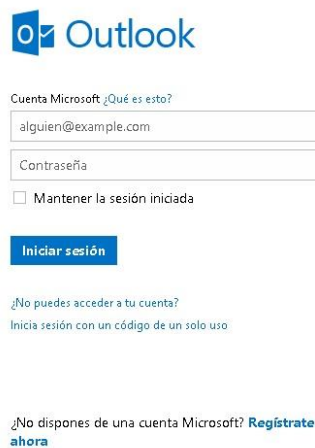
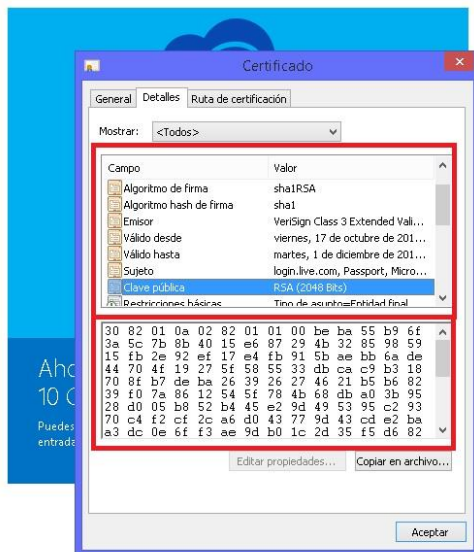
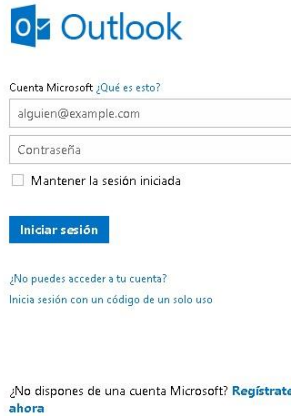
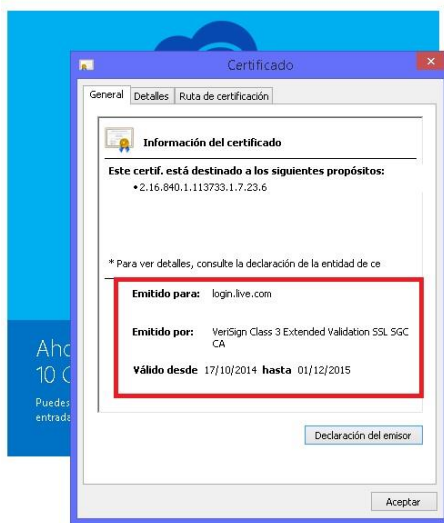


# Protocolo HTTP

## BLOQUE 1: ACTIVIDAD CON NAVEGADORES

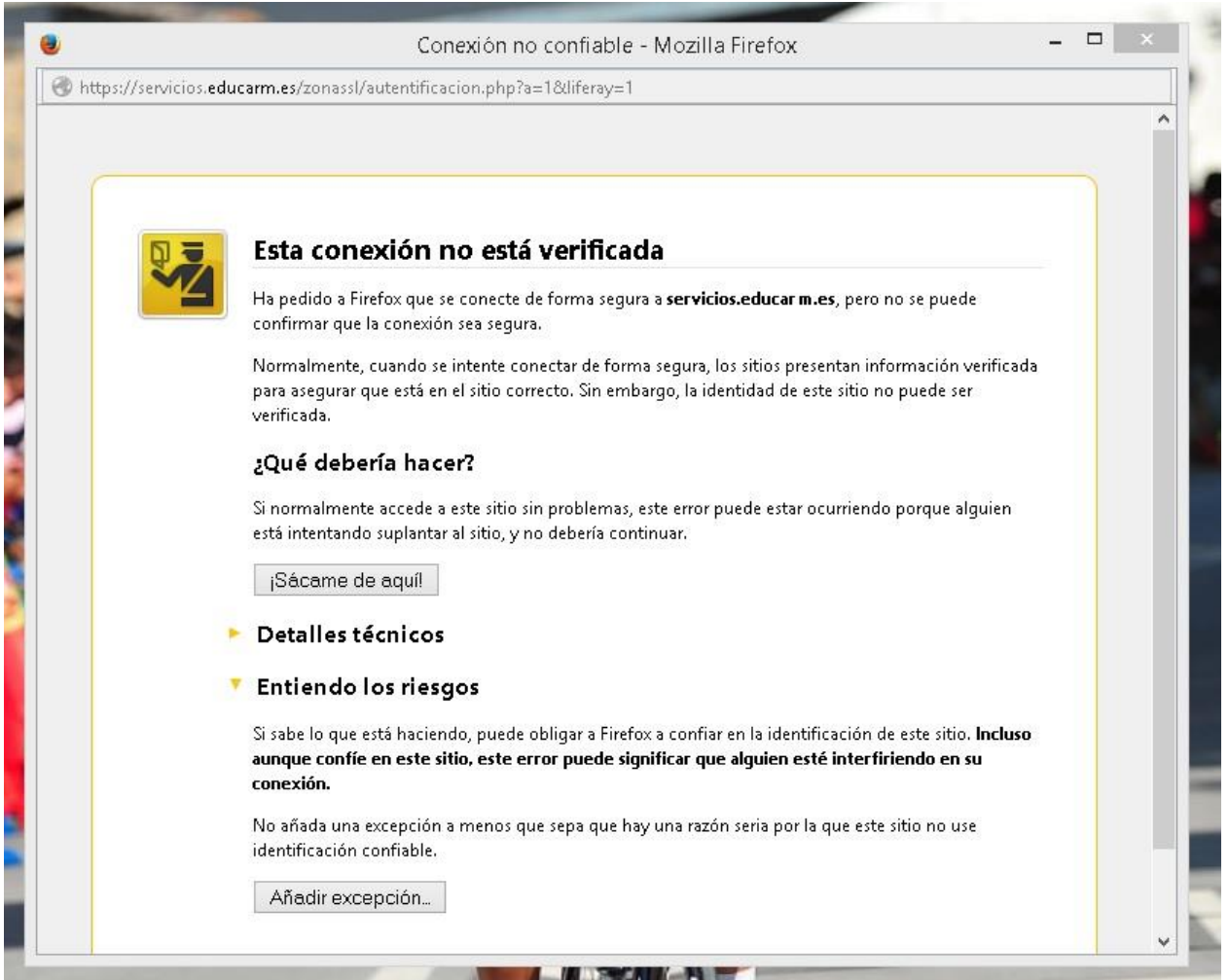
**1º Visita una Web que utilice HTTPS y comprueba los siguientes datos: Para quién se emitió el certificado, CA, algoritmo, fecha de caducidad y clave pública. Entregar datos obtenidos y pantallas de capturas**



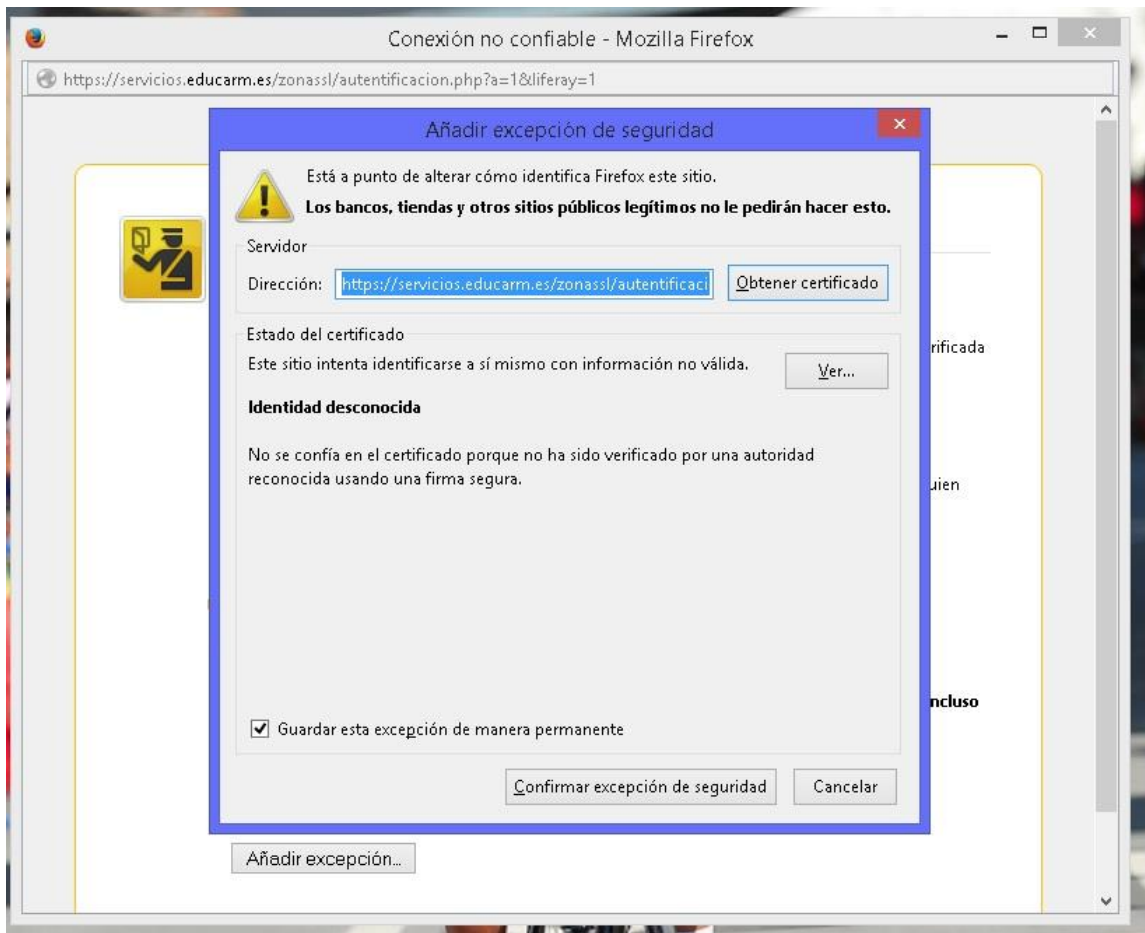
En las capturas podemos ver los datos obtenidos, clave pública, CA, algoritmo, etc...

**2º Instala el certificado de una Web segura, visita la Web y comprueba que no vuelve a pedirte autorización, borra el certificado y visítala de nuevo. (Windows y Linux). Entregar el proceso seguido paso a paso y con pantallas de capturas**

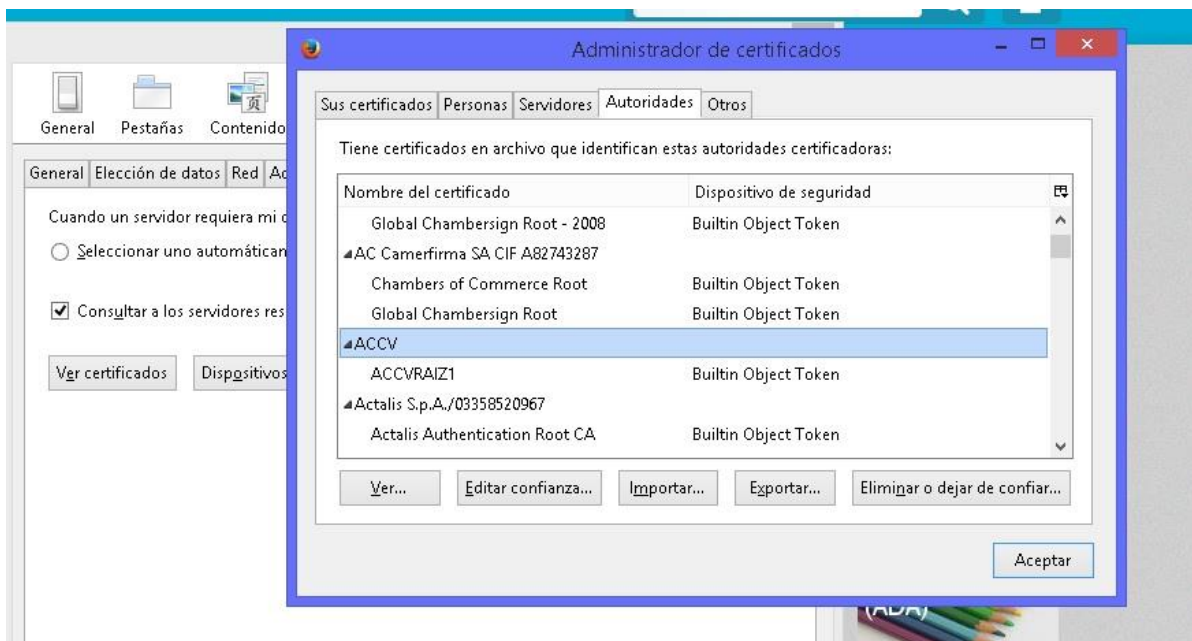
Primero entramos en una web segura, por ejemplo, en el área privada de educarm.es. Nos saldrá esto:



Le damos a añadir excepción y nos aparecerá esto:

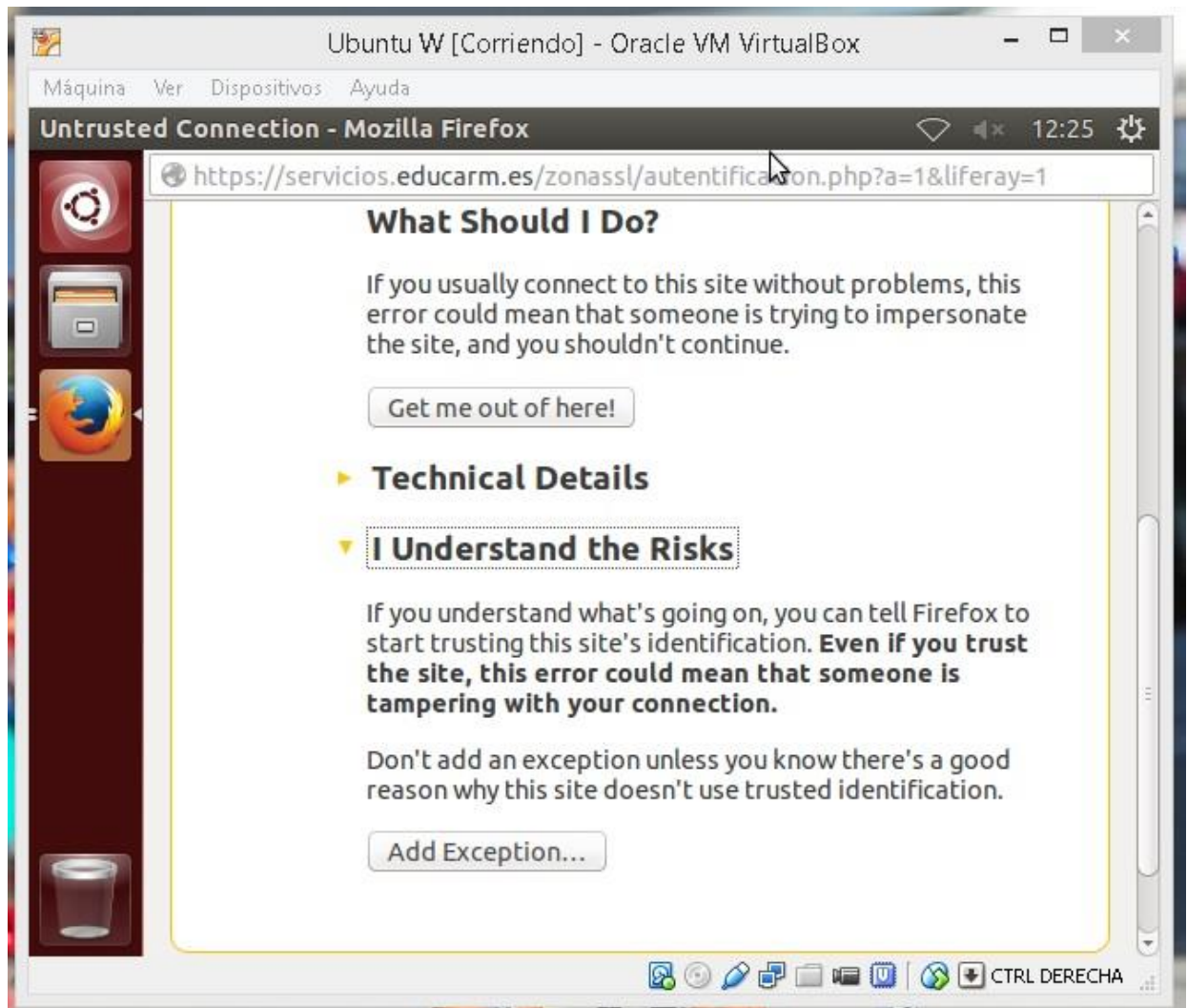


Pulsamos sobre **Obtener certificado** y **Confirmar excepción de seguridad**. Una vez que hayamos hecho esto, podremos entrar en la zona segura y se nos habrá añadido el certificado como se puede ver en la siguiente imagen:

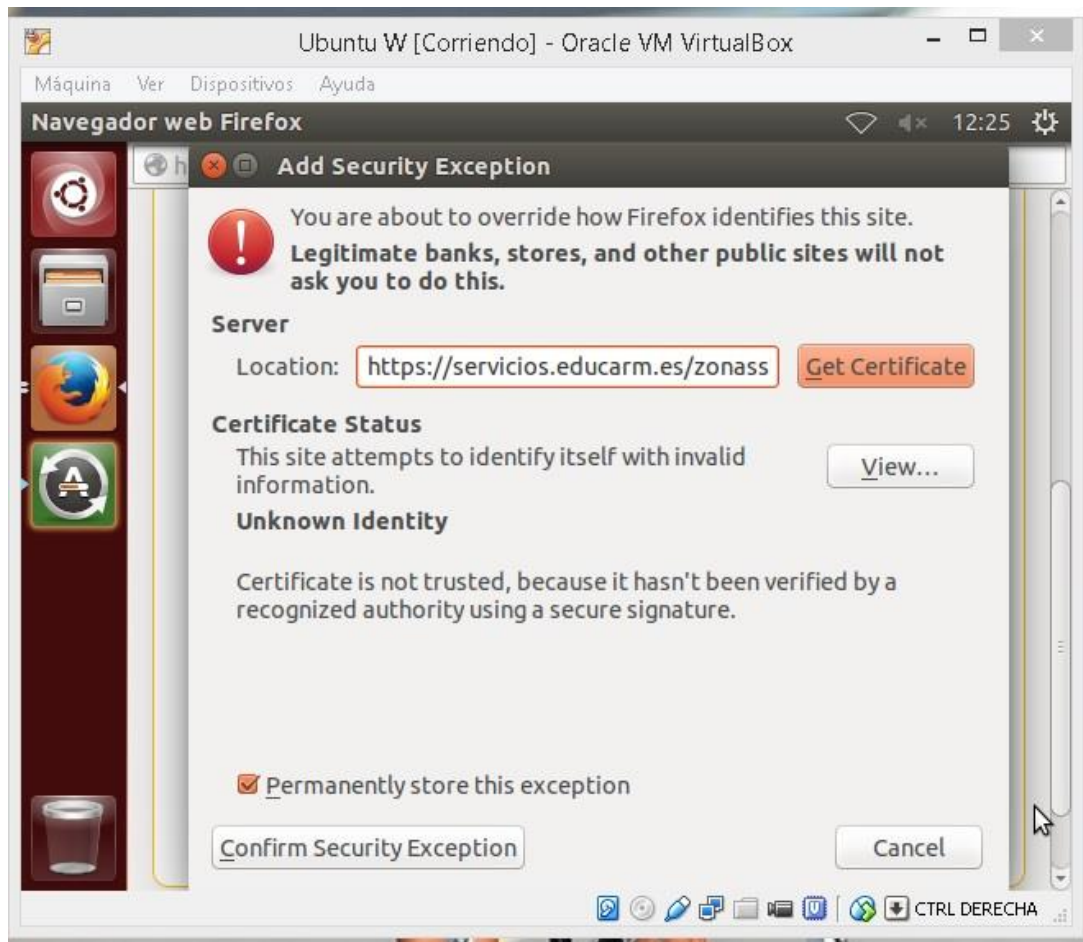


Ahora lo vemos con Linux.

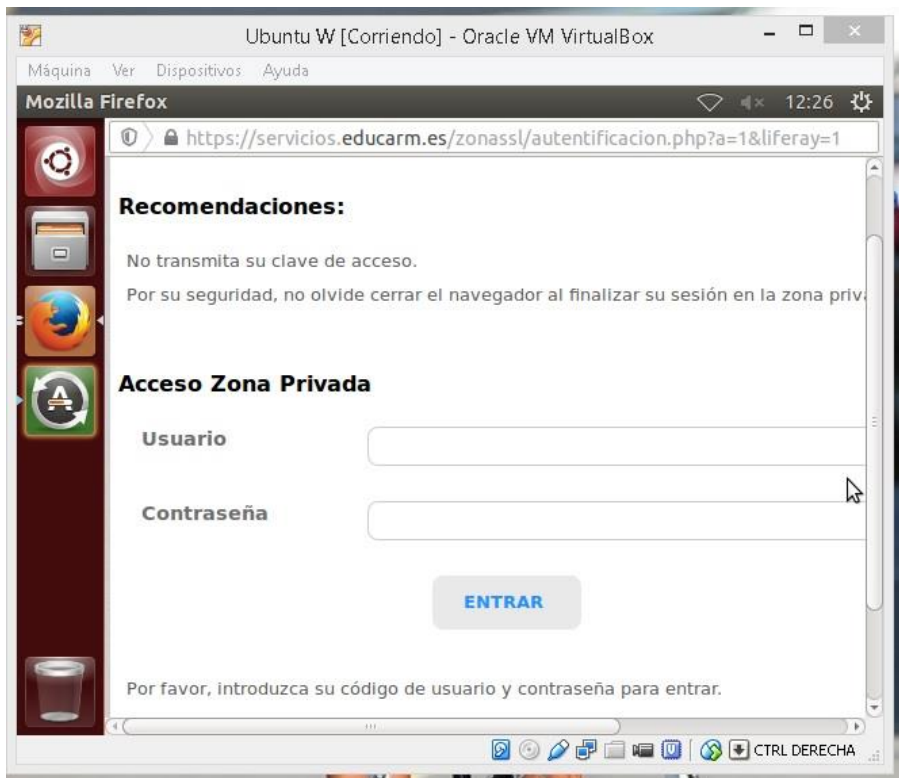
Entramos en la zona privada y nos sale esto:



Pinchamos sobre **Add Exception...** y nos saldrá lo siguiente:

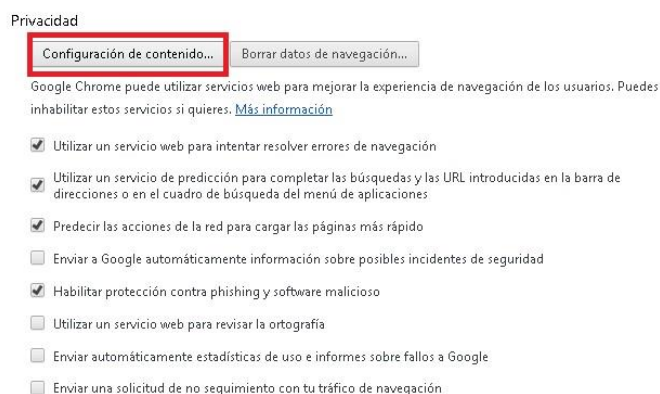


Ahora pulsamos sobre **Get Certificate** y **Confirm Security Exception**. Una vez hecho esto ya podremos entrar con normalidad a la zona privada, como podemos ver en la siguiente imagen.



**3º Configura el navegador para que acepte todas las cookies, navega unos minutos por tus Web favoritas y consulta las cookies que se han guardado en tu equipo. Una vez comprobadas bórralas para no dejar rastros de tu uso de internet y configura el navegador en su estado anterior. Entrega listado de cookies y pasos realizados.**

Entramos en Google Chrome y damos a **configuración**. Ahora nos vamos a la pestaña de **privacidad** y le damos a **configuración de contenido**.



Configuración de contenido

Cookies

Permitir que se almacenen datos locales (recomendado)

Conservar datos locales solo hasta que salgas del navegador

No permitir que se guarden datos de los sitios

Bloquear los datos de sitios y las cookies de terceros

Administrar excepciones... Todas las cookies y los datos de sitios...

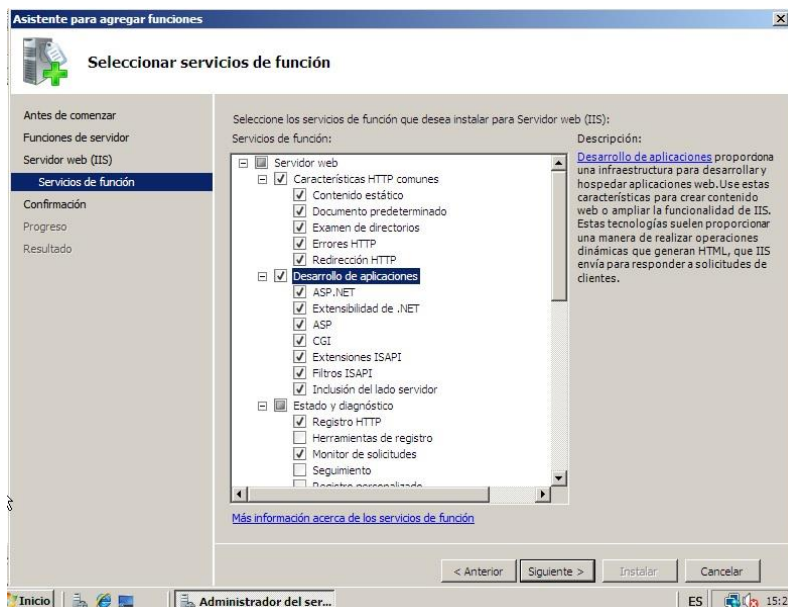
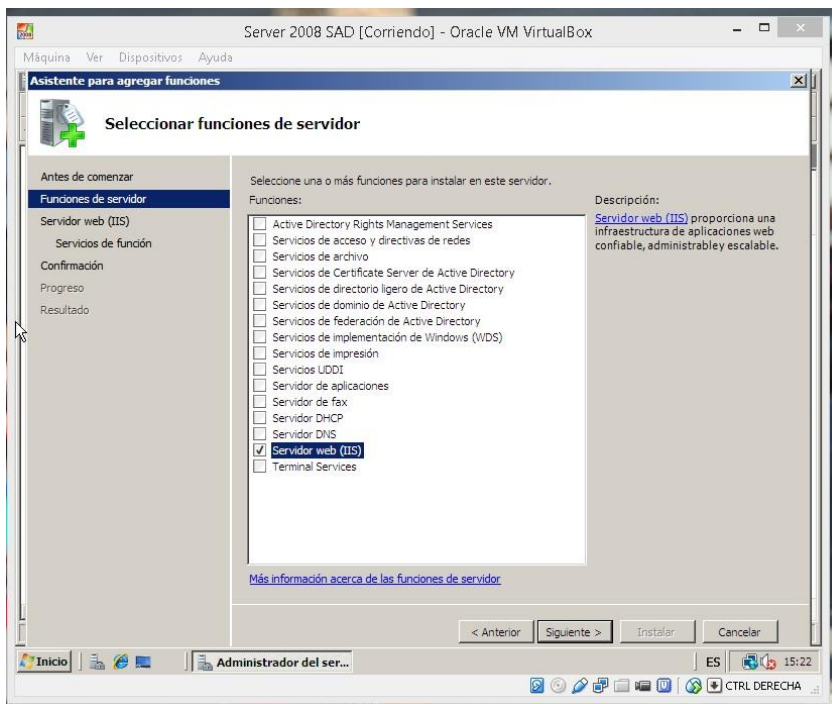


Como vemos, se han guardado las **cookies** correctamente. Ahora las borramos y dejamos el navegador como lo teníamos.

