

Hosts Virtuales

Autoridad de certificación

Sitios en misma y diferente IP

PHP: Páginas dinámicas

Índice

Apache2.....	2
Certificar.....	2
Autoridad de certificación.....	2
Clave privada.....	2
Certificado.....	2
Certificar un dominio.....	3
Habilitar modulo SSL.....	4
Host Virtuales.....	5
Sitios en la misma IP.....	5
Sitios en diferentes IPs.....	6
Habilitar Host Virtuales.....	6
Pruebas.....	7

Apache2

En esta práctica vamos a configurar cuatro hosts virtuales para que, dos atiendan a la misma IP, diferenciándose por su nombre, y dos atiendan a IPs diferentes.

Certificar

Como vamos a utilizar la conexión cifrada, necesitaremos un certificado por cada uno de los host virtuales (pues cada uno atiende a un nombre de dominio diferente).

Vamos a realizar el primero, siendo posible hacer el resto por analogía.

Autoridad de certificación.

Primero hemos de crear una autoridad de certificación o CA (*Certification Authority*, en inglés), pues necesitamos de esta para firmar los certificados que vamos a crear.



En la práctica, crear una **Autoridad de Certificación (CA)** no es realista. Estas autoridades son limitadas y, a grandes rasgos, es complicado crear una de forma individual. Se utilizan CA ya existentes para firmar certificados.

Clave privada

Creamos la clave privada del servidor.

```
openssl genpkey-algorithm RSA-pkeyopt rsa_keygen_bits:2048-pkeyopt  
rsa_keygen_pubexp:65537-out cakey.pem
```

Certificado

Y luego el certificado

```
openssl req-new-x509-key cakey.pem-out cacert.pem-days 365
```

Deberíamos ver un resultado parecido al siguiente:

```
root@vmi2144575: /etc/certificate/demoCA  
root@vmi2144575:/etc/certificate/demoCA# ls -l  
total 8  
-rw-r--r-- 1 root root 1375 Oct 29 11:54 cacert.pem  
-rw----- 1 root root 1704 Oct 29 11:49 cakey.pem  
root@vmi2144575:/etc/certificate/demoCA#
```

En este mismo directorio, deberíamos crear los siguientes recursos.

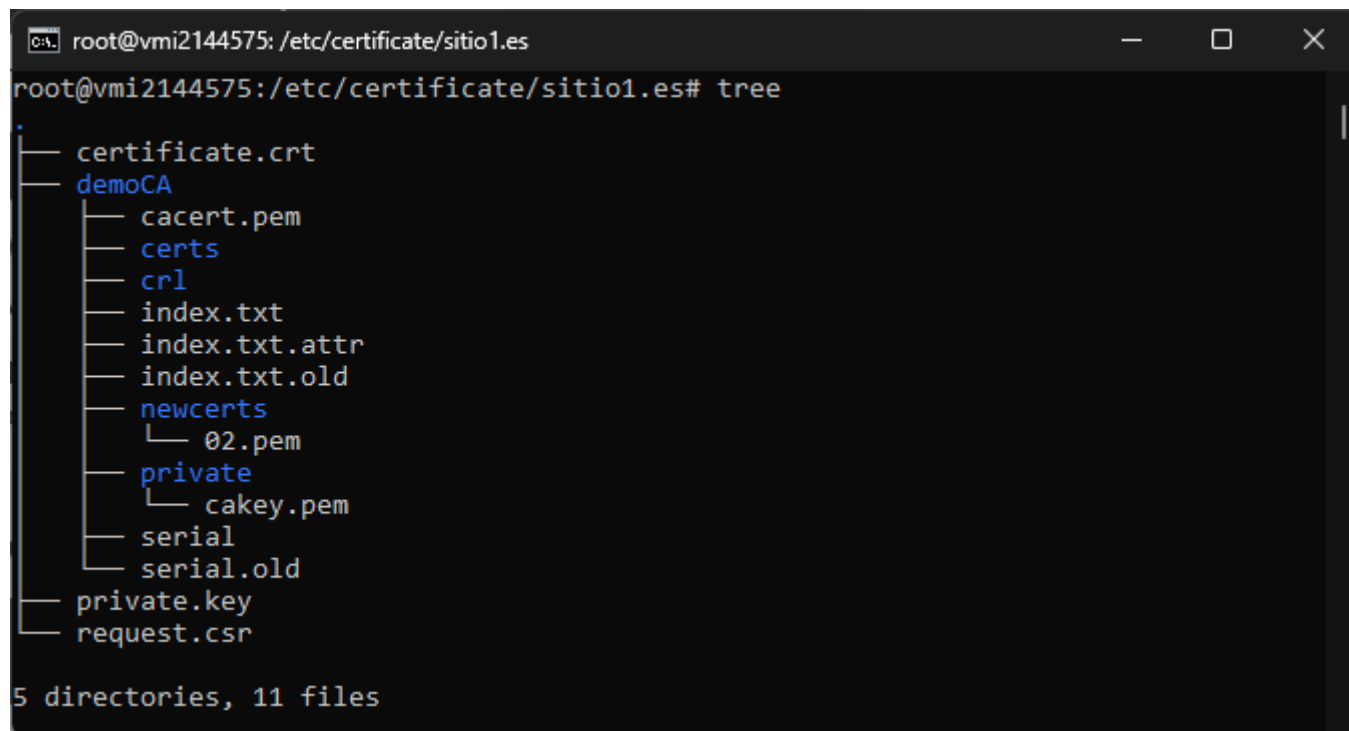
```
mkdir demoCA/certs  
mkdir demoCA/crl  
mkdir demoCA/newcerts
```

```
mkdir demoCA/private
touch demoCA/index.txt
echo 02 > demoCA/serial
```

