

## Cómo instalar Apache+SSL (+Tomcat)

En estas páginas vamos a explicar como añadir al servidor Apache el soporte para SSL (Secure Socket Layer de forma que tengamos un servidor seguro. Hay que dejar claro que este proceso difiere en bien poco de la instalación de Apache en solitario.

Aunque no sea "estrictamente" un tema relacionado con Java, ya que nos lo habéis pedido, y puesto que en la mayoría de las ocasiones Tomcat se usa junto con Apache, vamos a intentar explicar aquí como añadir al servidor Apache la capa de SSL.

Aunque es posible añadir SSL directamente a Tomcat, no es tan sencillo como añadirsele a Apache, y puesto que la comunicación "privada" es entre Apache y el browser, es ahí donde se debe cifrar la información, y Tomcat podrá recibir las peticiones seguras de todas formas.

Existen dos posibilidades para instalar SSL sobre Tomcat. La primera es mediante apache-ssl, desarrollada por la propia fundación apache, que es algo así como una versión parcheada del servidor. La segunda es instalarlo como un módulo cualquiera (como nuestro Tomcat), con el módulo mod\_ssl.

En esta guía de instalación nosotros trataremos la segunda, con mod\_ssl, puesto que es más sencilla de instalar. Es muy habitual que el Apache que instalemos trabaje con varios módulos (php, perl, python, Tomcat, etc.), lo que supone que la forma de instalar SSL mediante el parche pueda hacer que el servidor (o algún módulo) dehe de funcionar si aplicamos el parche en un mal momento.

En la sección de enlaces entontrará las direcciones desde las que descargar todos los programas que serán necesarios a lo largo de esta guía de instalación.

Para cualquier duda o sugerencia no dudes en ponerte en contacto con [nosotros](#).

## Cómo instalar Apache+SSL (+Tomcat) en Linux

1. [Preparacion](#)
2. [Configurando el servidor web Apache](#)
3. [Instalando OpenSSL](#)
4. [Configurando mod\\_ssl](#)
5. [Instalando el servidor web Apache](#)
6. [Comprobando que funciona](#)

### Preparacion

Suponemos que hemos bajado de internet todos los ficheros que necesitamos (servidor apache, mod\_ssl, openssl, y si se quiere los de Tomcat) Suponemos que los guardamos todos en un directorio temporal, por ejemplo /tmp/www/, y tomamos ese directorio como centro de operaciones

```
# cd /tmp/www
```

## Configurando el servidor web Apache

Lo primero que haremos, y puesto que otros pasos así lo requieren, será descomprimir nuestro servidor apache y configurarlo indicando en que directorio lo queremos instalar:

```
# tar xvzf apache_1.3.12.tar.gz
# cd apache_1.3.12
# ./config --prefix=/usr/local/apache
# cd ..
```

## Instalando OpenSSL

Una vez hemos descargado los ficheros de OpenSSL para Linux, lo descomprimos como siempre:

```
# tar xvzf openssl-0.9.5a.tar.gz
# cd openssl-0.9.5a
```

Lo configuramos indicando donde lo queremos instalar (opcion `prefix` del script `config`), lo compilamos y lo instalamos, como cualquier programa Linux que viene en forma de código fuente, nada nuevo:

```
# ./config --prefix=/usr/local/ssl
# make
# make test
# make install
# cd ..
```

## Configurando mod\_ssl

Tenemos que tener cuidado al descargar el fichero que contiene los fuentes de `mod_ssl` y hacerlo del que corresponda con nuestra versión de Apache. Estos ficheros (por ejemplo `mod_ssl-2.6.4-1.3.12.tar.gz`) traen dos números de la serie. El primero (2.6.4) indica la versión de `mod_ssl`, el segundo indica la versión de Apache a la que corresponde (1.3.12).

Rápidamente descomprimos el `mod_ssl` que hemos descargado de internet, y lo configuramos indicándole donde tenemos el código fuente del servidor Apache (que aún no hemos instalado). Por ejemplo:

```
# tar xvzf mod_ssl-2.6.4-1.3.12.tar.gz
# cd mod_ssl-2.6.4-1.3.12
# ./configure --with-apache=../apache_1.3.12
# cd ..
```

## Instalando el servidor web Apache

En este punto es donde podremos añadir a nuestro servidor Apache todos los módulos que queramos (Perl, PHP, Tomcat, etc), de la misma forma que se indica en sus respectivas guías de instalación.