## BLOQUE 2: ACTIVIDADES CON LOS SERVIDORES

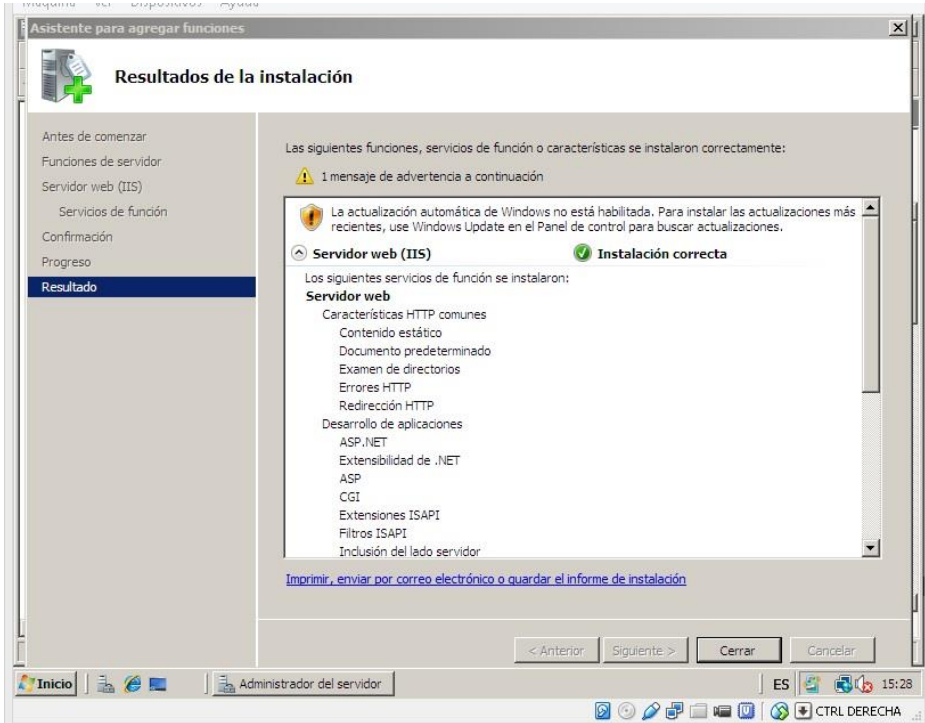
### Instalación de IIS en Windows

Entramos en **Administrar Servidor > Agregar características** y elegimos **Servicio Web (IIS)**. Lo dejamos como en las siguientes imágenes:

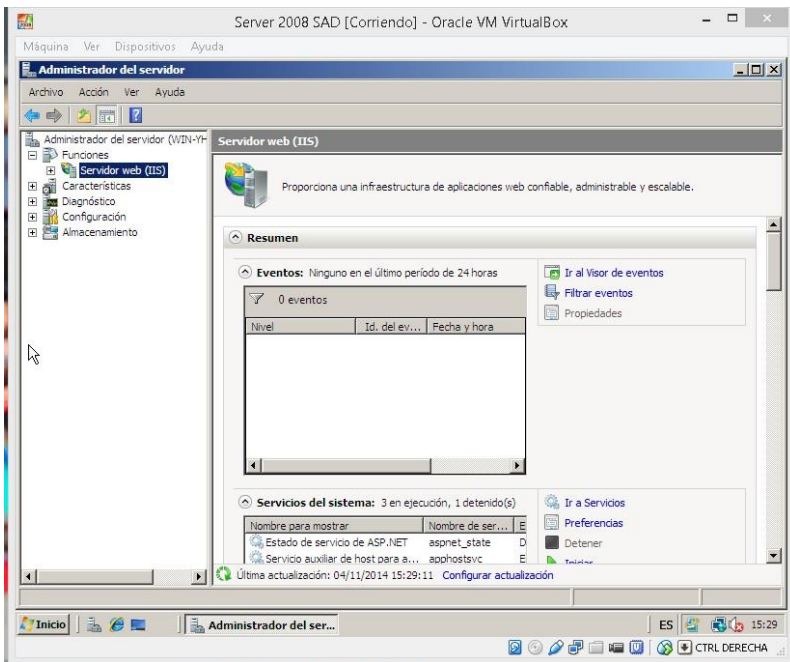


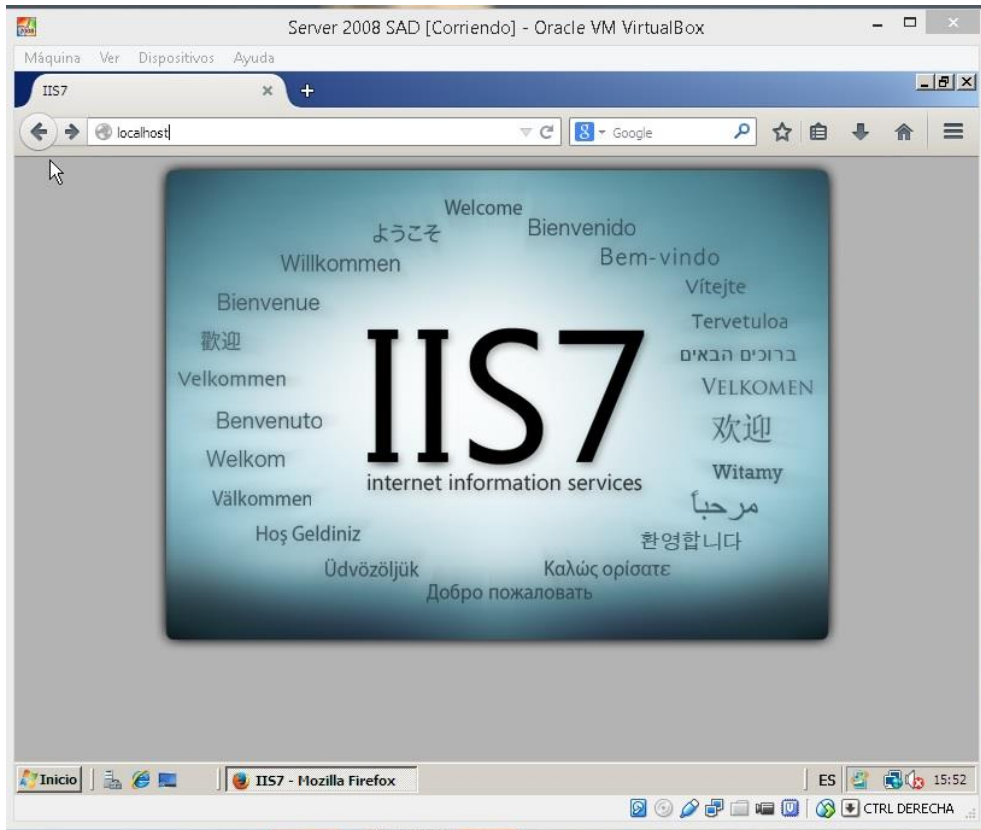


A continuación, le damos a **Siguiente** y a **Instalar**. Ahora esperemos a que la instalación acabe correctamente.



Ahora vemos como está nuestro servicio web funcionando.





## Apache2 en Linux

En el terminal y estando como **root** usamos:

```
Apt-get install apache2
```

Con esto instalaremos el apache2 y solo nos faltará configurarlo.

```
root@ubuntu:/home/adrian# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
 libaprutil1-ldap
Paquetes sugeridos:
 apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils
Se instalarán los siguientes paquetes NUEVOS:
 apache2 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
 libaprutil1-ldap
0 actualizados, 7 se instalarán, 0 para eliminar y 3 no actualizados.
Necesito descargar 1.267 kB de archivos.
Se utilizarán 5.238 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

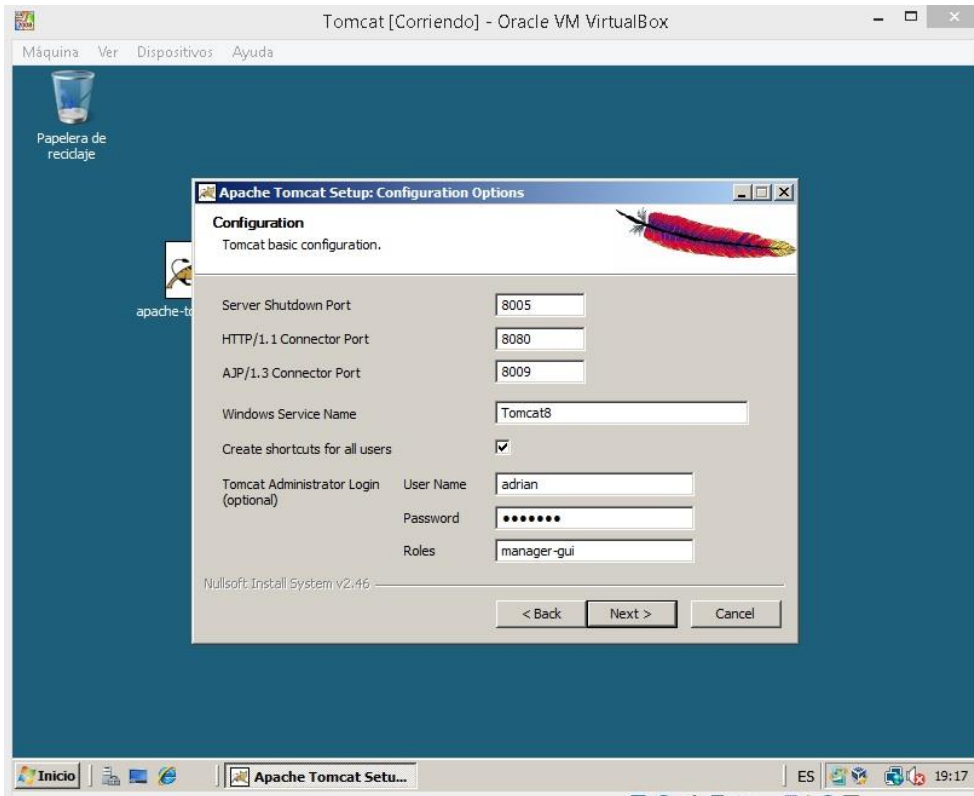
```
Máquina Ver Dispositivos Ayuda
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
* Starting web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
*
Processing triggers for libc-bin (2.19-0ubuntu6.3) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for ufw (0.34~rc-0ubuntu2) ...
root@ubuntu:/home/adrian#
```



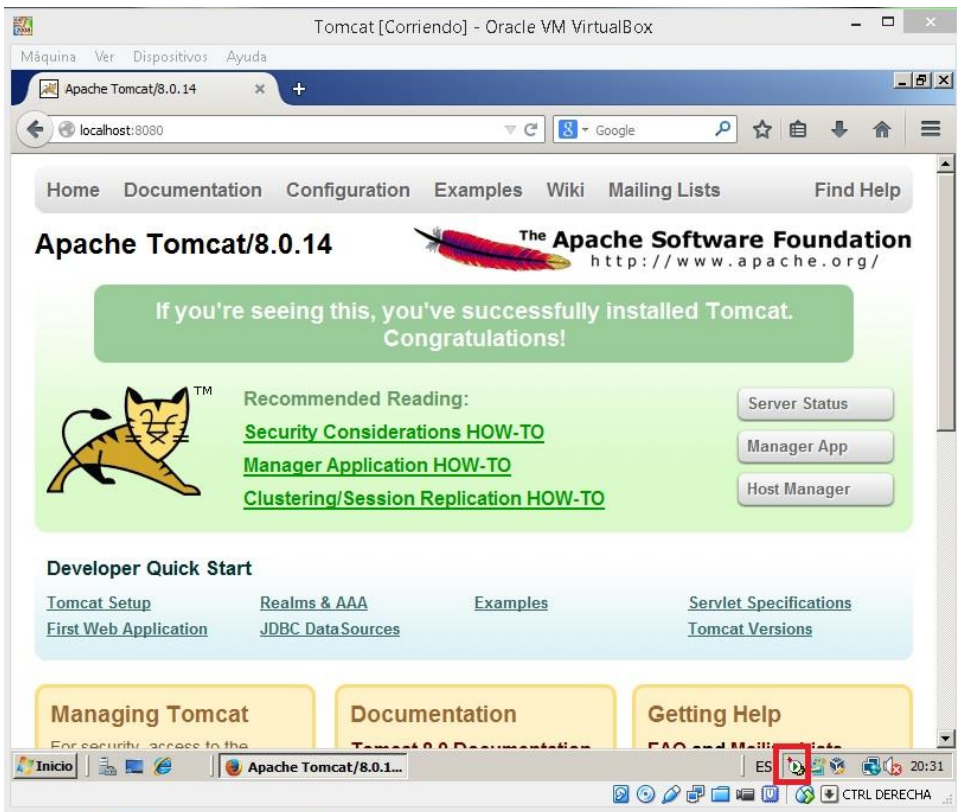
## Tomcat en Windows

Lo primero nos descargamos el Tomcat de <http://tomcat.apache.org/download-80.cgi>. Descargamos el fichero .exe para que sea más fácil la instalación y lo ejecutamos.





Ahora introducimos la ruta donde tenemos instalado JAVA y continuamos instalando. Una vez instalado, comprobamos que funciona.



## Tomcat en Linux

Lo primero que tenemos que hacer es instalar el **Tomcat** y **Java JDK**. Para ello usamos los siguientes comandos:

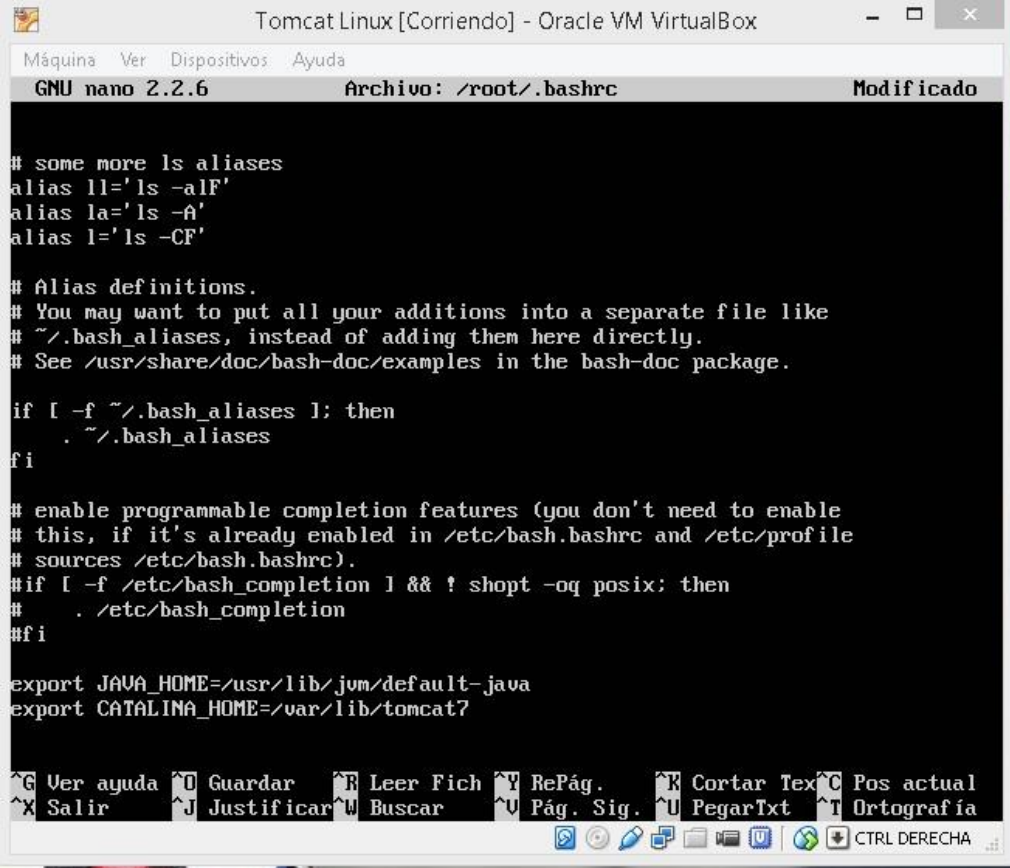
```
Sudo apt-get install tomcat7
```

```
Sudo apt-get install default-jdk
```

Ahora que los tenemos instalados, vamos al archivo `~/.bashrc` y añadimos estas líneas al final:

```
Export JAVA_HOME=/usr/lib/jvm/default-java
```

```
Export CATALINA_HOME=/var/lib/tomcat7
```



```
Tomcat Linux [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
GNU nano 2.2.6 Archivo: /root/.bashrc Modificado

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

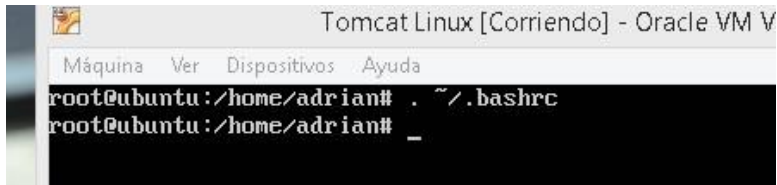
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi

export JAVA_HOME=/usr/lib/jvm/default-java
export CATALINA_HOME=/var/lib/tomcat7

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^U Pág. Sig. ^U PegarTxt ^T Ortografía
CTRL DERECHA
```

Ahora recargamos el archivo usando esto:

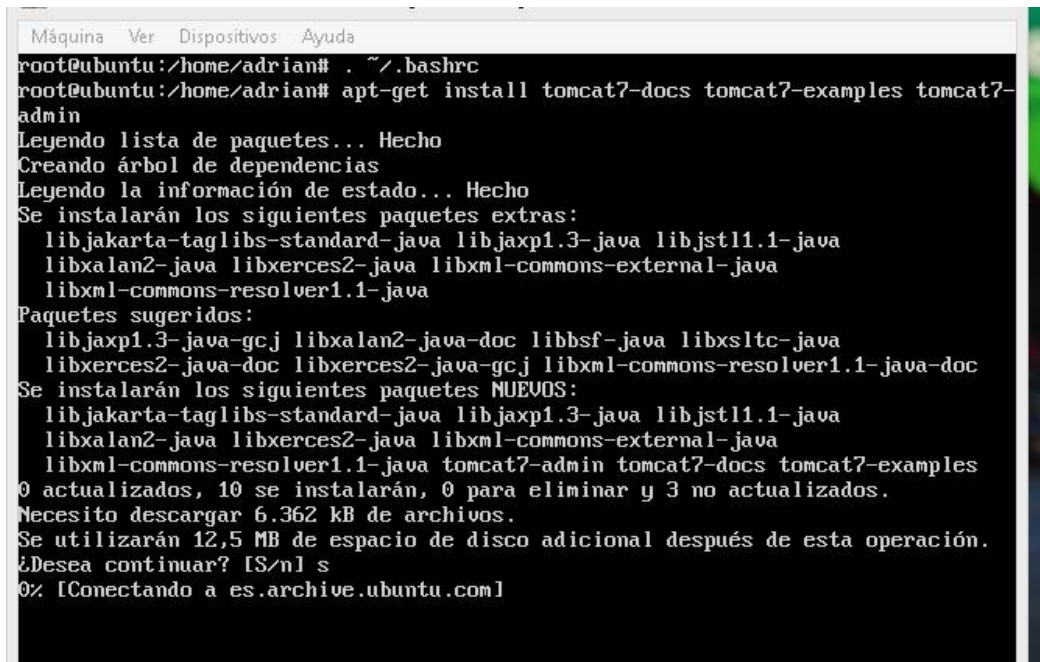
```
. ~/.bashrc
```



```
Tomcat Linux [Corriendo] - Oracle VM V
Máquina Ver Dispositivos Ayuda
root@ubuntu:/home/adrian# . ~/.bashrc
root@ubuntu:/home/adrian# _
```

Ahora podemos descargar documentación, etc.. usando lo siguiente:

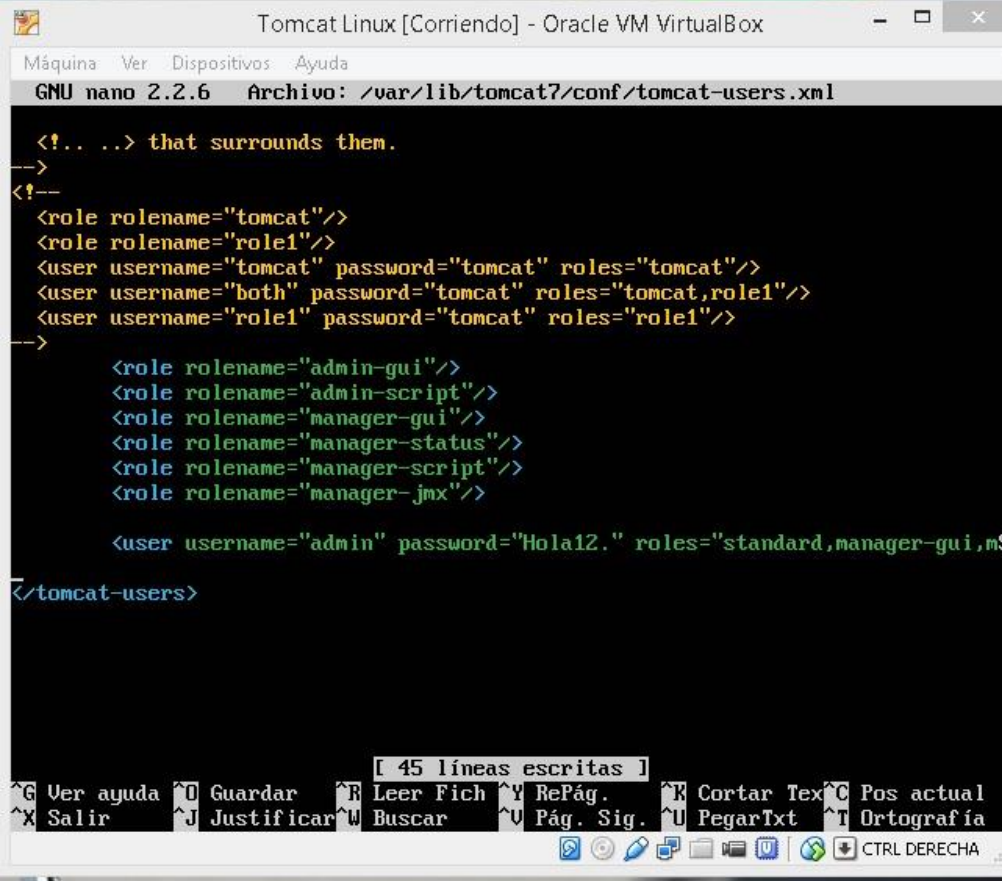
*Apt-get install tomcat7-docs tomcat7-examples tomcat7-admin*



```
Máquina Ver Dispositivos Ayuda
root@ubuntu:/home/adrian# . ~/.bashrc
root@ubuntu:/home/adrian# apt-get install tomcat7-docs tomcat7-examples tomcat7-
admin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libjakarta-taglibs-standard-java libjasp1.3-java libjstl1.1-java
 libxalan2-java libxerces2-java libxml-commons-external-java
 libxml-commons-resolver1.1-java
Paquetes sugeridos:
 libjasp1.3-java-gcj libxalan2-java-doc libbsf-java libxslt-java
 libxerces2-java-doc libxerces2-java-gcj libxml-commons-resolver1.1-java-doc
Se instalarán los siguientes paquetes NUEVOS:
 libjakarta-taglibs-standard-java libjasp1.3-java libjstl1.1-java
 libxalan2-java libxerces2-java libxml-commons-external-java
 libxml-commons-resolver1.1-java tomcat7-admin tomcat7-docs tomcat7-examples
0 actualizados, 10 se instalarán, 0 para eliminar y 3 no actualizados.
Necesito descargar 6.362 kB de archivos.
Se utilizarán 12,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
0% [Conectando a es.archive.ubuntu.com]
```

Ahora solo nos falta modificar el **archivo de usuarios**.

*Nano /var/lib/tomcat7/conf/tomcat-users.xml*



The screenshot shows a terminal window titled "Tomcat Linux [Corriendo] - Oracle VM VirtualBox". The terminal is running GNU nano 2.2.6, editing the file /var/lib/tomcat7/conf/tomcat-users.xml. The content of the file is as follows:

```
<!-- ... --> that surrounds them.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="tomcat" roles="tomcat"/>
<user username="both" password="tomcat" roles="tomcat,role1"/>
<user username="role1" password="tomcat" roles="role1"/>
-->

<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-status"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>

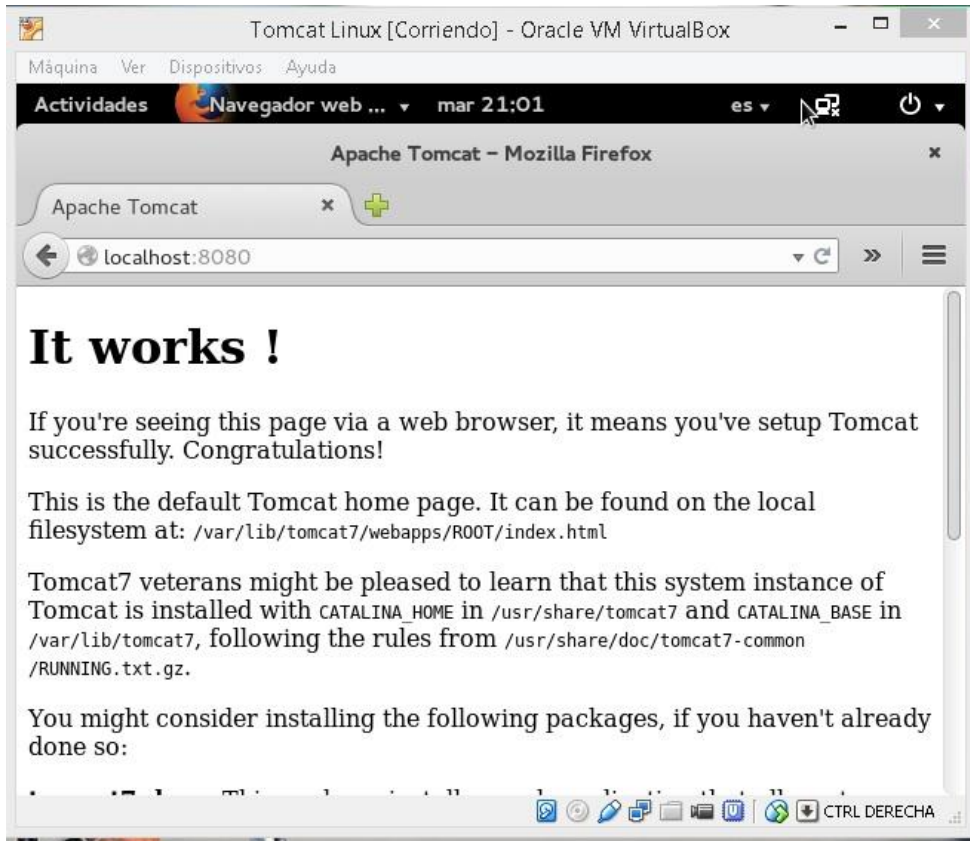
<user username="admin" password="Hola12." roles="standard,manager-gui,m$
</tomcat-users>
```

At the bottom of the terminal, there is a status bar indicating "[ 45 líneas escritas ]" and a menu of keyboard shortcuts for nano, such as "Ver ayuda", "Guardar", "Leer Fich", "Repág.", "Cortar Tex", "Pos actual", "Salir", "Justificar", "Buscar", "Pág. Sig.", "PegarTxt", and "Ortografía".

Ahora solo falta iniciar el servicio **Tomcat** para ello usamos:

*Service tomcat7 start*



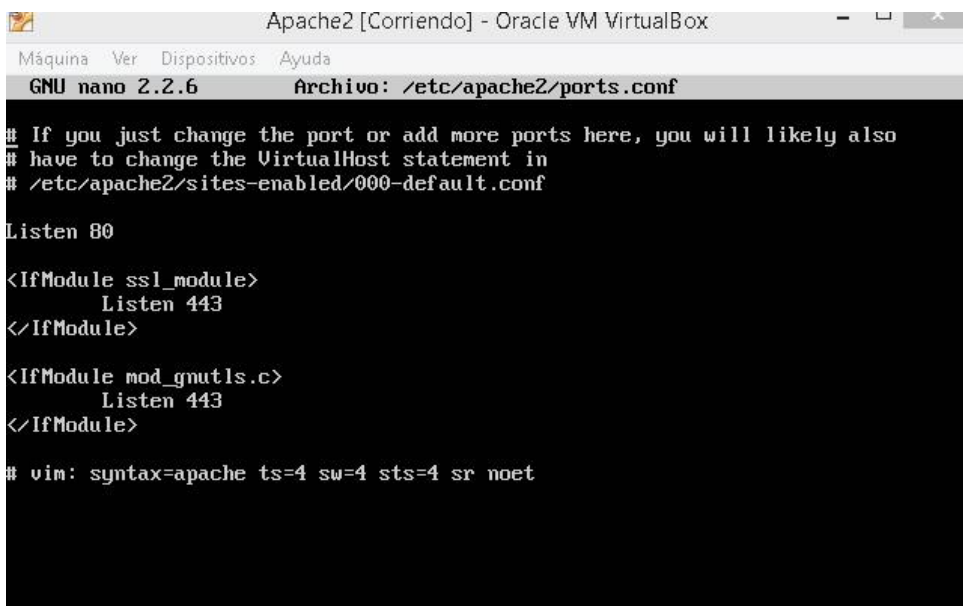


Como vemos funciona perfectamente nuestro servicio **Tomcat7** en Linux.

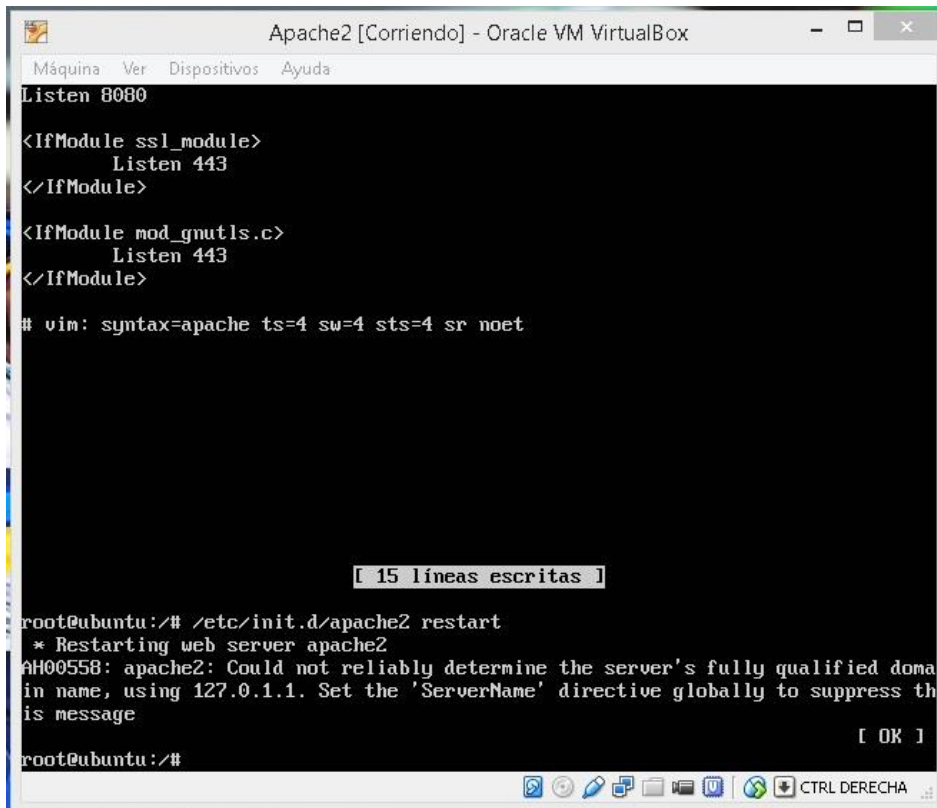
### Configuración de apache.

Para cambiar el puerto de escucha tenemos que modificar un archivo, para ello usamos:

Nano /etc/apache2/ports.conf



Tenemos que poner **Listen 80** al puerto que queramos, en nuestro caso el **8080**. Una vez que hemos editado el fichero, guardamos y reiniciamos **apache2**.



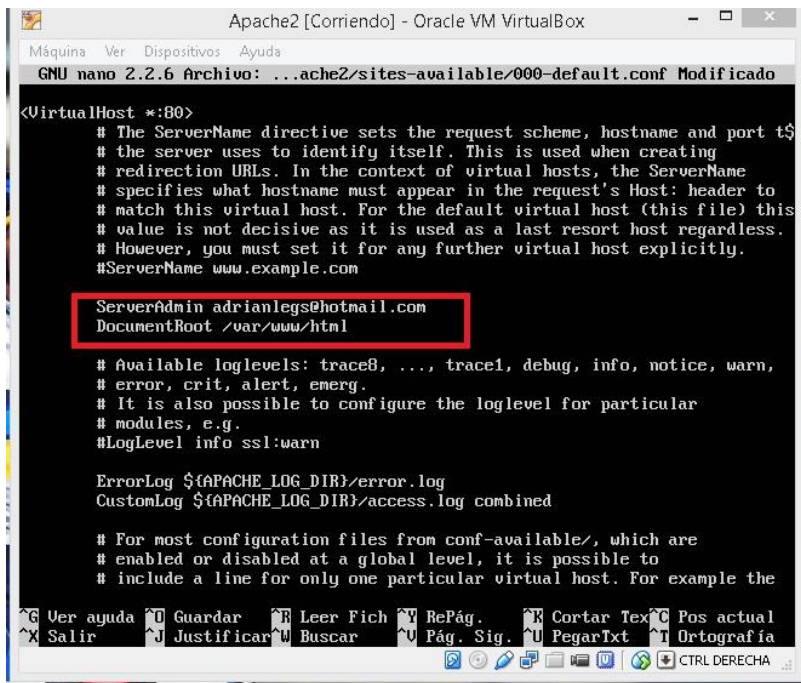
```
Apache2 [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
Listen 8080
<IfModule ssl_module>
  Listen 443
</IfModule>
<IfModule mod_gnutls.c>
  Listen 443
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

