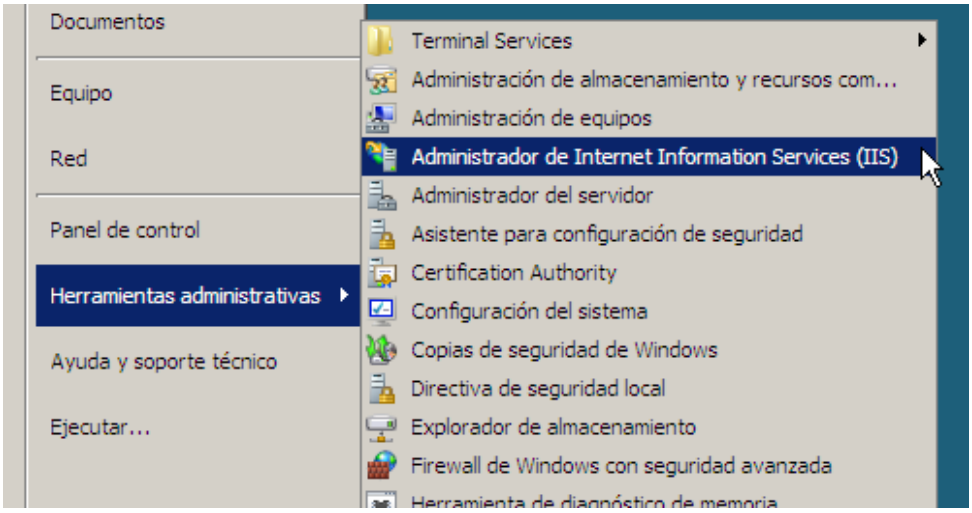
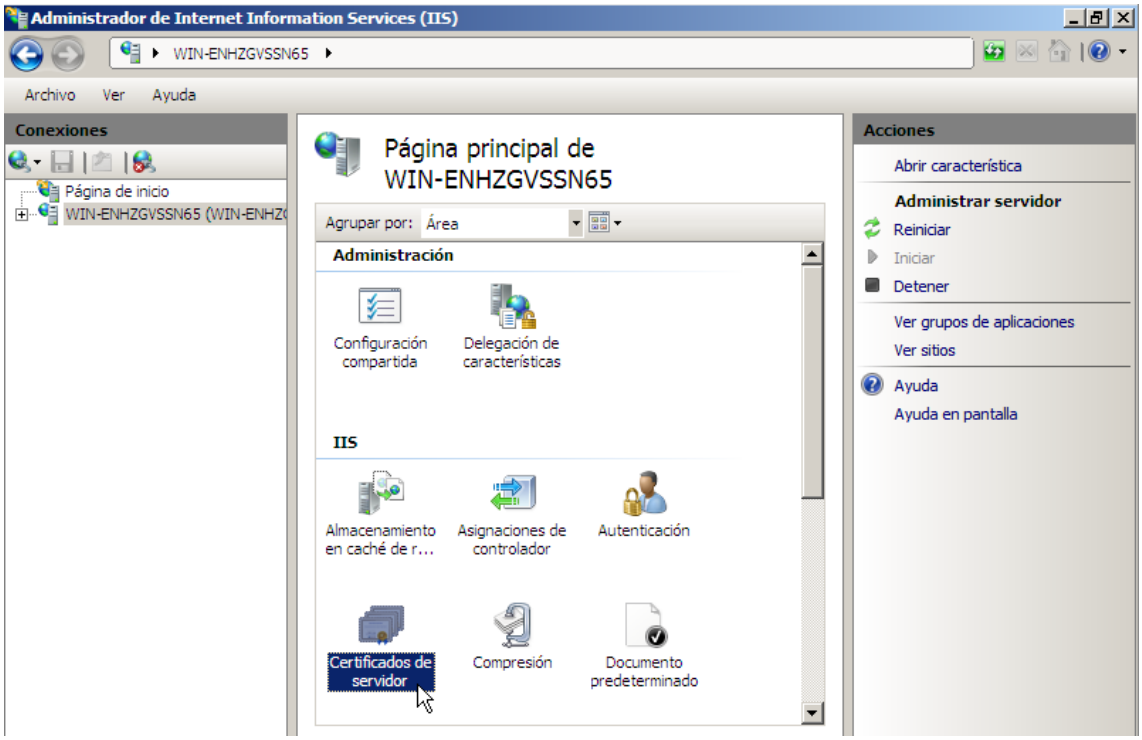
Crea un sitio Web seguro usando tu propio certificado digital (Windows y Linux).

