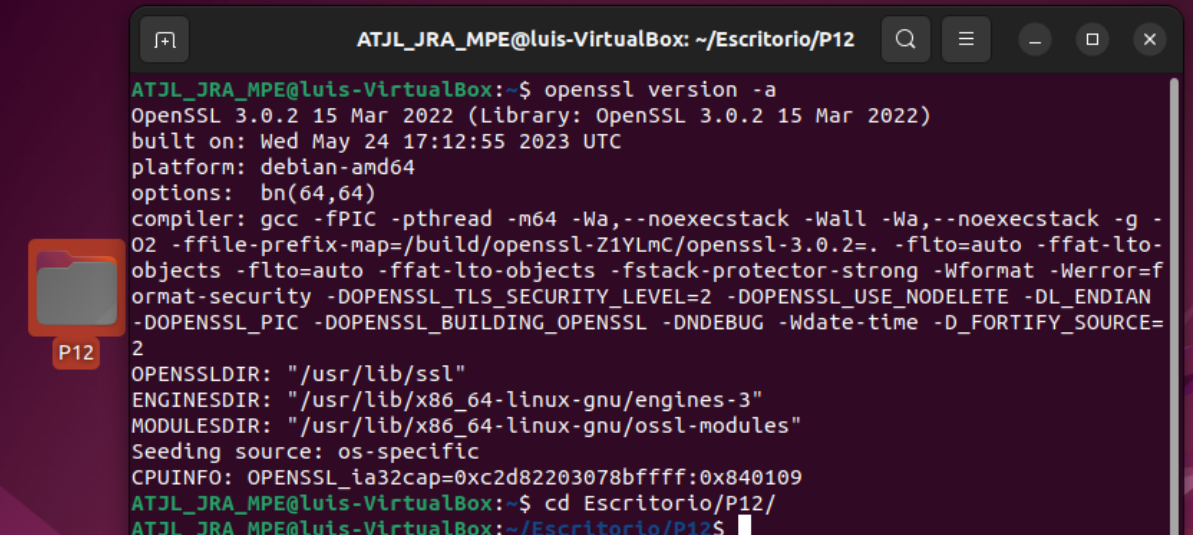
Crear un certificado digital.

1. Instale OpenSSL



```
ATJL_JRA_MPE@luis-VirtualBox: ~  
ATJL_JRA_MPE@luis-VirtualBox:~$ openssl version -a  
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)  
built on: Wed May 24 17:12:55 2023 UTC  
platform: debian-amd64  
options: bn(64,64)  
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -  
O2 -ffile-prefix-map=/build/openssl-Z1YLMC/openssl-3.0.2=. -flto=auto -ffat-lto-  
objects -flto=auto -ffat-lto-objects -fstack-protector-strong -Wformat -Werror=f  
ormat-security -DOPENSSL_TLS_SECURITY_LEVEL=2 -DOPENSSL_USE_NODELETE -DL_ENDIAN  
-DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=  
2  
OPENSSLDIR: "/usr/lib/ssl"  
ENGINESDIR: "/usr/lib/x86_64-linux-gnu/engines-3"  
MODULESDIR: "/usr/lib/x86_64-linux-gnu/openssl-modules"  
Seeding source: os-specific  
CPUINFO: OPENSSL_ia32cap=0xc2d82203078bffff:0x840109  
ATJL_JRA_MPE@luis-VirtualBox:~$
```