[ 15 líneas escritas ]
root@ubuntu:~# /etc/init.d/apache2 restart
* Restarting web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
[ OK ]
root@ubuntu:~#
```

Ya tenemos editado el puerto de escucha de apache a **8080**. Ahora al entrar a localhost, tendremos que poner **localhost:8080**



Ahora, vamos a cambiar el correo del administrador para ello vamos a **/etc/apache2/sites-available/000-default.conf** editamos ese archivo y donde pone **Serveradmin** ponemos nuestro correo.



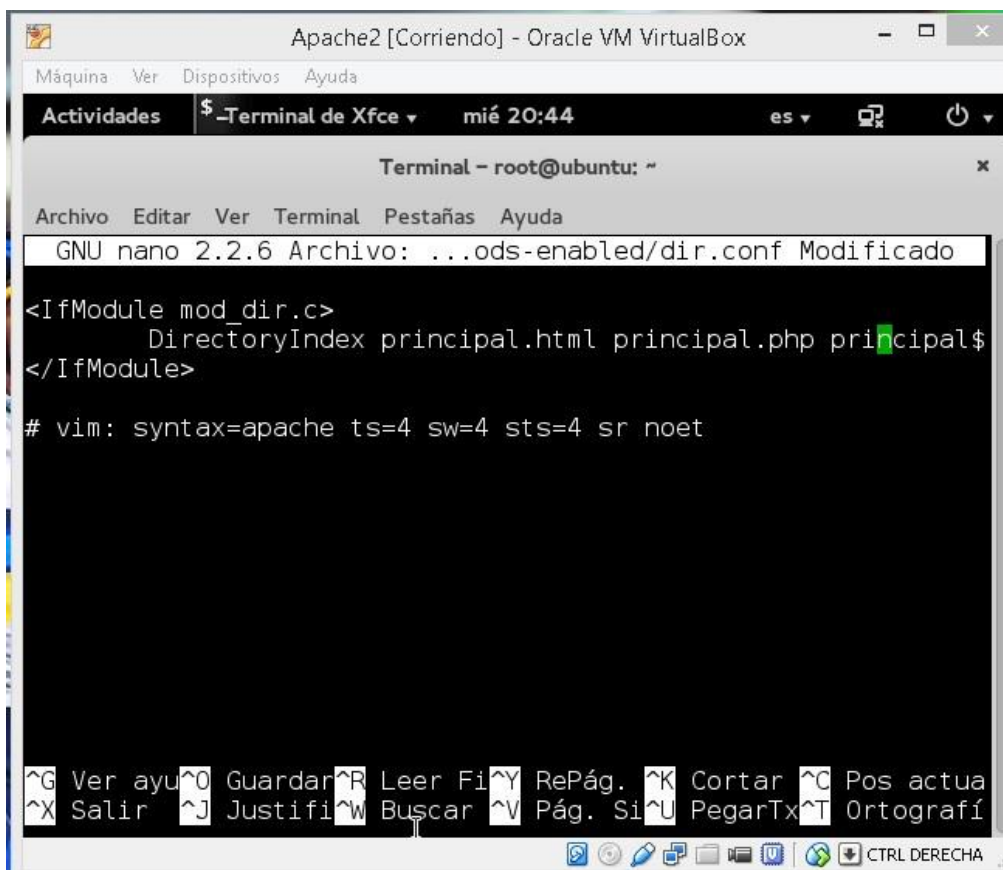
Guardamos y reiniciamos apache, como hemos visto antes.

`/etc/init.d/apache2 restart`

Ahora cambiamos el número máximo de clientes conectado, para ello vamos a **/etc/apache2/mods-enabled/mpm\_prefork.conf** , buscamos la directiva y la modificamos. (Comprobándolo no funciona muy bien, solo corta cuando la conexiones son muy altas, por ejemplo 10000)

```
<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers       10
    MaxRequestWorkers     50
    MaxConnectionsPerChild 0
</IfModule>
```

Para cambiar el fichero por defecto index.html vamos a **/etc/apache2/mods-enabled/dir.conf** y cambiamos el **DirectoryIndex** a principal.html



The screenshot shows a terminal window titled "Terminal - root@ubuntu: ~" with a menu bar including "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The terminal content shows the configuration of the `DirectoryIndex` directive in the `mpm_prefork.conf` file, which has been modified. The configuration is as follows:

```
GNU nano 2.2.6 Archivo: ...ods-enabled/dir.conf Modificado
<IfModule mod_dir.c>
    DirectoryIndex principal.html principal.php principal$
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: `^G Ver ayu`, `^O Guardar`, `^R Leer Fi`, `^Y RePág.`, `^K Cortar`, `^C Pos actua`, `^X Salir`, `^J Justifi`, `^W Buscar`, `^V Pág. Si`, `^U PegarTx`, `^T Ortografí`.

Ahora reiniciamos apache como hemos visto antes por si acaso y comprobamos si funciona bien. Antes tenemos que crear un archivo en **/var/www** llamándolo como hemos puesto en **DirectoryIndex**.

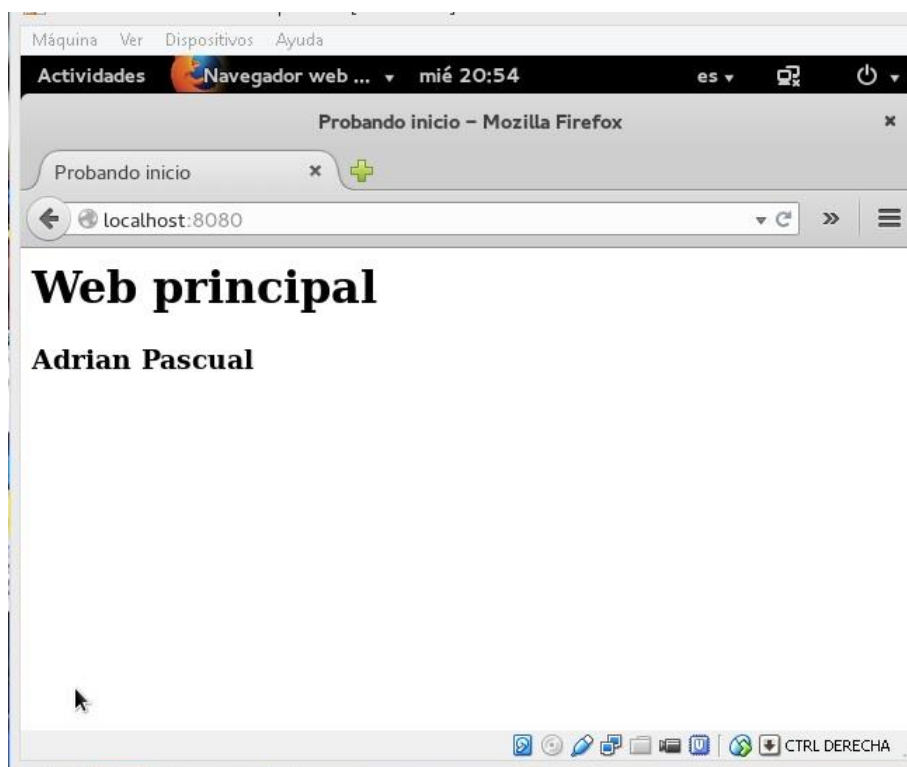
*Touch /var/www/principal.html*



```
GNU nano 2.2.6 Archivo: ...r/www/principal.html
<html>
<head><title>Probando inicio</title>
</head>
<body>
  <h1>Web principal</h1>
  <h3>Adrian Pascual</h3>
</body>
</html>
```

[ 8 líneas leídas ]

Ahora probamos a ver si funciona correctamente.

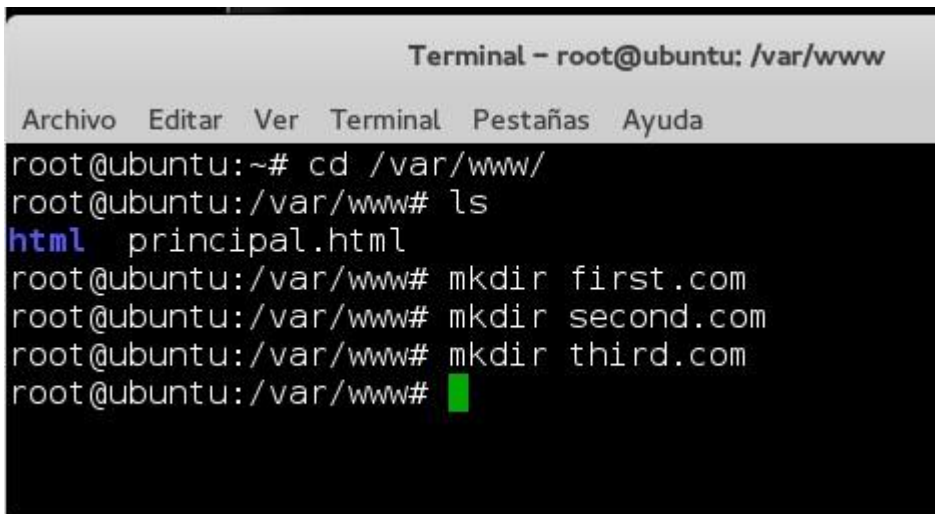


Como vemos, funciona todo bien. Entramos directamente a localhost:8080 y entra a **principal.html**

## Crear tres sitios.

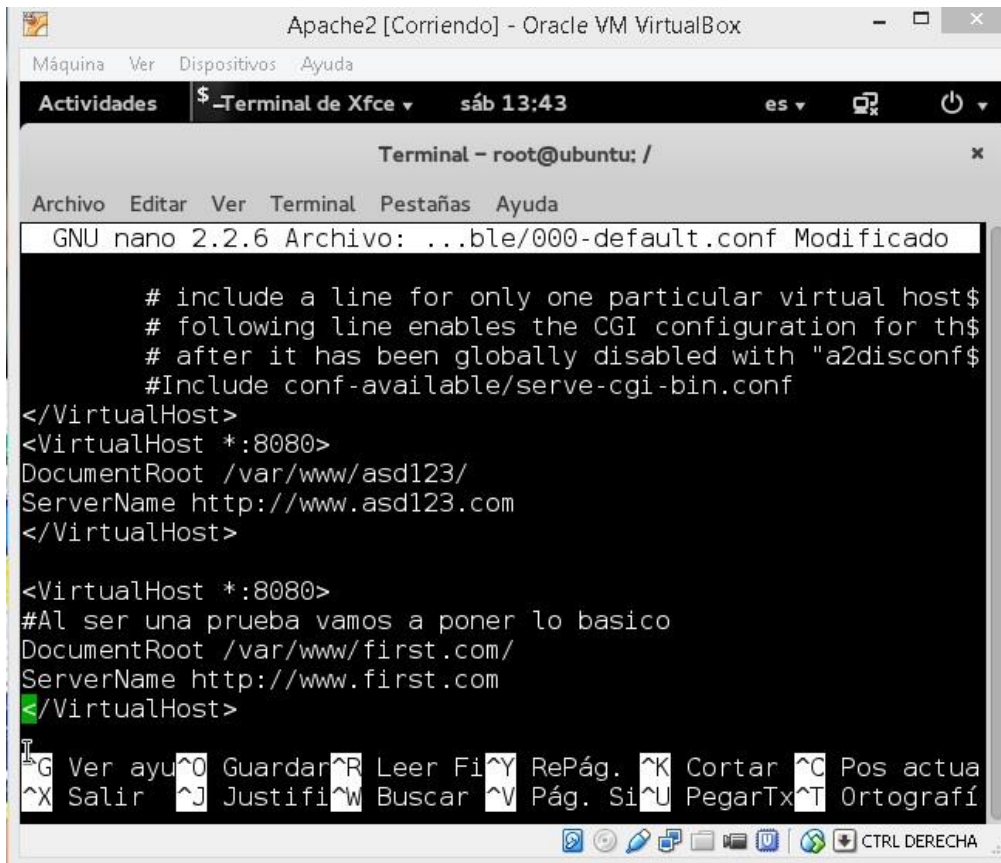
Lo primero que tenemos que hacer para crear los 3 sitios, es crear las carpetas donde estará todo, para ello vamos a **/var/www** y usamos:

*Mkdir first.com second.com third.com*



```
Terminal - root@ubuntu: /var/www
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
root@ubuntu:~# cd /var/www/
root@ubuntu:/var/www# ls
html  principal.html
root@ubuntu:/var/www# mkdir first.com
root@ubuntu:/var/www# mkdir second.com
root@ubuntu:/var/www# mkdir third.com
root@ubuntu:/var/www# █
```

Una vez que tenemos eso, nos vamos a **/etc/apache2/sites-available/000-default.conf** y lo abrimos con cualquier editor, lo que vamos a hacer es añadir un host virtual.



The screenshot shows a terminal window titled "Terminal - root@ubuntu: /" running the nano text editor. The editor is editing a file named "...ble/000-default.conf". The content of the file is as follows:

```
GNU nano 2.2.6 Archivo: ...ble/000-default.conf Modificado

# include a line for only one particular virtual host$
# following line enables the CGI configuration for th$
# after it has been globally disabled with "a2disconf$
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
<VirtualHost *:8080>
DocumentRoot /var/www/asd123/
ServerName http://www.asd123.com
</VirtualHost>

<VirtualHost *:8080>
#Al ser una prueba vamos a poner lo basico
DocumentRoot /var/www/first.com/
ServerName http://www.first.com
</VirtualHost>
```

The terminal window also shows a menu bar with various keyboard shortcuts like "Ver ayuda", "Guardar", "Leer", "Re Pág.", "Cortar", "Pos actua", "Salir", "Justifi", "Buscar", "Pág. Si", "PegarTx", and "Ortografí".

Ahora vamos a crear un archivo para que se vea en el navegador cuando pongamos [www.first.com](http://www.first.com) para ello usamos:

***Nano /etc/www/first.com/principal.html***

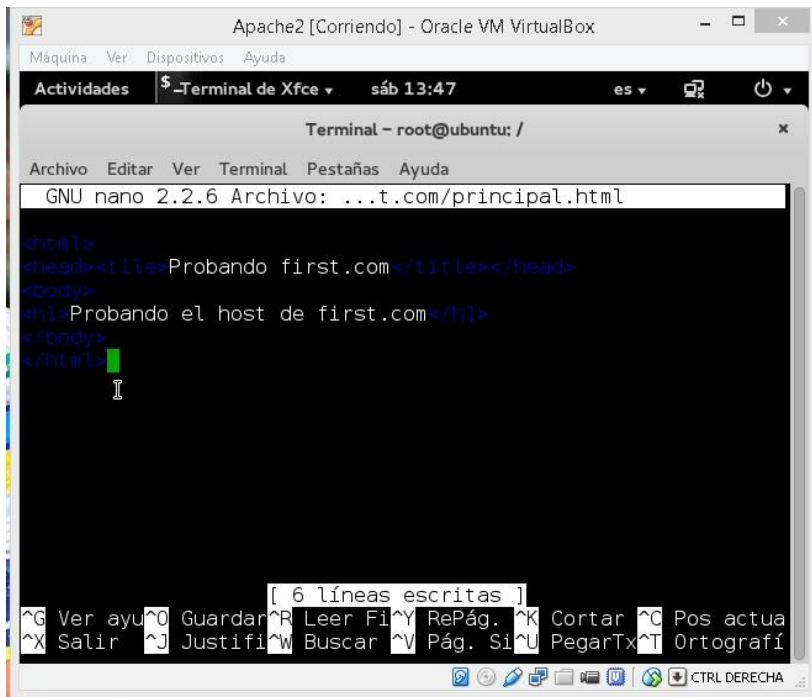
y creamos una página web sencilla:

```
<html>
```

```
<head><title>Probando first.com</title></head>
```

```
<body><h1>Probando el host first.com</h1></body>
```

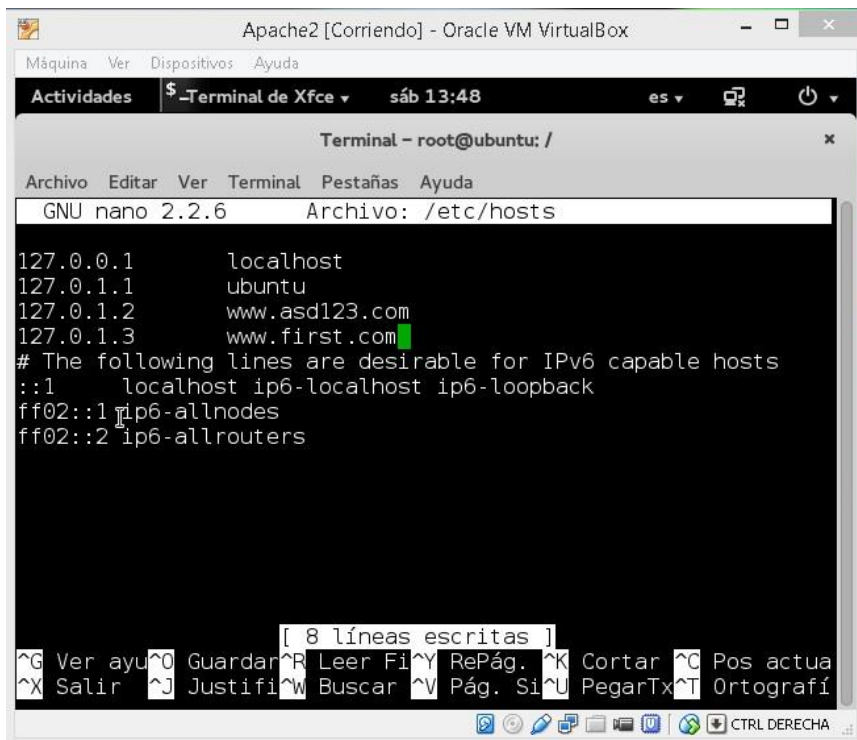
```
</html>
```



```
Apache2 [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
Actividades $-Terminal de Xfce sáb 13:47 es
Terminal - root@ubuntu: /
GNU nano 2.2.6 Archivo: ...t.com/principal.html
ctrl+g
ctrl+g
ctrl+g
ctrl+g
ctrl+g
ctrl+g
[ 6 líneas escritas ]
^G Ver ayuda ^O Guardar ^R Leer Fichero ^Y RePág. ^K Cortar ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U Pegar ^T Ortografía
CTRL DERECHA
```

Una vez que tenemos hecho eso, nos vamos a **/etc/hosts** y añadimos la siguiente línea:

127.0.1.3 [www.first.com](http://www.first.com) (enmi caso)



```
Apache2 [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
Actividades $-Terminal de Xfce sáb 13:48 es
Terminal - root@ubuntu: /
GNU nano 2.2.6 Archivo: /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu
127.0.1.2 www.asd123.com
127.0.1.3 www.first.com
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
[ 8 líneas escritas ]
^G Ver ayuda ^O Guardar ^R Leer Fichero ^Y RePág. ^K Cortar ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U Pegar ^T Ortografía
CTRL DERECHA
```



Por último, **reiniciamos apache (importantísimo)**.

```
/etc/init.d/apache2 restart
```

```
/etc/init.d/apache2 reload
```

Siempre hago los dos, porque leí en internet que muchas veces lee lo que ya tiene cargado anteriormente.

```
root@ubuntu:/# /etc/init.d/apache2 restart
* Restarting web server apache2 [ OK ]
root@ubuntu:/# /etc/init.d/apache2 reload
* Reloading web server apache2
*
root@ubuntu:/# █
```

Ahora vamos a comprobar que funciona nuestro primer host virtual, para ello vamos al navegador web y colocamos [www.first.com:8080](http://www.first.com:8080) (el 8080 es porque cambiamos el puerto de escucha anteriormente).



Listo, funciona correctamente nuestro host virtual.

Ahora vamos a realizar lo mismo pero que solo el usuario **curro** tenga acceso a este sitio, para ello hacemos los mismos pasos de antes pero cambiando el nombre a **second.com**.

Necesitamos una serie de requisitos que supuestamente nos deberían venir instalados con apache2, pero por si acaso pongo un enlace para ver cuáles son: <http://httpd.apache.org/docs/current/howto/auth.html>

Una vez estamos donde el paso anterior, empezaremos a cifrarlo, para ello necesitamos un archivo con contraseñas que lo crearemos con un comando:

```
Htpasswd -c /etc/apache2/passwords curro
```

**Importante: Este archivo no puede ser abierto por nadie que acceda a la web, hay que ponerlo en un sitio no accesible por los usuarios.**

```
root@ubuntu:/# htpasswd -c /etc/apache2/passwords curro
New password:
Re-type new password:
Adding password for user curro
root@ubuntu:/# █
```

Una vez que hemos hecho eso, nos vamos a **/var/www/second.com** y ahí creamos un archivo llamado **.htaccess** que contendrá lo siguiente:

```
AuthUserFile /etc/apache2/passwords
```

```
AuthName "Solo personal autorizado"
```

```
AuthType Basic
```

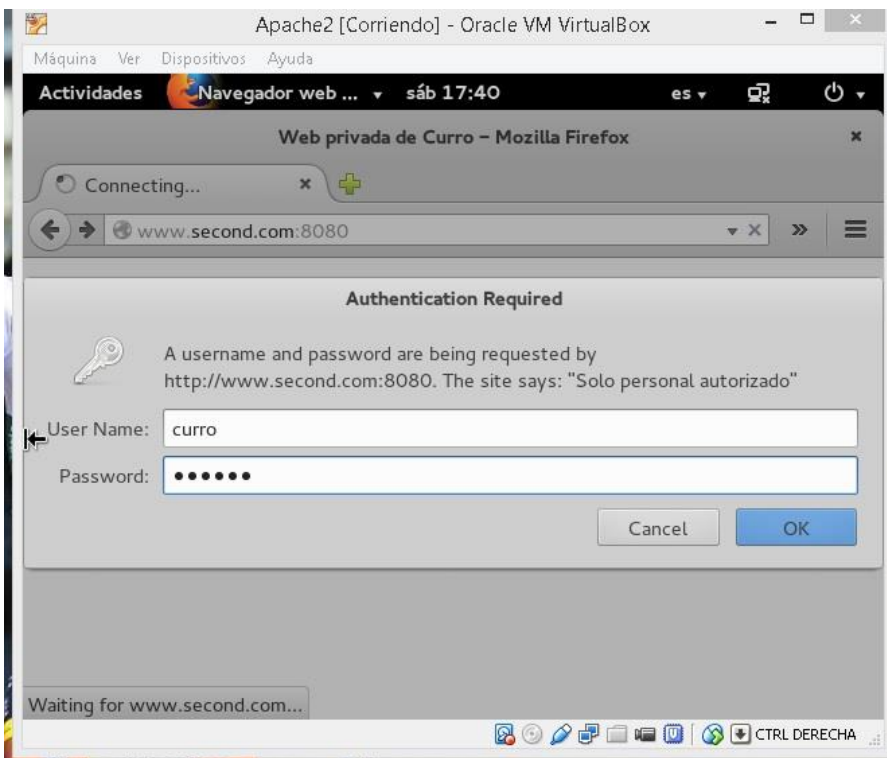
```
Require valid-user
```

Guardamos y le damos permisos:

```
Chmod 644 .htaccess
```

```
Terminal - root@ubuntu: /
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Archivo: ...second.com/.ht
AuthUserFile /etc/apache2/passwords
AuthName "Solo personal autorizado"
AuthType Basic
Require valid-user
```

Una vez hecho esto, vamos al navegador y entramos en [www.second.com:8080](http://www.second.com:8080) nos debería aparecer un “login” pidiéndonos los datos anteriormente puestos. Si no nos sale, reiniciamos apache y probamos de nuevo.



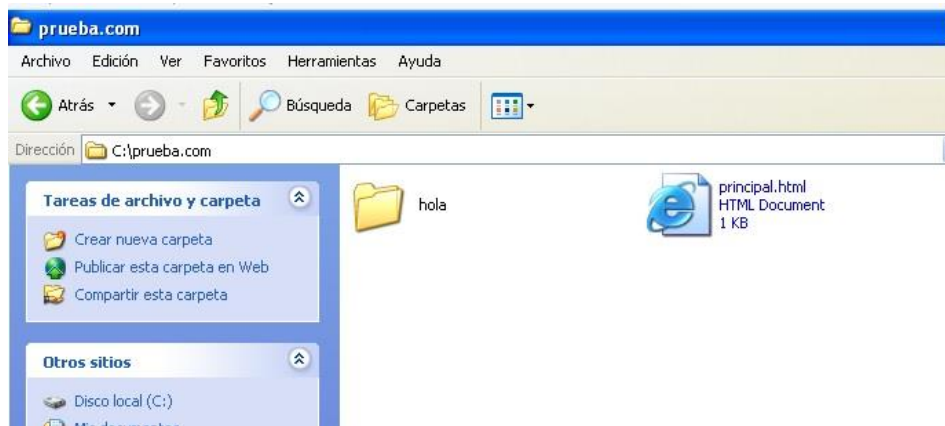
Ponemos los datos y le damos a **OK** y ya entraremos en la web.



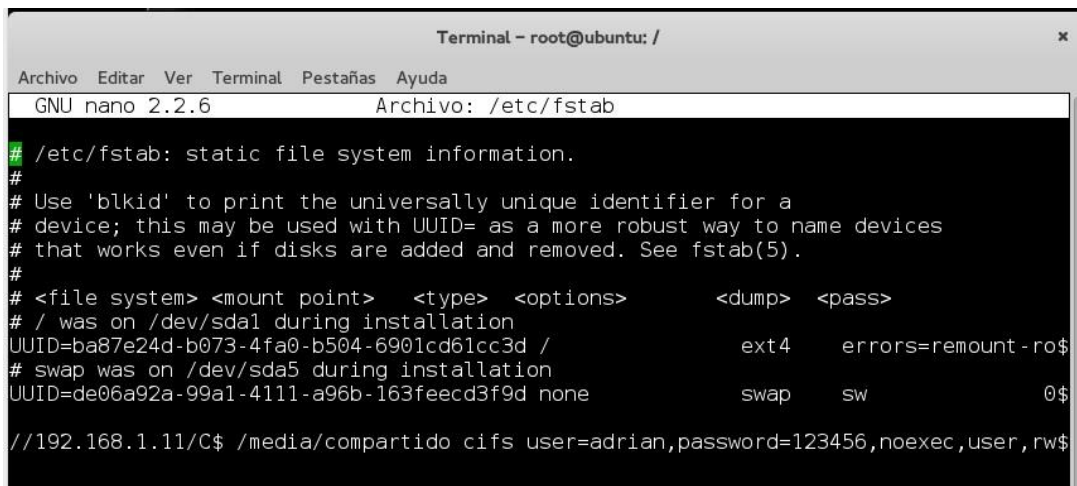
Como hemos visto funciona correctamente el acceso al segundo sitio.

Ahora toca crear el tercer sitio pero con los documentos en otro ordenador. Lo primero de todo tenemos que ver como compartir carpetas entre Ubuntu y Windows.

Primero creamos en el otro equipo el host con su archivo principal.



