

SERVIDORES WEB

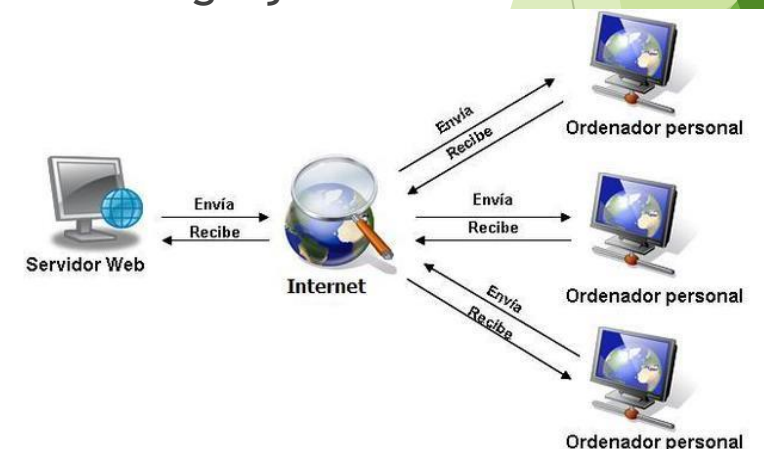


Héctor Morán

Despliegue de Aplicaciones Web – CIFP Juan de Colonia – 2021/2022

Servidor web

- ▶ Un servidor web es una aplicación de servidor cuyo cometido es responder a las peticiones mediante protocolo http de los clientes con el recurso web solicitado (típicamente una página web en html). En otras palabras, es el software que permite a los usuarios acceder a una página web para visualizarla en su navegador web.
- ▶ Los servidores web pueden estar configurados para permitir el acceso a través de Internet, o únicamente a los usuarios de una red local (intranet).
- ▶ Existen multitud de servidores web, elegir el que utilicemos dependerá tanto de la plataforma sobre la que va a ser instalado el servidor web, como del lenguaje de programación que se utilice para programar la aplicación web.



Funcionamiento básico de un servidor web



- ▶ El esquema básico de una petición de página web desde un navegador es el siguiente:
 - ▶ En la primera consulta de un usuario a una web, se establece una conexión entre el ordenador y el servidor DNS, que responde con la dirección IP del servidor web donde está alojado el contenido al que queremos acceder.
 - ▶ Una vez localizado el servidor web, se le envía una solicitud del contenido mediante http (puerto 80) o https (puerto 443).
 - ▶ El servidor web recibe la solicitud de contenido del cliente web, busca el contenido y lo procesa (ejecutando código en el caso de que la página tenga contenido dinámico).
 - ▶ Finalmente, envía el contenido solicitado (típicamente en forma de html) al cliente web que lo solicitó.

Servidores web

- ▶ Las cuatro aplicaciones que actúan como servidores web más populares en la actualidad son Apache, Nginx, IIS y LiteSpeed, cada una con sus peculiaridades:

- ▶ Apache: durante muchísimos años ha sido el más utilizado de manera abrumadora, aunque en los últimos tiempos ha ido perdiendo cuota hasta ser superado por Nginx en 2021. Es un servidor de código abierto, multiplataforma y muy robusto, aunque en ciertos entornos tiene limitaciones de velocidad que pueden hacer que su rendimiento no sea el óptimo.



- ▶ Nginx: es un servidor web de software libre multiplataforma que se ha hecho muy popular debido a que funciona especialmente bien en entornos con un gran número de peticiones simultáneas, con bajo consumo de recursos incluso en momentos de acceso concurrente de múltiples usuarios.



Servidores web

- ▶ Las cuatro aplicaciones que actúan como servidores web más populares en la actualidad son Apache, Nginx, IIS y LiteSpeed, cada una con sus peculiaridades:

- ▶ IIS: servidor web propietario, orientado a sistemas Windows, fundamentalmente dirigido a aplicaciones Microsoft (.NET o ASP)



- ▶ LiteSpeed: este servidor web nació como sustituto de Apache, para mejorar el rendimiento del servidor web en entornos de alto tráfico, ofreciendo un alto rendimiento y una gran escalabilidad. Es compatible con muchas características utilizadas habitualmente en Apache, con lo que la transición desde este servidor web resulta sencilla.



Instalación de un servidor web Apache



- ▶ En la página web de Apache (<https://httpd.apache.org>) podemos encontrar los archivos necesarios para iniciar la instalación, en función del sistema operativo que tengamos. En sistemas Windows el proceso de instalación depende del proveedor de binarios que se escoja, pero suele consistir en únicamente descomprimir los binarios en una carpeta del sistema y ejecutar *httpd.exe* para iniciar el servidor web.
- ▶ Típicamente el servidor web se debería iniciar como servicio, para registrarlo como tal debemos ejecutar el comando *httpd.exe -k install*, y a través del gestor de servicios de Windows podremos arrancarlo, pararlo o reiniciarlo. Para ver si está funcionando correctamente podemos comprobar que está escuchando con el comando netstat o más fácilmente escribiendo en la barra de direcciones del navegador <http://localhost/>

Configuración de un servidor web Apache



- ▶ Todo servidor se configura realizando modificaciones sobre su archivo de configuración. En el caso de Apache ese fichero es [httpd.conf](#). En ese fichero se da valor a las diferentes directivas que nos permiten adecuar el funcionamiento del servidor web a nuestros intereses. Estas directivas pueden aplicar a todo el servidor web si están establecidas directamente en el fichero, o a una parte específica si están entre las siguientes etiquetas:
 - ▶ Directory: estas directivas solo aplican a un directorio determinado.
 - ▶ Files: estas directivas solo aplican a un fichero determinado.
 - ▶ Location: estas directivas solo aplican a una url determinada.
 - ▶ VirtualHost: estas directivas solo aplican a un host virtual determinado.