2. Cree una carpeta para guardar el programa OpenSSL que será la carpeta de trabajo y acceda a ella.

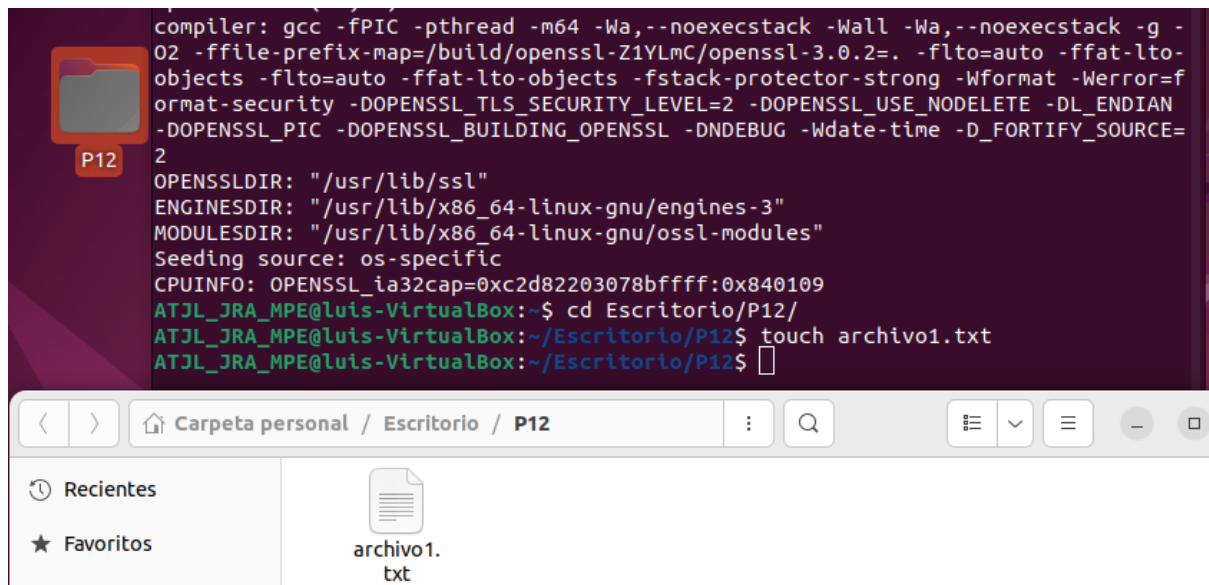


```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12  
ATJL_JRA_MPE@luis-VirtualBox:~$ openssl version -a  
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)  
built on: Wed May 24 17:12:55 2023 UTC  
platform: debian-amd64  
options: bn(64,64)  
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -  
O2 -ffile-prefix-map=/build/openssl-Z1YLMC/openssl-3.0.2=. -flto=auto -ffat-lto-  
objects -flto=auto -ffat-lto-objects -fstack-protector-strong -Wformat -Werror=f  
ormat-security -DOPENSSL_TLS_SECURITY_LEVEL=2 -DOPENSSL_USE_NODELETE -DL_ENDIAN  
-DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=  
2  
OPENSSLDIR: "/usr/lib/ssl"  
ENGINESDIR: "/usr/lib/x86_64-linux-gnu/engines-3"  
MODULESDIR: "/usr/lib/x86_64-linux-gnu/openssl-modules"  
Seeding source: os-specific  
CPUINFO: OPENSSL_ia32cap=0xc2d82203078bffff:0x840109  
ATJL_JRA_MPE@luis-VirtualBox:~$ cd Escritorio/P12/  
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

3. Compruebe que se instaló correctamente OpenSSL.

En la captura del punto 1 se puede visualizar que se instaló correctamente.

4. Cree un archivo de texto sin contenido.

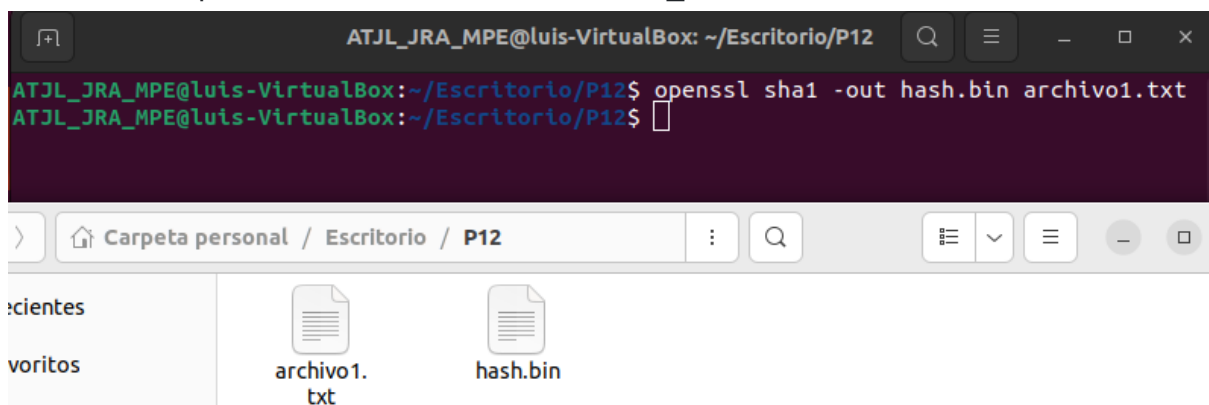


The screenshot shows a terminal window with the following commands and output:

```
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -O2 -ffile-prefix-map=/build/openssl-Z1YLMC/openssl-3.0.2=. -flto=auto -ffat-lto-objects -flto=auto -ffat-lto-objects -fstack-protector-strong -Wformat -Werror=format-security -DOPENSSL_TLS_SECURITY_LEVEL=2 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=2
OPENSSLDIR: "/usr/lib/ssl"
ENGINESDIR: "/usr/lib/x86_64-linux-gnu/engines-3"
MODULESDIR: "/usr/lib/x86_64-linux-gnu/openssl-modules"
Seeding source: os-specific
CPUINFO: OPENSSL_ia32cap=0xc2d82203078bffff:0x840109
ATJL_JRA_MPE@luis-VirtualBox:~$ cd Escritorio/P12/
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ touch archivo1.txt
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

Below the terminal, a file manager window shows the directory 'Carpeta personal / Escritorio / P12'. It contains a file named 'archivo1.txt'.

5. Aplique una función hash (md5, sha1 o sha265) al archivo anterior y guárdelo en un archivo *.bin openssl sha1 -out hash.bin nombre_archivo.txt