Ahora en el Linux vamos a **/etc/fstab** y agregamos la siguiente línea:



```
Terminal - root@ubuntu: /
GNU nano 2.2.6 Archivo: /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=ba87e24d-b073-4fa0-b504-6901cd61cc3d / ext4 errors=remount-ro$
# swap was on /dev/sda5 during installation
UUID=de06a92a-99a1-4111-a96b-163feecd3f9d none swap sw 0$
//192.168.1.11/C$ /media/compartido cifs user=adrian,password=123456,noexec,user,rw$
```

Ahora creamos la carpeta **/media/compartido**

Reiniciamos la máquina completa, y abrimos el siguiente archivo **/etc/apache2/sites-available/000-default.conf** y lo dejamos así:

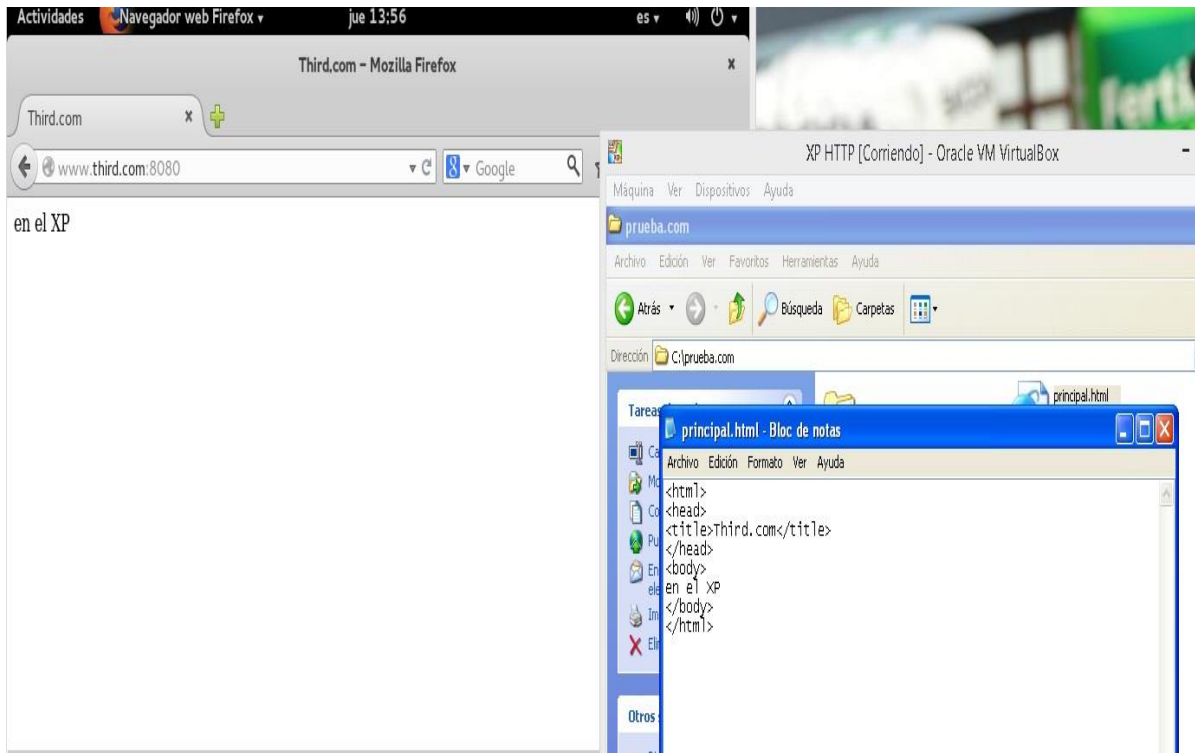


```
<VirtualHost www.third.com:8080>
DocumentRoot /media/compartido/prueba.com
#Alias /tercero.com /tercero.com
ServerName http://www.third.com
<Directory /media/compartido/prueba.com>
Order allow,deny
Allow from all
Require all granted
</Directory>
</VirtualHost>
```

Ahora reiniciamos apache2 con

**/etc/init.d/apache2 restart**

Y probamos a ver si funciona.



Como vemos, funciona correctamente ya tenemos el third.com funcionando.

### **Cambiar el directorio de Linux a Windows y viceversa.**

Para cambiar los directorios de apache de Linux a Windows y de Windows a Linux tenemos que entrar en ***/etc/apache2/apache2.conf***

*Nano /etc/apache2/apache2.conf*

Una vez estemos ahí tenemos que configurar el **ServerRoot** de la siguiente forma:

*ServerRoot <dirección Windows>/<etiqueta de disco>/AppServ/Apache/*

GNU nano 2.2.6 Archivo: /etc/apache2/apache2.conf Modificado  
#  
# NOTE! If you intend to place this on an NFS (or otherwise network)  
# mounted filesystem then please read the Mutex documentation (available  
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);  
# you will save yourself a lot of trouble.  
#  
# Do NOT add a slash at the end of the directory path.  
#  
#ServerRoot "<ruta servidor w.>/<etiqueta disco>/AppServ/Apache"  
ServerName www.adrian.com  
#  
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.  
#  
Mutex file:\${APACHE\_LOCK\_DIR} default  
#  
# PidFile: The file in which the server should record its process  
# identification number when it starts.  
# This needs to be set in /etc/apache2/envvars  
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual  
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía

Ahora para hacerlo al contrario, nos vamos al archivo de configuración de apache2 en Windows y cambiamos el **ServerRoot**

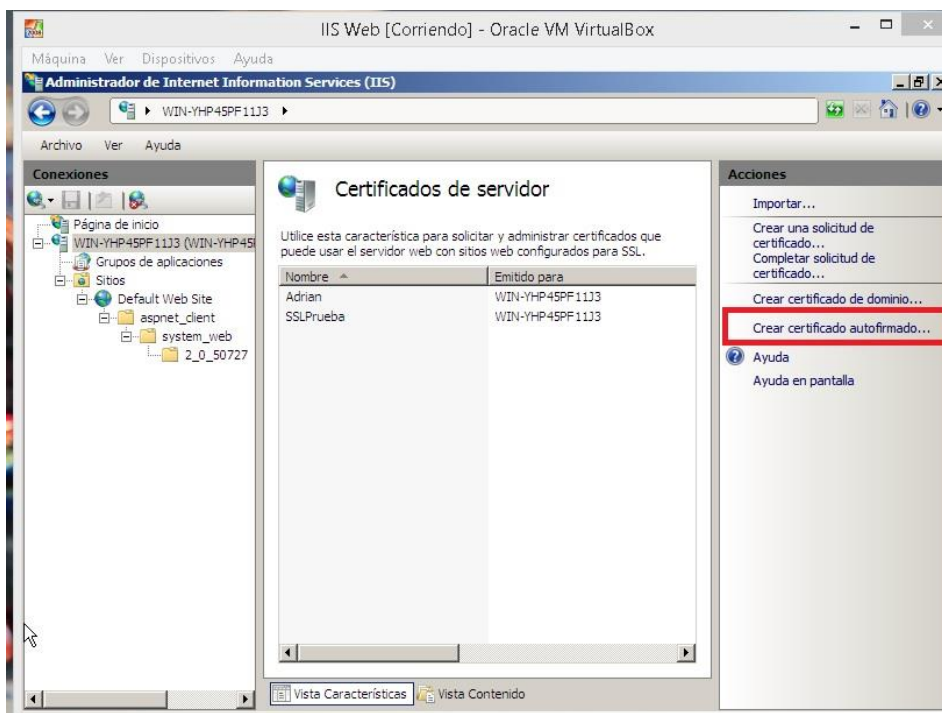
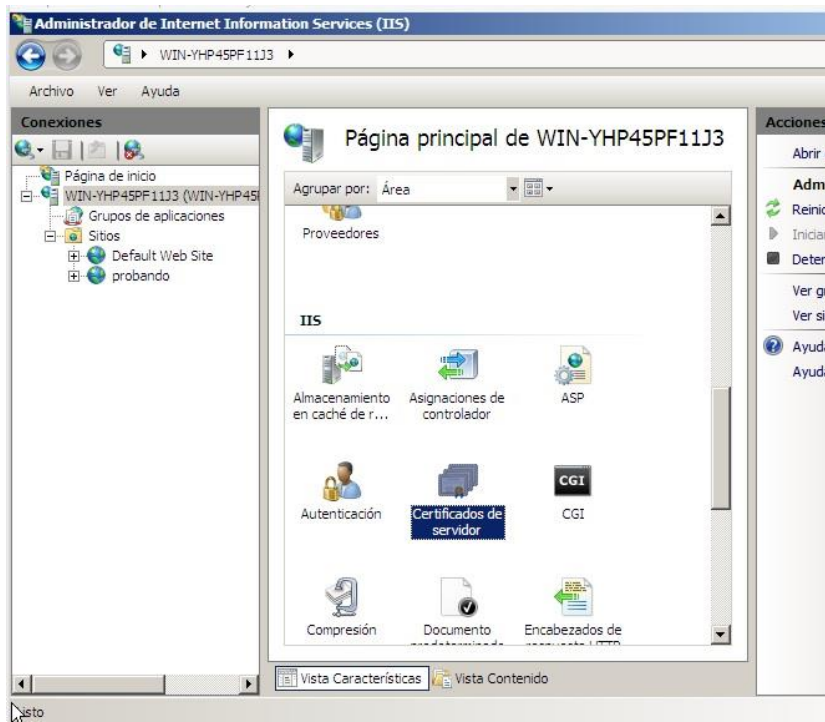
*ServerRoot <dirección Linux>/etc/apache2*

**(No pongo foto porque no tengo instalado apache2 en Windows)**

## Instalación de Certificados digitales.

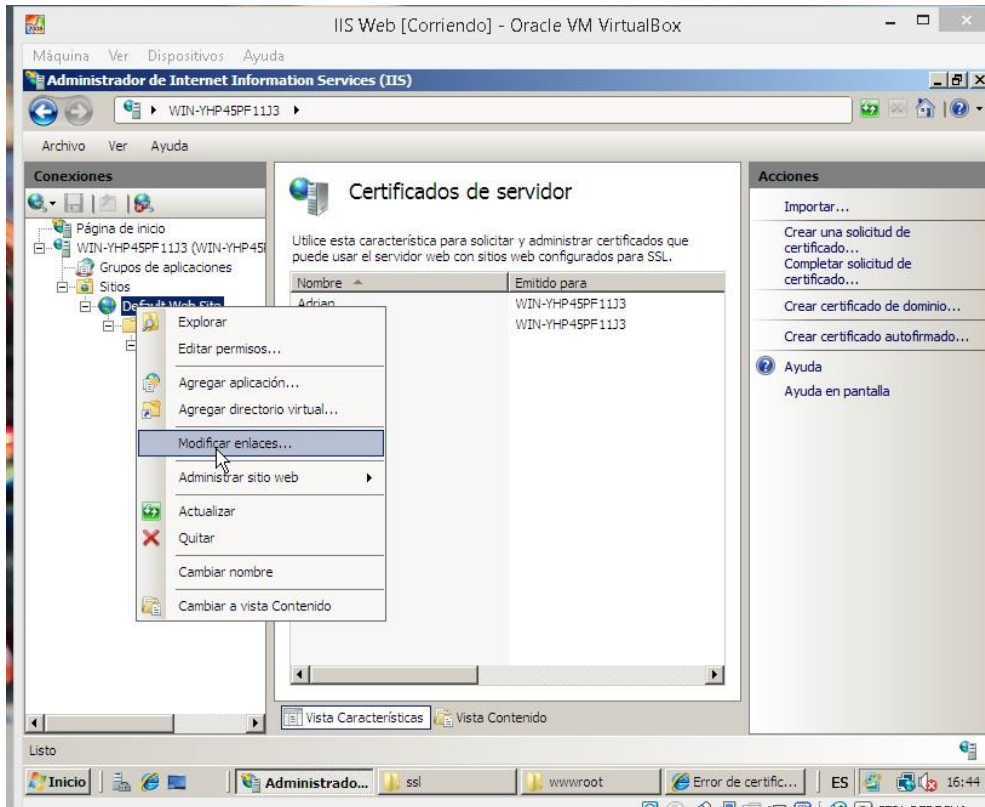
### *IIS – Server 2008*

Una vez tenemos instalado el servicio IIS y un sitio web funcionando, vamos a la herramienta y entramos en **Certificado de Servidores** y a **Crear una solicitud de certificado**.

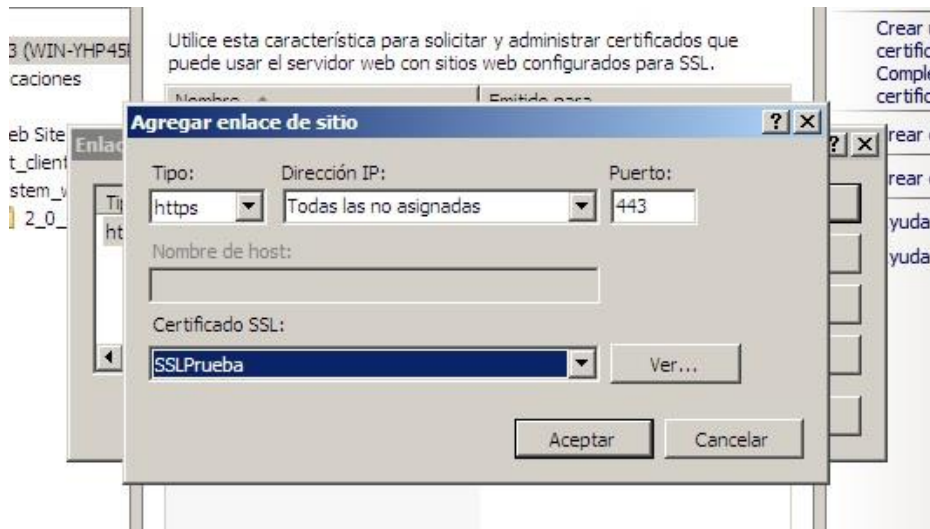


Ahora, creamos el certificado poniendo el nombre el nombre del certificado.  
Ahora vamos a **Default Web Site**, botón derecho y **Modificar enlaces...**

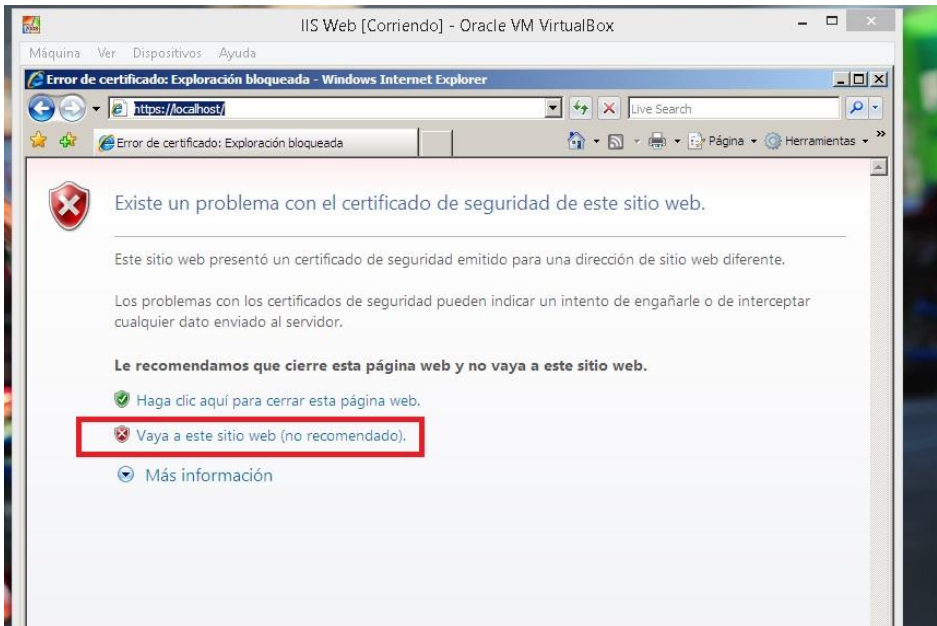




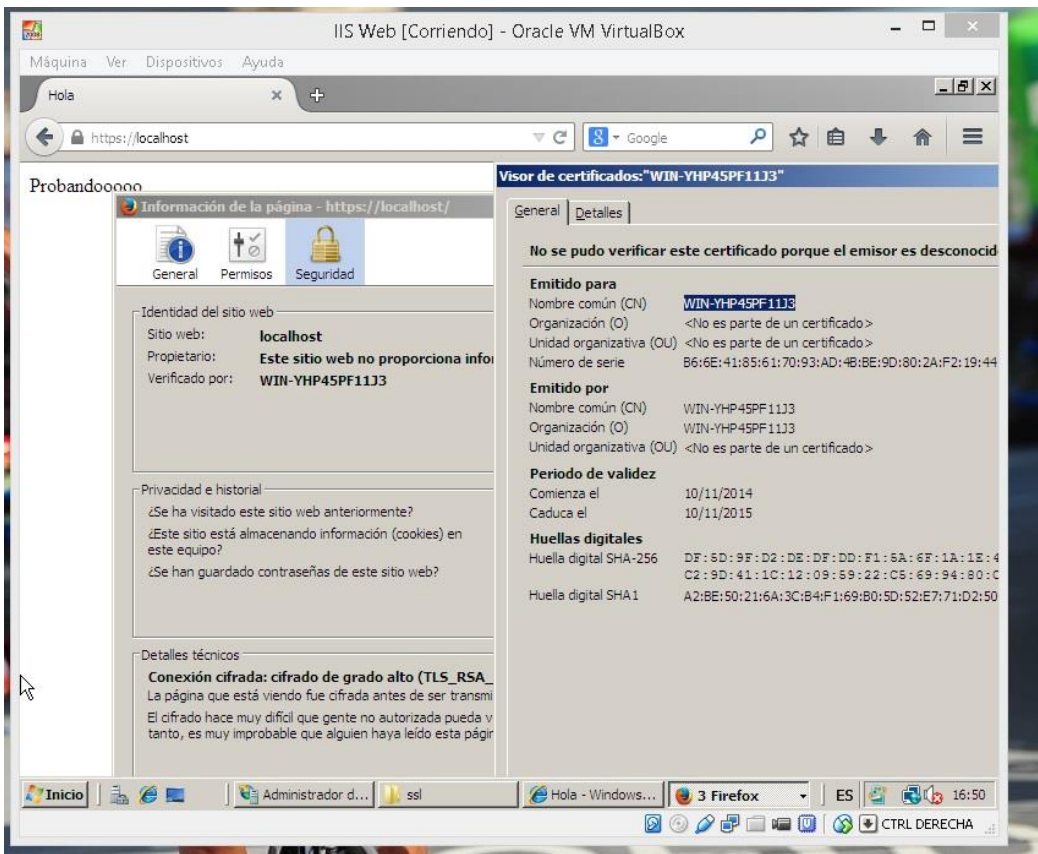
Ahora pulsamos sobre **Agregar** y añadimos el enlace como en la siguiente foto. Después reiniciamos y comprobamos.



Ahora entramos en el navegador, poniendo <https://localhost> y ahora nos saldrá un “error” de que tenemos que aceptar la entrada en ese sitio.



Como vemos funciona correctamente, ahora entramos y comprobamos el certificado.




## Certificado digital propio Apache2

Lo primero que tenemos que hacer es tener nuestro host virtual instalado y configurado, ahora vamos a crear una llave privada, para ello usamos el siguiente comando: (es necesario tener instalado openssl)

Si no tenemos instalado OpenSSL usamos:

```
Apt-get install openssl
```

```
Openssl genrsa -out servidor.key 1024
```



```
Terminal - root@ubuntu: /
; Archivo Editar Ver Terminal Pestañas Ayuda
root@ubuntu:/# apt-get install openssl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssl ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 62 no actualizados.
root@ubuntu:/# op
open      openssl      openssl      openssl
root@ubuntu:/# op
open      openssl      openssl      openssl
root@ubuntu:/# openssl genrsa -out servidor.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
root@ubuntu:/# █
```

Ahora que ya tenemos la clave, vamos a crear una base para el certificado SSL, para ello usamos:

```
Openssl req -new -key servidor.key -out servidor.csr
```

```

Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
root@ubuntu:/# openssl req -new -key servidor.key -out servidor.csr
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]:Es
State or Province Name (full name) [Some-State]:Murcia
Locality Name (eg, city) []:Lorca
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Adrian
Organizational Unit Name (eg, section) []:Adrian
Common Name (e.g. server FQDN or YOUR name) []:www.first.com
Email Address []:adrianlegs@hotmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:
root@ubuntu:/# █

```

Nos preguntará unos datos, se los damos. El más importante es el de **Common name** que debe ser igual que el nombre de dominio donde lo vamos a instalar.

Ahora vamos a generar el certificado SSL.

*openssl x509 -req -days 365 -in servidor.csr -signkey servidor.key -out servidor.crt*

```

Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
root@ubuntu:/# op
open      openssl    openvt     opldecode
root@ubuntu:/# openssl x509 -req -days 365 -in servidor.csr -signkey servidor.
y -out servidor.crt
Signature ok
subject=/C=Es/ST=Murcia/L=Lorca/O=Adrian/OU=Adrian/CN=www.first.com/emailAddress=adrianlegs@hotmail.com
Getting Private key
root@ubuntu:/# █

```

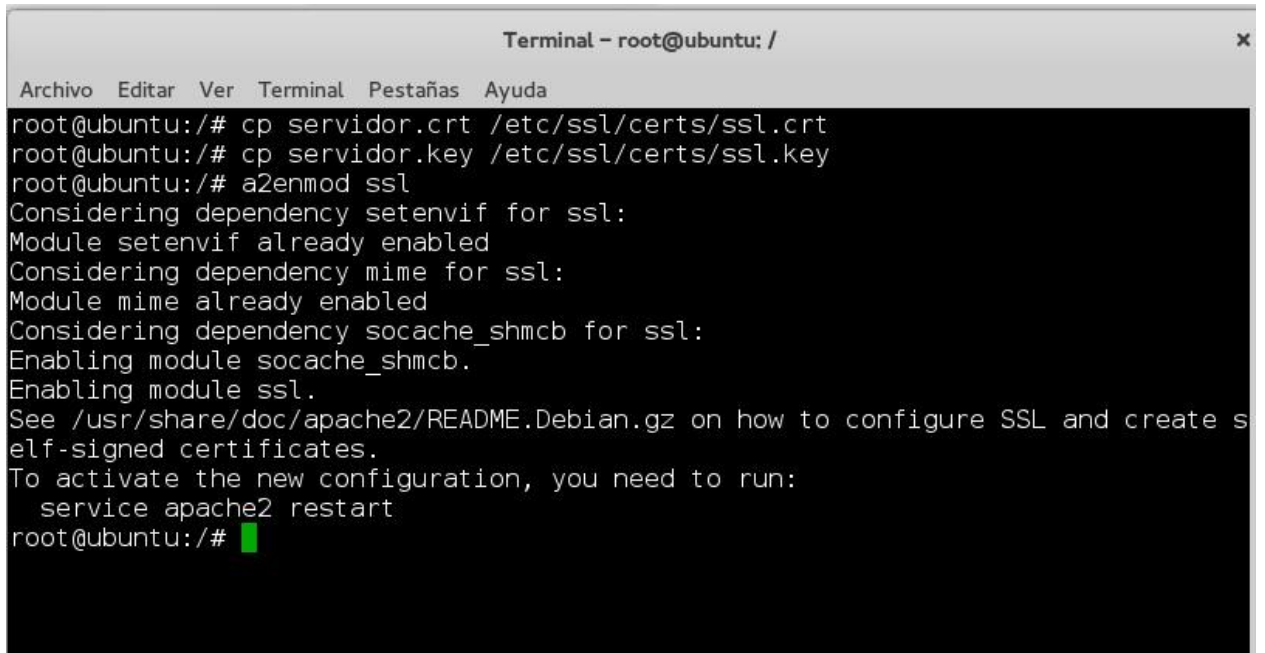
Copiamos el certificado y la clave privada a **/etc/ssl/certs** usamos:

```
Cp servidor.crt /etc/ssl/certs/ssl.crt
```

```
Cp servidor.key /etc/ssl/certs/ssl.key
```

Y activamos el **módulo SSL** de apache, usando:

```
a2enmod ssl
```



```
Terminal - root@ubuntu: /
Archivo Editar Ver Terminal Pestañas Ayuda
root@ubuntu:/# cp servidor.crt /etc/ssl/certs/ssl.crt
root@ubuntu:/# cp servidor.key /etc/ssl/certs/ssl.key
root@ubuntu:/# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@ubuntu:/# █
```

Abrimos con un editor **/etc/apache2/sites-available/default-ssl** y lo dejamos así:

```
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerName www.first.com
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/first.com

    <Directory />
      Options FollowSymLinks
      AllowOverride None
    </Directory>
</VirtualHost>
<Directory /var/www>
  Options -Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow, deny
  allow from all
</Directory>
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined
SSLEngine on
SSLCertificateKeyFile /etc/ssl/certs/ssl.key
SSLCertificateFile /etc/ssl/certs/ssl.crt

BrowserMatch "MSIE [2-6]" \
  nokeepalive ssl-unclean-shutdown \
  downgrade-1.0 force-response-1.0

BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
```

Ahora lo activamos con el siguiente comando:

```
a2ensite default-ssl
```

```
root@ubuntu:/# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@ubuntu:/# █
```

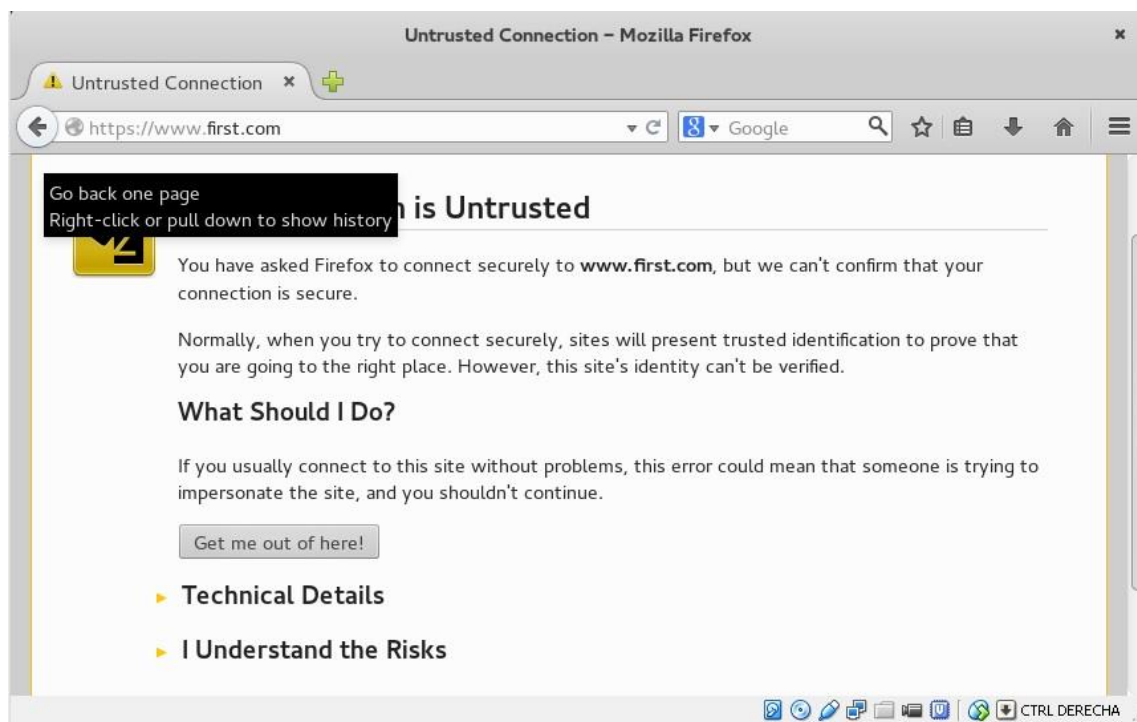
Para que la configuración tenga efecto usamos los siguientes comandos y comprobamos:

```
/etc/init.d/apache2 reload
```

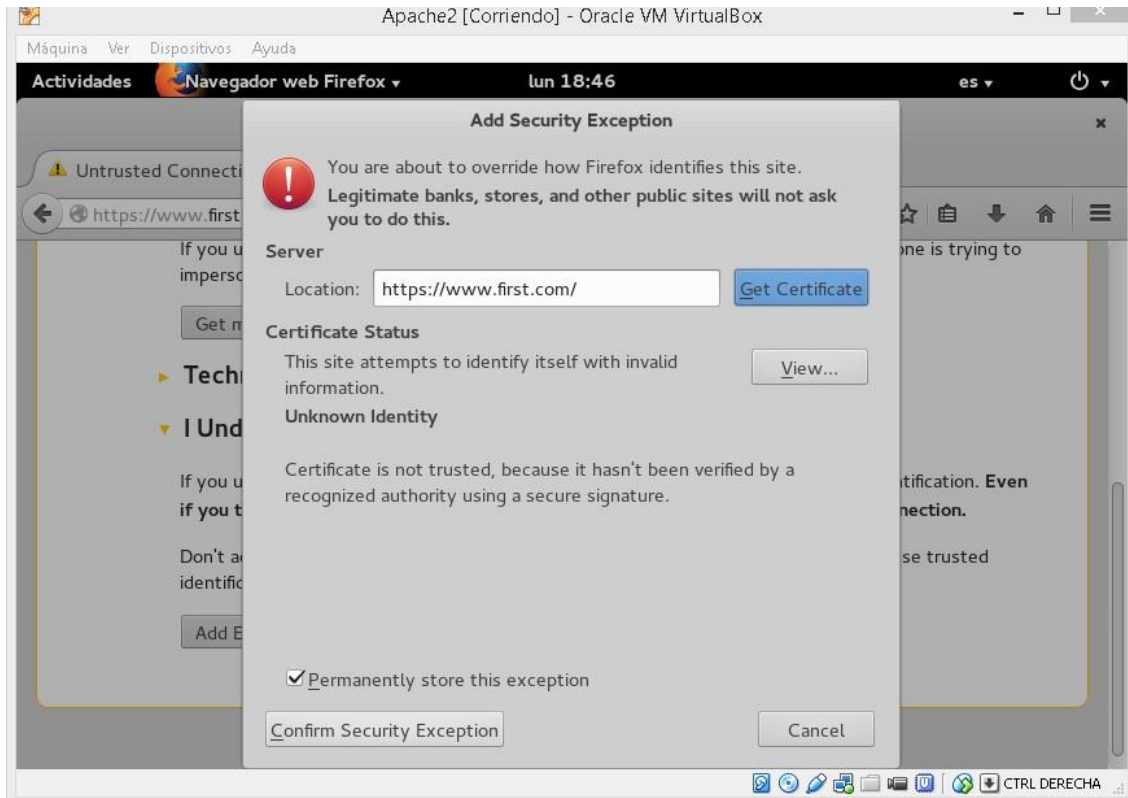
```
/etc/init.d/apache2 restart
```

Ahora entramos al navegador y comprobamos.

```
root@ubuntu:/# apache2ctl configtest
Syntax OK
root@ubuntu:/# /etc/init.d/apache2 reload
* Reloading web server apache2
*
root@ubuntu:/# /etc/init.d/apache2 restart
* Restarting web server apache2
root@ubuntu:/# [ OK ]
```



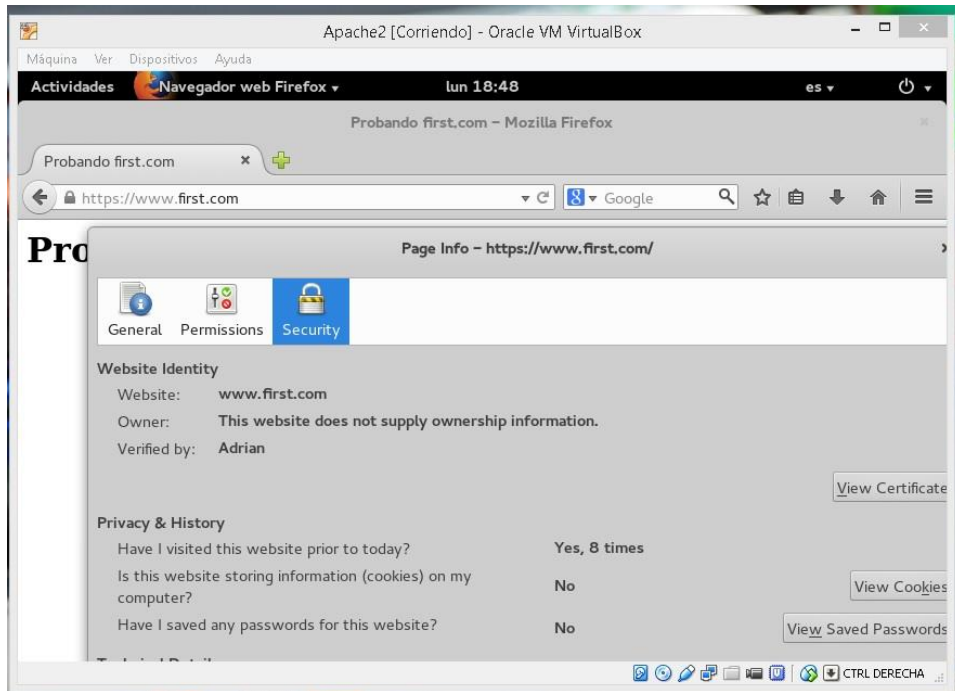
Añadimos la excepción:



Y ya tendremos nuestro certificado digital.