Pero bueno, en esta guía tratamos el modulo SSL y en eso estamos. Primero le indicamos donde hemos descomprimido OpenSSL y despues le indicamos los distintos modulos que queremos usar:

```
SSL_BASE=../openssl-0.9.5a ./configure --prefix=/usr/local/apache \
  --enable-module=ssl --enable-shared=ssl
  (aqui meteriamos el modulo de Tomcat (--enable-module=so), php, etc.)
```

Ya tenemos Apache preparado para instalarlo, asi que lo compilamos:

```
# make
```

Creamos nuestro certificado y llave, contestando un montón de preguntas sobre nuestra "empresa" (recordar que tiene que tener el mismo nombre que nuestro servidor):

```
# make certificate TYPE=custom
```

Y finalmente instalamos el Apache como siempre:

```
# make install
```

Si todo ha ido bien, tendremos al final de la instalación un cuadro indicandonoslo, y mostrando lo que tenemos que escribir para ejecutar el servidor apache "normal" y "seguro".

```
(servidor normal)
# /usr/local/apache/bin/apachectl start

(servidor seguro)
# /usr/local/apache/bin/apachectl startssl
```

Si arrancamos la versión segura nos pedira el password que hallamos dado a nuestra llave SSL, y listo, ya podemos acceder a nuestro servidor seguro desde nuestro browser con la direccion:

```
https://127.0.0.1/
```

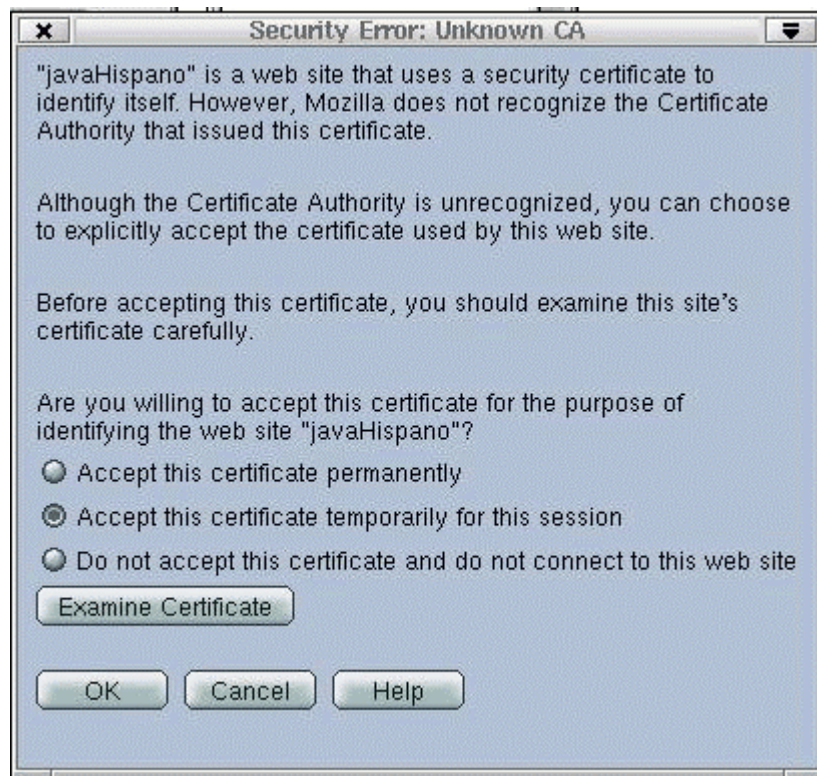
## Comprobando que funciona

Simplemente dirigimos nuestro navegador, en mi caso mi flamante opensource Mozilla 0.9 (descárgalo desde <http://www.mozilla.org>) a la dirección segura de nuestro servidor, como hemos dicho con:

https://127.0.0.1/

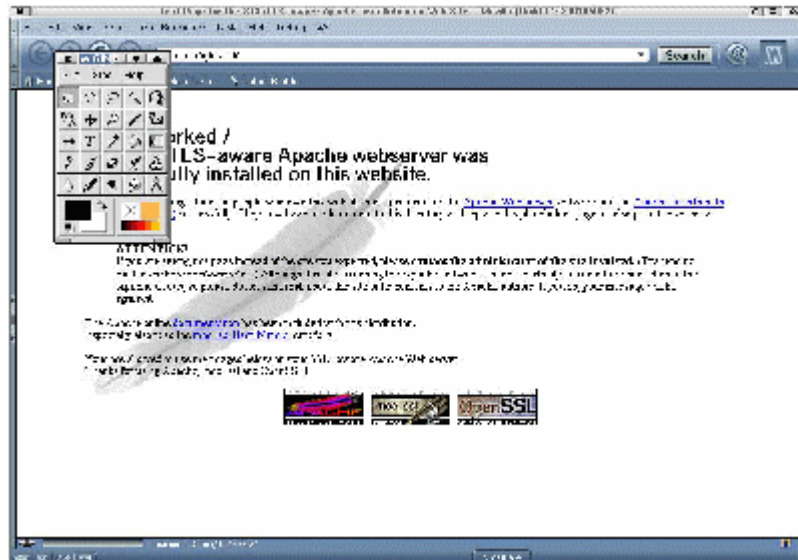
Y recibiremos la siguiente ventana informándonos de la entrada en la "zona segura" y de que el certificado no esta aprobado por ningún organismo oficial, por lo que podría no ser válido.

NOTA: posiblemente, si no eres un raro como yo, y vives en España, el texto te aparecerá en otro idioma, seguramente en español, pero me temo que mi Linux habla inglés ;-). Si alguien tiene ganas, que me mande la versión en castellano de estas dos ventanas.



Aceptamos el certificado (lo examinamos si queremos), y nos sale la página de presentación de Apache con la información de SSL.

NOTA: la misma página sale aunque hallas arrancado el servidor normal, asi que no te asustes. Tienes que fijarte en la direccion indicada para llegar a ella. Perdonar por la baja calidad, pero una imagen de 1024x768 era demasiado grande ;-).



NOTA: vale!!, ya se que se me ha colado una ventana del **Gimp** en el pantallazo, pero cuando me di cuenta no tenia tiempo para cambiarlo :-).

## Cómo instalar Apache+SSL (+Tomcat) en Windows

1. [Palabras previas](#)
2. [Instalando el servidor web Apache](#)
3. [Creando nuestro certificado](#)
4. [Configurando Apache con mod\\_ssl](#)
5. [Comprobando que funciona](#)

### Palabras previas

Quede dicho de antemano, que hablar de un servidor seguro en Windows puede ser un tema de broma, conocidas las limitaciones en cuanto a cuestiones de seguridad de dicho sistema operativo. Además el servidor Apache en su versión Windows esta considerado como "de calidad beta", por lo que no es recomendable su uso en ese sistema más allá que para un test del sistema.

Asi pues, si realmente necesitas un servidor web opensource y además seguro, considera pasarte a la versión GNU/Linux.

### Instalando el servidor web Apache

Para Windows podemos descargar directamente la última versión de Apache en formato autoextraíble (exe), o en .msi (En estos momentos está la versión 1.3.19 en [http://httpd.apache.org/dist/binaries/win32/apache\\_1.3.19-win32-no\\_src-r2.msi](http://httpd.apache.org/dist/binaries/win32/apache_1.3.19-win32-no_src-r2.msi), visite <http://httpd.apache.org> para ver cual es la versión más reciente). Para que Windows reconozca estos ficheros deberá tener instalado el Windows Installer. El programa de instalación es exactamente igual al resto de los programas Windows, es decir, le preguntará donde quiere instalar el servidor, el tipo de instalación, etc.

Una vez se ha terminado la instalación, como siempre, deberemos configurar un par de cosas. Para ello editamos el archivo `PATH_APACHE/conf/httpd.conf` y buscamos y editamos las propiedades `ServerName`, `Port` y `Listen`. En la primera ponemos el nombre de nuestra maquina, en la segunda, el puerto SSL, normalmente el 443, y en la segunda (indica donde esperará peticiones el servidor), ponemos dos valores, el puerto 443 para peticiones seguras, y el 80 para peticiones normales (o los valores que queramos usar).

```
. . .
ServerName javahispano
. . .
Port 443
. . .
Listen 80
Listen 443
. . .
```

Ahora podemos arrancar el servidor y probar que funciona con los dos puertos con las direcciones

```
http://127.0.0.1:80/
http://127.0.0.1:443/
```

En ambos casos debemos obtener la página de bienvenida de Apache.

A partir de este punto (también podemos hacerlo al final del proceso) es cuando podemos instalar los módulos de Apache que queramos, para Tomcat, para php, etc, puesto que nuestro servicio SSL será un módulo más.