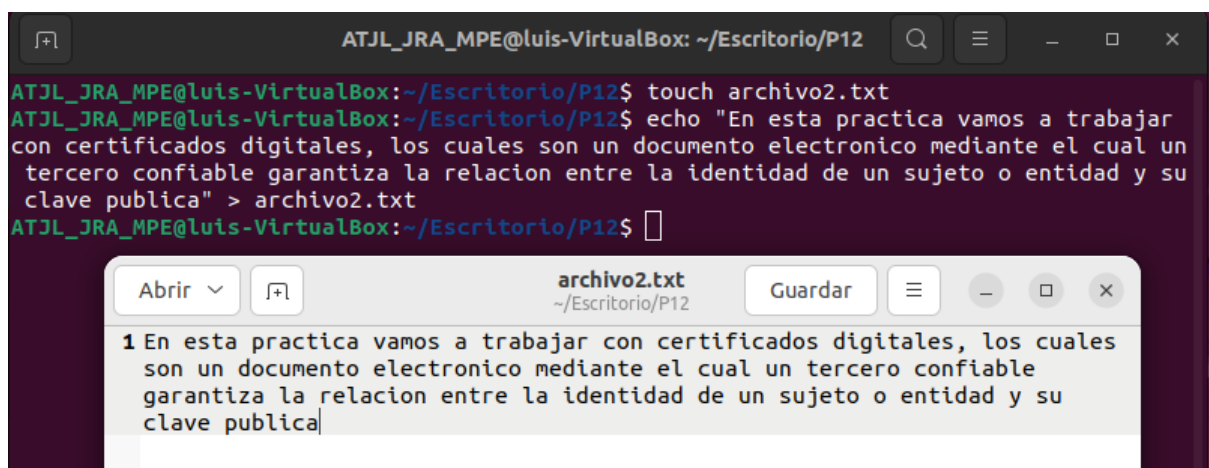


The screenshot shows a terminal window with the following commands and output:

```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl sha1 -out hash.bin archivo1.txt
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

Below the terminal, a file manager window shows the directory 'Carpeta personal / Escritorio / P12'. It contains two files: 'archivo1.txt' and 'hash.bin'.

6. Cree un nuevo archivo de texto con el nombre "archivo2.txt". En esta ocasión que sí tenga contenido.



The screenshot shows a terminal window with the following commands and output:

```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ touch archivo2.txt
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ echo "En esta practica vamos a trabajar con certificados digitales, los cuales son un documento electronico mediante el cual un tercero confiable garantiza la relacion entre la identidad de un sujeto o entidad y su clave publica" > archivo2.txt
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

Below the terminal, a text editor window shows the file 'archivo2.txt' with the following content:

```
1 En esta practica vamos a trabajar con certificados digitales, los cuales son un documento electronico mediante el cual un tercero confiable garantiza la relacion entre la identidad de un sujeto o entidad y su clave publica
```

7. Ingrese el siguiente comando y explique la función de cada parámetro.

`openssl enc -des3 -pbkdf2 -in archivo2.txt -out cifra_a.bin -pass pass:12345`

openssl enc: Permiten cifrar o descifrar datos utilizando varios cifrados de bloque y flujo utilizando claves basadas en contraseñas o proporcionadas explícitamente.

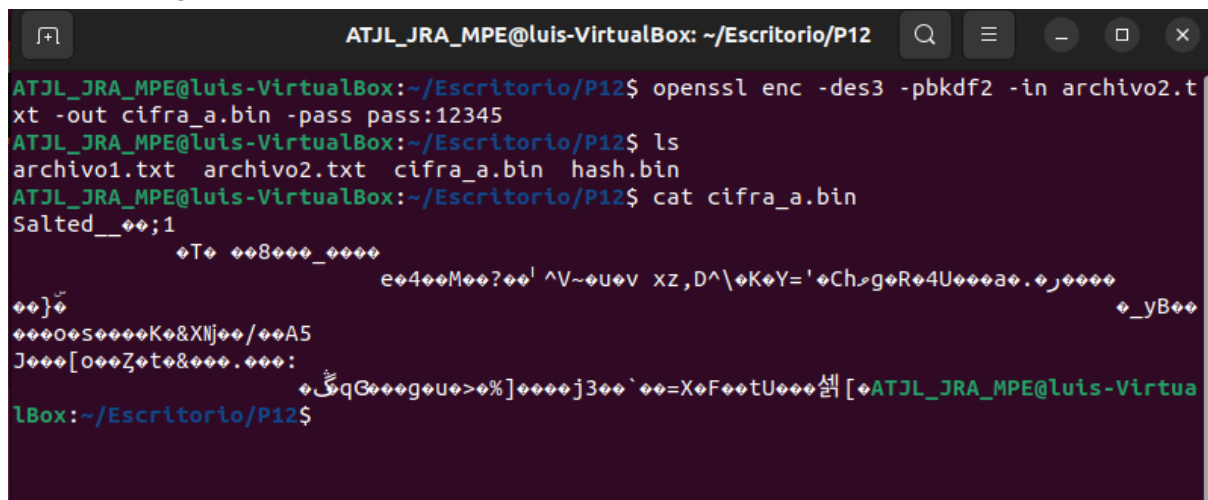
-des3: Es un tipo de cifrado, el cual habilita la contraseña para la clave privada. Este es un parámetro opcional. También puede habilitar la contraseña para una clave privada existente (como el parámetro -in).

-pbkdf2: Indica que se usa el algoritmo PBKDF2 con una cuenta predeterminada de iteraciones de 10000, a menos que se especifique lo contrario en la opción de línea de comandos con el parámetro -iter.

-in archivo2.txt: El nombre del archivo de entrada, entrada estándar por defecto. En este caso, el archivo es archivo2.txt

-out cifra_a.bin: El nombre del archivo de salida, salida estándar por defecto(cifra_a.bin)

-pass pass:12345: La fuente de contraseña. Puede o no tomar argumentos, en este caso se toma el argumento pass que es la contraseña dada



```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl enc -des3 -pbkdf2 -in archivo2.txt -out cifra_a.bin -pass pass:12345
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ ls
archivo1.txt  archivo2.txt  cifra_a.bin  hash.bin
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ cat cifra_a.bin
Salted__
      T 8_
      e4M? ^V~u v xz,D^\K Y='Ch_gR4Uaa. j
}
osK&Xl j/A5
]ooZt&. :
qGgou>%]j3` =XFeetU 섹 [ATJL_JRA_MPE@luis-Virtua
lBox:~/Escritorio/P12$
```

8. Ingrese el siguiente comando y explique la función de cada parámetro.

`openssl enc -des-ede3-cbc -pbkdf2 -in cifra_a.bin -out descifrado.txt`

open enc -cipher: Permite cifrar o descifrar datos utilizando varios cifrados de bloque y flujo utilizando claves basadas en contraseñas o proporcionadas explícitamente.