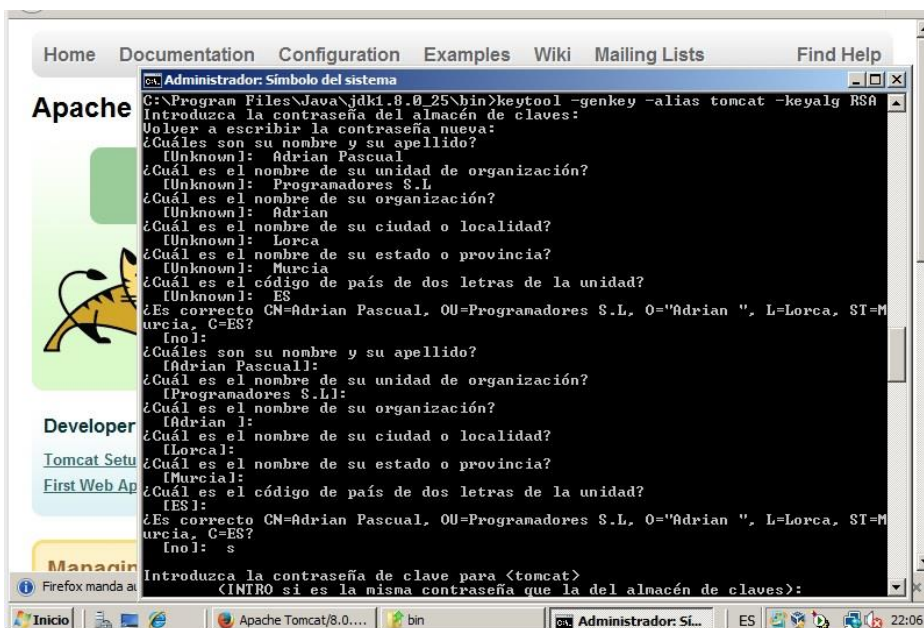


Como vemos tenemos nuestro certificado creado por nosotros mismos funcionando correctamente.

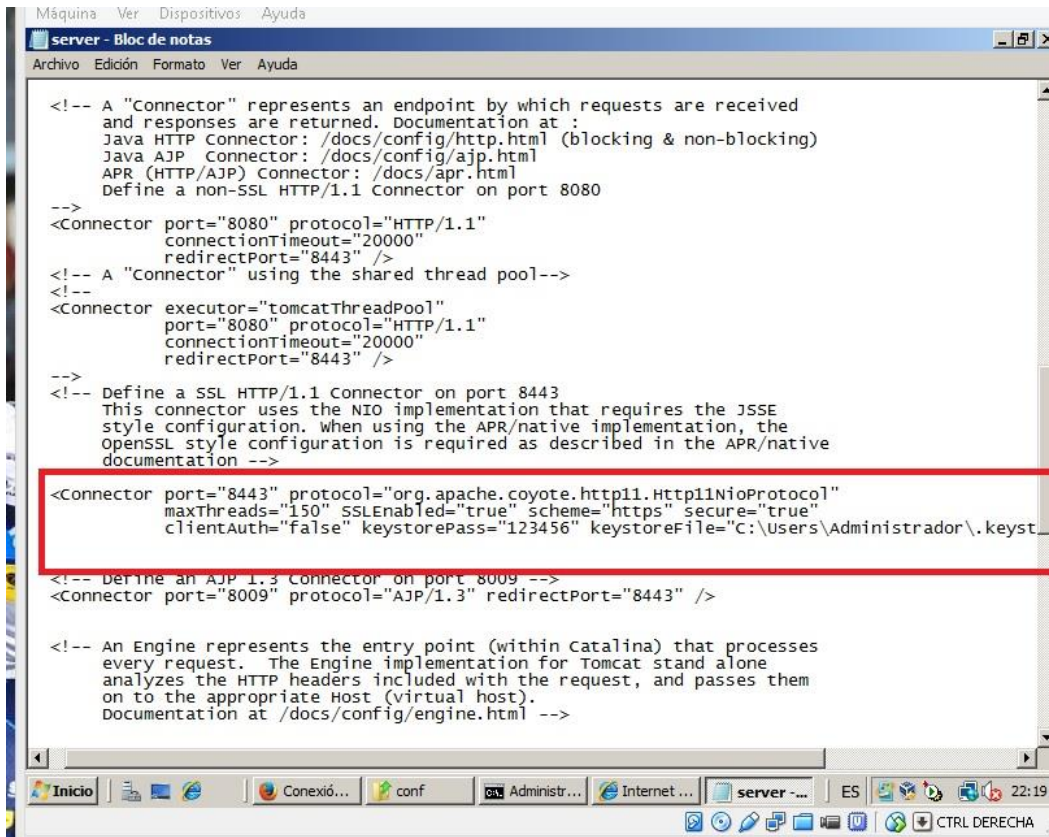
## Certificado digital en Tomcat (Windows)

Para generar el certificado, nos vamos a la ruta de JAVA y ejecutamos el siguiente comando:

*Keytool -genkey -alias tomcat -keyalg RSA*



Tenemos que activar el conector para el modo seguro, para ello nos vamos a la configuración y editamos el archivo **server.xml** y lo dejamos así:

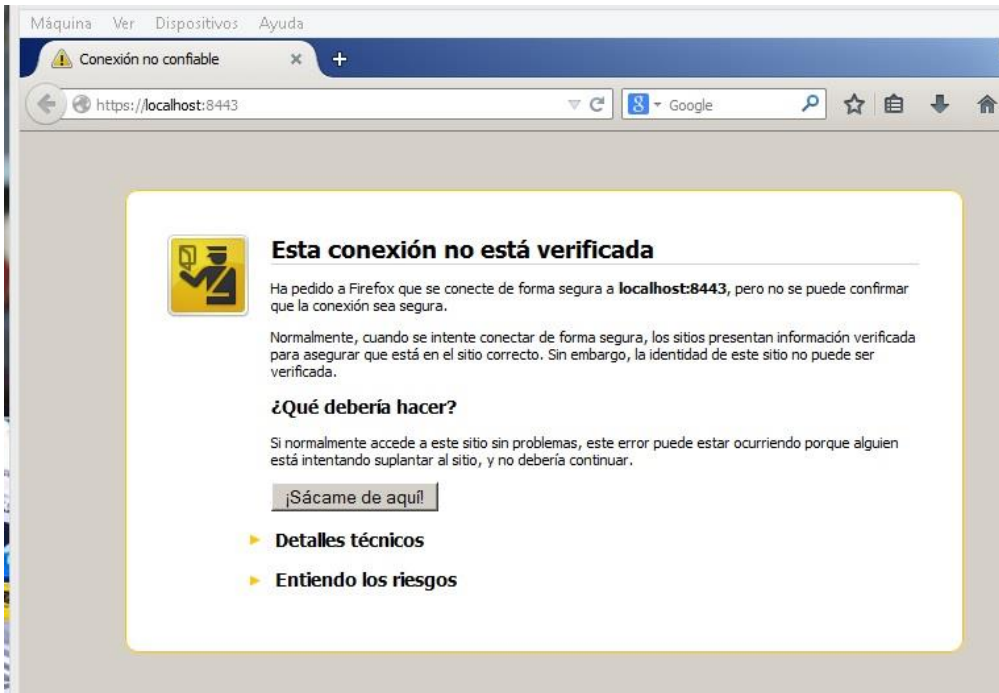


```
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
-->
<!-- Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation that requires the JSSE
style configuration. When using the APR/native implementation, the
openssl style configuration is required as described in the APR/native
documentation -->
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" keystorePass="123456" keystoreFile="C:\Users\Administrador\keyst...
-->
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

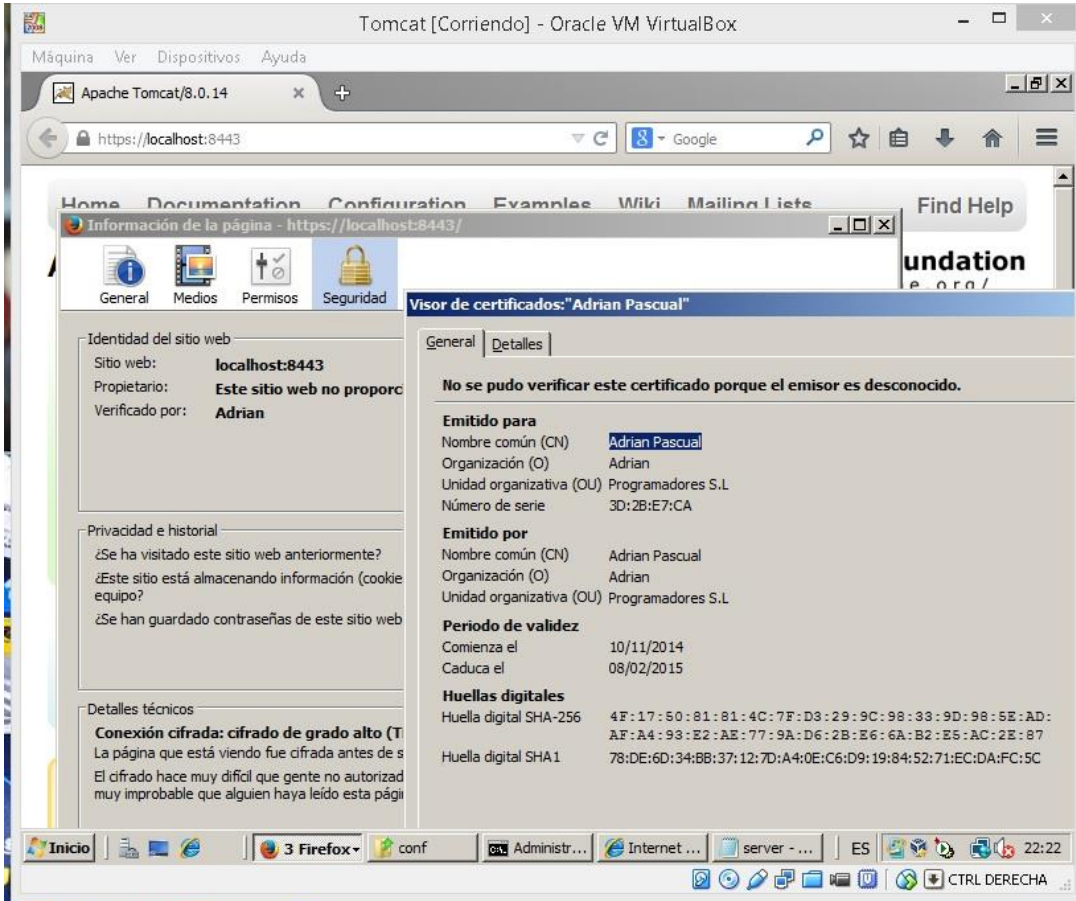
<!-- An Engine represents the entry point (within Catalina) that processes
every request. The Engine implementation for Tomcat stand alone
analyzes the HTTP headers included with the request, and passes them
on to the appropriate Host (virtual host).
Documentation at /docs/config/engine.html -->
```

Debemos poner keystorepass si la contraseña introducida al generar el certificado es distinta a la que pusimos en Tomcat. Por otro lado usamos keystoreFile para indicar dónde está el certificado.

Ahora iniciamos Tomcat y entramos al navegador usando <https://localhost:8443>



Como estamos viendo, parece que funciona correctamente. Ahora vamos a agregar el certificado y a verificarlo.



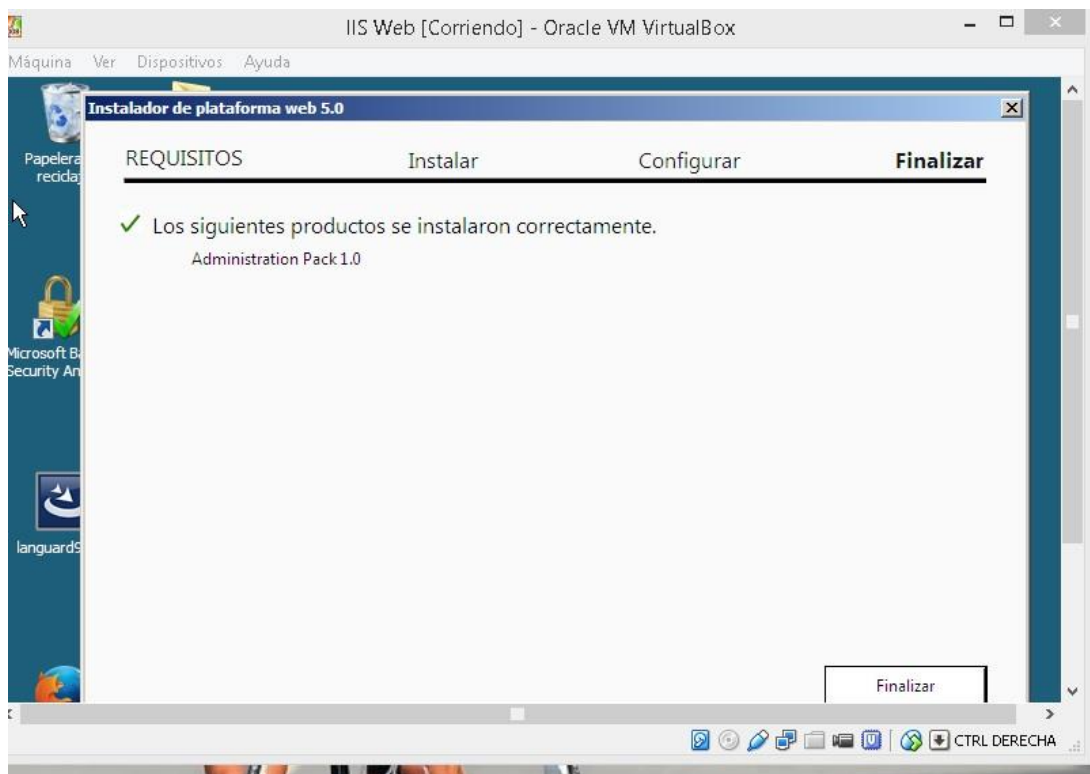
Ya tenemos nuestro certificado digital funcionando en Tomcat con Windows Server 2008.

## Instalación de módulos PHP

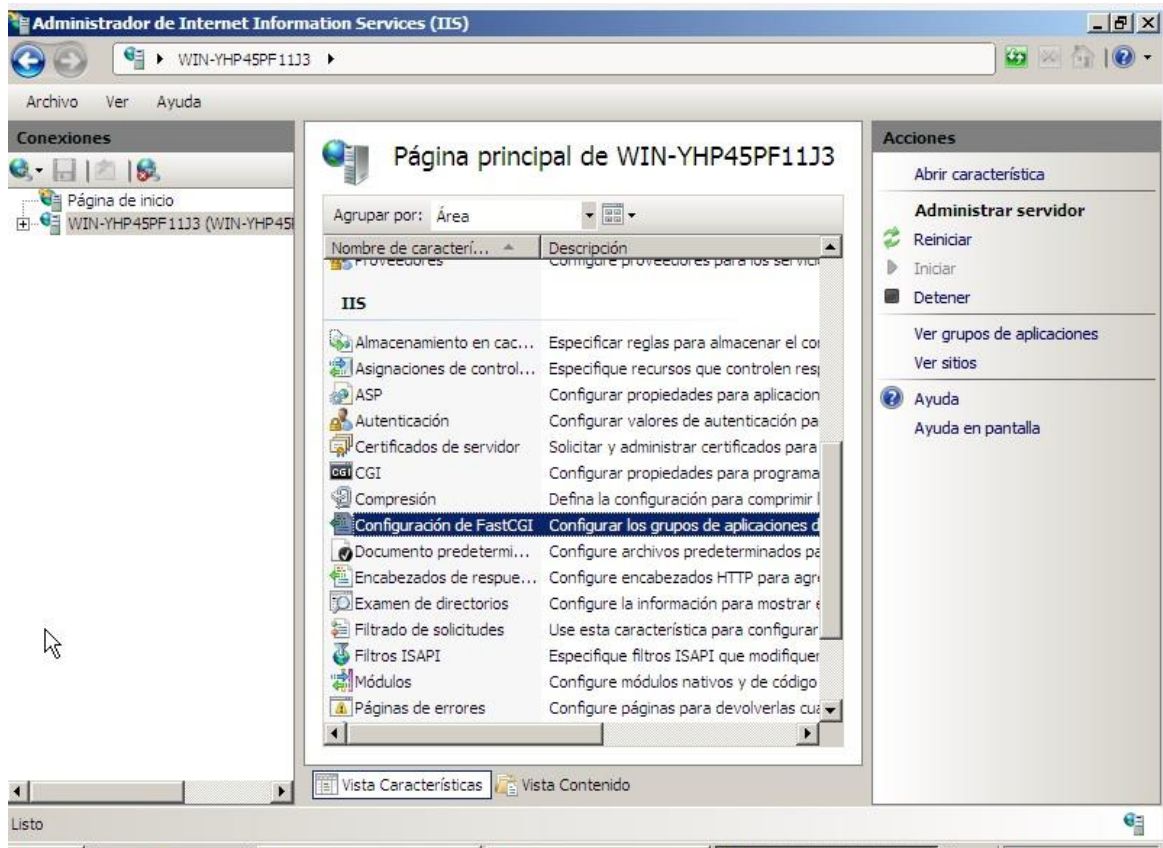
### *IIS Windows Server 2008*

Necesitamos tener instalado el módulo GCI, yo lo instale al instalar el servicio IIS, sino tendríamos que agregárselo. Una vez hecho esto, vamos a actualizarlo.

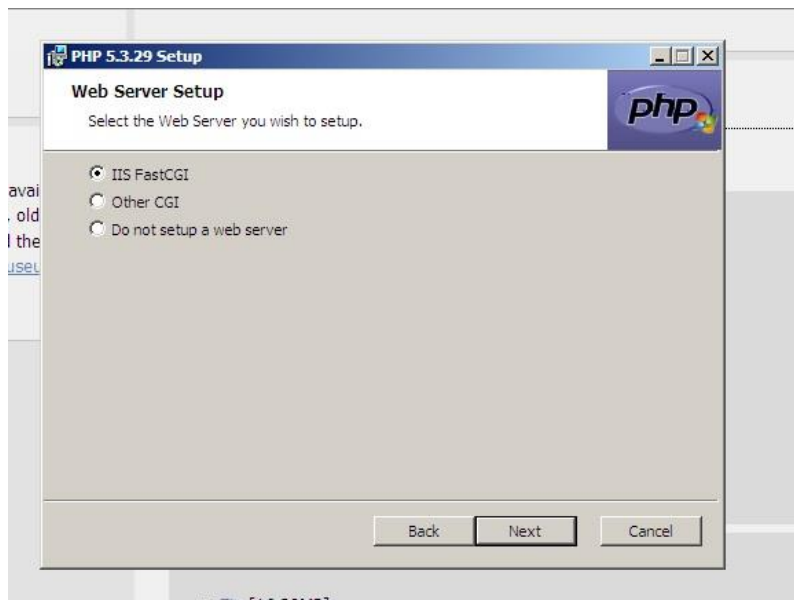
Entramos en [www.iist.net/downloads/administrationpack](http://www.iist.net/downloads/administrationpack) lo descargamos e instalamos.

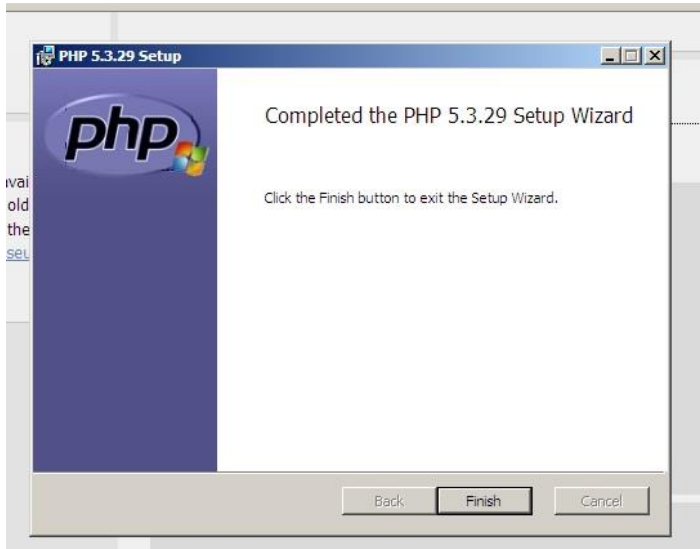


Una vez que termine la instalación entramos en la administración de IIS y buscamos **Configuración de Fast CGI**

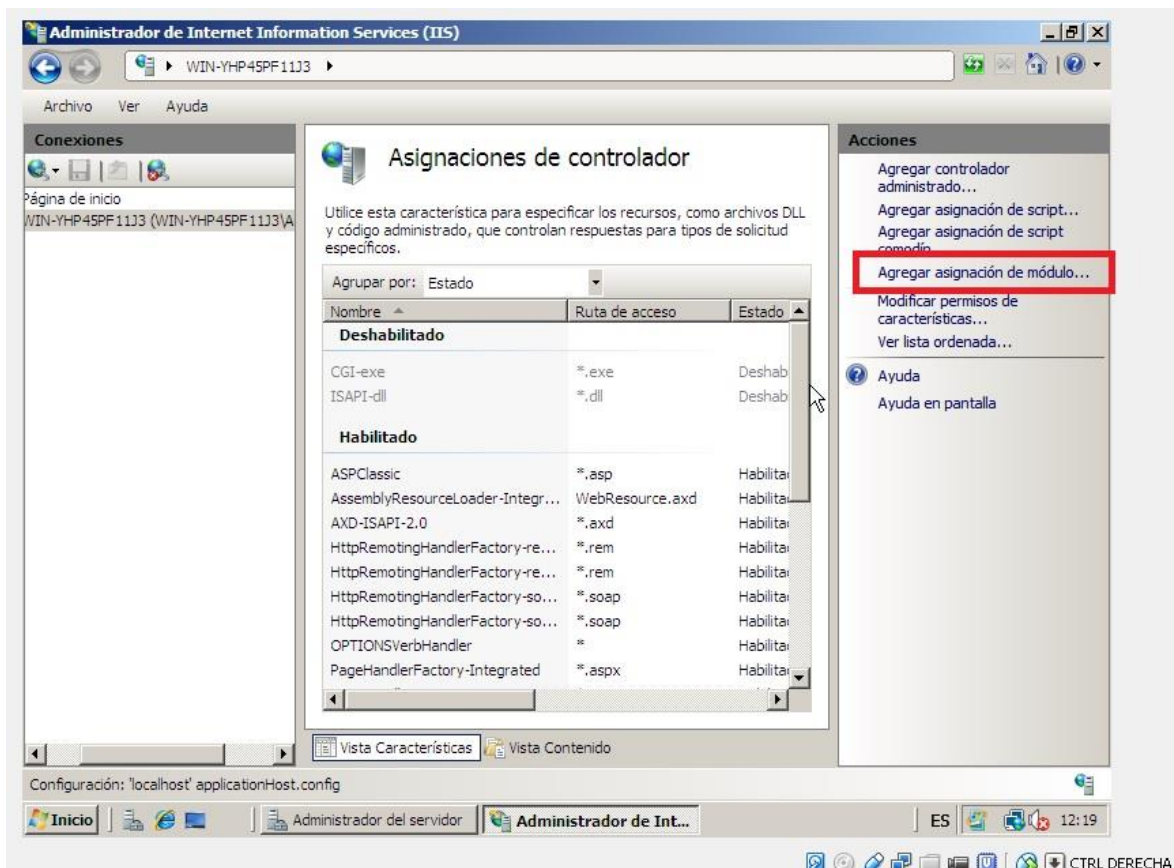


Toca descargarse PHP, para ello entramos en [Windows.php.net/download](http://Windows.php.net/download) y nos los descargamos. A continuación lo instalamos.

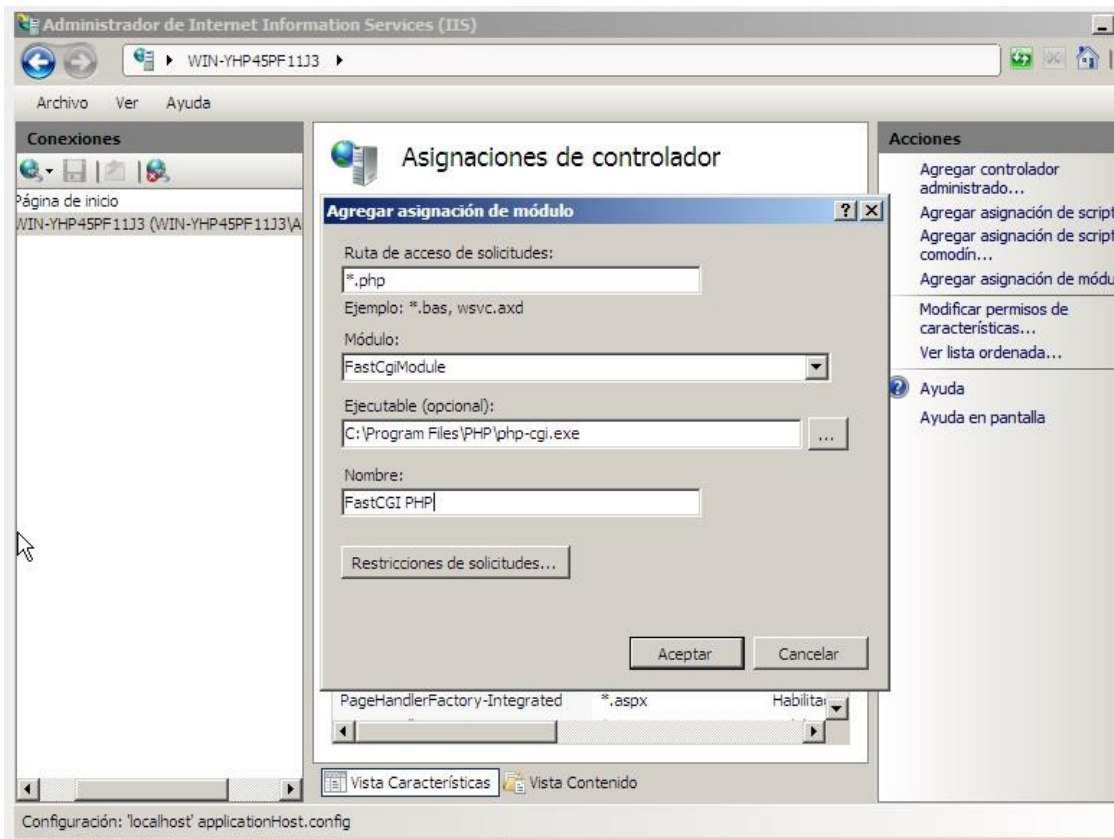




Ahora nos vamos a la administración de IIS y entramos en **Asignación de controladores**. Después a **Agregar asignación a modulo...**



Y lo rellenamos así:



Con esto ya podríamos usar PHP, para ello nos vamos a crear un archivo que contenga algo en PHP por ejemplo:

```
<?php
```

```
phpInfo();
```

```
?>
```

Y comprobamos a ver si funciona.

