



**Universidad de Valladolid**

**Escuela de Ingeniería Informática**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática

(Mención Tecnologías de la Información)

**Manual de instalación para el  
administrador.**

Autor:

**Dña. Lydia Ruiz Miravalles**

## Tabla de contenido

1. INSTALACIÓN EN MACOS SIERRA .....	3
2. INSTALACIÓN EN LINUX, UBUNTU.....	7
3. MIGRACIÓN DE LA WEB DE MACOS A LINUX.....	11
4. INSTALACIÓN CERTIFICADO SSL.....	12

# 1. Instalación en MacOS Sierra

A continuación, se explica como se ha llevado a cabo la instalación en el sistema MacOS Sierra:

Para la instalación del servidor Apache debemos introducir en la consola del sistema los siguientes comandos:

```
sudo apachectl start
```

La ruta que tendrá este servidor por defecto en el navegador es:  
<http://localhost/>

A continuación, se crea la carpeta "Sites" en el directorio raíz. Editamos el fichero de usuarios de apache:

```
sudo vi /etc/apache2/users/username.conf
```

Sí no existe este fichero, se crea uno nuevo con este nombre. Añadimos las siguientes líneas en este fichero cambiando el "username" por nuestro nombre de usuario:

```
<Directory "/Users/username/Sites/">  
AllowOverride All  
Options Indexes MultiViews FollowSymLinks  
Require all granted  
</Directory>
```

Después le damos los permisos necesarios al fichero con el comando:

```
sudo chmod 644 /etc/apache2/users/username.conf
```

A continuación, modificamos el fichero de configuración de http para permitir algunos módulos:

```
sudo vi /etc/apache2/httpd.conf
```

Y en ese fichero descomentamos o añadimos, si no existen, las siguientes líneas:

```
LoadModule authz_core_module libexec/apache2/mod_authz_core.so  
LoadModule authz_host_module libexec/apache2/mod_authz_host.so  
LoadModule userdir_module libexec/apache2/mod_userdir.so  
LoadModule include_module libexec/apache2/mod_include.so  
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
```

```
LoadModule php5_module libexec/apache2/libphp5.so
Include /private/etc/apache2/extra/httpd-userdir.conf
```

Modificamos otro fichero de configuración de apache:

```
sudo vi /etc/apache2/extra/httpd-userdir.conf
```

Y descomentamos la siguiente línea:

```
Include /private/etc/apache2/users/*.conf
```

Reiniciamos el sistema con:

```
sudo apachectl restart
```

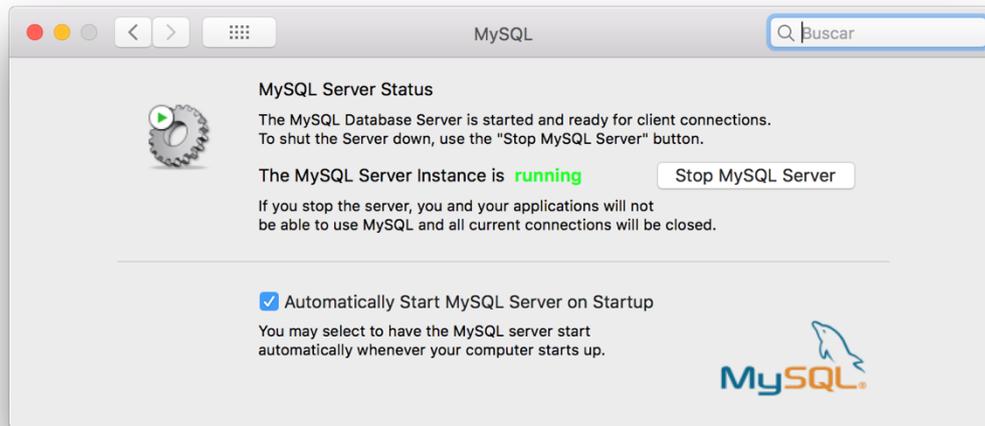
Comprobamos que funciona en la dirección: `http://localhost/username/`

Una vez que tengamos el servidor Apache funcionando, pasamos a **instalar PHP**.