-des-ede3-cbc: Triple DES con tres claves, usado en modo CBC , con relleno no especificado.

-pbkdf2: Indica que se usa el algoritmo PBKDF2 con una cuenta predeterminada de iteraciones de 10000, a menos que se especifique lo contrario en la opción de línea de comandos con el parámetro -iter.

-in cifra_a.bin: El nombre del archivo de entrada, entrada estándar por defecto(cifra_a.bin)

-out descifrado.txt: El nombre del archivo de salida, salida estándar por defecto(descifrado.txt).

```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl enc -des-ede3-cbc -pbkdf2 -in cifra_a.bin -out descifrado.txt
enter DES-EDE3-CBC encryption password:
Verifying - enter DES-EDE3-CBC encryption password:
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ ls
archivo1.txt  archivo2.txt  cifra_a.bin  descifrado.txt  hash.bin
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

Crear una Autoridad Certificadora.

9. Se creará una AC para certificados X.509 utilizando el algoritmo de cifrado RSA de 2048 bytes, almacenando llaves públicas y privadas en diferentes archivos.

9.1. Investigue las características del certificado X.509.

Uso de la Clave Pública: Indica para qué propósito puede utilizarse la clave pública del sujeto, como firma digital, cifrado, etc.

Extensiones: Los certificados X.509 pueden contener extensiones que proporcionan información adicional.

El certificado contiene información sobre la entidad a la que se le ha emitido el certificado. Esto puede incluir el nombre distinguido (DN) del sujeto, que generalmente incluye información como el nombre común (CN), la organización (O), la unidad organizativa (OU), la localidad (L), el estado o provincia (ST), el país (C), y otros atributos.

Número de Serie: Cada certificado X.509 tiene un número de serie único que lo identifica de manera exclusiva.

Validez: El certificado indica el período de tiempo durante el cual es válido. Esto se especifica mediante dos fechas: la fecha de inicio (Not Before) y la fecha de Firma Digital: La firma digital es un valor criptográfico que se genera aplicando un algoritmo de firma digital a los datos del certificado (Not After).

9.2. **Ingrese el siguiente comando y explique cada uno de los parámetros.**

openssl req: Crea y procesa principalmente solicitudes de certificados en formato PKCS#10 y crea certificados autofirmados para usarlos.

-x509: Genera un certificado autofirmado en lugar de una solicitud de certificado. Normalmente se utiliza para generar un certificado de prueba o una CA raíz autofirmada. Las extensiones agregadas al certificado (si las hay) se especifican en el archivo de configuración.

-newkey rsa: 2048: Crea una nueva solicitud de certificado y una nueva clave privada. El argumento adopta una de varias formas. `rsa:nbits`, donde `nbits` es el número de bits, genera una clave RSA de tamaño `nbits`. Si se omite `nbits`, es decir, se especifica `-newkey rsa`, se utiliza el tamaño de clave predeterminado, especificado en el archivo de configuración.

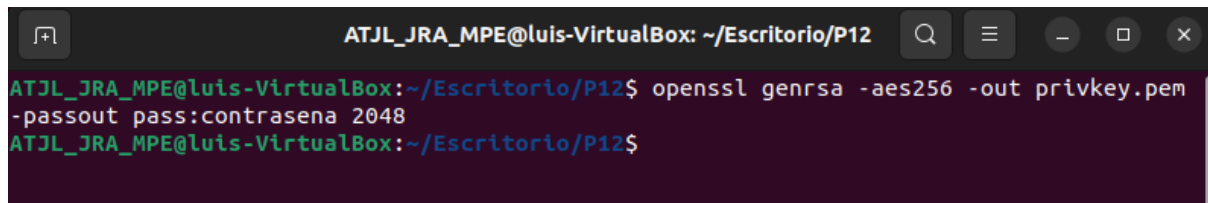
-keyout cakey.pem: Proporciona el nombre del archivo en el que se escribe la clave privada recién creada. Si no se especifica esta opción, se utiliza el nombre de archivo presente en el archivo de configuración.

-days 365: Cuando se utiliza la opción `-x509`, esto especifica la cantidad de días para certificar el certificado (30 días por default para nosotros 365).

-out cacert.pem: El nombre del archivo de salida (`cacert.pem`).

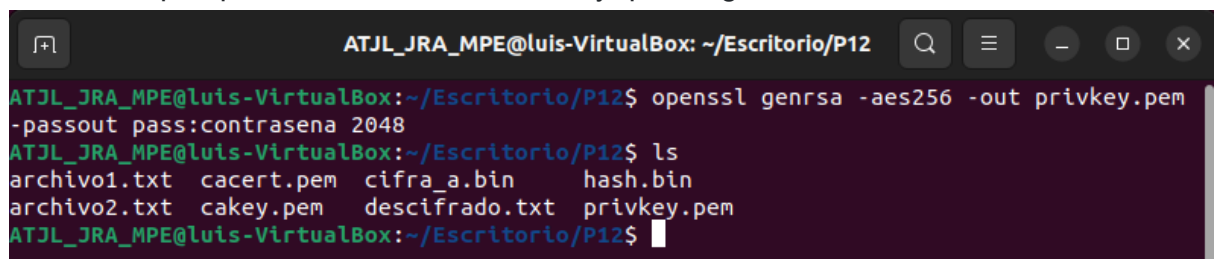
Generación de certificados

10. El tipo de certificado a crear es Certificate Sign Request que puede ser utilizado para dar soporte a sitios Web o sockets. El primer paso es crear la clave privada.
`openssl genrsa -aes256 -out privkey.pem -passout pass:contraseña 2048`