Apache
HTTP



Principales directivas Apache



- ▶ Algunas de las principales directivas en la configuración de Apache (fichero httpd.conf) son:
 - ▶ DocumentRoot: establece el directorio base donde hay que almacenar las páginas para que el servidor web las encuentre. En este directorio podremos crear más subcarpetas y clasificar así nuestros contenidos. Generalmente es *htdocs*.
 - ▶ ServerName: nombre del servidor.
 - ▶ ServerAlias: permite establecer nombres adicionales para el servidor.
 - ▶ ServerRoot: raíz de instalación de Apache.
 - ▶ Listen: puerto de escucha (típicamente 80).
 - ▶ DirectoryIndex: nombre del archivo índice del directorio. Normalmente es *index.html*.
 - ▶ ErrorDocument: archivo que se muestra cuando ocurre un error.
 - ▶ ErrorLog: ruta al archivo log.
 - ▶ LogLevel: nivel de los mensajes que se almacenan en los ficheros log.

Fichero .htaccess

- ▶ Además del fichero de configuración de Apache, también se pueden incluir opciones especiales de configuración en los ficheros .htaccess, que estarán situados en la ruta donde colocaremos las webs (htdocs por defecto) y que aplican al directorio concreto donde están situados.
- ▶ Se recomienda que la configuración mediante ficheros .htaccess se realice únicamente cuando no tengamos acceso para editar el fichero de configuración de Apache (httpd.conf), ya que en caso de tener acceso es más eficiente configurar las directivas a nivel de directorio.
- ▶ Además, por defecto la directiva AllowOverride de la configuración de Apache (httpd.conf) está establecida a None, por lo que los ficheros .htaccess son ignorados.



Directivas Apache

- Las directivas que hemos visto anteriormente no tienen por qué aplicarse a todo el servidor web, se pueden aplicar únicamente a determinados directorios, determinados ficheros o determinados host virtuales, por ejemplo:

```
<Directory /usr/local/httpd/htdocs>  
    Options Indexes FollowSymLinks  
</Directory>
```

```
<Directory />  
    AllowOverride none  
    Require all denied  
</Directory>
```

```
<Files ".ht*">  
    Require all denied  
</Files>
```

```
<VirtualHost *:80>  
    DocumentRoot "C:/xampp/htdocs"  
</VirtualHost>
```

```
<VirtualHost externalapp.local:80>  
    DocumentRoot "E:/LaravelApps/my_project_name/public"  
    ServerName externalapp.local  
    <Directory "E:/LaravelApps">  
        Require all granted  
    </Directory>  
</VirtualHost>
```

```
<Directory "${SRVROOT}/cgi-bin">  
    AllowOverride None  
    Options None  
    Require all granted  
</Directory>
```

```
<LocationMatch "^/private">  
    Require all denied  
</LocationMatch>
```



Directivas Apache

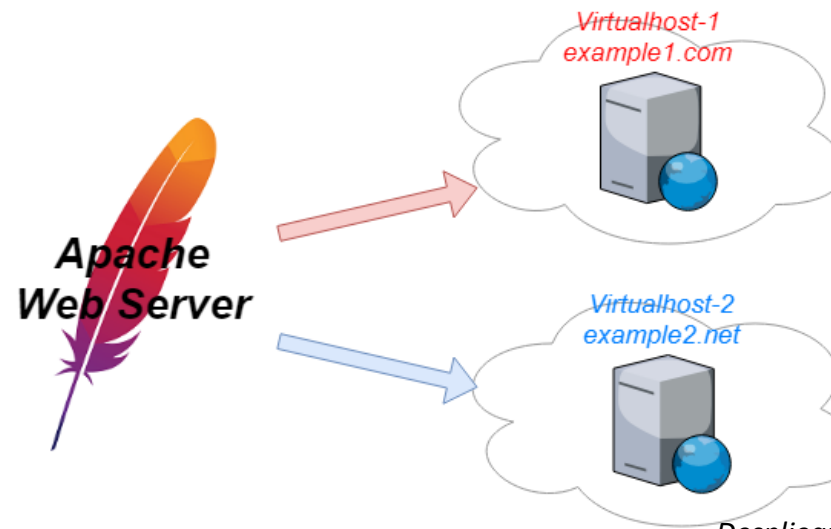


- ▶ Entre las principales directivas que aplican principalmente a directorios, están las siguientes:
 - ▶ *Require all granted*: se permite el acceso de todo el mundo a ese directorio/fichero/url.
 - ▶ *Require all denied*: se deniega el acceso de todo el mundo a ese directorio/fichero/url.
 - ▶ *Options Indexes*: si en la carpeta que solicita el cliente no existe un fichero por defecto, Apache muestra el contenido de la carpeta.
- ▶ Otra directiva bastante utilizada que tiene que ver con directorios aunque no se coloca dentro de las etiquetas de directorio es *Alias*. Permite mapear URLs con rutas del sistema de ficheros, por lo que permite que se acceda a ficheros que están fuera del DocumentRoot.

```
Alias "/image" "/ftp/pub/image"  
<Directory "/ftp/pub/image">  
    Require all granted  
</Directory>
```