También, movemos la clave privada al directorio `private`.

```
mv cakey.pem ./private/.
```

Para cada sitio, deberíamos crear un directorio para trabajar con facilidad. Pero, en cada directorio, deberemos copiar la carpeta `demoCA`. Resultando así:



```
root@vmi2144575: /etc/certificate/sitio1.es
root@vmi2144575:/etc/certificate/sitio1.es# tree
.
├── certificate.crt
├── demoCA
│   ├── cacert.pem
│   ├── certs
│   ├── crl
│   ├── index.txt
│   ├── index.txt.attr
│   ├── index.txt.old
│   ├── newcerts
│   │   └── 02.pem
│   ├── private
│   │   └── cakey.pem
│   ├── serial
│   └── serial.old
├── private.key
└── request.csr

5 directories, 11 files
```

Certificar un dominio

Ahora podemos certificar nuestro dominio, en este caso, `sitio1.es`.

Como antes, creamos una clave privada para el certificado.

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -pkeyopt
rsa_keygen_pubexp:65537 -out private.key
```

Luego, creamos una petición (*request*, en inglés) de firma de nuestro certificado.

```
openssl req -new -key private.key -out request.csr
```

Nosotros, como Autoridad de Certificación, firmamos el certificado.

```
openssl ca -in request.csr -out certificate.crt
```

El resultado debería ser tres archivos, siendo el de petición desechable.

```
root@vmi2144575: /etc/certificate/sitio1.es
-rw-r--r-- 1 root root 4382 Oct 29 12:02 certificate.crt
drwxr-xr-x 6 root root 4096 Oct 29 12:02 demoCA
-rw----- 1 root root 1704 Oct 29 11:55 private.key
-rw-r--r-- 1 root root 1021 Oct 29 11:56 request.csr
root@vmi2144575:/etc/certificate/sitio1.es#
```

Si queremos verificar que nuestro certificado es funcional. Podemos ejecutar:

```
openssl ca -in request.csr -out certificate.crt
```

```
root@vmi2144575: /etc/certificate/sitio1.es
root@vmi2144575:/etc/certificate/sitio1.es# openssl verify -CAfile demoCA/cacert.p
em certificate.crt
certificate.crt: OK
root@vmi2144575:/etc/certificate/sitio1.es#
```

Una vez creados. Debemos mover los certificados y las claves a un sitio más estandarizado.

```
cp certificate.crt /etc/ssl/certs/
cp demoCA/cacert.pem /etc/ssl/certs/
cp private.key /etc/ssl/private/
```

Habilitar modulo SSL

Recordemos que es importante habilitar el módulo SSL si no lo está.