The screenshot shows a web browser window with the address bar containing 'localhost/hola.php'. The main content area displays 'PHP Version 5.3.29' with the PHP logo. Below this is a table of system and configuration details:

System	Windows NT WIN-YHP45PF11J3 6.0 build 6001 (Windows Server 2008 Enterprise Edition Service Pack 1) i586
Build Date	Aug 15 2014 19:01:45
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	csript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--with-enchant=shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\Program Files\PHP\php.ini
Scan this dir for	(none)

The Windows taskbar at the bottom shows several open applications, including 'hola.php - Bloc ...' which is highlighted with a red box.

Como vemos está funcionando perfectamente PHP en nuestro servidor.

## PHP en Apache2

Es mucho más fácil que en IIS, solo tenemos que poner

Apt-get install php5

Apt-get install libapache2-mod-php5

Y ahora reiniciamos apache.

```

root@ubuntu:~# apt-get install php5
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libapache2-mod-php5 php5-cli php5-common php5-json php5-readline
Paquetes sugeridos:
  php-pear php5-user-cache
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php5 php5 php5-cli php5-common php5-json php5-readline
0 actualizados, 6 se instalarán, 0 para eliminar y 68 no actualizados.
Necesito descargar 4.860 kB de archivos.
Se utilizarán 20,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu/ trusty/main php5-json amd64 1.3.2-2build1
  [34,4 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu/ trusty-updates/main php5-common amd64 5.5.9+dfsg-1ubuntu4.5 [442 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu/ trusty-updates/main php5-cli amd64 5.5.9+dfsg-1ubuntu4.5 [2.164 kB]
26% [3 php5-cli 804 kB/2.164 kB 37%]

```

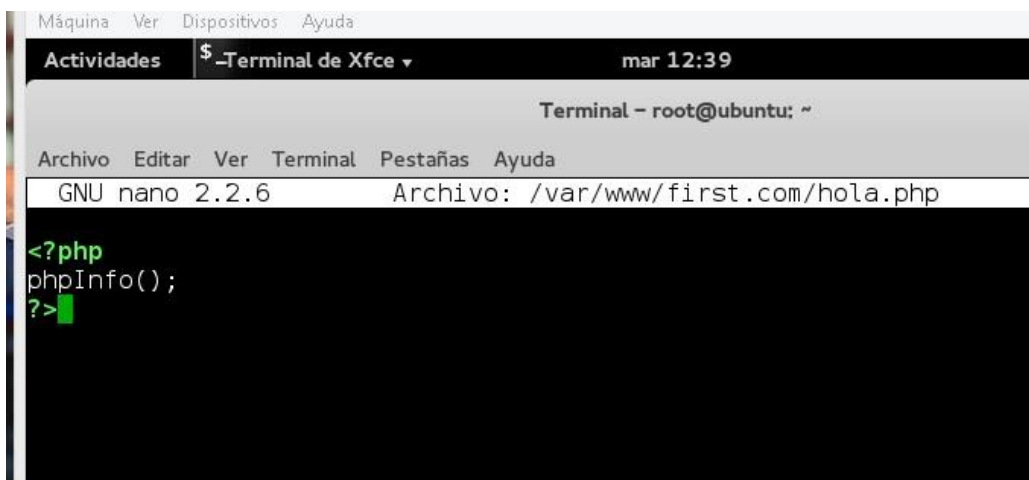


```
root@ubuntu:~# apt-get install libapache2-mod-php5
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
libapache2-mod-php5 ya está en su versión más reciente.
fijado libapache2-mod-php5 como instalado manualmente.
0 actualizados, 0 se instalarán, 0 para eliminar y 68 no actualizados.
root@ubuntu:~# █
```

Ahora reiniciamos apache con

```
/etc/init.d/apache2 restart
```

Y creamos un archivo PHP, por ejemplo en ***/var/www/first.com/hola.php***



The screenshot shows a terminal window titled "Terminal - root@ubuntu: ~" with a menu bar including "Máquina", "Ver", "Dispositivos", and "Ayuda". The terminal content shows the nano editor editing the file `/var/www/first.com/hola.php`. The editor's status bar indicates "GNU nano 2.2.6" and the file path. The code being edited is:

```
<?php
phpInfo();
?> █
```

Vamos al navegador y comprobamos a ver si funciona.



Funciona correctamente, ya tenemos instalado PHP en Apache2 y IIS.

## Comunicación con Tomcat

### *Apache2 con Tomcat*

En los apartados anteriores hemos visto como instalar Tomcat con Apache2, pero Apache2 no es capaz de resolver el JSP por lo que necesitamos que Tomcat lo resuelva y se lo devuelva a apache2 para mostrarlo.

Empezamos por mover el proyecto de ejemplo que trae Tomcat a la carpeta de apache del host que hemos creado. Comprobamos a ver si funciona y nos mostrará el código en JSP o Java.

Lo primero que hacemos es descargarnos el mod\_jk para apache, para ello usamos

*Apt-get install libapache2-mod-jk*

Una vez lo tenemos instalado vamos a configurar el conector. Para ello vamos a **/etc/libapache2-mod-jk/** y abrimos `workers.properties`

*Nano /etc/libapache2-mod-jk/workers.properties*

En ese archivo tendremos que indicar la ruta de JAVA, de Tomcat y añadir algunas líneas que son estas:

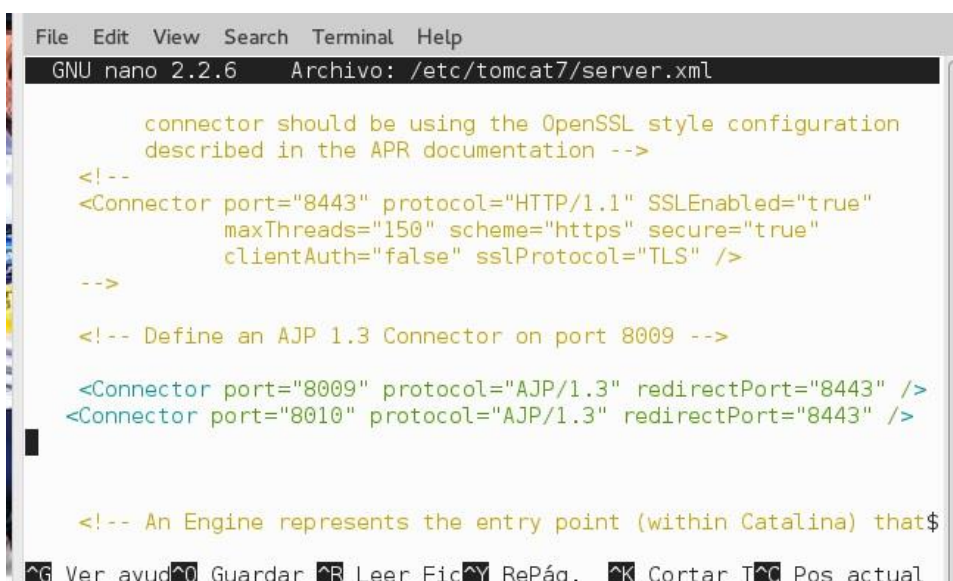
`Worker.list=ajp13_worker, adrian`

Y luego esto:

```
worker.ajp13_worker.port=8009
worker.ajp13_worker.host=localhost
worker.ajp13_worker.type=ajp13
worker.adrian.port=8010
worker.adrian.host=localhost
worker.adrian.type=ajp13
#
# Specifies the load balance factor when used with
# a load balancing worker.
# Note:
# ----> lbfactor must be > 0
# ----> Low lbfactor means less work done by the worker.
worker.ajp13_worker.lbfactor=1
worker.adrian.lbfactor=1
#
```

Entramos en **server.xml** y activamos los conectores en Tomcat

*Nano /etc/tomcat7/server.xml*



```
File Edit View Search Terminal Help
GNU nano 2.2.6 Archivo: /etc/tomcat7/server.xml

    connector should be using the OpenSSL style configuration
    described in the APR documentation -->
<!--
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS" />
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
<Connector port="8010" protocol="AJP/1.3" redirectPort="8443" />

<!-- An Engine represents the entry point (within Catalina) that$

^G Ver ayud ^O Guardar ^R Leer Fic ^Y RePág. ^K Cortar T ^C Pos actual
```

Ahora tenemos que indicar dónde está el archivo `workers.properties`.

Accedemos y abrimos **`/etc/apache2/mods-enabled/jk.conf`**

*Nano /etc/apache2/mod-enabled/jk.conf*

Y lo dejamos así:

```
<IfModule jk_module>

    # We need a workers file exactly once
    # and in the global server
    JkWorkersFile /etc/libapache2-mod-jk/workers.properties

    # Our JK error log
    # You can (and should) use rotatelogs here
    JkLogFile /var/log/apache2/mod_jk.log

    # Our JK log level (trace,debug,info,warn,error)
    JkLogLevel info

    # Our JK shared memory file
    JkShmFile /var/log/apache2/jk-runtime-status
```

Entramos al fichero de configuración de host virtuales

*Nano /etc/apache2/sites-available/000-default.conf*

Y en el `VirtualHost` que queremos añadimos:

`JkMount /examples* adrian`

```
<VirtualHost *:80>
    ServerName www.hola.com
    DocumentRoot /var/www/hola
    JkMount /examples* adrian
</VirtualHost>
```

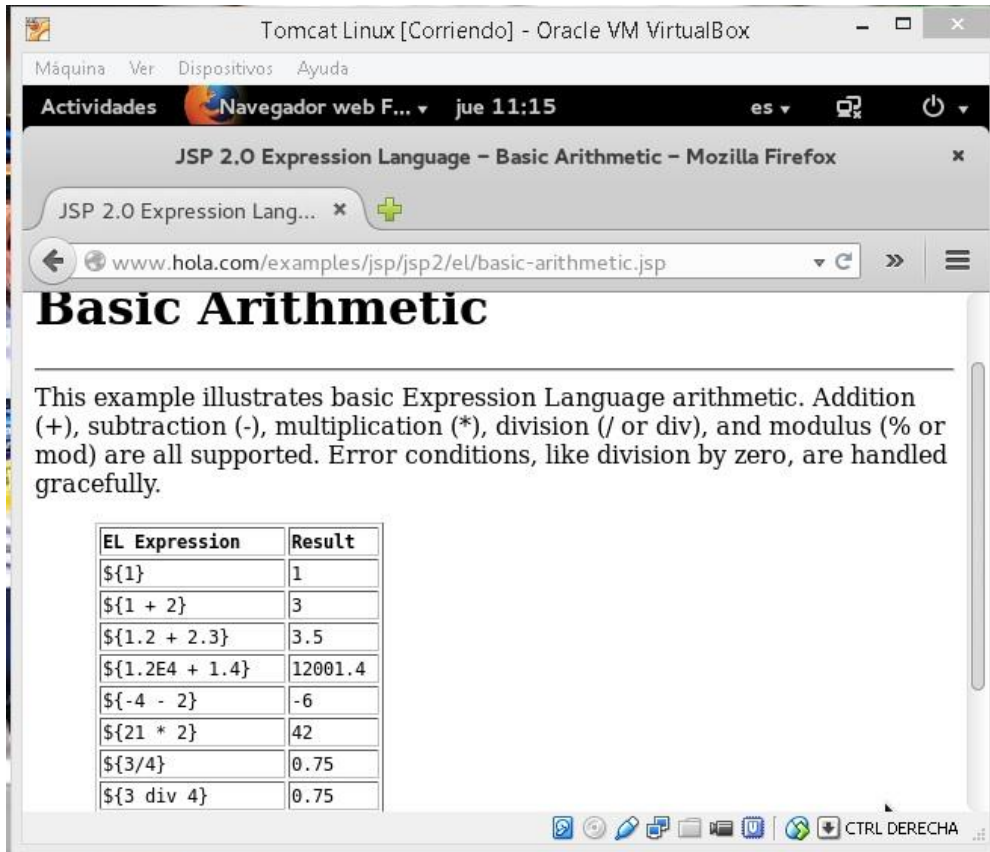
Examples es la carpeta que copiamos anteriormente de Tomcat a Apache y `adrian` es el `worker.list` que definimos antes.

Por último, reiniciamos los dos servicios.

`/etc/init.d/tomcat7 restart`

`/etc/init.d/apache2 restart`

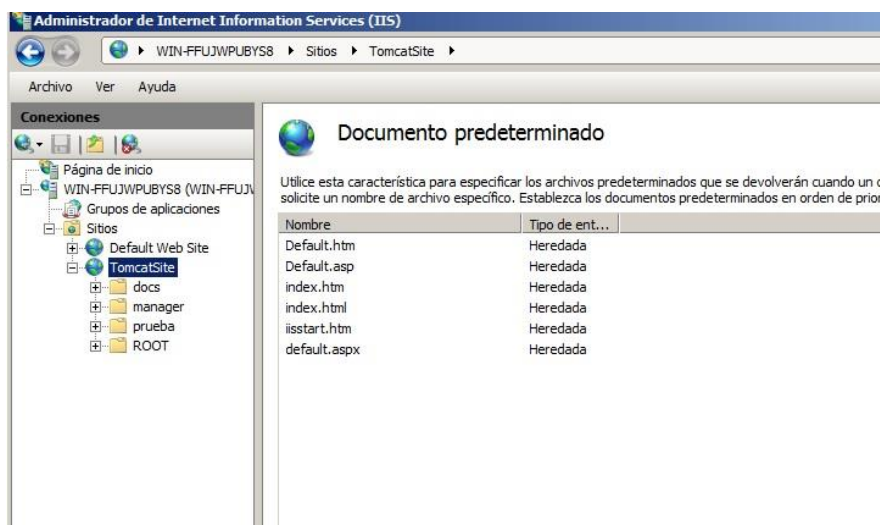
Y comprobamos a ver si funciona.



Como vemos funciona correctamente la comunicación. **(La carpeta debe ser un proyecto creado, por ejemplo con NetBeans o Eclipse)**

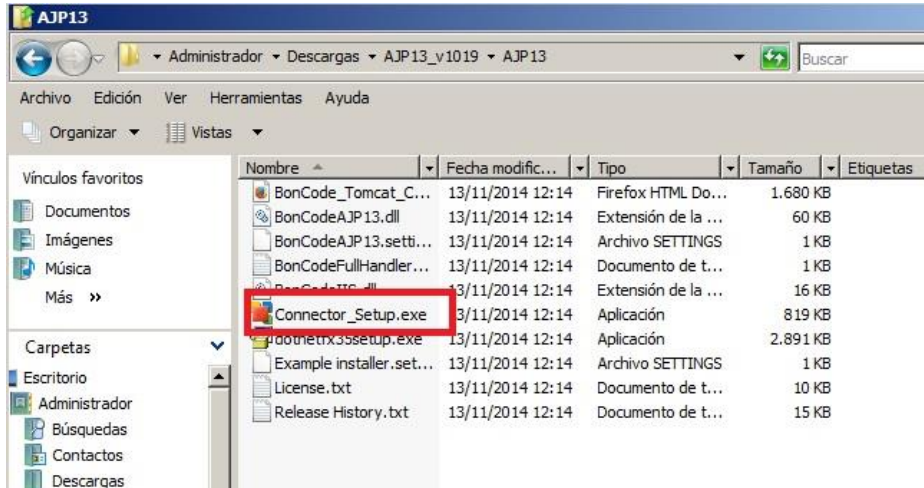
### IIS con Tomcat

Teniendo instalado Tomcat e IIS vamos a ver como conectarlos. Para ello nos vamos a IIS y creamos un sitio que apunte a los directorios de Tomcat



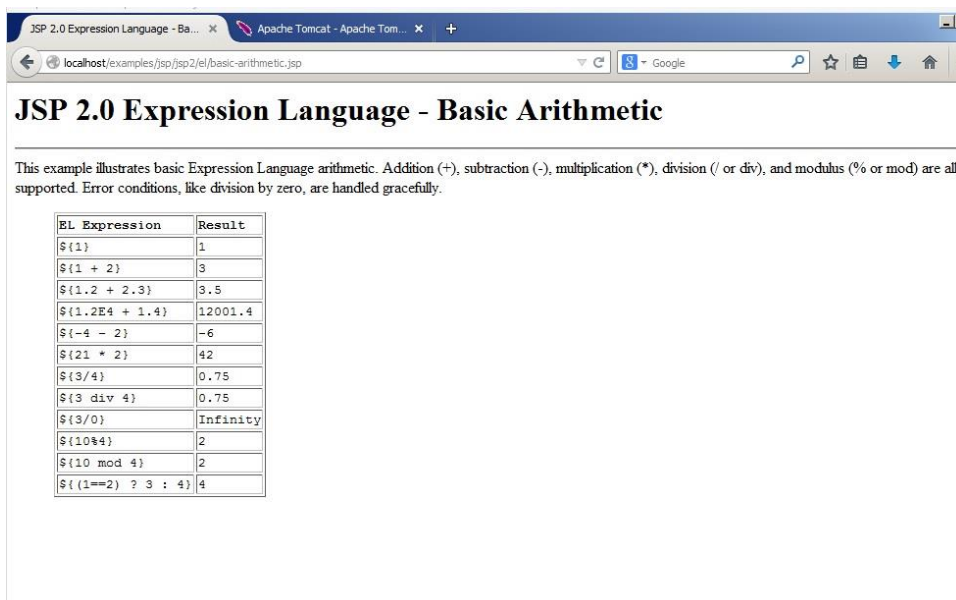
Intentamos acceder a ver si nos conecta pero nos dará un error de que necesitamos el conector. Vamos a descargar un conector de <http://tomcatiiis.riaforge.org> que nos lo hará todo indicándole unas opciones.

Una vez nos lo descargamos, lo extraemos y lo ejecutamos.



Al instalarlo tenemos que elegir la opción de que solo lo haga en el sitio deseado, en mi caso **TomcatSite**.

Ahora reiniciamos IIS y Tomcat e intentamos entrar a ver si nos funciona.



Como vemos funciona correctamente Tomcat con IIS.

## BLOQUE 3: ACTIVIDADES CON SERVIDORES WEB Y CMS

**Cosas básicas para la instalación de los CMS**

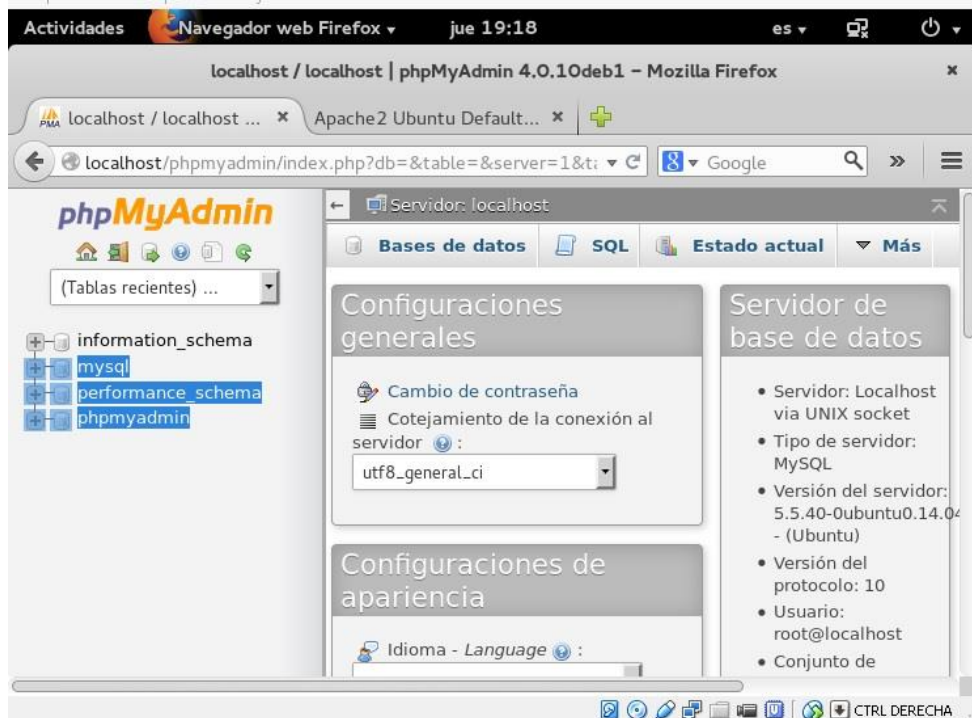
Lo primero que necesitamos es una máquina con conexión a internet. Ahora vamos a instalar apache2, php y mysql con phpmyadmin para gestionar mejor nuestra base de datos. Aplicamos este comando:

```
Apt-get install apache2 mysql-server5.5 phpmyadmin php5-common  
libapache2-mod-php5 php5-mysqlnd
```

Con este comando instalamos lo básico que necesitaremos. Si queremos tener todos los módulos de php utilizamos:

```
Apt-get install php5-mysql php5-curl php-gd php5-idn php-pear php5-imagick  
php5-imap php5-mcrypt php5-memcache php5-ming php5-ps php5-pspell  
php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
```

Ahora vamos a comprobar si funciona tanto apache como phpmyadmin.



Ya tenemos las cosas básicas funcionando ahora vamos a empezar a instalar los hosts virtuales y a continuación el CMS.

## Creación de host virtuales.

Primero creamos las carpetas en **/var/www/**

mkdir joomla, wordpress, Moodle, prestashop

Modificamos el archivo **/etc/hosts** dejándolo así:

```
GNU nano 2.2.6 Archivo: /etc/hosts
127.0.0.1    localhost
127.0.1.1    ubuntu
127.0.1.2    www.joomla.com
127.0.1.3    www.wordpress.com
127.0.1.4    www.moodle.com
127.0.1.5    www.prestashop.com
```

Ahora, nos vamos al directorio de Apache2 a **/etc/apache2/sites-available/000-default.conf** y añadimos los host virtuales, dejando el archivo así:

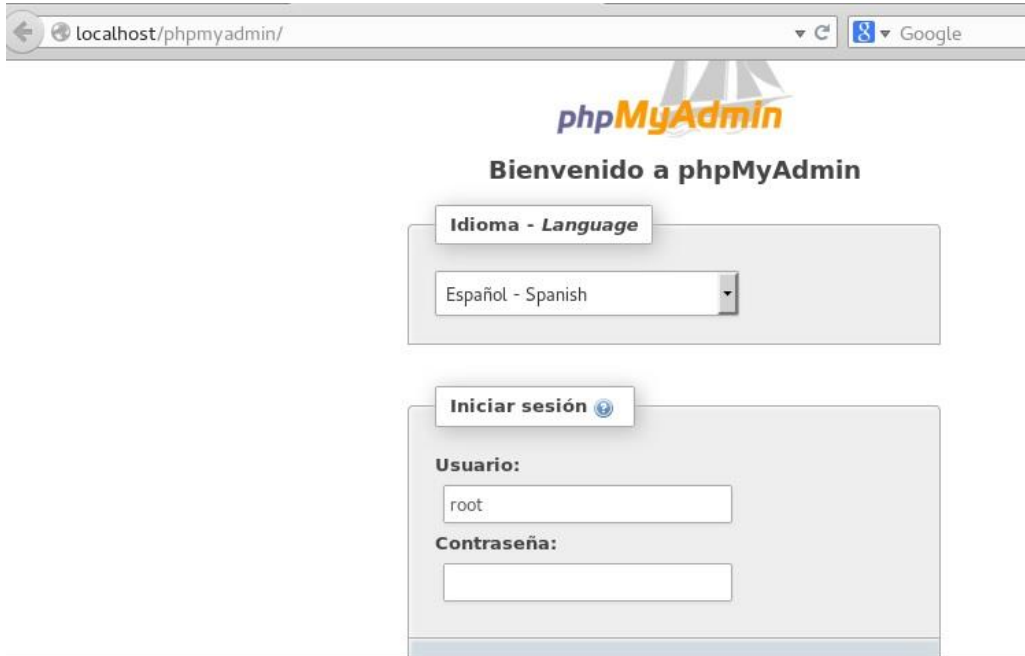
```
root@ub
File Edit View Search Terminal Help
GNU nano 2.2.6 Archivo: ...ache2/sites-a

<VirtualHost www.joomla.com:80>
DocumentRoot /var/www/joomla
ServerName www.joomla.com
</VirtualHost>
<VirtualHost www.wordpress.com:80>
DocumentRoot /var/www/wordpress
ServerName www.wordpress.com
</VirtualHost>
<VirtualHost www.moodle.com:80>
DocumentRoot /var/www/moodle
ServerName www.moodle.com
</VirtualHost>
<VirtualHost www.prestashop.com:80>
DocumentRoot /var/www/prestashop
ServerName www.prestashop.com
</VirtualHost>
```



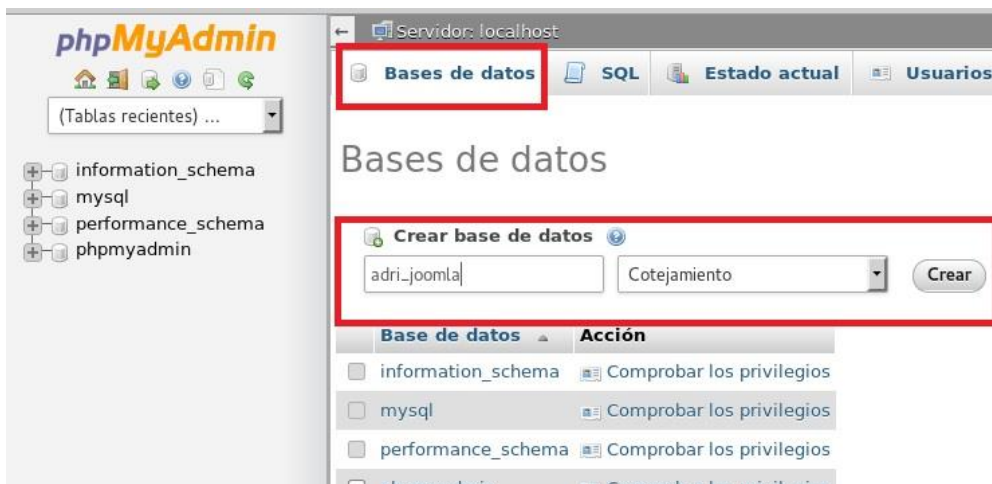
Tenemos los cuatro virtual host creados. Reiniciamos el apache2 y empezamos a descargar los CMS. Antes vamos a crear la base de datos con el **PHPMYADMIN** para acceder, entramos con:

localhost/phpmyadmin.



El usuario por defecto es root y la contraseña la que pusimos en el proceso de instalación.

Una vez dentro, pulsamos sobre **Bases de datos** y la creamos. (Voy a crear una para cada CMS)





Ya tengo las cuatro bases de datos creadas, ahora vamos a proceder con los CMS.

## JOOMLA

Entramos en la página web de Joomla o en la comunidad hispana de Joomla que nos darán Joomla en castellano sin tener que instalar ningún paquete adicional.

[www.joomlaspanish.org](http://www.joomlaspanish.org)



Una vez lo tenemos descargado, extraemos su contenido en **/var/www/joomla**

Unzip `-d /var/www/joomla /root/Descargar/joomla.zip`

Le estamos diciendo que descomprima en **/var/www/Joomla** ese archivo.

```
root@ubuntu:~# ls /var/www/joomla/
administrator  components  includes  language  logs  README.md  tmp
bin            CONTRIBUTING.md  index.php  layouts   media  README.txt  web.config.txt
cache         htaccess.txt  installation  libraries  modules  robots.txt.dist
cli          images        joomla.xml  LICENSE.txt  plugins  templates
root@ubuntu:~#
```

Ya tenemos los archivos, ahora vamos a darle permisos a las carpetas y archivos necesarios.