Virtual host

- ▶ El término Virtual host en un servidor web hace referencia a la funcionalidad que permite hospedar más de una web (por ejemplo parte1.empresa.com / parte2.empresa.com) en una única máquina.
- ▶ Estos hosts virtuales pueden estar basados en la IP (con lo que tendríamos una IP diferente para cada sitio web) o por nombre de dominio (con lo que tendríamos varios sitios web en una única IP).
- ▶ La directiva en la que se encapsula la configuración de un virtual host en Apache es `<VirtualHost> ... </VirtualHost>`

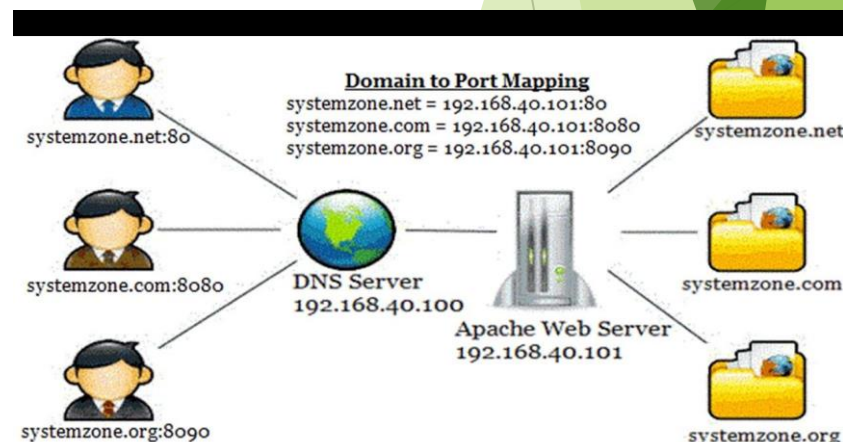
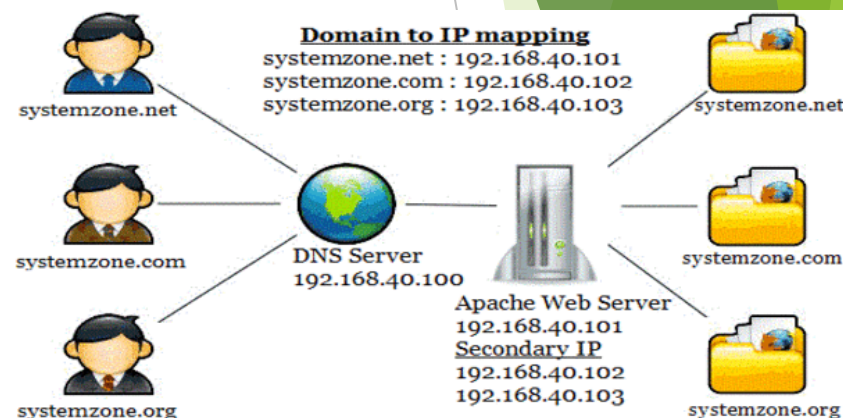


Virtual host basado en IP / puertos

- En los virtual host basados en IP/puertos cada uno de los virtual host se corresponde con una combinación única de IP/puerto. Aunque puede ser útil en un servidor con varios interfaces de red, en la mayoría de las ocasiones lo que queremos es utilizar el sistema basado en nombres.

```
<VirtualHost 172.20.30.40:80>
  ServerAdmin webmaster@www1.example.com
  DocumentRoot "/www/vhosts/www1"
  ServerName www1.example.com
  ErrorLog "/www/logs/www1/error_log"
  CustomLog "/www/logs/www1/access_log" combined
</VirtualHost>

<VirtualHost 172.20.30.50:80>
  ServerAdmin webmaster@www2.example.org
  DocumentRoot "/www/vhosts/www2"
  ServerName www2.example.org
  ErrorLog "/www/logs/www2/error_log"
  CustomLog "/www/logs/www2/access_log" combined
</VirtualHost>
```