```
a2enmod ssl
apachectl restart
```

Host Virtuales

Lo primero es crear los hosts virtuales. Podemos hacerlo de la siguiente manera.

Sitios en la misma IP

Para el `sitio1.es..`

```
root@vmi2144575: /etc/apache2/sites-available
<IfModule mod_ssl.c>
  <VirtualHost 192.168.1.100:443>
    ServerAdmin angel@sitio1.es
    DocumentRoot /var/www/sitio1.es
    ServerName sitio1.es

    SSLEngine on

    SSLCertificateFile /etc/ssl/certs/certificate.sitio1.es.crt
    SSLCertificateKeyFile /etc/ssl/private/private.sitio1.es.key
    SSLCACertificateFile /etc/ssl/certs/demo.cacert.pem

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
  </VirtualHost>
</IfModule>
"sitio1.es.conf" 17L, 463B written 17,11 All
```

Para el `sitio2.es..`

```
root@vmi2144575: /etc/apache2/sites-available
<IfModule mod_ssl.c>
  <VirtualHost 192.168.1.100:443>
    ServerAdmin angel@sitio2.es
    DocumentRoot /var/www/sitio2.es
    ServerName sitio2.es

    SSLEngine on

    SSLCertificateFile /etc/ssl/certs/certificate.sitio2.es.crt
    SSLCertificateKeyFile /etc/ssl/private/private.sitio2.es.key
    SSLCACertificateFile /etc/ssl/certs/demo.cacert.pem

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
  </VirtualHost>
</IfModule>
"sitio2.es.conf" 17L, 463B 13,0-1 All
```

Sitios en diferentes IPs

Para el `sitio3.es`.

```
root@vmi2144575: /etc/apache2/sites-available
<IfModule mod_ssl.c>
    <VirtualHost 192.168.1.101:443>
        ServerAdmin angel@sitio3.es
        DocumentRoot /var/www/sitio3.es
        ServerName sitio3.es

        SSLEngine on

        SSLCertificateFile /etc/ssl/certs/certificate.sitio3.es.crt
        SSLCertificateKeyFile /etc/ssl/private/private.sitio3.es.key
        SSLCACertificateFile /etc/ssl/certs/demo.cacert.pem

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
</IfModule>
"sitio3.es.conf" 17L, 463B 17,11 All
```

Para el `sitio4.es`.

```
root@vmi2144575: /etc/apache2/sites-available
<IfModule mod_ssl.c>
    <VirtualHost 192.168.1.102:443>
        ServerAdmin angel@sitio4.es
        DocumentRoot /var/www/sitio4.es
        ServerName sitio4.es

        SSLEngine on

        SSLCertificateFile /etc/ssl/certs/certificate.sitio4.es.crt
        SSLCertificateKeyFile /etc/ssl/private/private.sitio4.es.key
        SSLCACertificateFile /etc/ssl/certs/demo.cacert.pem

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
</IfModule>
"sitio4.es.conf" 17L, 463B 10,62-76 All
```

Habilitar Host Virtuales

Recordemos que hay que habilitarlos. Podemos hacerlo con el comando `a2ensite`.