Windows

Abrimos la ventana del *Administrador de IIS*, desde *Herramientas administrativas*.

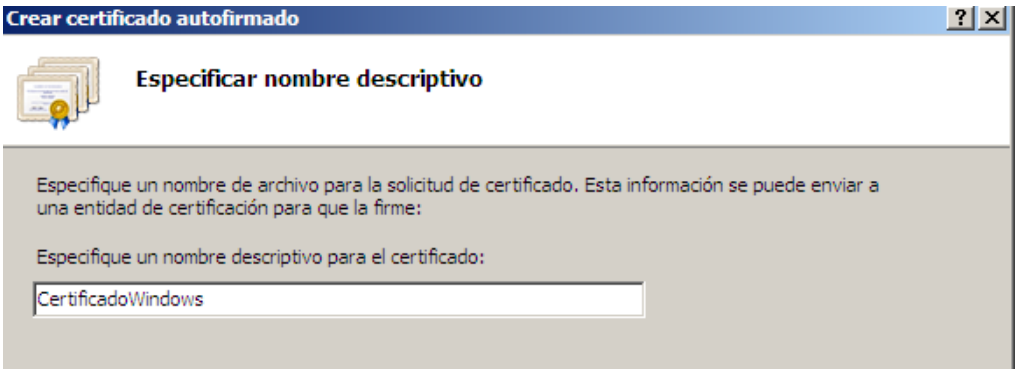


Entramos en Certificados de servidor.