```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl genrsa -aes256 -out privkey.pem -passout pass:contrasena 2048
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

10.1. Verifique que se ha creado el archivo y que tenga el contenido correcto.



```
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl genrsa -aes256 -out privkey.pem -passout pass:contrasena 2048
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ ls
archivo1.txt  cacert.pem  cifra_a.bin  hash.bin
archivo2.txt  cakey.pem   descifrado.txt  privkey.pem
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

11. El siguiente paso es definir al propietario. Se hace una petición donde se especifica a quién pertenece. Se indica la clave privada y la contraseña.

11.1. Ingrese el siguiente comando y **explique cada uno de los parámetros.**
`openssl req -new -subj "/DC=edu.xbe.mx/OU=RedesComp/CN=FC" -key privkey.pem -passin pass:contraseña -out peticion.pem`

openssl req: Crea y procesa principalmente solicitudes de certificados en formato PKCS#10. También puede crear certificados autofirmados para usarlos, como CA raíz.

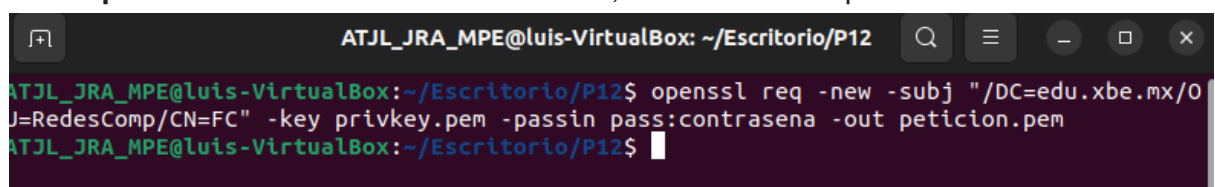
-new: Genera una nueva solicitud de certificado. Le solicitará al usuario los valores de campo relevantes. Los campos reales solicitados y sus tamaños máximo y mínimo se especifican en el archivo de configuración y en cualquier extensión solicitada.

-subj /DC=edu.xbe.mx/OU=RedesComp/CN=FC": Reemplaza el campo de asunto de la solicitud de entrada con datos específicos y genera la solicitud modificada. El argumento debe tener el formato /type0=value0/type1=value1/type2=. . . , los caracteres se pueden escapar mediante (barra invertida) y no se omiten espacios.

-key privkey.pem: Especifica el archivo que leerá la clave privada. También acepta claves privadas en formato PKCS#8 para archivos en formato PEM.

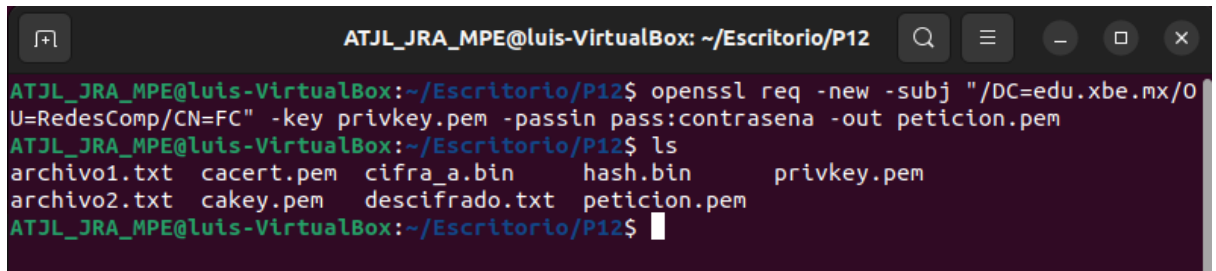
-passin pass:contraseña: Argumento de contraseña para contraseñas de entrada, puede o no tomar argumentos.

-out peticion.pem: El nombre del archivo de salida, salida estándar por defecto.



```
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl req -new -subj "/DC=edu.xbe.mx/OU=RedesComp/CN=FC" -key privkey.pem -passin pass:contrasena -out peticion.pem
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```