```
sudo a2ensite sitio1.es.conf
sudo a2ensite sitio2.es.conf
sudo a2ensite sitio3.es.conf
sudo a2ensite sitio4.es.conf
```

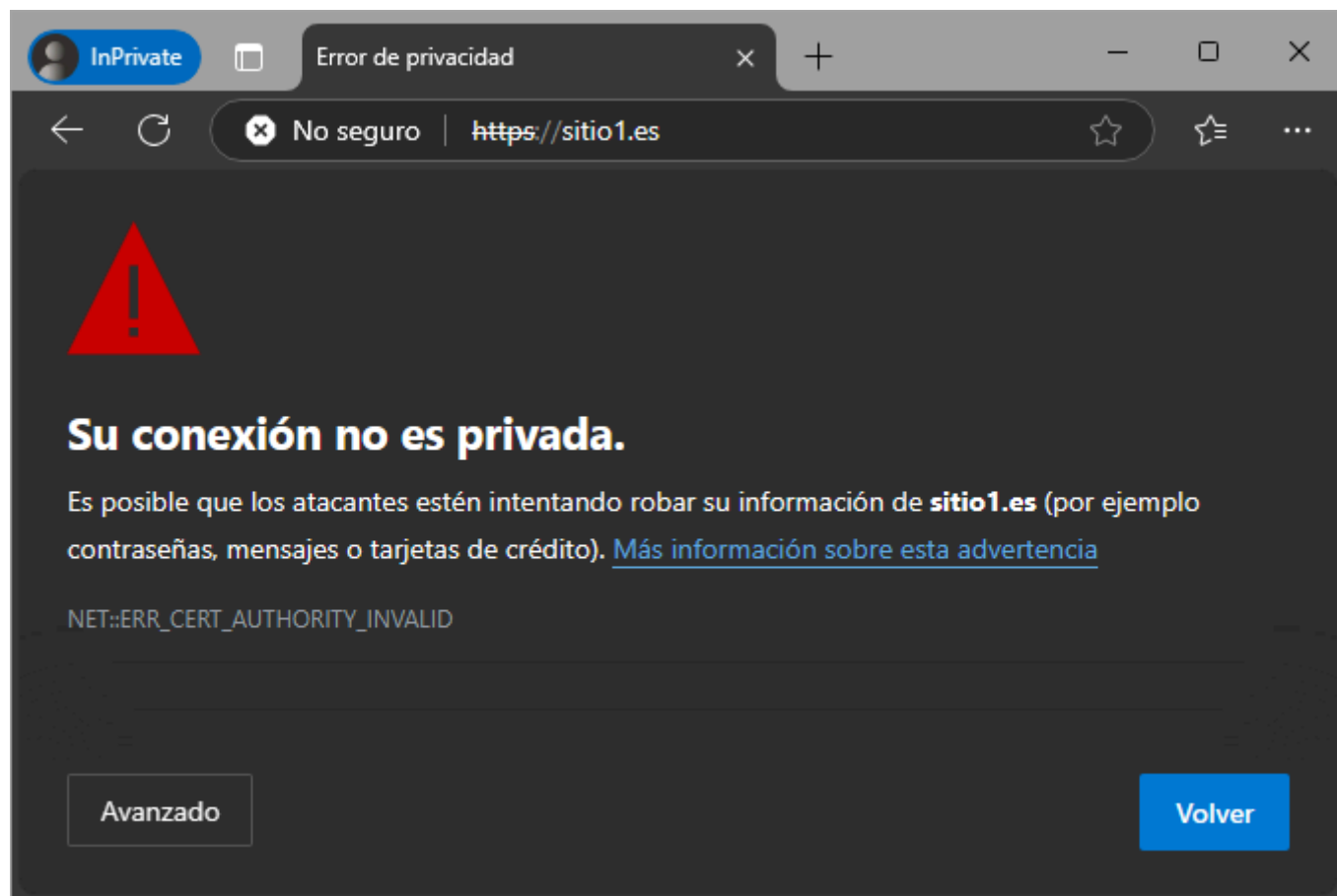
Pruebas

Para las pruebas, hemos creado unas páginas web de ejemplo.

Ahora podemos, en teoría, acceder a las páginas web.



Para que podamos escribir el nombre de dominio en el buscador, cuando en realidad no existe dicho dominio, hay que editar el archivo **hosts** del sistema operativo y añadir dicho dominio y su respectiva IP.



Como vemos, nos sale una advertencia que la conexión no es privada, insegura. Esto es normal, pues la autoridad de certificación no existe.

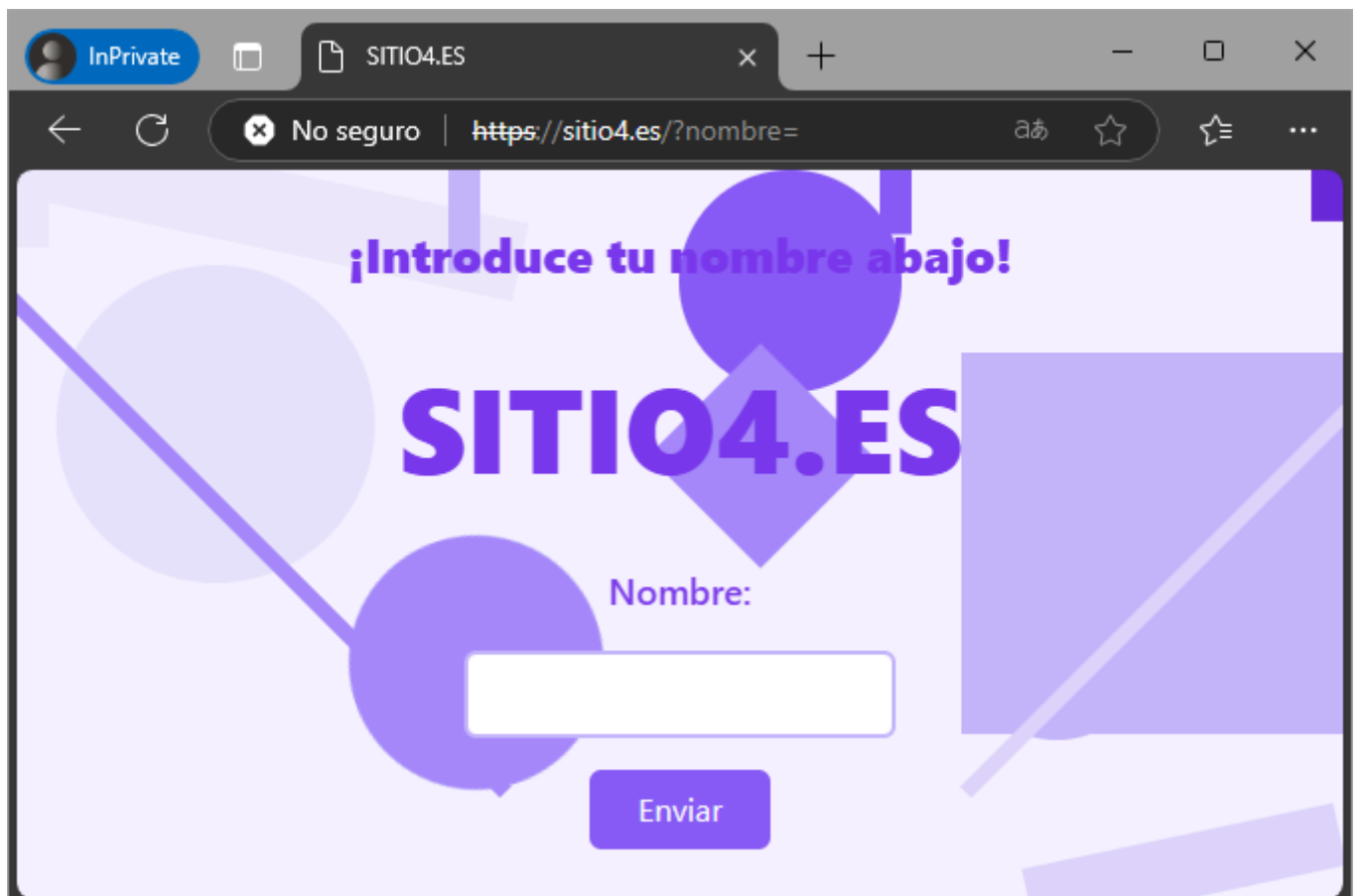