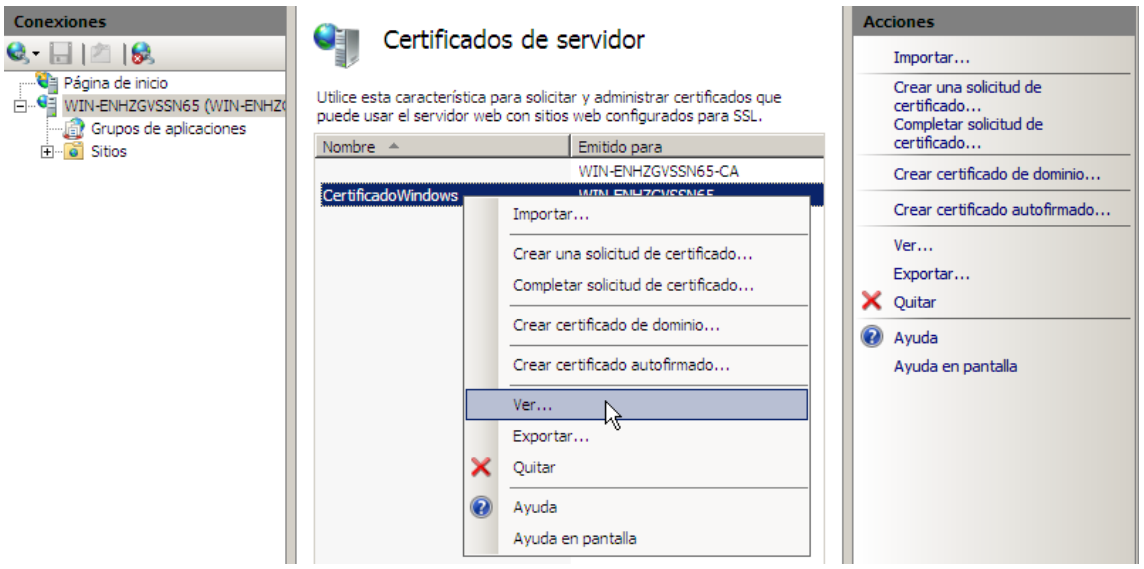


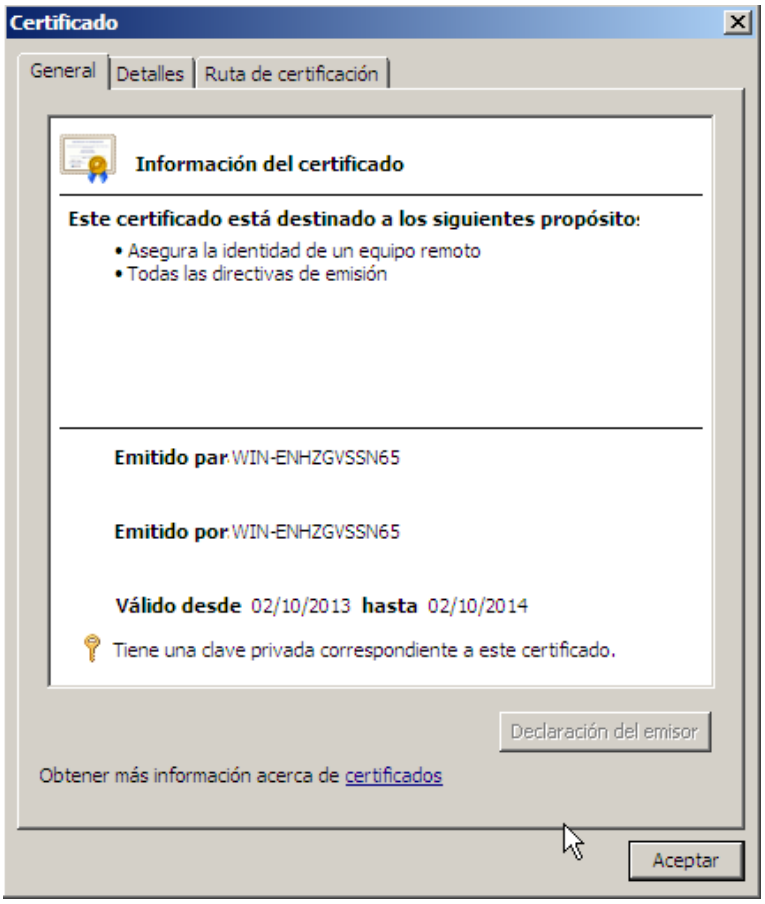
En el menú de la derecha, hacemos clic en **Crear certificado autofirmado...**

Le especificamos un nombre al certificado.

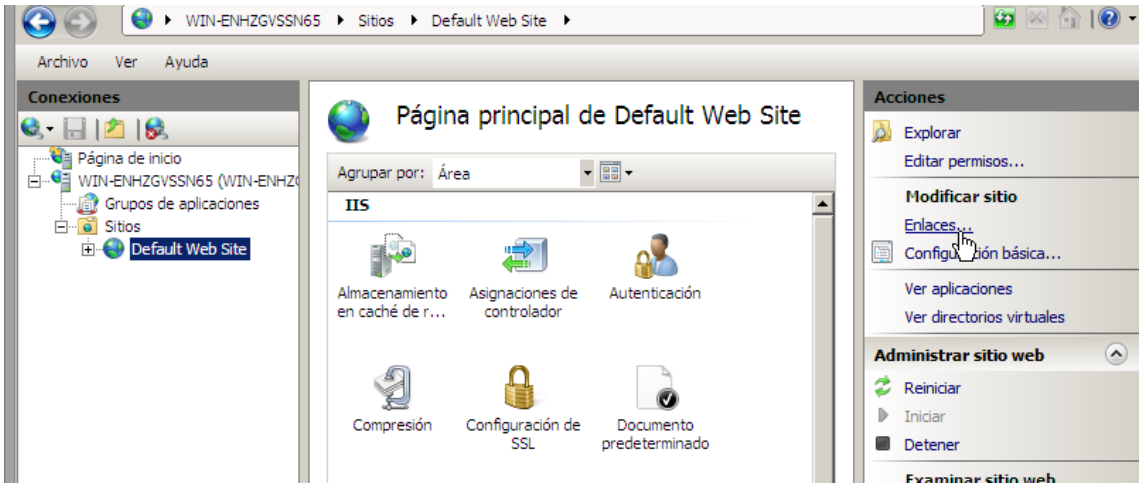


Si hacemos *clic derecho* > *Ver* sobre el certificado creado, podemos ver sus características.