11.2. Verifique que el nuevo archivo generado sea correcto.



```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl req -new -subj "/DC=edu.xbe.mx/O=RedesComp/CN=FC" -key privkey.pem -passin pass:contrasena -out peticion.pem
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ ls
archivo1.txt  cacert.pem  cifra_a.bin  hash.bin     privkey.pem
archivo2.txt  cakey.pem   descifrado.txt  peticion.pem
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

Firma del Certificado Digital.

12. Ingrese el siguiente comando que genera el certificado firmado por la AC ya creada y explique cada uno de los parámetros.

openssl x509: Genera un certificado autofirmado en lugar de una solicitud de certificado. Se utiliza para generar un certificado de prueba o una CA raíz autofirmada.

-CA cacert.pem: Es una aplicación de CA mínima. Se puede utilizar para firmar solicitudes de certificados en una variedad de formas y generar CRL. También mantiene una base de datos de texto de los certificados emitidos y su estado.

-CAkey cakey.pem: Utiliza el archivo cakey.pem para obtener la firma de CA ya creada.

-req: Crea y procesa principalmente solicitudes de certificados en formato PKCS#10.

Además, puede crear certificados autofirmados para usarlos.

-in peticion.pem: Especifica la fuente de contraseña del archivo de entrada(peticion.pem).

-days 365: Cuando se utiliza la opción -x509 , esto especifica la cantidad de días para certificar el certificado. El valor predeterminado es 30 días (nosotros 365).

-sha1: Este parámetro es una función hash criptográfica con una salida de 160 bits. En versiones posteriores a OpenSSL 3.0 ha quedado obsoleta, y en su lugar basta definiendo OPENSSL API COMPAT con un valor de versión adecuado.

-CAcreateserial: Asignará el número de serie 01 al certificado firmado y luego creará este archivo de número de serie con el siguiente número de serie 02.

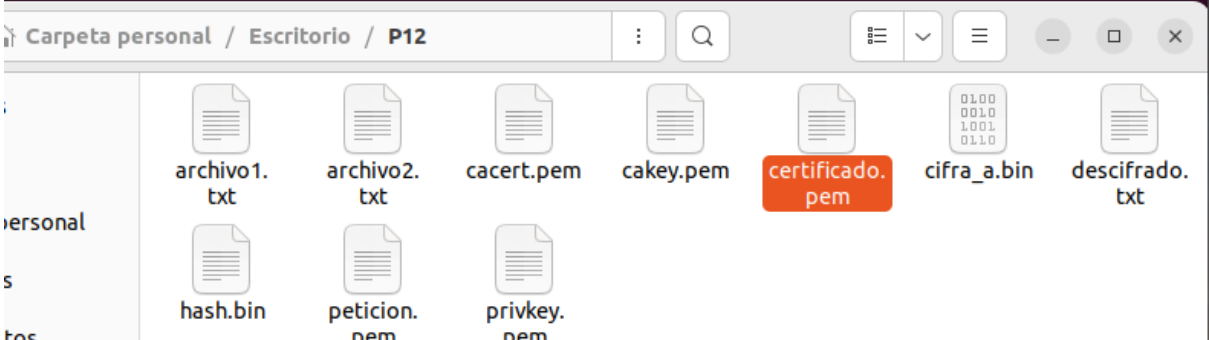
-out certificado.pem: El nombre del archivo de salida, salida está por defecto(certificado.pem)

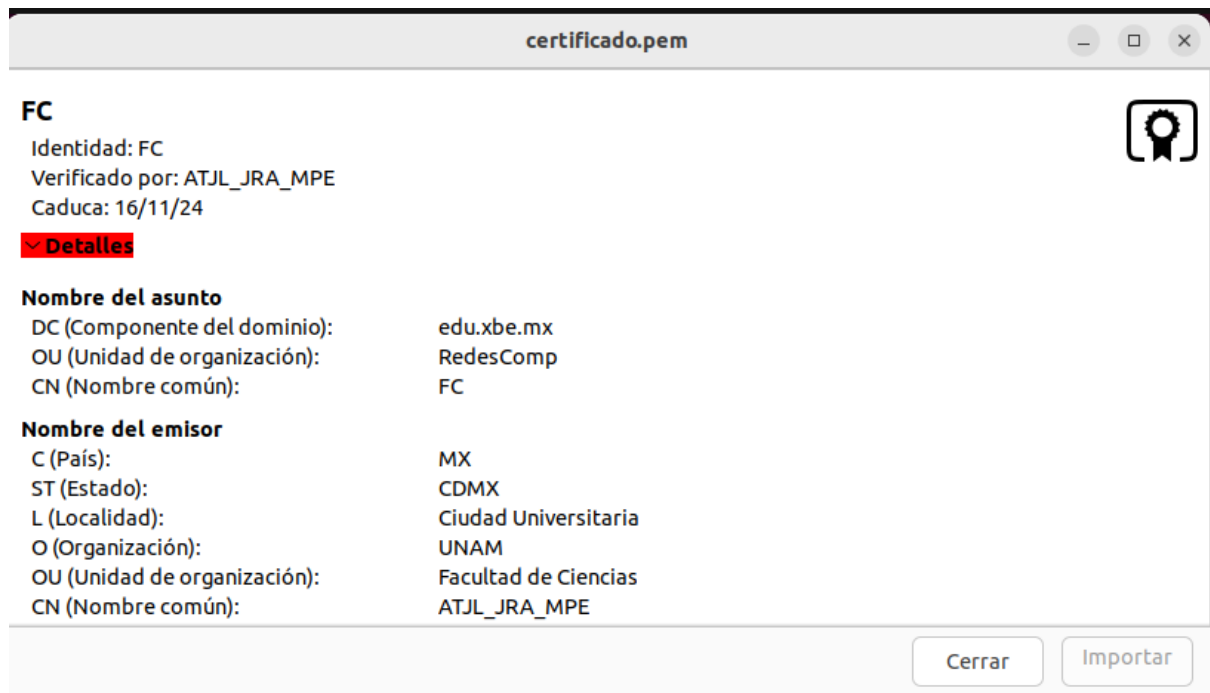
12.1. Ingrese la contraseña creada en el punto 9.3 `openssl x509 -CA cacert.pem -CAkey cakey.pem -req -in petition.pem -days 365 -sha1 -CAcreateserial -out certificado.pem`