Para instalar PHP únicamente tendremos que cambiar la configuración de apache del fichero que indicamos a continuación, descomentar la línea indicada y reiniciar el servidor:

```
sudo vi /etc/apache2/httpd.conf
LoadModule php5_module libexec/apache2/libphp5.so (descomentar
esta línea)
sudo apachectl restart
```

El siguiente paso será **instalar MySQL**. Para esto únicamente tendremos que descargarlo de la web oficial de MySQL y seguir la instalación que indica. Después desde las preferencias del sistema pincharemos en MySQL y después en Start. Una vez en funcionamiento obtendremos lo siguiente:



*Ilustración 1. Pantalla MySQL Server Status en MacOS.*

Una vez instalado configuramos el PATH del sistema para poder llamar a MySQL sin escribir la ruta completa. Introducimos los siguientes comandos en la consola:

```
cd ; vi .bash_profile
```

Y escribimos:

```
export PATH="/usr/local/mysql/bin:$PATH"
```

Después ejecutamos:

```
source ~/.bash_profile
```

Una vez hecho esto ya podremos acceder a la base de datos directamente con el comando:

```
mysql -u root -p
```

Ya tenemos funcionando apache, php y mysql. Ahora **instalaremos phpMyAdmin**. Para ello ejecutamos los comandos:

```
sudo mkdir /var/mysql  
sudo ln -s /tmp/mysql.sock /var/mysql/mysql.sock
```

Descargamos, de la pagina web de phpMyAdmin, el zip correspondiente. Creamos la carpeta de configuración y le damos permisos:

```
mkdir ~/Sites/phpmyadmin/config  
chmod o+w ~/Sites/phpmyadmin/config
```

Una vez aquí debemos ir al navegador, a la dirección:  
http://localhost/~username/phpmyadmin/setup/ ó http://localhost/phpmyadmin/setup/

Aquí debemos configurar el servidor donde se encuentra la base de datos que queremos tener en este administrador. Para ello pinchamos en la opción “Nuevo servidor”:

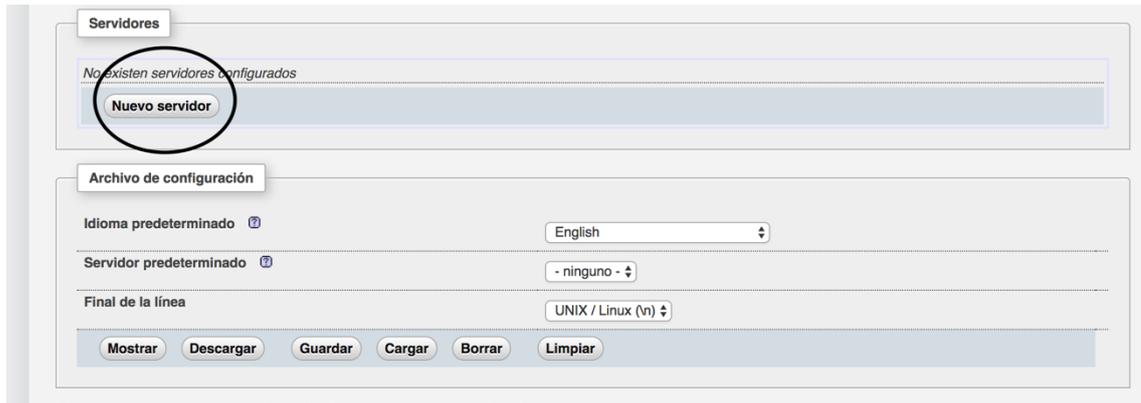


Ilustración 2. Configuración phpMyAdmin 1.