**El archivo de configuration.php le damos permisos 777 y al acabar la instalación le ponemos 444.**

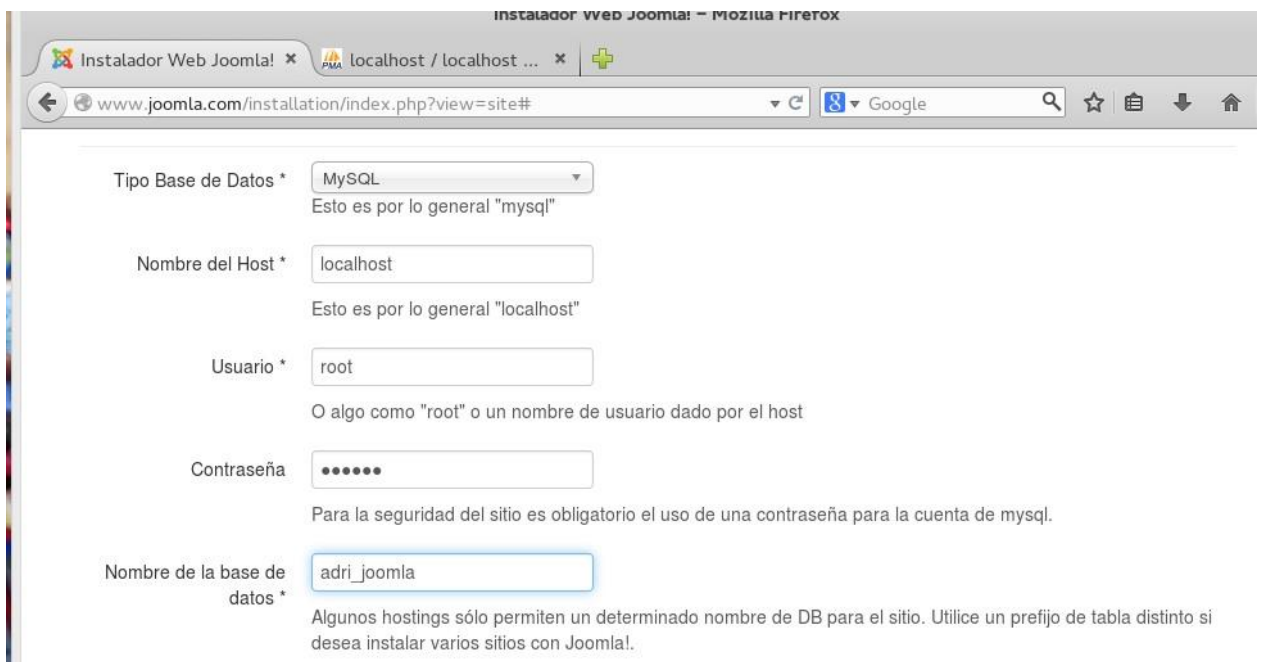
Una vez dado los permisos, vamos con la instalación, para ello entramos a [www.joomla.com](http://www.joomla.com) (que es nuestro virtual host creado anteriormente)



The screenshot shows the Joomla! installation interface in Spanish. The browser address bar shows 'www.joomla.com/installation/index.php?view=site'. The language is set to 'Spanish (Español)'. The 'Configuración principal' section contains the following fields:

- Nombre del sitio \***: Sitio de Adrian Pascual. Subtext: 'Introduzca un nombre para su sitio Web.'
- Meta Descripción**: Probando el CMS Joomla. Subtext: 'Introduzca una descripción general del sitio web para los motores de búsqueda. Por lo general, un máximo de 20 palabras es lo óptimo.'
- E-mail del Administrador \***: adrianlegs@hotmail.com. Subtext: 'Introduzca su dirección de e-mail. Esta será la dirección de correo electrónico del sitio Web vinculada al Super Administrador.'
- Usuario del Administrador \***: Adrian. Subtext: 'Usted puede cambiar el nombre de usuario predeterminado **admin**.'
- Contraseña del Administrador \***: [Redacted]. Subtext: 'Establezca la contraseña para la cuenta de administrador.'

Presionamos sobre **Siguiente**.



The screenshot shows the Joomla! installation interface in Spanish, specifically the database configuration step. The browser address bar shows 'www.joomla.com/installation/index.php?view=site#'. The 'Configuración de base de datos' section contains the following fields:


- Tipo Base de Datos \***: MySQL. Subtext: 'Esto es por lo general "mysql"'
- Nombre del Host \***: localhost. Subtext: 'Esto es por lo general "localhost"'
- Usuario \***: root. Subtext: 'O algo como "root" o un nombre de usuario dado por el host'
- Contraseña**: [Redacted]. Subtext: 'Para la seguridad del sitio es obligatorio el uso de una contraseña para la cuenta de mysql.'
- Nombre de la base de datos \***: adri\_joomla. Subtext: 'Algunos hostings sólo permiten un determinado nombre de DB para el sitio. Utilice un prefijo de tabla distinto si desea instalar varios sitios con Joomla!.'


Este paso es el más importante, ya que es la configuración de la base de datos. Rellenamos los datos como en la foto anterior y presionamos sobre **Siguiente**.

Nos llevará a una página para si queremos gestionar el FTP en nuestro caso, no lo queremos gestionar, por lo que presionamos sobre **Siguiente**.

Joomla!® es software libre distribuido bajo [Licencia Pública General GNU](#).  
Pack creado por Joomla! Spanish 2005-2014 - Series Joomla! 3.2.x

## Instalando...



- Realizar copias de seguridad de las antiguas tablas de la base de datos
- Creación de tablas de la base de datos
- Instalación de los datos de ejemplo
- Creando el archivo de configuración 

Esperamos a que termine de instalar y por fin tendremos nuestro sitio en Joomla.

**¡Felicidades! el pack de Joomla! Spanish ya está instalado.**

POR FAVOR RECUERDE ELIMINAR COMPLETAMENTE EL DIRECTORIO DE INSTALACION.  
Usted no será capaz de seguir más allá de este punto hasta que el directorio de instalación sea eliminado. Esta es una característica de seguridad de Joomla!.

[Eliminar la carpeta de instalación \(installation\)](#)

Nos vamos a **/var/www/joomla** y modificamos el nombre de la carpeta instalación, por ejemplo, a2sqfinstalacionasd23d esto es para que nadie pueda ejecutarlo, o bien lo eliminamos.

Cuando lo hayamos eliminado, ya entramos en nuestro sitio web.



## Wordpress

Hacemos igual que antes, nos vamos a google y buscamos **WordPress** y nos lo descargamos de la página oficial, si es posible en castellano.

¡Bienvenida! ¡Bienvenido!

¡Bienvenid@ a WordPress España!

WordPress es una avanzada plataforma semántica de publicación personal orientada a la estética, los estándares web y la usabilidad. WordPress es libre y, al mismo tiempo, gratuito.

Dicho de forma más sencilla, WordPress es el sistema que utilizas cuando deseas trabajar con tu herramienta de publicación en lugar de pelearte con ella.



## Descargar

Aquí puedes descargar la versión completa de WordPress en español de España. Para

Descargar WordPress 4.0  
zip — 6.7 MB

Descargar .tar.gz — 6.2 MB

Una vez que lo tenemos descargado, lo descomprimos en la carpeta **/var/www/wordpress** que es la indicada para este gestor de contenidos.

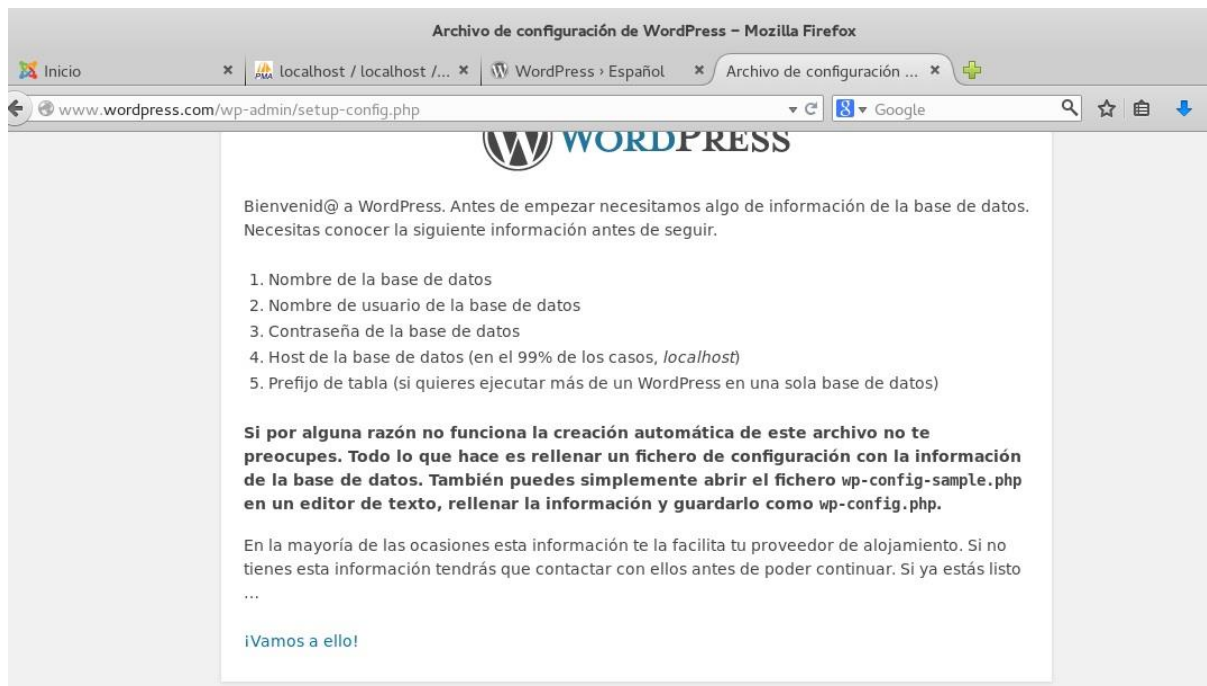
Unzip -d /var/www/wordpress /root/Descargar/wordpress.zip

```

root@ubuntu:/var/www/wordpress# ls
index.php          wp-admin          wp-cron.php       wp-mail.php
licencia.txt      wp-blog-header.php wp-includes       wp-settings.php
license.txt       wp-comments-post.php wp-links-opml.php wp-signup.php
readme.html      wp-config-sample.php wp-load.php       wp-trackback.php
wp-activate.php  wp-content       wp-login.php      xmlrpc.php
root@ubuntu:/var/www/wordpress# █

```

Ahora que ya los tenemos, vamos a instalarlo. Necesitamos la base de datos, pero yo ya la cree anteriormente. También podríamos usar la de Joomla, pero sería muchísimas tablas y mejor por separado.



Pulsamos sobre **Vamos a ello!**

<b>Nombre de la base de datos</b>	<input type="text" value="adri_wordpress"/>	El nombre de la base de datos en la que quieres que se ejecute WP.
<b>Nombre de usuario</b>	<input type="text" value="root"/>	Tu nombre de usuario de MySQL.
<b>Contraseña</b>	<input type="text" value="123456"/>	...y la contraseña de MySQL.
<b>Host de la base de datos</b>	<input type="text" value="localhost"/>	Si no funciona localhost tendrás que contactar con tu proveedor de alojamiento para que te diga cual es.
<b>Prefijo de tabla</b>	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

Enviar

**Posiblemente nos indicará que creemos el archivo nosotros, eso es debido a que no tenemos permisos de escritura y no lo puede crear el automáticamente.**

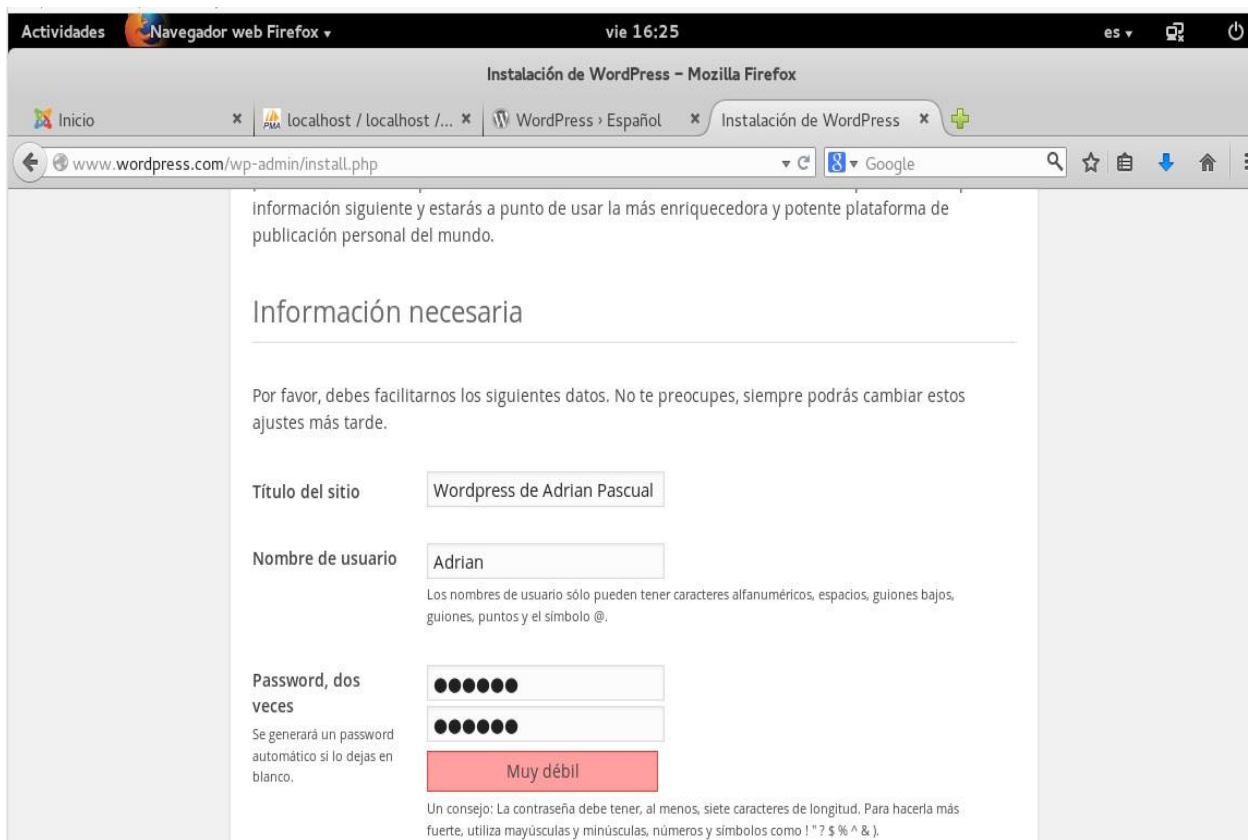
Introducimos los datos de la base de datos, como vemos, hay un fallo de “seguridad” de WordPress permitiendo que se visualice la contraseña en la instalación.



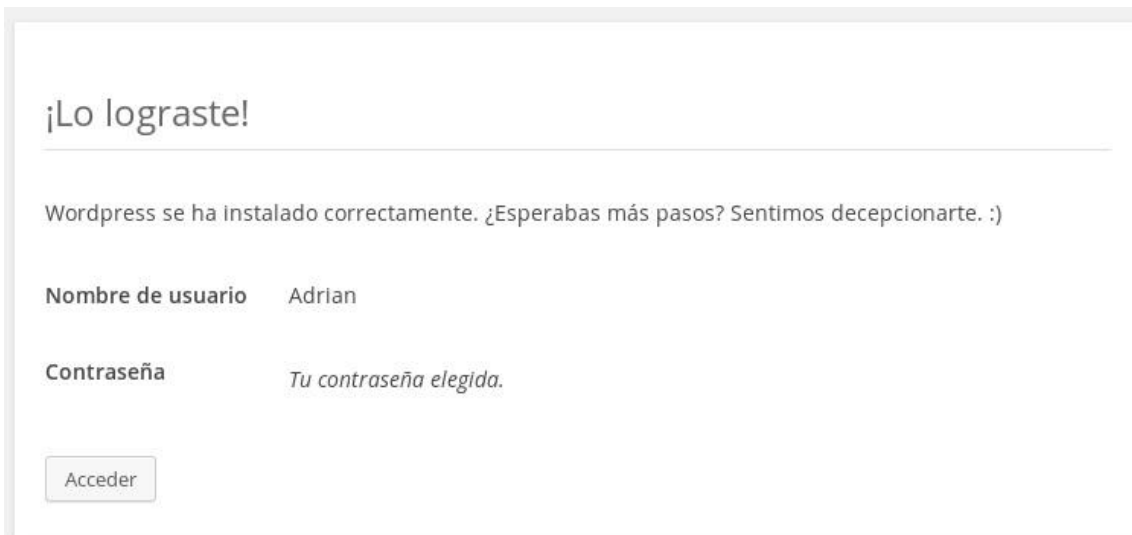
¡Todo correcto! Ya has terminado esta parte de la instalación. Ahora WordPress puede comunicarse con tu base de datos. Si estás preparado es momento de ...

[Iniciar la instalación](#)

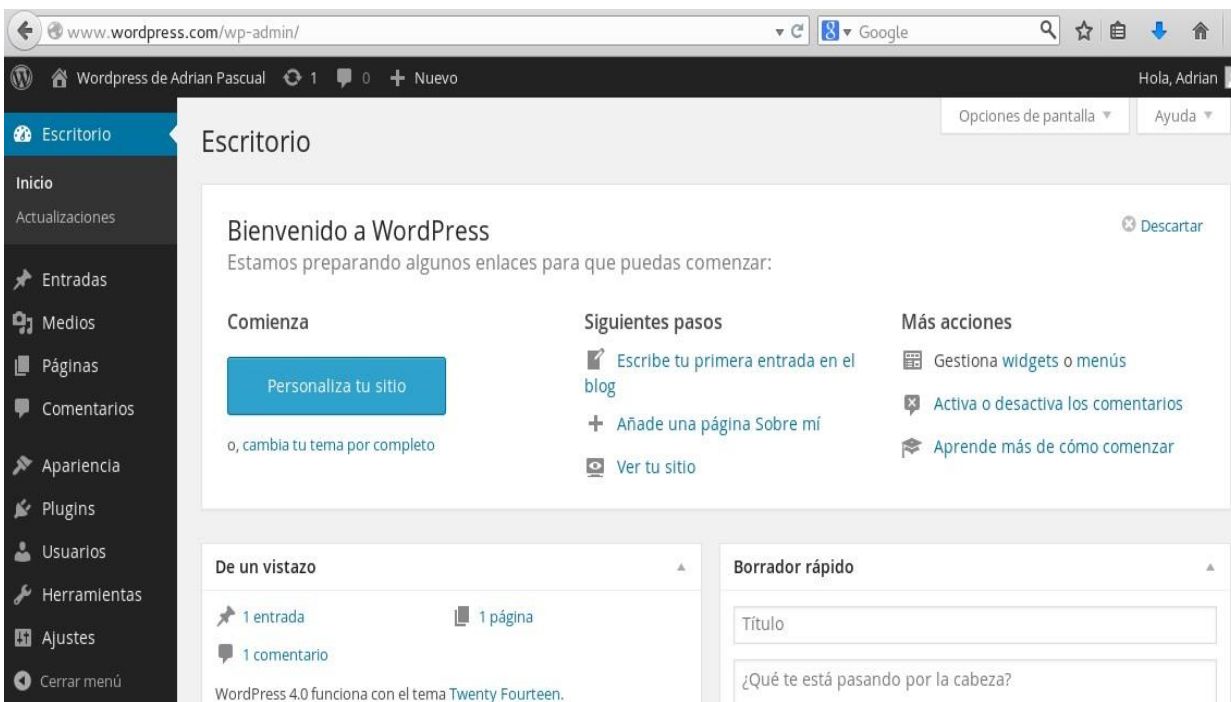
Seguimos con la instalación.



Presionamos sobre continuar, debajo de la página.

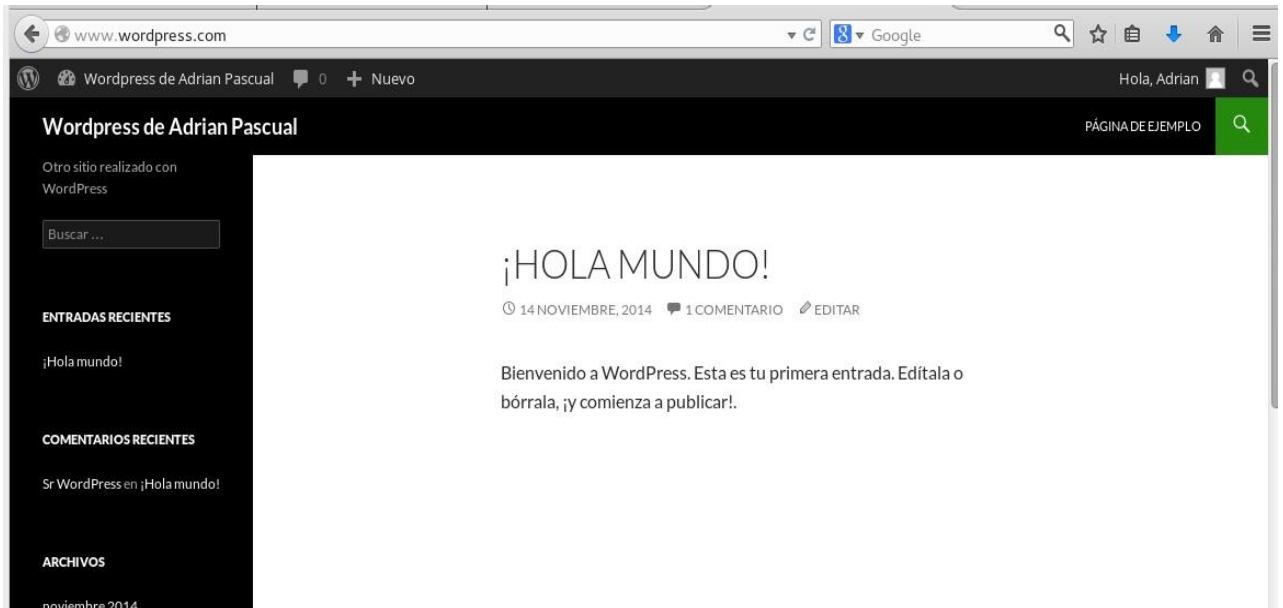


Ya tenemos WordPress instalado, ahora vamos a comprobar que funciona correctamente.



Aquí podemos ver la zona de administración y por último vemos el portal del sitio.





Ya tenemos WordPress funcionando correctamente.

## Moodle

Entramos en la página web de Moodle, y descargamos la última versión, acto seguido lo descomprimos en la carpeta **/var/www/Moodle**.

Unzip `-d /var/www/Moodle /root/Descargar/Moodle.zip`

```

root@ubuntu:/var/www/moodle# ls
admin          COPYING.txt    iplookup      portfolio
auth          course        lang          question
availability  draftfile.php lib           rating
backup        enrol         local        README.txt
badges        error         login        report
behat.yml.dist file.php      mdeploy.php  repository
blocks        files        mdeploytest.php rss
blog          filter       message      tag
brokenfile.php githash.php  mnet         tags.txt
cache         grade        mod          theme
calendar     group        my           TRADEMARK.txt
cohort       help_ajax.php notes        user
comment      help.php     phpunit.xml.dist userpix
completion   index.php    pix          version.php
composer.json install.php   plagiarism   webservice
config-dist.php install.php  pluginfile.php
root@ubuntu:/var/www/moodle#
  
```

Una vez tenemos los archivos, vamos al navegador y empezamos la instalación.

## Instalación

Idioma

### Seleccionar idioma

Por favor, seleccione un idioma para el proceso de instalación. Este idioma se usará también como idioma por defecto del sitio, si bien puede cambiarse más adelante.

Idioma [Siguiente »](#)

**Necesitamos tener instalada la versión php5-curl de php5.**

Seleccionamos el controlador de base de datos **MySql**.

Servidor de la base de datos	<input type="text" value="localhost"/>
Nombre de la base de datos	<input type="text" value="adri_moodle"/>
Usuario de la base de datos	<input type="text" value="root"/>
Contraseña de la base de datos	<input type="text" value="123456"/>
Prefijo de tablas	<input type="text" value="mdl_"/>
Puerto de la base de datos	<input type="text" value="3306"/>
Socket Unix	<input type="text"/>

[« Anterior](#) [Siguiente »](#)

Presionamos siguiente y aceptamos los términos.

## Instalación

### Moodle - Modular Object-Oriented Dynamic Learning Environment

#### Copyright

Copyright (C) 1999 en adelante, Martin Dougiamas (<http://moodle.com>)

Este programa es software libre: usted puede redistribuirlo y/o modificarlo bajo los términos de la GNU (General Public License) publicada por la Fundación para el Software Libre, ya sea la versión 3 de dicha Licencia, o (a su elección) cualquier versión posterior.

Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA; sin la garantía implícita de COMERCIALIZACIÓN o IDONEIDAD PARA UN PROPÓSITO PARTICULAR.

Vea la página de información de Licencia de Moodle para más detalles: <http://docs.moodle.org/en/License>

¿Ha leído y comprendido los términos y condiciones?

Si nuestro entorno pasa los entornos mínimos nos mostrará un mensaje en verde y presionamos sobre **Continuar**.

#### logstore\_database

Éxito

#### logstore\_legacy

Éxito

#### logstore\_standard

Éxito

Si se nos ha instalado bien, presionamos sobre **Continuar**.

## ▼ General

Nombre de usuario\*

Escoger un método de identificación: ?

Cuentas manuales

La contraseña debería tener al menos 8 caracter(es), al menos 1 dígito(s), al menos 1 minúscula(s), al menos 1 mayúscula(s), al menos 1 caracter(es) no alfanuméricos

Nueva contraseña\* ?

 DesenmascararForzar cambio de contraseña ? 

Nombre\*

Apellido(s)\*

Editamos los datos del administrador y presionamos en **Actualizar**.

## Nuevos ajustes - Ajustes de la página principal

Nombre completo del sitio  
fullnameNombre corto para el sitio (una  
palabra)  
shortnameDescripción de la página  
principal  
summary

Rellenamos los datos básicos del sitio y le damos a **Guardar Cambios**. Ya tenemos Moodle instalado.

The screenshot shows the Moodle website interface. At the top, the browser address bar displays 'www.moodle.com'. The user is logged in as 'Adrián Pascual'. The main heading is 'Moodle Adrián Pascual'. On the left, there are two navigation menus: 'NAVEGACIÓN' (Navigation) and 'ADMINISTRACIÓN' (Administration). The 'NAVEGACIÓN' menu includes 'Página Principal', 'Área personal', 'Páginas del sitio', 'Mi perfil', and 'Cursos'. The 'ADMINISTRACIÓN' menu includes 'Ajustes de la página principal', 'Activar edición', 'Editar ajustes', 'Usuarios', 'Filtros', 'Informes', 'Copia de seguridad', and 'Restaurar'. The main content area is titled 'Cursos disponibles' and features a button 'Agregar un nuevo curso'. On the right, there is a 'CALENDARIO' (Calendar) for November 2014, showing the days of the week and the dates.

Como vemos funciona correctamente nuestro CMS Moodle.

### PrestaShop.

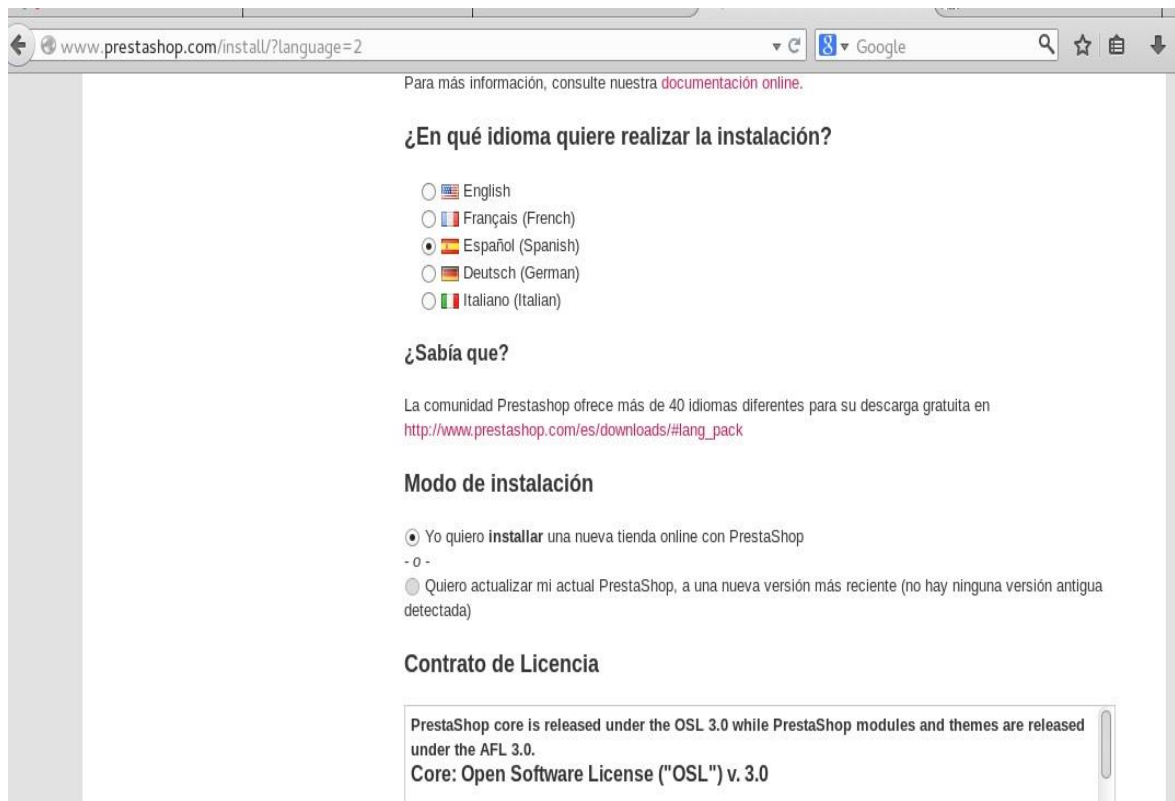
Para empezar, entramos a google y nos descargamos PrestaShop de la web oficial. Una vez lo tengamos descargado lo extraemos en

**`/var/www/prestashop`**

*`Unzip -d /var/root/prestashop /root/Descargas/prestashop.zip`*

```
root@ubuntu:/var/www/prestashop# ls
404.php                guest-tracking.php    order-slip.php
addresses.php          header.php            override
address.php           history.php           pagination.php
admin                 identity.php         password.php
attachment.php        images.inc.php       pdf-invoice.php
authentication.php    img                  pdf-order-return.php
best-sales.php        index.php             pdf-order-slip.php
cache                 init.php              prices-drop.php
cart.php              install               product.php
category.php          js                    products-comparison.php
changepcurrency.php  log                   product-sort.php
classes               mails                  search.php
cms.php               manufacturer.php      sitemap.php
config                modules                sitemap.xml
contact-form.php     my-account.php       statistics.php
controllers           new-products.php     stores.php
css                   order-confirmation.php supplier.php
docs                  order-detail.php     themes
download              order-follow.php     tools
footer.php            order-opc.php         translations
get-file.php          order.php             upload
root@ubuntu:/var/www/prestashop#
```

Ahora vamos al navegador y comenzamos la instalación.