```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl x509 -CA cacert.pem -CAkey cakey
.pem -req -in petition.pem -days 365 -sha1 -CAcreateserial -out certificado.pem
Certificate request self-signature ok
subject=DC = edu.xbe.mx, OU = RedesComp, CN = FC
Enter pass phrase for cakey.pem:
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```

13. Verifique que se ha creado correctamente el certificado.

```
ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl x509 -CA cacert.pem -CAkey cakey
.pem -req -in petition.pem -days 365 -sha1 -CAcreateserial -out certificado.pem
Certificate request self-signature ok
subject=DC = edu.xbe.mx, OU = RedesComp, CN = FC
Enter pass phrase for cakey.pem:
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ ls
archivo1.txt  cacert.pem  certificado.pem  descifrado.txt  petition.pem
archivo2.txt  cakey.pem   cifra_a.bin     hash.bin        privkey.pem
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$
```





14. Ingrese el siguiente comando para obtener información del certificado creado.
openssl x509 -in servidorcert.pem -text -noout

```

ATJL_JRA_MPE@luis-VirtualBox: ~/Escritorio/P12
ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$ openssl x509 -in certificado.pem -text -noout
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            30:cb:13:2a:52:7c:b7:a6:8d:db:3e:a7:41:4d:78:fc:14:d8:34:84
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C = MX, ST = CDMX, L = Ciudad Universitaria, O = UNAM, OU = Facultad de Ciencias, CN = ATJL_JRA_MPE, emailAddress = jose-luis@ciencias.unam.mx
        Validity
            Not Before: Nov 17 07:37:44 2023 GMT
            Not After : Nov 16 07:37:44 2024 GMT
        Subject: DC = edu.xbe.mx, OU = RedesComp, CN = FC
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
            Modulus:
                00:cd:fb:6b:cb:d9:cf:ee:b1:09:0d:35:98:f2:44:
                a9:68:d1:cd:6a:14:01:59:fd:72:a4:18:0c:31:f0:
                da:c9:7d:b3:4b:15:23:4a:ad:f2:ad:b3:5a:2f:15:
                f4:d0:5c:05:58:18:4c:63:50:4e:03:6f:81:04:b1:
                6e:5d:9c:01:00:a5:26:e0:93:d2:ad:1c:7e:9f:c9:
                6b:9b:4c:35:60:85:f3:20:fb:9b:81:d9:9e:51:bb:
                4c:ee:fb:c2:a0:65:4b:d7:ac:ac:9a:98:de:ec:46:
                e8:48:a2:bd:25:f3:ef:39:2f:b6:a6:ea:9b:7a:68:
                d2:05:23:14:48:90:61:1f:59:f0:ce:4c:24:c1:81:
                24:bb:1e:7b:ca:02:b0:46:24:41:c9:d7:05:d6:79:
                c5:b9:71:76:f7:c5:ca:14:e2:18:01:72:9d:69:a5:
                00:82:fb:8d:60:d8:ff:ee:b3:25:8f:00:ae:cf:93:
                28:88:53:00:76:22:cf:fa:95:a9:6a:6b:be:c7:e1:
                cc:a0:53:c5:8a:e6:4a:f5:32:66:f0:7e:12:ba:a0:
                4d:59:e4:eb:26:62:15:59:bd:08:48:4f:82:d9:47:
                09:02:f7:02:c0:72:98:01:a0:8f:26:3f:95:ac:17:
                be:4f:d8:4a:eb:4c:a6:86:4d:3d:84:8b:fe:5f:41:
                c0:a9
            Exponent: 65537 (0x10001)
        Signature Algorithm: sha1WithRSAEncryption
        Signature Value:

```

Signature Value:

```

29:23:96:f1:0f:d0:74:e5:2c:1c:3c:39:11:97:f9:62:01:90:
36:17:e9:2f:4b:84:45:46:55:ef:8d:50:58:90:95:bb:66:83:
f4:70:f5:78:49:82:4c:68:2b:e1:da:54:dd:95:ae:ab:47:98:
09:f9:be:29:09:f0:5a:64:9a:35:78:09:2c:17:f2:f8:dd:fb:
c4:a1:f0:00:3c:c9:0b:e2:20:ca:c1:21:40:50:08:15:f2:a6:
58:89:ca:29:d8:86:9a:43:b8:69:a0:b7:8b:84:27:e5:13:71:
6f:be:eb:53:fb:6d:c6:23:36:00:4d:03:6e:db:12:ee:19:f5:
ef:d3:66:72:57:18:ce:f8:03:42:97:b3:74:98:be:ed:a6:72:
cc:4b:9b:ba:0e:8b:43:c2:6c:51:80:e2:40:49:98:d3:e0:a9:
af:e2:8a:1b:63:45:7a:4d:5e:e0:e8:38:28:8a:4f:e6:d3:85:
fd:20:c5:ef:8a:3b:74:58:4c:f4:cd:88:7b:1a:21:6f:fe:5c:
56:7e:b1:87:7c:17:ad:dc:a8:82:14:04:85:de:10:1a:f2:8a:
4b:55:d6:6a:50:a3:f5:8e:2d:a2:29:d4:e0:51:db:31:49:39:
d5:59:db:b9:71:5a:3c:69:d4:02:0e:20:a6:42:52:95:cf:67:
45:9a:4e:bb

```

```

ATJL_JRA_MPE@luis-VirtualBox:~/Escritorio/P12$

```

14.1. Explique la información mostrada por el comando anterior.

- openssl: Es el comando para invocar OpenSSL, una herramienta de código abierto para la administración de certificados SSL/TLS y criptografía en general.
- x509: Es la suborden que indica que se va a trabajar con certificados X.509.
- certificado.pem: Especifica el nombre del archivo de certificado del cual se desea obtener información.
- text: Indica que se desea mostrar el contenido del certificado de una manera legible.
- noout: Se usa para evitar que se muestre la salida predeterminada (como la codificación del certificado).