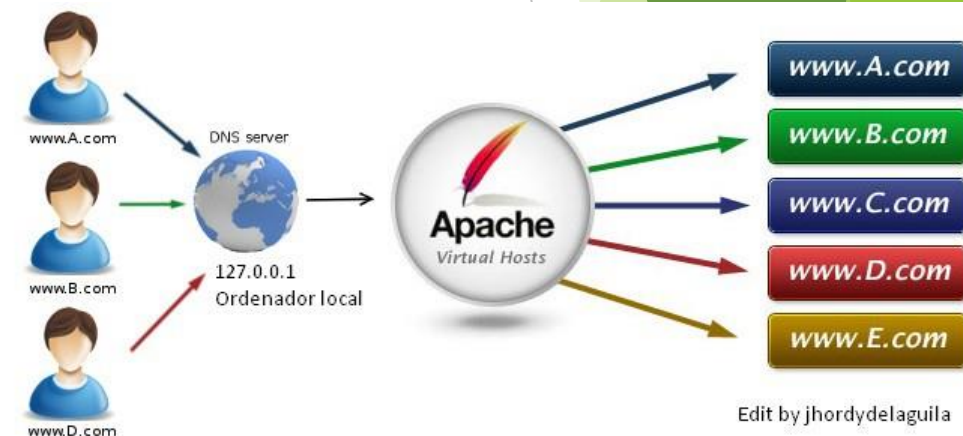


Virtual host basado en nombres

- ▶ En los virtual host basados en IP/nombres cada uno de los virtual host se corresponde con un dominio diferente, con lo que de acuerdo al dominio solicitado por el cliente en la petición http, se servirá un sitio web u otro, estando todos ellos en la misma máquina física.

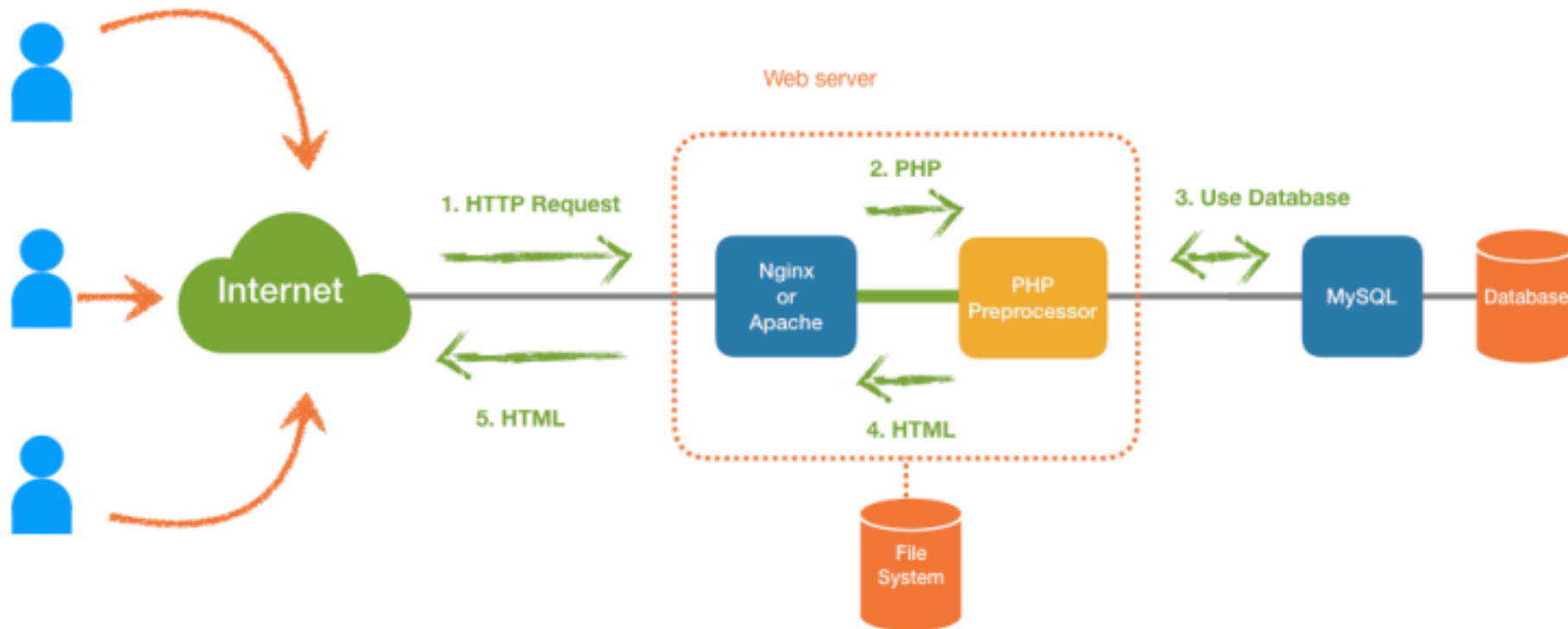
```
<VirtualHost *:80>
# This first-listed virtual host is also the default for *:80
ServerName www.example.com
ServerAlias example.com
DocumentRoot "/www/domain"
</VirtualHost>

<VirtualHost *:80>
ServerName other.example.com
DocumentRoot "/www/otherdomain"
</VirtualHost>
```



Apache + PHP

- Todo lo anterior nos permite tener un servidor web Apache funcionando para servir páginas estáticas, para servir páginas dinámicas debemos integrar un lenguaje como PHP en nuestro servidor web para que haga el preprocesado de las páginas con código PHP que generen el html final que se enviará al navegador web del cliente.



Instalación PHP

- ▶ Desde la página oficial de php (<https://www.php.net/>) puedes obtener los enlaces a las versiones binarias del lenguaje de programación. Es importante descargar una versión de los binarios de PHP compatible con la versión de Apache instalada. La instalación consiste simplemente en descomprimir la carpeta en nuestro disco duro.