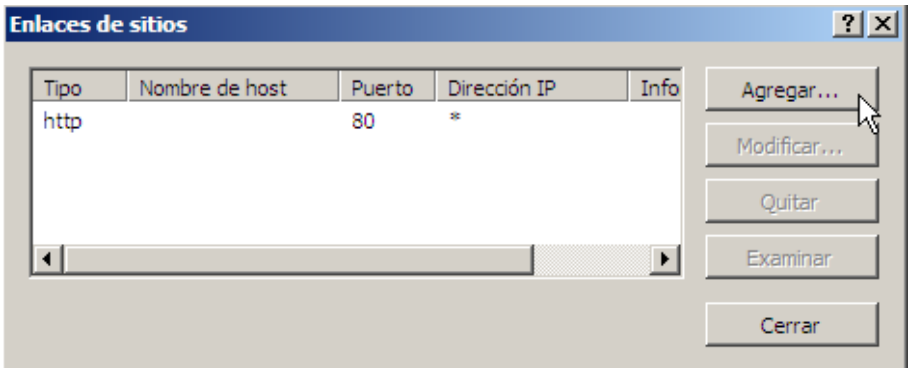




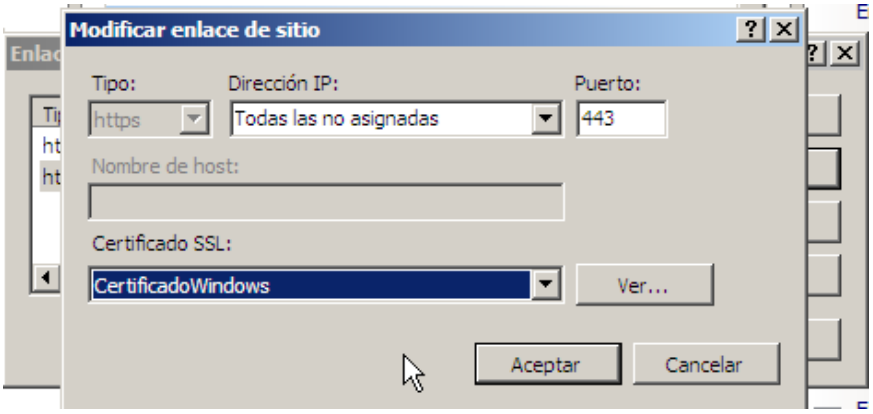
Vamos a Default Web Site, y entramos en **Enlaces...** en el menú de la derecha.