## Creando nuestro certificado

Una vez hemos obtenido la versión de `mod_ssl` que corresponde a nuestro Apache de la dirección o ftp de `mod_ssl` (<ftp://ftp.modssl.org/contrib/>) (cada fichero indica la versión de Apache a la que corresponde, así como la versión de `mod_ssl` y la versión de `openssl` que **incluye**) los descomprimos en un directorio temporal.

Importante: Podemos ver que en su versión binaria Windows, `mod_ssl` incluye ya `openssl`. No hay que buscarlo y bajarlo de ningún sitio. En la carpeta donde lo habeis descomprimido, encontrareis un directorio llamado `openssl` que incluye las librerías `dll` y archivos necesarios.

Copiamos entonces los ficheros `ssleay32.dll` y `libeay32.dll` al directorio `System32` de nuestro Windows, ya sea `c:\windows\system32` para Windows 9x/ME o `c:\winnt\system32` para Windows NT y Windows 2000.

Ahora necesitamos un fichero de configuración de `openssl`, si no os viene con el `mod_ssl`, lo podeis descargar de [aquí mismo](#). Lo descargais y lo poneis en la misma carpeta donde esta el programa `openssl.exe` (algo así como `directorio_descrompresion\openssl\bin\`).

Después de eso podeis hacer doble click sobre el fichero `openssl.exe`. Vereis que os aparece una ventana de línea de comandos (la típica negra con letras blancas) con el `PROMPT openssl>`. Es el interfaz de `openssl` para crear nuestro certificado. Así que lo vamos creando con los siguientes comandos:

NOTA: es importante que el certificado tenga el mismo nombre que nuestra máquina, en nuestro caso "javahispano", pero deberias cambiar ese nombre por el hallais puesto al configurar Apache.

```
req -config openssl.cnf -new -out javahispano.csr

// aqui vamos contestando las preguntas típicas (nombre, organizacion,
etc.)

rsa -in privkey.pem -out javahispano.key

x509 -in javahispano.csr -out javahispano.cert -req -signkey
javahispano.key -days 365
//aquí podemos cambiar la opción "days" por otro valor.
//este indica que nuestro certificado caduca en un año.

quit //para cerrar el programa openssl
```

Ya tenemos nuestro certificado (aunque no tiene valor), así que copiamos los ficheros `javahispano.cert` y `javahispano.key` a algún sitio donde los encuentre Apache. Lo ideal será crear un subdirectorio llamado `ssl` dentro de la carpeta `PATH_APACHE/conf/` y meterlos allí.

## Configurando Apache con `mod_ssl`

Lo primero que hacemos es copiar '**TODOS**' (leer el párrafo siguiente antes de hacerlo) los ficheros que venían con nuestro paquete `mod_ssl` en el directorio donde tenemos instalado Apache. Algunos se sobrescribirán, pero no importa, aunque luego quizás tengamos que retocar algunos parámetros de nuestro servidor.

En las versiones modernas el `mod_ssl` incluye un fichero `conf/httpd.conf`, por lo que lo indicado al principio de los puertos y el nombre del servidor se hace necesario de nuevo. Mala suerte, pero es importante hacerlo al principio para saber que el servidor Apache funciona correctamente antes de instalar SSL. Algunos otros parámetros que tendremos que reconfigurar son `ROOT` con el directorio de la instalación del servidor Apache (nombrado en otras partes de este artículo como `PATH_APACHE`), teniendo en cuenta que hay que usar `/` en lugar de `\`, el directorio de usuarios `UserDir` (comentarlo para no usarlo en Windows), el de documentos `DocumentRoot` y un poco más abajo `Directory` el valor algo así como `PATH_APACHE/htdocs`, y por supuesto lo referente a la configuración de otros módulos, como por ejemplo Tomcat (carga de `mod_jk` y inclusión de su configuración).

No he probado a no copiar este fichero (lo copié antes de saber las consecuencias) y cambiar las cosas necesarias manualmente, pero en fin, si alguien lo hace que me cuente si funciona. Yo os recomiendo que lo intentéis antes así!, aunque el *desastre* no es irreparable tampoco hay porque pegarse un tiro en el pie, ¿no?.

Ahora volvemos a editar el fichero `PATH_APACHE/conf/httpd.conf` para añadirle unas cuantas cosas.