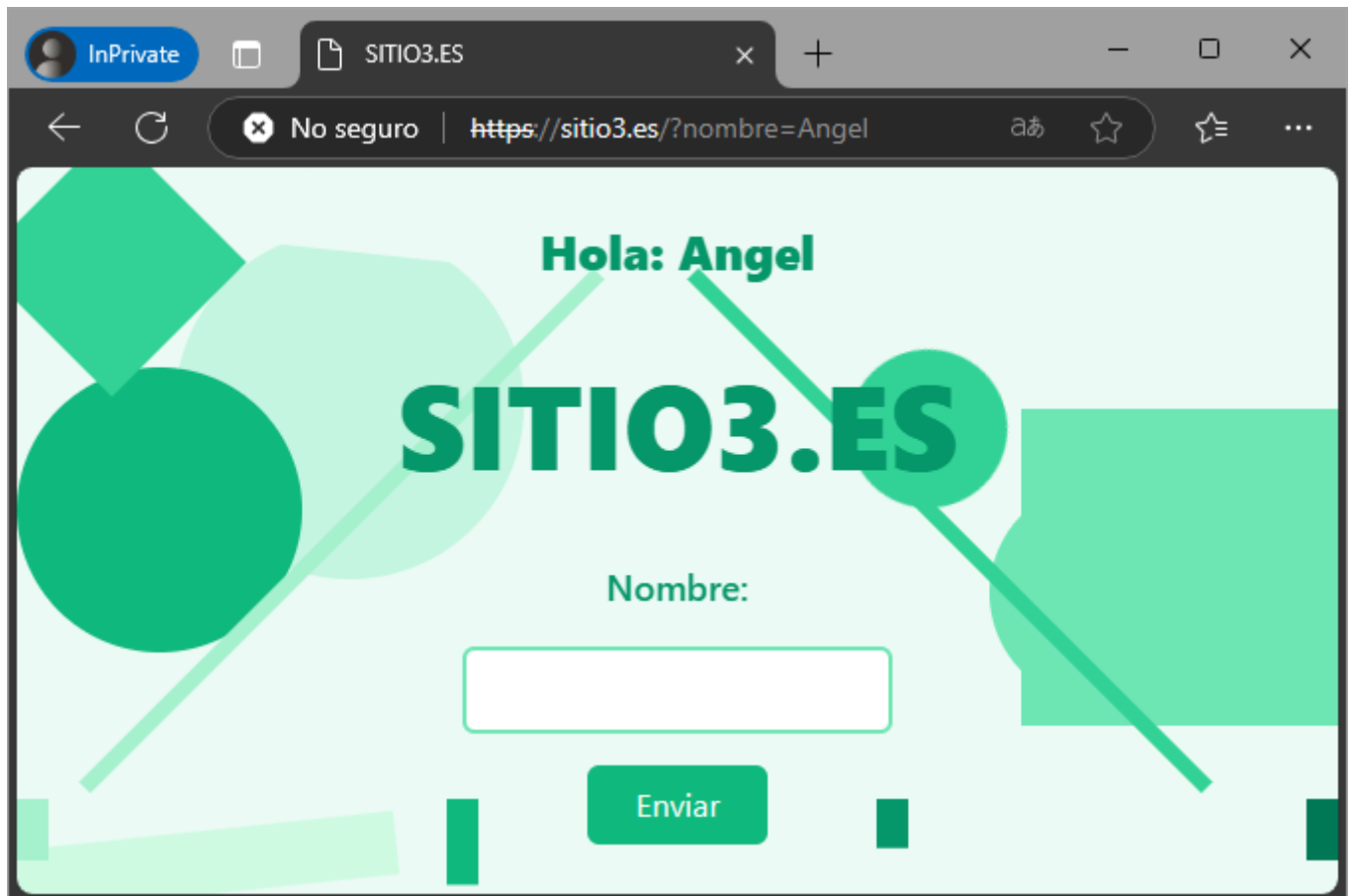
Igualmente le podemos dar a **Continuar a sitio1.es (no seguro)**.

Este servidor no puede demostrar que es **sitio1.es**; su certificado de seguridad no es válido para el sistema operativo de tu equipo. Esto puede deberse a un error de configuración o a un intruso esté interceptando la conexión.

[Continuar a sitio1.es \(no seguro\)](#)

Una vez dentro, podemos ver la página. Son páginas dinámicas que utilizan PHP para mostrar el nombre introducido en un formulario.





Aquí podemos ver el certificado de la Autoridad de Certificación que nos hemos inventado.

Visor de certificados: sitio1.es ✕

General Detalles

Enviado a

Nombre común (CN)	sitio1.es
Organización (O)	IES Las Galletas
Unidad organizativa (OU)	DAW2

Emitido por

Nombre común (CN)	localhost
Organización (O)	IES Las Galletas
Unidad organizativa (OU)	DAW2

Período de validez

Emitido el	martes, 29 de octubre de 2024, 11:01:49
Vencimiento el	miércoles, 29 de octubre de 2025, 11:01:49

Huellas digitales SHA-256

Certificado	3b69ce83b591a67fc9876dfc69d280a618009d7e9a04a357c9cdeba1d47f01d3
Clave pública	eee3371bc015b79429e353d924a10b4c736f08eee00cfd353779fb87f2f0c348

Extra

Como extra, para poder redirigir cualquier petición HTTP a HTTPS. Podemos configurar lo siguiente en, por ejemplo, el archivo `.htaccess`:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{SERVER_NAME}/$1 [R,L]
```