Vamos a la pestaña de Autenticación y añadimos el usuario y la contraseña. Hecho esto damos al botón aplicar y después a guardar.

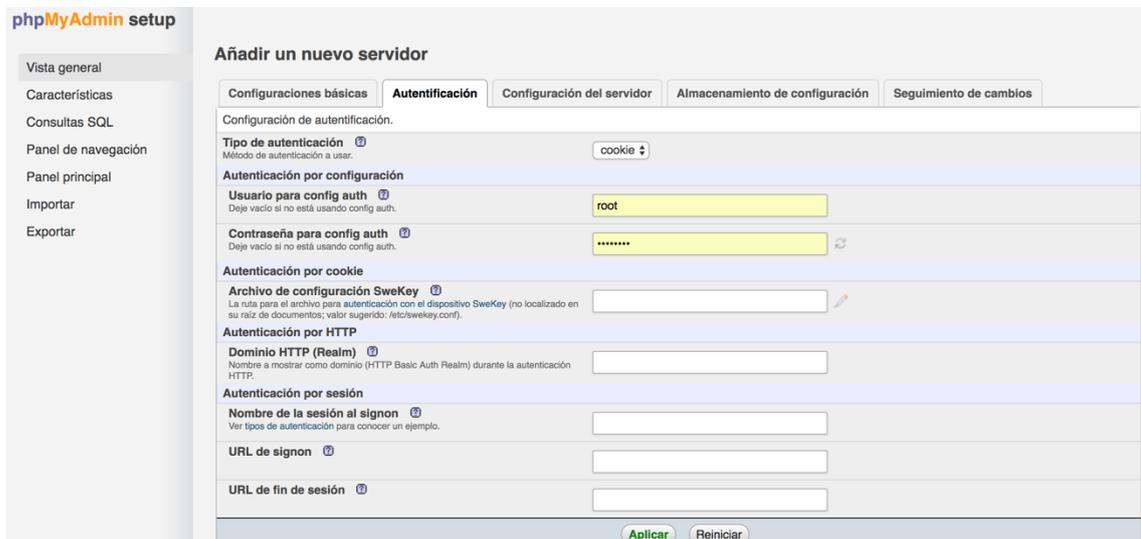


Ilustración 3. Configuración phpMyAdmin 2.

Para finalizar movemos el archivo “config.inc.php” al directorio /phpmyadmin y elimina el directorio /config dado que estará vacío. Para acceder a phpmyadmin se hará a través de la dirección:  
http://localhost/~username/phpmyadmin/.

## 2. Instalación en Linux, Ubuntu.

Primero, instalamos el **servidor de Apache** mediante los comandos:

```
sudo apt -get update  
sudo apt -get install apache2
```

Configuramos el ServerName en el fichero de configuración de apache:

```
sudo vi /etc/apache2/apache2.conf
```

Al final del documento añadimos la línea:

```
ServerName virtual.lab.inf.uva.es:20082
```

Revisamos la sintaxis:

```
sudo apache2ctl configtest
```

A lo que obtenemos: Syntax OK

Reiniciamos Apache para fijar los cambios:

```
sudo systemctl restart apache2
```

Comprobamos que el servidor funciona accediendo desde un navegador a la dirección <http://virtual.lab.inf.uva.es:20082/> y obtenemos esto:

**Apache2 Ubuntu Default Page**

ubuntu

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
|

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

**Document Roots**

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, `public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

**Reporting Problems**

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Ilustración 4. Página Apache2 Ubuntu Default Page.