- ▶ Una vez descomprimido, debemos renombrar el fichero *php.ini-development* (si no encontramos en un entorno de desarrollo) o *php.ini-production* (entorno de producción) a php.ini, que es el nombre por defecto del fichero de configuración de PHP.
- ▶ Podemos ejecutar php desde línea de comandos, pero lo habitual es que el servidor web sea el encargado de ejecutarlo cuando se encuentre con código PHP incrustado en páginas web.



Configuración de Apache para PHP



- ▶ Para enlazar el servidor web con el intérprete de PHP debemos editar el archivo de configuración de Apache (httpd.conf) añadiendo el módulo de PHP.

php8

- ▶ LoadModule php_module "C:/PHP/php8apache2_4.dll"
- ▶ Añadimos las siguientes líneas de configuración del módulo de PHP para que el intérprete trate los archivos con extensión .php

- ▶

```
<IfModule php_module>  
    PHPINIDir "C:/PHP"  
    AddHandler application/x-httpd-php .php  
</IfModule>
```

- ▶ Por último, añadimos index.php a las páginas por defecto, y reiniciamos el servidor.

- ▶

```
<IfModule dir_module>  
    DirectoryIndex index.html index.php  
</IfModule>
```

Configuración de Apache para PHP



- Para comprobar que la configuración ha sido realizada correctamente, crear un archivo con extensión .php con el siguiente código (es una llamada a una función que nos mostrará una tabla con información sobre el intérprete de PHP instalado) y accede a él mediante el navegador web.

► `<?php`

`phpinfo();`

`?>`

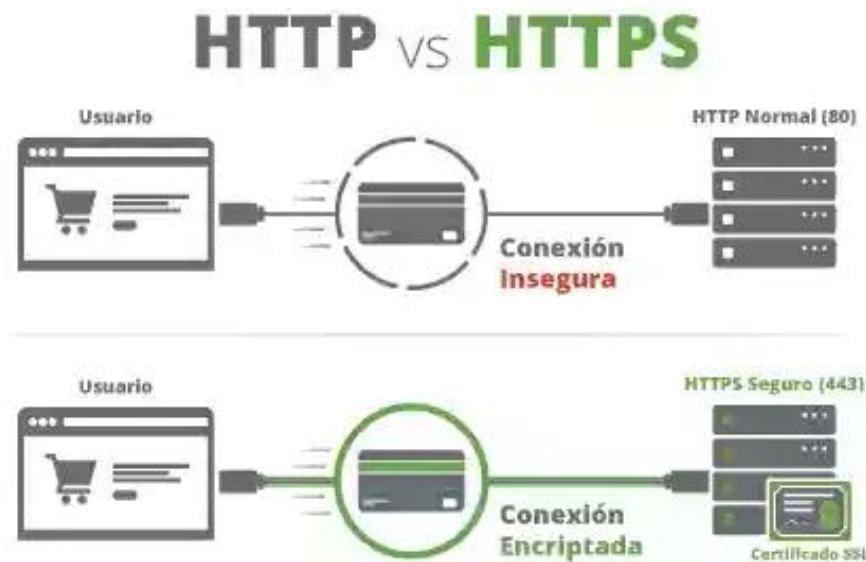
PHP Version 8.0.12	
System	Windows NT WSHMB 6.2 build 9200 (Windows Server 2012 Datacenter Edition) AMD64
Build Date	Oct 19 2021 11:17:33
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:javascript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\instantclientsdk,shared" "--with-oci8-19=..\..\..\instantclientsdk,shared" "--enable-object-out-dir=../obj" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\PHP\php.ini

Certificados (https)

- En la actualidad, la mayoría de los sitios web utilizan conexiones seguras mediante el protocolo https, basadas en SSL (Security Socket Layer).

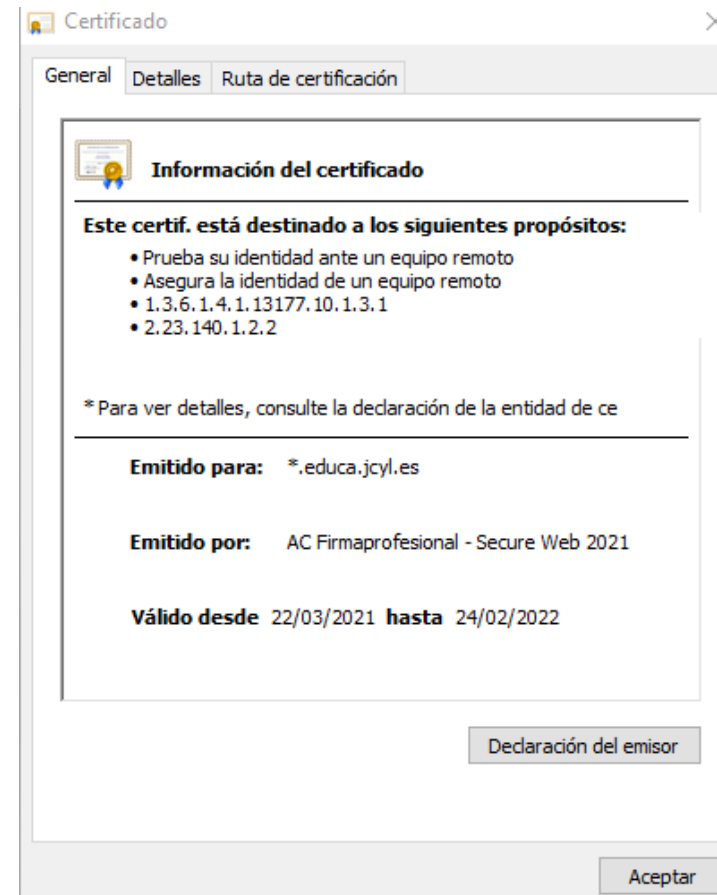
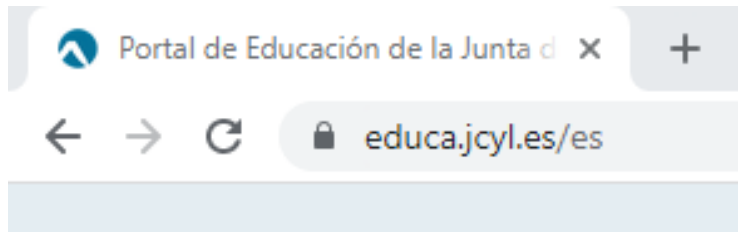