En la sección `LoadModule` añadimos (o simplemente descomentamos, porque seguramente ya esta puesto):

<http://www.javahispano.com>

```
. . .  
LoadModule ssl_module modules/ApacheModuleSSL.dll  
# o según corresponda, uno de los dos  
LoadModule ssl_module modules/ApacheModuleSSL.so  
. . .
```

Nota: En otras versiones de Apache + mod\_ssl puede tener otro nombre. **Victor** nos informa que para la versión 1.3.19 el archivo se llama **mod\_ssl.so**.

Y al final del fichero añadimos:

```
# ver http://www.modssl.org/docs/2.4/ssl\_reference.html  
# para más información.  
SSLMutex sem  
SSLRandomSeed startup builtin  
SSLSessionCache none  
  
SSLLog logs/ssl.log  
SSLLogLevel info  
  
<Virtualhost javahispano:443>  
SSLEngine on  
SSLCertificateFile conf/ssl/javahispano.cert  
SSLCertificateKeyFile conf/ssl/javahispano.key  
</Virtualhost>
```

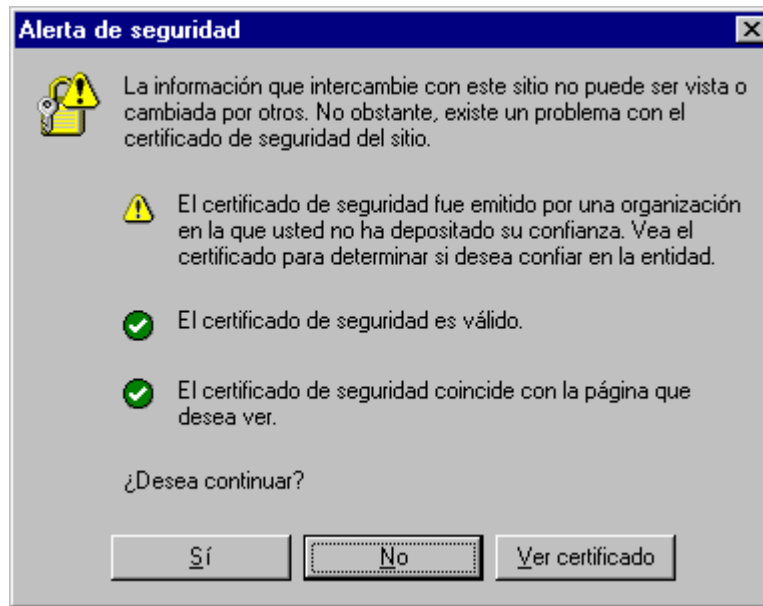
Ya está todo. Fácil, ¿no?

## Comprobando que funciona

Simplemente dirigimos nuestro navegador a la dirección segura de nuestro servidor, en mi caso

```
https://127.0.0.1/
```

Y recibiremos la siguiente ventana informándonos de la entrada en la "zona segura" y de que el certificado no está aprobado por ningún organismo oficial, por lo que podría no ser válido.



Podemos examinar el certificado pulsando sobre el tercer botón, algo así como "mostrar certificado". Y nos saldrá la información que introdujimos al crearlo.



NOTA: mis agradecimientos a **Victor** por las imágenes en castellano.

NOTA: al hacer esta revisión he visto que existe en las nuevas versiones de mod\_ssl existe un fichero casi igual que este pero en inglés. Que conste que es solo cuestión de que sólo hay un proceso y que no conocía dicho fichero, así que no se puede considerar a esto una traducción.

## Enlaces

- La página del servidor web Apache: <http://httpd.apache.org>
- La página del apache+ssl: <http://www.apache-ssl.org>
- La página de mod-ssl: <http://www.mod-ssl.org>
- El ftp de mod-ssl (la página falla en ocasiones): <ftp://ftp.modssl.org/contrib/>
- La página de openssl: <http://www.openssl.org>

**Alberto Molpeceres** es ahora mismo desarrollador de aplicaciones en ámbito cliente/servidor para la empresa **T-Systems - debis Systemhaus** en Munich (Alemania). Cuando no está trabajando o "metiendo caña" al resto de los integrantes de javaHispano intenta pasear con su novia, buscar la desaparecida lógica del idioma alemán o intenta olvidar la *pesadilla* que es buscar piso en Munich. Para cualquier duda o tirón de orejas, e-mail a: [al@javahispano.com](mailto:al@javahispano.com)