Elegimos el idioma, seleccionamos que queremos una nueva tienda online y aceptamos los términos, y presionamos sobre continuar.

Nos saldrán errores diciendo que tenemos que darle permisos a las carpetas, se los damos con CHMOD.

✓ Bienvenido

► **Compatibilidad sistema**

Configuración de base de datos

Configuración de la tienda

¡Instalación terminada!

**Asegúrese por favor de que cada uno de los siguientes parámetros ha sido validado.**

Si tiene alguna duda o pregunta, echele un vistazo a nuestra [Documentación en Wiki](#) y/o [en nuestro foro comunitario](#).

✓ **Su configuración es válida, haga clic en "Siguiente" para continuar**

PHP parámetros (para asistencia, pregúntale a su proveedor de alojamiento web)

**Configure su base de datos llenando los siguientes campos:**

Por favor crear un base de datos MySQL y despues verifique sus parámetros debajo, para asistencia por favor preguntale a su proveedor de alojamiento web.

Servidor:	<input type="text" value="localhost"/>
Nombre de la base de datos:	<input type="text" value="adri_prestashop"/>
Inicio de la base de datos	<input type="text" value="root"/>
Contraseña de la base de datos	<input type="password" value="•••••"/>
Tipo de base de datos:	<input type="text" value="InnoDB"/>
Prefijo de las tablas:	<input type="text" value="ps_"/>

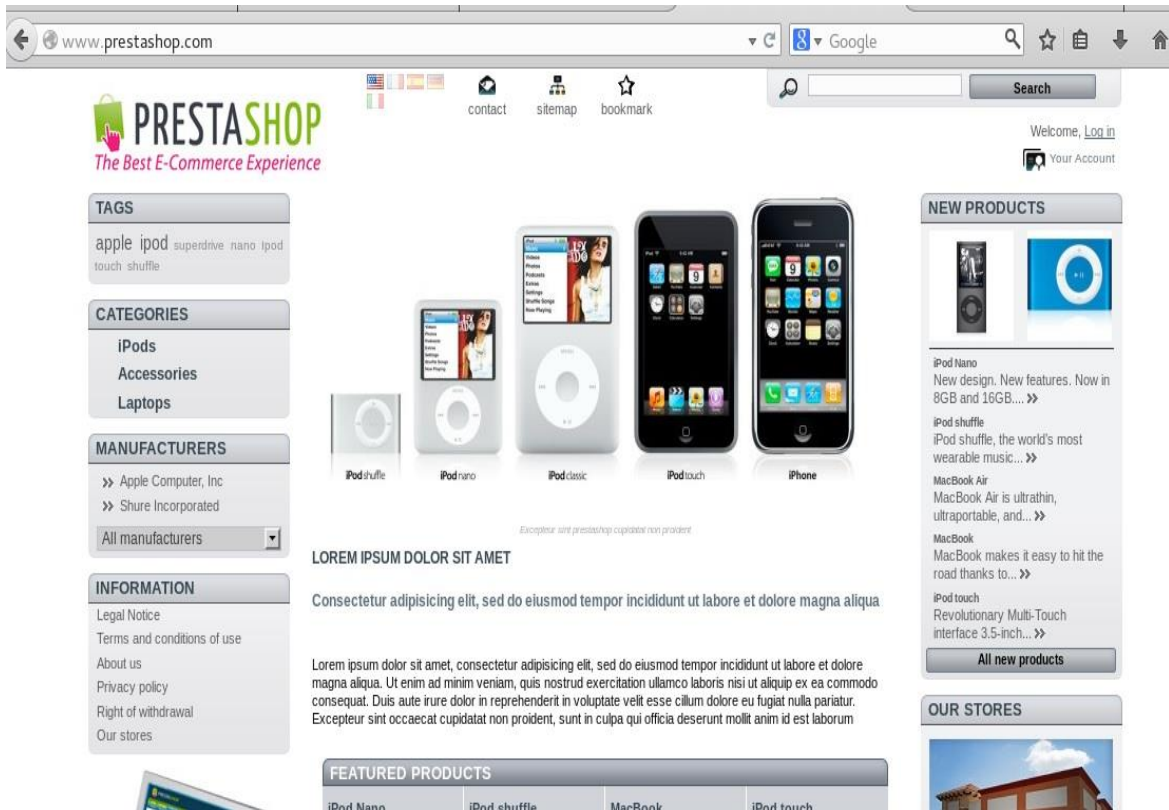
**¡Pruebe ahora!**

✓ ¡La base de datos se ha encontrado!

Rellenamos los datos de la base de datos y le damos a **Siguiente**.

Modo Catálogo:	<input checked="" type="radio"/> Sí <input type="radio"/> No	<i>Si activa esta opción, se desactivarán todas las aplicaciones de compra. Puede activar dicha opción posteriormente en el panel de administración.</i>
Nombre:	<input type="text" value="Adrian"/> *	
Apellido:	<input type="text" value="Pascual Pareja"/> *	
Dirección de email:	<input type="text" value="adrianlegs@hotmail.com"/> *	
Contraseña de la tienda:	<input type="password" value="•••••"/> *	
Confirmar la contraseña:	<input type="password" value="•••••"/> *	
<input type="checkbox"/> Recibe estos datos por email		<i>Esta opción se puede bloquear si su configuración de correo electrónico está incorrecta.</i>

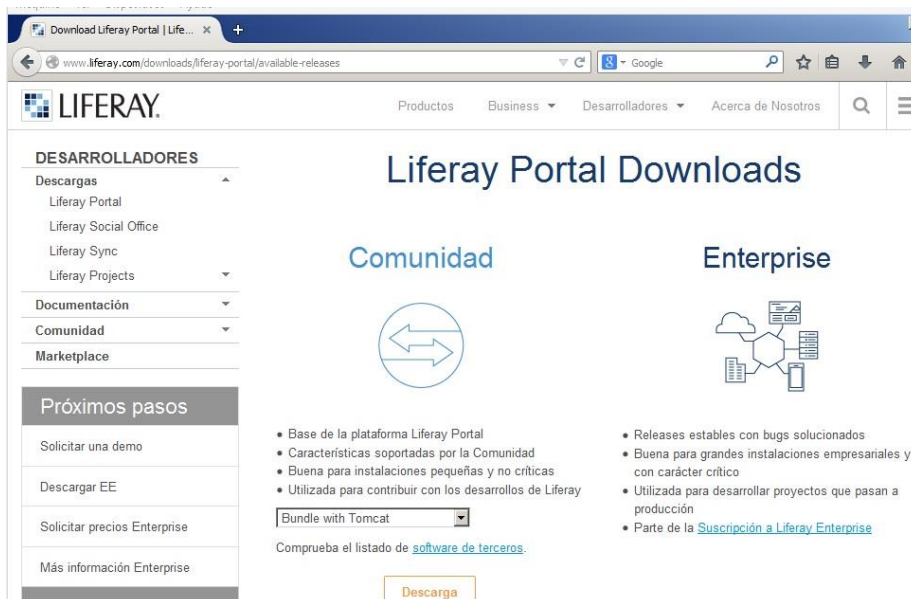
Y continuamos.



Ya tenemos nuestro PrestaShop instalado y funcionando correctamente.

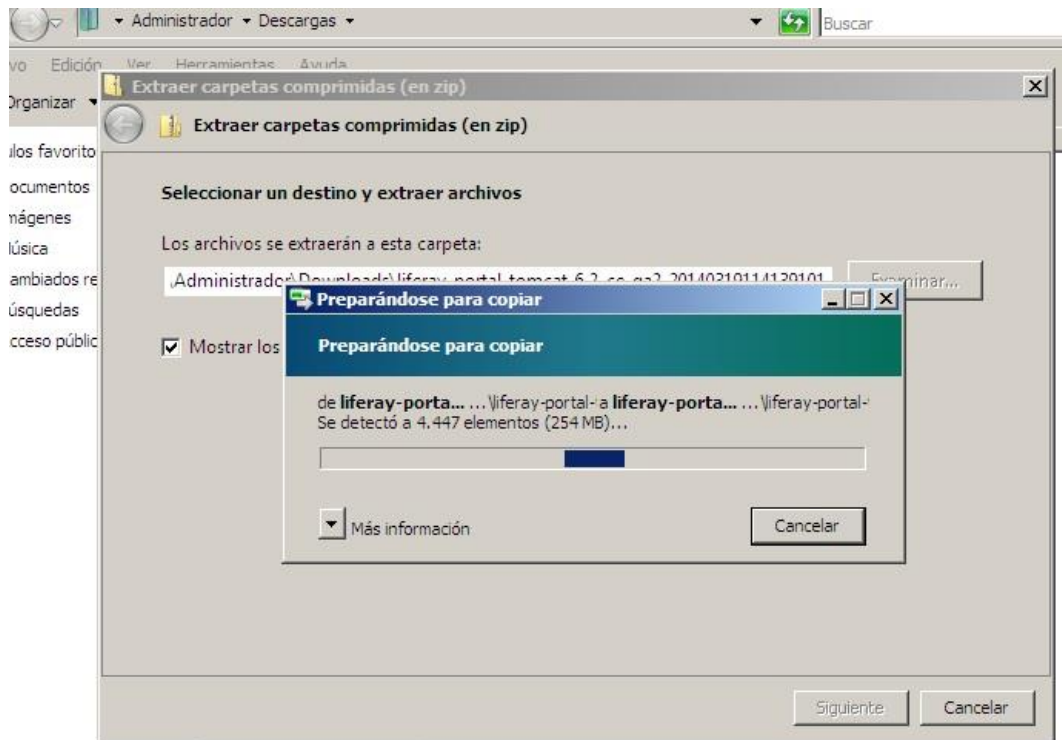
## Instalación de CMS Java Liferay con Tomcat

Teniendo Tomcat instalado, entramos en la página de CMS Java Liferay y lo descargamos.

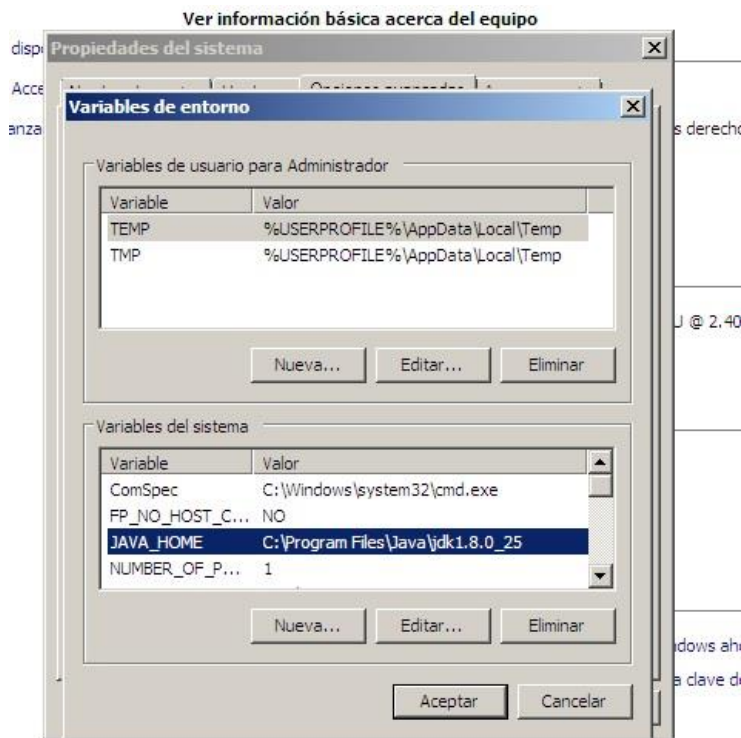




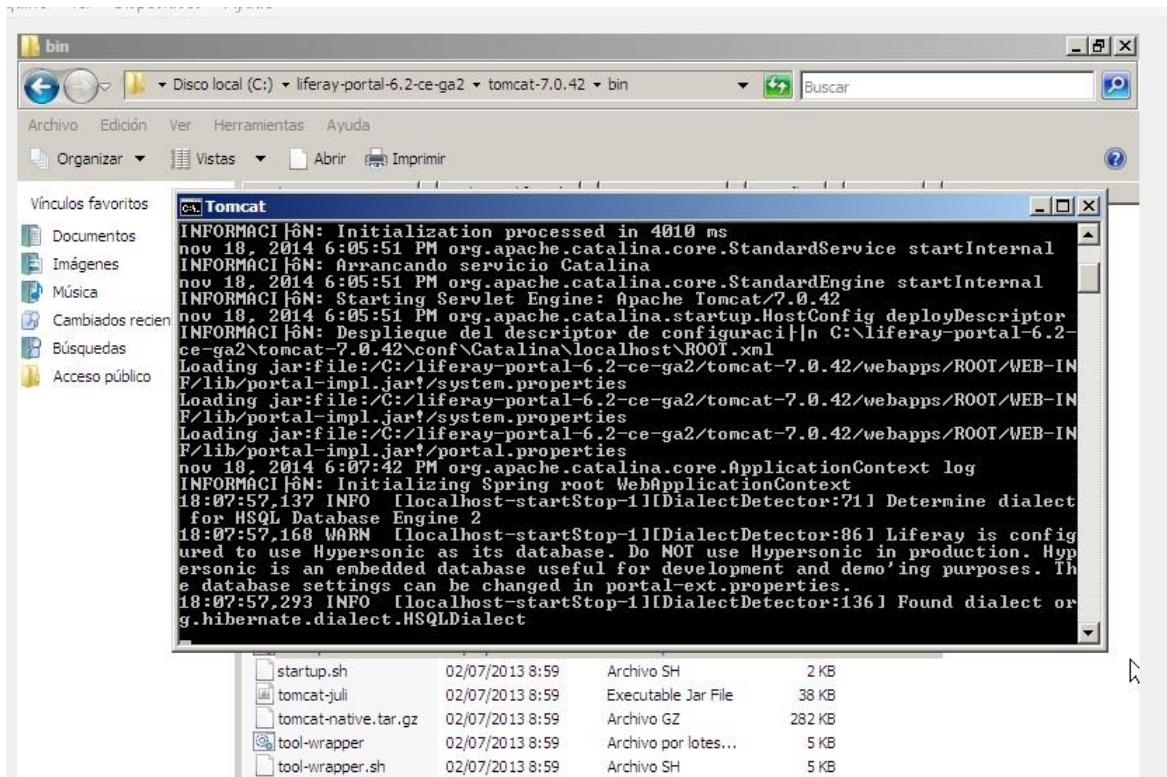
Una vez lo tenemos descargado, lo descomprimos.



Nos descargamos el Java JDK y lo instalamos, ahora creamos la variable de entorno JAVA\_HOME en **Equipo, Propiedades y valores de entorno**.



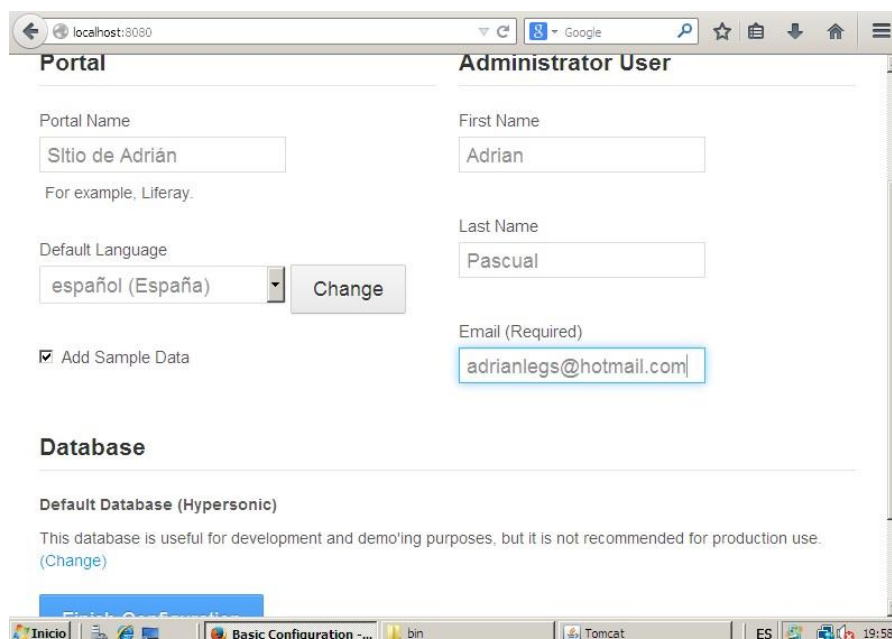
Este CMS lleva incluido el Tomcat por lo que será más fácil, entramos dentro del CMS en la carpeta Tomcat y luego en **bin**, acto seguido, ejecutamos **startup.bat**.



```

Tomcat
-----
INFORMACIÓN: Initialization processed in 4010 ms
nov 18, 2014 6:05:51 PM org.apache.catalina.core.StandardService startInternal
INFORMACIÓN: Arrancando servicio Catalina
nov 18, 2014 6:05:51 PM org.apache.catalina.core.StandardEngine startInternal
INFORMACIÓN: Starting Servlet Engine: Apache Tomcat/7.0.42
nov 18, 2014 6:05:51 PM org.apache.catalina.startup.HostConfig deployDescriptor
INFORMACIÓN: Despliegue del descriptor de configuración C:\liferay-portal-6.2-ce-ga2\tomcat-7.0.42\conf\Catalina\localhost\ROOT.xml
Loading jar:file:/C:/liferay-portal-6.2-ce-ga2\tomcat-7.0.42\webapps\ROOT\WEB-INF\lib\portal-impl.jar!/system.properties
Loading jar:file:/C:/liferay-portal-6.2-ce-ga2\tomcat-7.0.42\webapps\ROOT\WEB-INF\lib\portal-impl.jar!/system.properties
Loading jar:file:/C:/liferay-portal-6.2-ce-ga2\tomcat-7.0.42\webapps\ROOT\WEB-INF\lib\portal-impl.jar!/portal.properties
nov 18, 2014 6:07:42 PM org.apache.catalina.core.ApplicationContext log
INFORMACIÓN: Initializing Spring root WebApplicationContext
18:07:57.137 INFO [localhost-startStop-1][DialectDetector:711] Determine dialect for HSQL Database Engine 2
18:07:57.168 WARN [localhost-startStop-1][DialectDetector:861] Liferay is configured to use Hypersonic as its database. Do NOT use Hypersonic in production. Hypersonic is an embedded database useful for development and demo'ing purposes. The database settings can be changed in portal-ext.properties.
18:07:57.293 INFO [localhost-startStop-1][DialectDetector:1361] Found dialect org.hibernate.dialect.HSQLDialect
-----
File Name      Date Modified      Size      Type
-----
startup.sh     02/07/2013 8:59    2 KB     Archivo SH
tomcat-juli    02/07/2013 8:59    38 KB     Executable Jar File
tomcat-native.tar.gz 02/07/2013 8:59    282 KB    Archivo GZ
tool-wrapper   02/07/2013 8:59    5 KB     Archivo por lotes...
tool-wrapper.sh 02/07/2013 8:59    5 KB     Archivo SH
  
```

Cuando termine, se nos iniciara el instalador del CMS, a continuación rellenamos los datos que nos pide dejándolo así:



Portal Name:

For example, Liferay.

Default Language:

Add Sample Data

Administrator User

First Name:

Last Name:

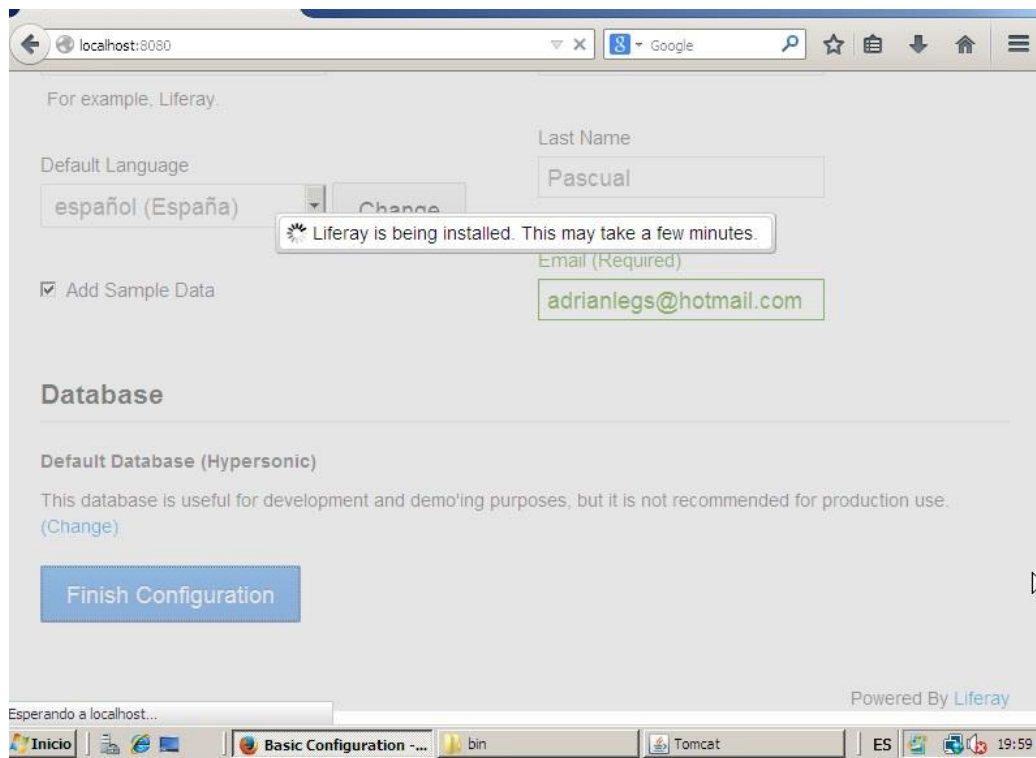
Email (Required):

Database

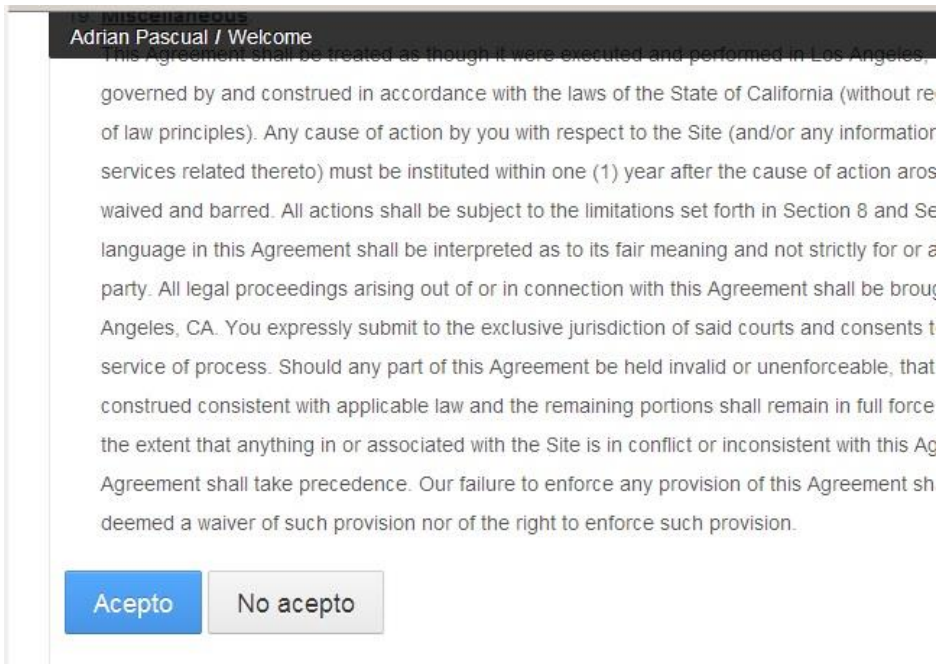
Default Database (Hypersonic)

This database is useful for development and demo'ing purposes, but it is not recommended for production use. [\(Change\)](#)

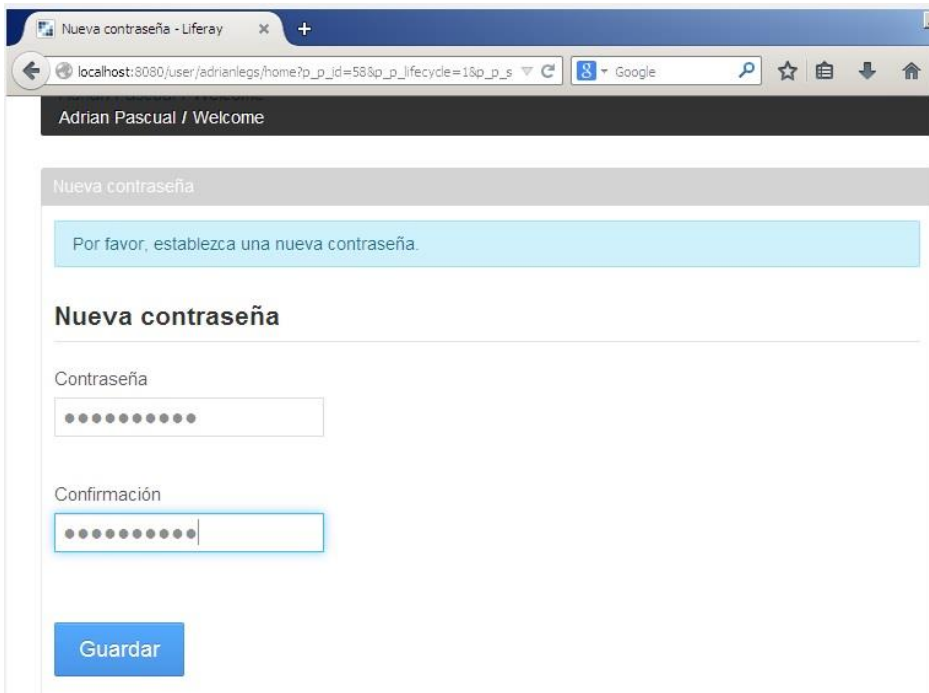
A continuación le damos a **Finish Configuration**.



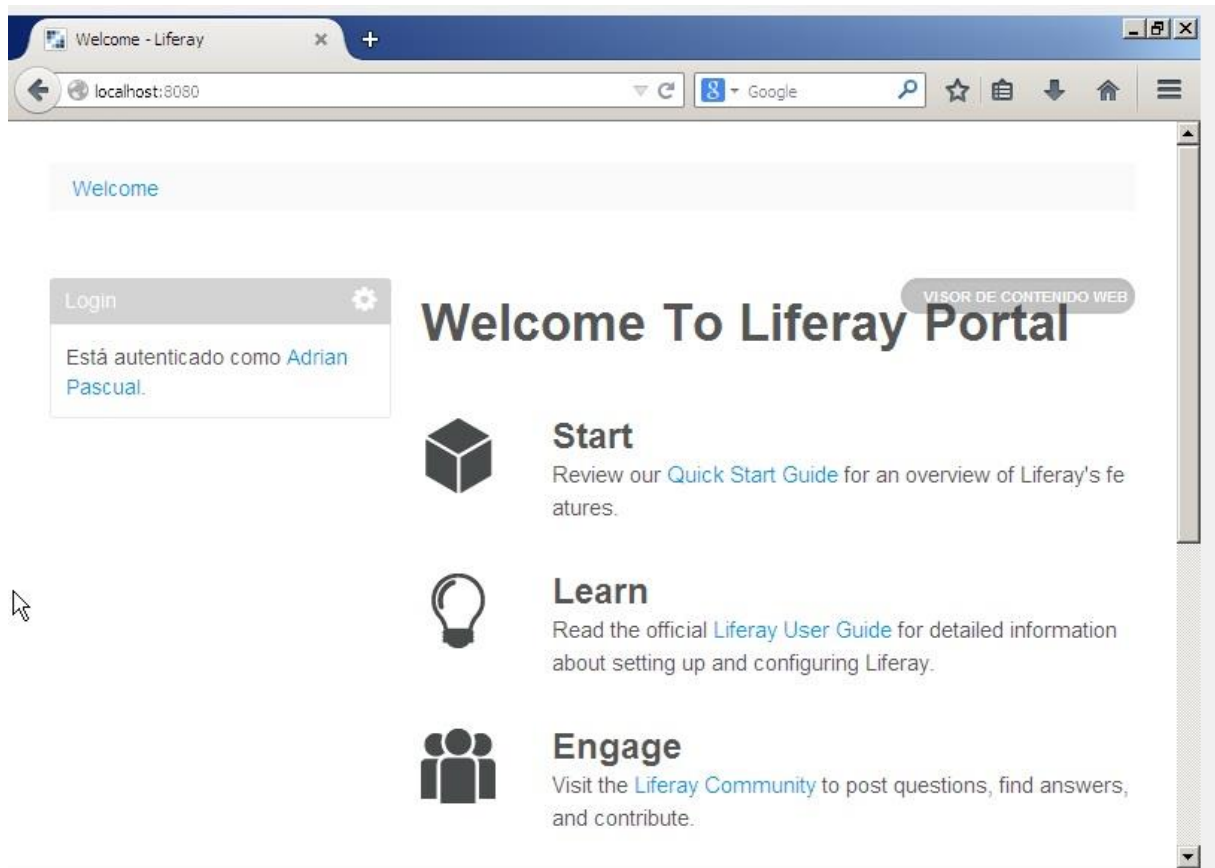
Presionamos en **Ir a mi Portal** y nos saldrá para aceptar los términos, los aceptamos para continuar.



Establecemos una contraseña para nuestra cuenta.



Y le damos a guardar. Nos preguntara una palabra clave para resetear la contraseña en caso de perdida y continuamos. Una vez que tengamos eso, ya tendremos LifeRay instalado.



Ya tenemos CMS Liferay instalado en Windows Server 2008 y funcionando correctamente.