- El método SSL permite establecer un canal seguro para la interacción entre el navegador del cliente y el servidor web a través de una red local o de Internet, en el que la información irá encriptada y protegida.



Certificados (https)

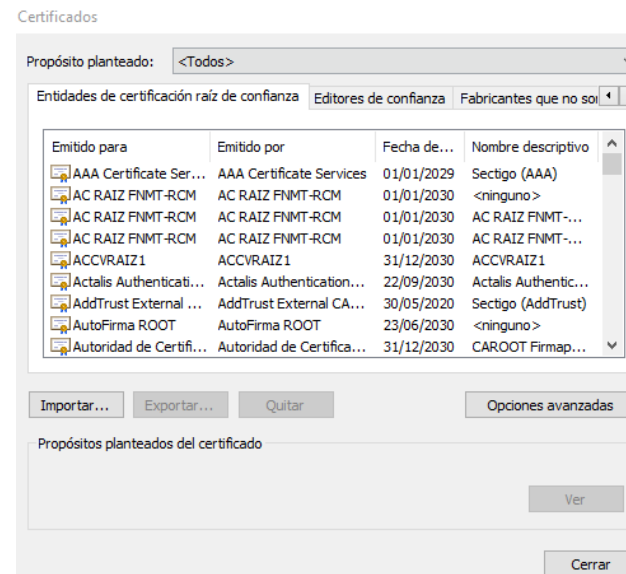
- Los navegadores web suelen informar al usuario de que una conexión a un sitio web es segura mostrando un candado en la barra de direcciones. Si pulsamos el candado podemos ver el certificado digital del sitio web, que es el que tiene la llave criptográfica que cifra la comunicación entre el navegador y el servidor web.



Certificados (https)

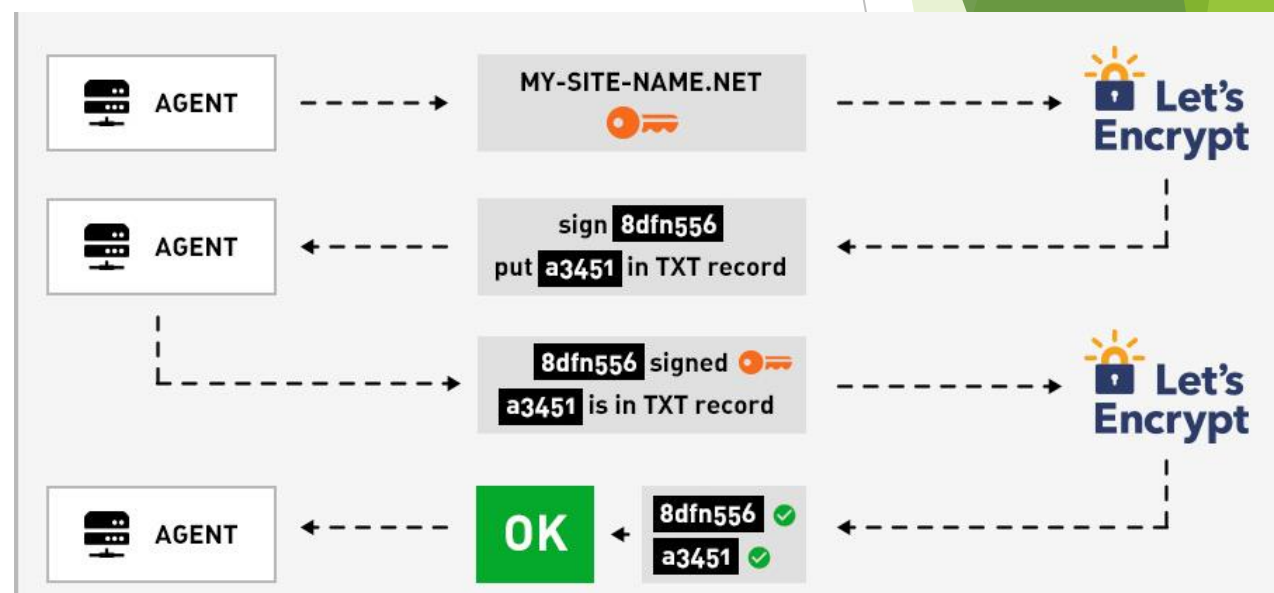


- ▶ Estos certificados utilizados en el servidor web son documentos electrónicos que sirven para comprobar la propiedad del sitio, es decir, confirmar que el sitio web pertenece a quien dice, y no es una página falsa que busca engañar a los usuarios. Los certificados incluyen información sobre la identidad del propietario, y una firma digital de una entidad emisora que verificó que los contenidos del certificado están correctos.
- ▶ Los navegadores web confían en los certificados de varias entidades de certificación de confianza, que podemos ver en los ajustes de cada uno de ellos.