Una vez que tenemos Apache, pasamos a **instalar PHP**. Para ello introducimos el comando:

```
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

Configuramos el gestor de directorios de apache:

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

Dejándolo de la siguiente manera:

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl
    index.xhtml index.htm
</IfModule>
```

Guardamos los cambios reiniciando apache:

```
sudo systemctl restart apache2
```

Comprobamos el estado del servidor con el comando:

```
sudo systemctl status apache2
```

Vemos que este es correcto:

```
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since mar 2018-05-22 15:35:06 CEST; 41min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 21671 ExecReload=/etc/init.d/apache2 reload (code=exited, status=0/SUCCESS)
    Tasks: 6
   Memory: 18.0M
      CPU: 561ms
   CGroup: /system.slice/apache2.service
           └─17739 /usr/sbin/apache2 -k start
             └─21706 /usr/sbin/apache2 -k start
               └─21707 /usr/sbin/apache2 -k start
                 └─21708 /usr/sbin/apache2 -k start
                   └─21709 /usr/sbin/apache2 -k start
                     └─21710 /usr/sbin/apache2 -k start

may 22 15:35:06 virtual apache2[17722]: *
may 22 15:35:06 virtual systemd[1]: Started LSB: Apache2 web server.
may 22 15:41:36 virtual systemd[1]: Reloading LSB: Apache2 web server.
may 22 15:41:36 virtual apache2[19070]: * Reloading Apache httpd web server apache2
may 22 15:41:36 virtual apache2[19070]: *
may 22 15:41:36 virtual systemd[1]: Reloaded LSB: Apache2 web server.
may 22 15:43:19 virtual systemd[1]: Reloading LSB: Apache2 web server.
may 22 15:43:19 virtual apache2[21671]: * Reloading Apache httpd web server apache2
may 22 15:43:19 virtual apache2[21671]: *
may 22 15:43:19 virtual systemd[1]: Reloaded LSB: Apache2 web server.
```

*Ilustración 5. Resultado comando "systemctl status apache2".*

Para comprobar que todo es correcto creamos un fichero en la dirección con el siguiente código:

```
sudo vi /var/www/html/info.php
```

```
<?php
phpinfo();
?>
```

Después vamos en el navegador a la dirección

<http://virtual.lab.inf.uva.es:20082/info.php>

Y obtenemos el siguiente resultado:



System	Linux virtual 4.4.0-124-generic #148-Ubuntu SMP Wed May 2 13:00:18 UTC 2018 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqld.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/15-xml.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gd.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mbstring.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-simplexml.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini, /etc/php/7.0/apache2/conf.d/20-wddx.ini, /etc/php/7.0/apache2/conf.d/20-xmlreader.ini, /etc/php/7.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.0/apache2/conf.d/20-xsl.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled

Ilustración 6. Página de información de PHP.

Para la **instalación de MySQL** :

```
sudo apt-get install mysql-server
```

Seguimos los pasos que nos aparecen para cambiar la contraseña del usuario root. Ejecutamos el siguiente comando para comprobar que funciona correctamente:

```
mysql -u root -p
```

Una vez tenemos la base de datos, instalamos el administrador phpmyadmin:

```
sudo apt-get update
sudo apt-get install phpmyadmin
```

Para crear el acceso vía web a ello tenemos que mover la carpeta al directorio de acceso web:

```
sudo mv /usr/share/phpmyadmin/ /var/www/html
```

Para acceder a ello utilizamos la dirección <http://virtual.lab.inf.uva.es:20082/phpmyadmin> y escribimos el usuario (normalmente root) y la contraseña de acceso de MySQL.

### 3. Migración de la web de MacOS a Linux

Este proyecto inicialmente fue creado en un sistema MacOS y más adelante decidió migrarse a una máquina virtual de Ubuntu. Por tanto, se tuvo que pasar tanto los scripts con la programación de la web como el contenido de la base de datos.