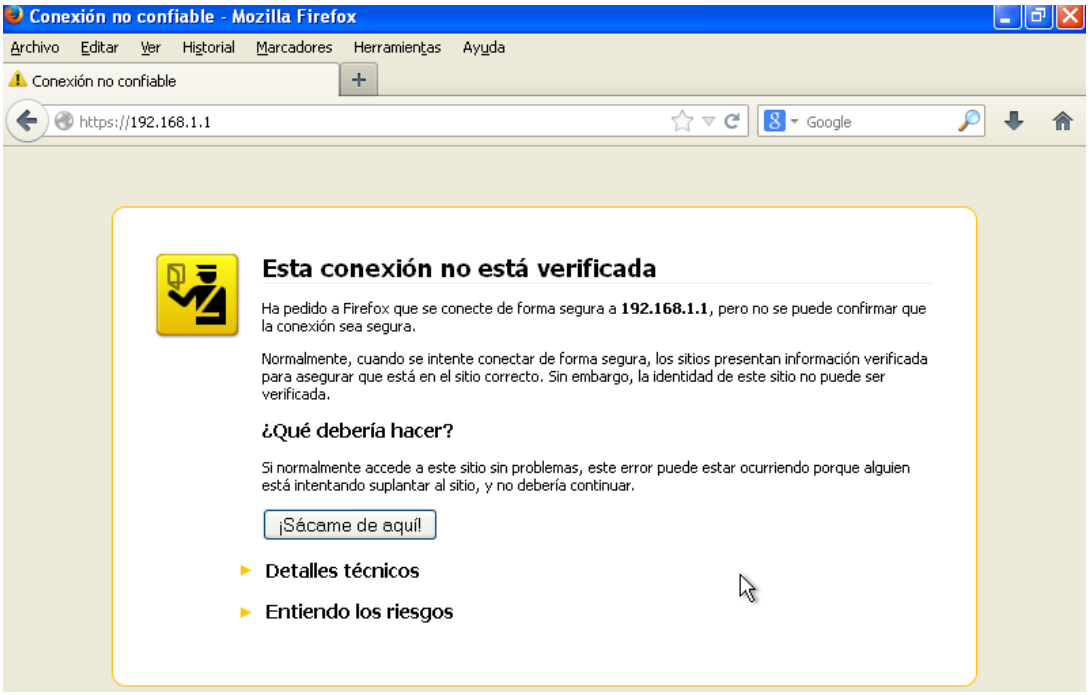
En la nueva ventana que se abre, pinchamos en el botón **Agregar...**



Seleccionamos el tipo HTTPS y elegimos nuestro certificado. Aceptamos y ya tenemos nuestro certificado instalado en Windows.



Vamos a un cliente para entrar a nuestro servidor y comprobamos que el certificado funciona correctamente.



Linux

El primer paso es crear nuestra clave privada. Abrimos el terminal, y escribimos el comando **openssl genrsa > clave.key**. Esto generará una clave privada y la guardará en un fichero llamado clave.key

```
servidorubuntu@ubuntu:~$ openssl genrsa > clave.key
Generating RSA private key, 512 bit long modulus
..++++++++++++
.......++++++++++++
e is 65537 (0x10001)
```

La clave pública se encuentra incluida dentro de la clave privada, y para extraerla usaremos el comando **openssl rsa –in clave.key –pubout –out publica.key**. Estará en un fichero llamado publica.key

```
servidorubuntu@ubuntu:~$ openssl rsa -in clave.key -pubout -out publica.key
writing RSA key
```

Usando ambas claves, crearemos nuestro certificado digital en un archivo llamado servidor.pem usando el comando **openssl req –new –key clave.key –x509 –days 365 –out servidor.pem**.

```
servidorubuntu@ubuntu:~$ openssl req -new -key clave.key -x509 -days 365 -out se
rvidor.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

Para instalar el certificado en nuestro servidor Apache, llevamos la clave privada clave.key al directorio /etc/ssl/private y el certificado servidor.pem al directorio /etc/ssl/certs

```
servidorubuntu@ubuntu:~$ sudo cp clave.key /etc/ssl/private/clave.key
servidorubuntu@ubuntu:~$ sudo cp servidor.pem /etc/ssl/certs/servidor.pem
```

Modificamos el fichero /etc/apache2/sites-available/default-ssl para incluir las rutas de la clave privada y el certificado, tal como se indica en la captura.

```
 GNU nano 2.2.6  Archivo: /etc/apache2/sites-available/default-ssl Modificado

        </Directory>

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile    /etc/ssl/certs/servidor.pem
        SSLCertificateKeyFile /etc/ssl/private/clave.key
```

Escribimos el comando **sudo a2enmod ssl** y después reiniciamos el servicio apache2.

```
servidorubuntu@ubuntu:~$ sudo a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
 create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
servidorubuntu@ubuntu:~$ sudo service apache2 restart
 * Restarting web server apache2
 ... waiting                                                          [ OK ]
```

Después, el comando **sudo a2ensite default-ssl** y **sudo service apache2 reload**.

```
servidorubuntu@ubuntu:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
servidorubuntu@ubuntu:~$ sudo service apache2 reload
 * Reloading web server config apache2                                 [ OK ]
```

Usamos un cliente para comprobar que el certificado funciona correctamente.