Certificados (https)

- En la actualidad, es sencillo obtener un certificado de confianza gratuito para nuestro sitio web, y la mayoría de plataformas de hosting lo proporcionan a través de Let's Encrypt, una autoridad de certificación gratuita, automatizada y abierta que ha logrado que la mayoría de la web actual sea segura mediante conexiones https.



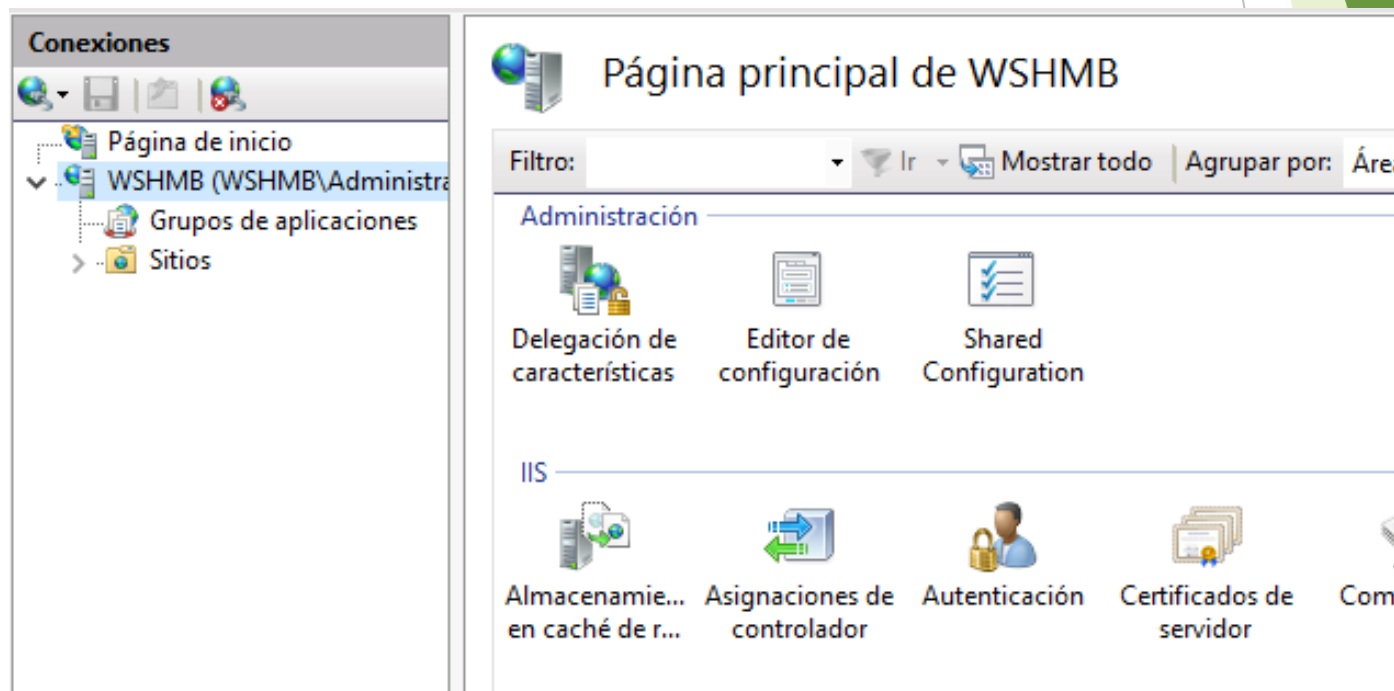
Generar Certificado (https)

- También podemos generar nuestro propio certificado digital e implementarlo en nuestro servidor web con herramientas como OpenSSL, o con la herramienta de generación de certificado del IIS, que es la que probaremos en este ejemplo. El problema con estos certificados es que el navegador web no los reconocerá como certificados de confianza, y mostrará una advertencia al visitante de la página web.