La información mostrada por el comando anterior es:

- Número de Serie: Identificador único asignado al certificado.
- Emisor: La entidad que emitió el certificado.
- Titular: La entidad a la que se emitió el certificado.
- Validez: Fechas de inicio y vencimiento del certificado.
- Algoritmo de Clave Pública: Tipo de algoritmo utilizado para la clave pública.
- Llave Pública: La propia llave pública (generalmente codificada).
- Método de encriptación: RSA Encryption.
- Tamaño de bits que contiene: 2048
- La llave misma

Conclusiones

José Luis:

En esta práctica aprendimos acerca de los certificados digitales y firmas digitales, los cuales son muy importantes para la autenticidad y la comunicación digital. Vimos que un certificado digital, emitido por una Autoridad de Certificación (CA), garantiza la relación entre la identidad de un sujeto y su clave pública, proporcionando una manera confiable para verificar la autenticidad de programas, documentos y comunicaciones cifradas.

El contenido de un certificado digital, que incluye información como el nombre distinguido del propietario, la clave pública y la firma de la CA emisora, es fundamental para su validez y seguridad, además el proceso de firma digital utiliza algoritmos criptográficos, como RSA, para vincular al autor al documento a través de una clave secreta única.

En esta práctica aprendimos el uso del programa OpenSSL, con la cual calculamos funciones hash (SHA-1) para generar firmas digitales seguras y verificar la autenticidad de los certificados digitales, esta herramienta ayuda a la seguridad y confianza en la comunicación digital.

Abraham: Aprendimos sobre este programa OpenSSL que yo nunca había utilizado pero con ayuda de las instrucciones resultó sencillo, la utilidad que yo le vi en particular es que puede encriptar y crear tus propias claves públicas y privadas para servidores web.

No olvidemos que también creamos el certificado digital, los certificados sirven para verificar la autenticidad de los servidores de un dominio web, de modo que sus dueños sean reconocidos. Y también hay que resaltar las funciones hash ya que son importantes para la criptografía. Para esta práctica no tuvimos ningún inconveniente

Esaú: En esta práctica generamos un certificado digital que es un documento firmado electrónicamente por un prestador de servicios de certificación, considerado como una autoridad para este tipo de contenido, que vincula unos datos de verificación de firma a un firmante, de forma que únicamente puede firmar este firmante, y confirma su identidad, así es que verifica la autenticidad de programas, documentos de la red, correos cifrado o firmado digitalmente, permitiendo hacer uso de una firma digital segura. Para ello hicimos uso de OpenSSL, creando un archivo de texto sin contenido y luego con contenido y aplicando diversas funciones hash almacenando llaves públicas y privadas, para llegar a crear nuestro propio certificado con características como el emisor, titular y la llave pública.

Referencias

- jam. (2020, 12 julio). *Cómo instalar y actualizar OpenSSL en Ubuntu 20.04 2021* [Video]. YouTube. <https://www.youtube.com/watch?v=jk7Ni-Eam3s>
- IBM documentation. (s. f.). <https://www.ibm.com/docs/es/rstfsq/9.1.0?topic=overview-creating-digital-certificate-openssl>
- SSH Shopper. (2014). Los comandos OpenSSL más comunes | SSH Shopper. Recuperado el 17 de noviembre de 2023, de <https://www.sshshopper.com/article-most-common-openssl-commands.html>
- Vladimir Kaplarevic. (2019). How to Check the OpenSSL Version Number | Phoenix NAP. Recuperado el 17 de noviembre de 2023, de <https://phoenixnap.com/kb/how-to-check-openssl-version>
- Mauricio. (2022). Comando cat linux: ejemplos y práctica | DONGEE. Recuperado el 16 de noviembre de 2023, de <https://www.dongee.com/tutoriales/comando-cat-linux/>
- OpenSSL. (2020). OpenSSL Software Foundation | OpenSSL. Recuperado de el 16 de noviembre de 2023, de <https://www.openssl.org/docs/man3.0/man1/openssl-enc.html>
- OpenSSL. (2020). Opciones de frase de contraseña de openssl | OpenSSL. Recuperado el 16 de noviembre de 2023, de <https://www.openssl.org/docs/man3.0/man1/openssl-passphrase-options.html>
- OpenSSL. (2020). x509 | OpenSSL. Recuperado el 16 de noviembre de 2023, de <https://www.openssl.org/docs/man1.1.1/man1/x509.html>
- OpenSSL. (2020). openssl | OpenSSL. Recuperado el 16 de noviembre de 2023, de <https://www.openssl.org/docs/man1.0.2/man1/openssl.html>
- OpenSSL. (2020). req | OpenSSL. Recuperado el 16 de noviembre de 2023, de <https://www.openssl.org/docs/man1.0.2/man1/openssl-req.html>
- Tecnocrática. (). Certificados PEM. Cómo crear un archivo .pem para instalaciones de certificados SSL | Tecnocrática. Recuperado el 16 de noviembre de 2023, de <https://tecnocratica.net/wikicratica/books/certificados/page/certificados-pem>