Para pasar el contenido de los ficheros simplemente se utilizó el comando scp de los sistemas Unix:

```
scp -P [puerto] -r carpeta dirMaquinaVirtual:dirCopia
```

La migración de la base de datos se hace muy sencilla gracias a phpMyAdmin. En este gestor, en el sistema de partida, el macOS, seleccionamos la opción exportar:

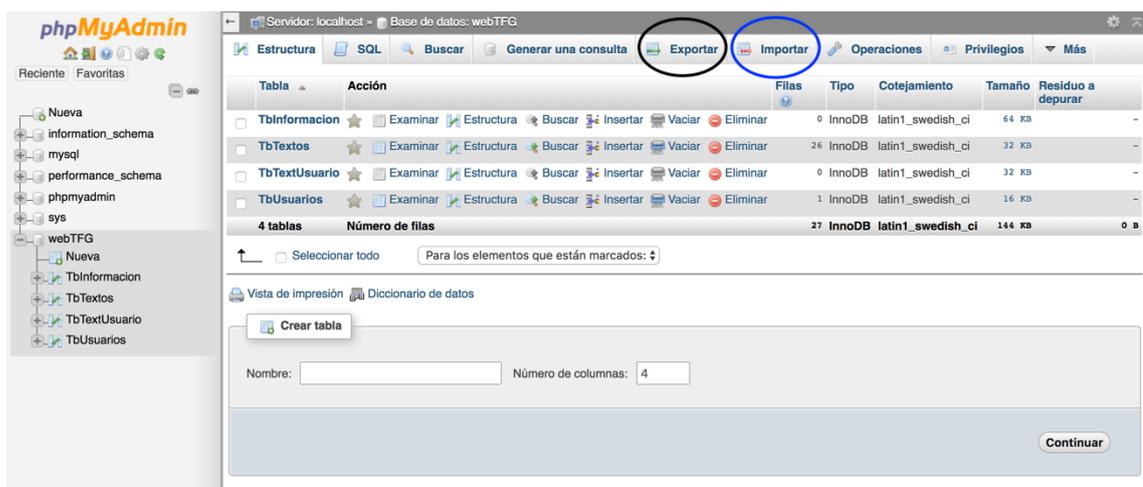


Ilustración 7. Migración de base de datos a través de phpMyAdmin.

Le indicamos el nombre, el método de exportación rápido y el formato SQL.

*Ilustración 8. Exportación base de datos con phpMyAdmin.*

Esto nos generara un fichero con extensión .sql. En la máquina destino tenemos dos opciones para importarlo:

- a) Desde phpMyAdmin con la opción importar.
- b) Desde la terminal, con el comando scp explicado anteriormente enviamos el archivo, abrimos mysql y ejecutamos:

```
source [direccionFichero.sql];
```

## 4. Instalación certificado SSL.

En este apartado se explicará las configuraciones en el servidor para obtener el servidor web con Certificado SSL.

En primer lugar, debemos actualizar el servidor con los siguientes comandos:

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

A continuación, activamos el módulo SSL:

```
sudo a2enmod ssl
```

```
sudo service apache2 restart
```

Creamos el certificado SSL autofirmado en una nueva carpeta:

```
sudo mkdir /etc/apache2/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

Configuramos el servidor Apache para el uso de SSL, para ello o bien modificamos el fichero “/etc/apache2/sites-available/default-ssl.conf” o bien lo eliminamos y creamos uno nuevo con el mismo nombre con el siguiente contenido:

```
rm /etc/apache2/sites-available/default-ssl.conf
```

```
sudo vi /etc/apache2/sites-available/default-ssl.conf
```

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
ServerName virtual.lab.inf.uva.es:20082
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
SSLEngine on
# SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
# SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key

<FilesMatch "\.(cgi|shtml|phtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>

<Directory /usr/lib/cgi-bin>
SSLOptions +StdEnvVars
</Directory>

BrowserMatch "MSIE [2-6]" \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0
BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
</VirtualHost>
</IfModule>
```

*Ilustración 9. Contenido fichero "/etc/apache2/sites-available/default-ssl.conf".*

Por último, activamos el host virtual SSL y reiniciamos el servidor:

```
sudo a2ensite default-ssl.conf
```

```
sudo service apache2 restart
```