Generar Certificado (https) en IIS

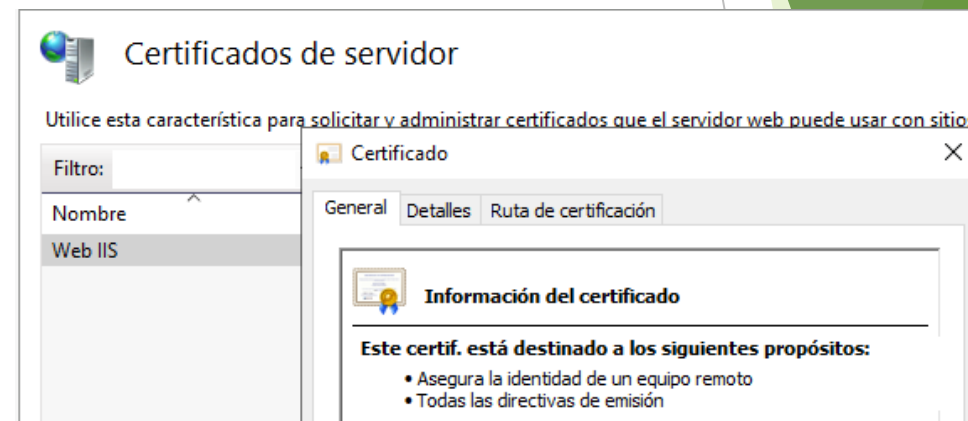
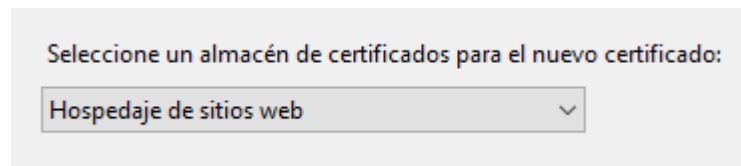
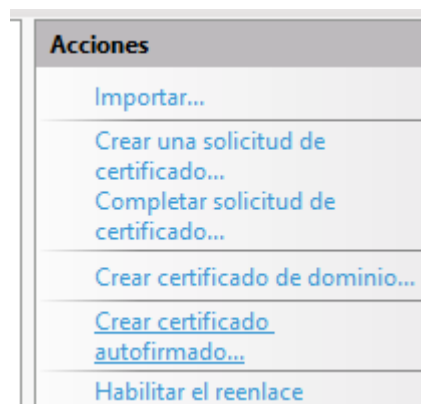
- En nuestra herramienta de Administración de IIS en Windows Server entre las diferentes configuraciones que podemos aplicar a nuestro servidor web está la opción Certificados de servidor. Aquí podremos crear nuestro propio certificado digital o importar para su uso un certificado que hayamos obtenido por otros medios (como Let's Encrypt o una entidad certificadora de confianza).



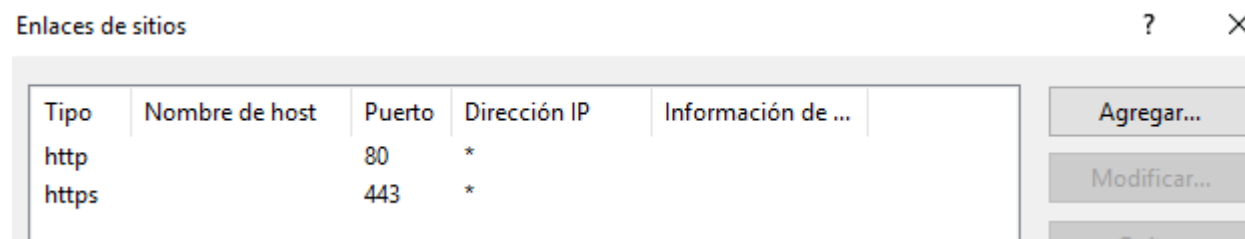
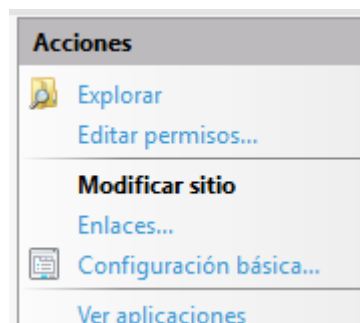
Generar Certificado (https) en IIS



- Para probar https en nuestro servidor podemos crear un certificado auto-firmado, que guardaremos en el almacén de certificados destinados al Hospedaje de sitios web.

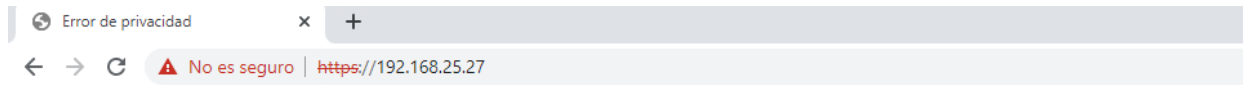


- Posteriormente, dentro de la configuración del sitio web, añadimos en Enlaces la conexión mediante https (puerto 443).



Generar Certificado (https) en IIS

- ▶ De esta manera ya podremos acceder al sitio web de nuestro IIS mediante https, pero el navegador nos advertirá de que el certificado no es de confianza y nos aconsejará no entrar.



La conexión no es privada

Es posible que los atacantes estén intentando robar tu información de **192.168.25.27** (por ejemplo, contraseñas, mensajes o tarjetas de crédito). [Más información](#)

NET::ERR_CERT_AUTHORITY_INVALID

Este servidor no ha podido probar que su dominio es **192.168.25.27**, el sistema operativo de tu ordenador no confía en su certificado de seguridad. Este problema puede deberse a una configuración incorrecta o a que un atacante haya interceptado la conexión.

[Acceder a 192.168.25.27 \(sitio no seguro\)](#)

- ▶ Para que no nos aparezca el error tendremos que exportar el certificado y añadirlo en los ajustes del navegador a la lista de entidades de certificados de confianza.

