

HOWTO

Manual para configurar Subversion + Apache + SSL en openSUSE



Índice de contenido

1 INTRODUCCIÓN.....	3
2 OBJETIVO.....	3
3 INSTALACIÓN.....	3
3.1 Paquetes Necesarios:.....	3
4 CONFIGURACIONES.....	4
4.1 Creación de Usuario y Grupo SVN:.....	4
4.2 Configuración de Apache:.....	4
4.3 Creación de Repositorios por cada Proyecto:.....	6
4.4 Creación de Archivos de Autorización:.....	7
4.4.1 Accesos para el repositorio Proyecto1:.....	7
4.4.2 Accesos para el repositorio Proyecto2:.....	7
4.5 Conexiones seguras mediante APACHE+SSL:.....	8
4.5.1 Requisitos:.....	8
4.5.2 Generación del Certificado.....	10
4.5.3 Generación de Certificado con Información Adicional:.....	11

1 INTRODUCCIÓN

En el trabajo surgió la necesidad de llevar un mejor control de las versiones de los fuentes de los diversos sistemas que se tienen implementado, baaa..., en realidad no existe ningún control, ya saben, archivos por doquier, 3 copias en una misma máquina, 1 copia de los fuentes en el servidor de aplicaciones, otra en la máquina del vecino y así un sin número de “backups” de fuentes que al final no sirven para nada porque se desconoce si están completos, actualizados o desactualizados, en fin, así surgió esta iniciativa de investigar acerca de este tema.

Subversion, como algunos ya saben, es una herramienta que nos permite llevar un control de versiones de cualquier tipo de archivo, en especial de códigos fuente, ya que permite mantener todo el historial de modificaciones que realicemos a los mismos, además nos permite documentar los cambios agregando comentarios y explicaciones acerca de los motivos de dichas alteraciones, para aquellos que aún no conocen o no entienden de que se trata este software les recomiendo visitar el siguiente enlace: <http://es.wikipedia.org/wiki/Subversion>

2 OBJETIVO

La idea es crear 2 repositorios de ejemplo, uno de ellos que exigirá autenticación por parte del usuario tanto para leer el repositorio como para escribir en él, el otro repositorios también exigirá autenticación para la escritura pero permitirá la lectura del repositorio para cualquier usuario sin la necesidad de autenticarse, esta última configuración es muy interesante si se quiere hacer público un repositorio con un proyecto de desarrollo para que todos los interesados tengan acceso a los fuentes de forma libre pudiendo luego enviar sus modificaciones o correcciones en forma de “parches” para que los encargados del proyecto puedan analizar si las modificaciones enviadas son correctas y necesarias para incorporarlas a una nueva versión del proyecto.

También, para poner en marcha este software, es necesario configurar Apache para publicar los repositorios vía web y dar soporte SSL para que los usuarios puedan conectarse de forma segura a los repositorios.

3 INSTALACIÓN

3.1 ***Paquetes Necesarios:***

Los paquetes que deberán ser instalación son los que se citan a continuación:

- apache2
- apache2-doc
- apache2-prefork
- libapr1
- libapr-util1
- neon

- subversion
- subversion-doc
- subversion-server

Todos los paquetes mencionados se encuentran disponibles en los repositorios oficiales de openSuSE (y también en el CD o DVD de instalación, aunque posiblemente algo desactualizados).

4 CONFIGURACIONES

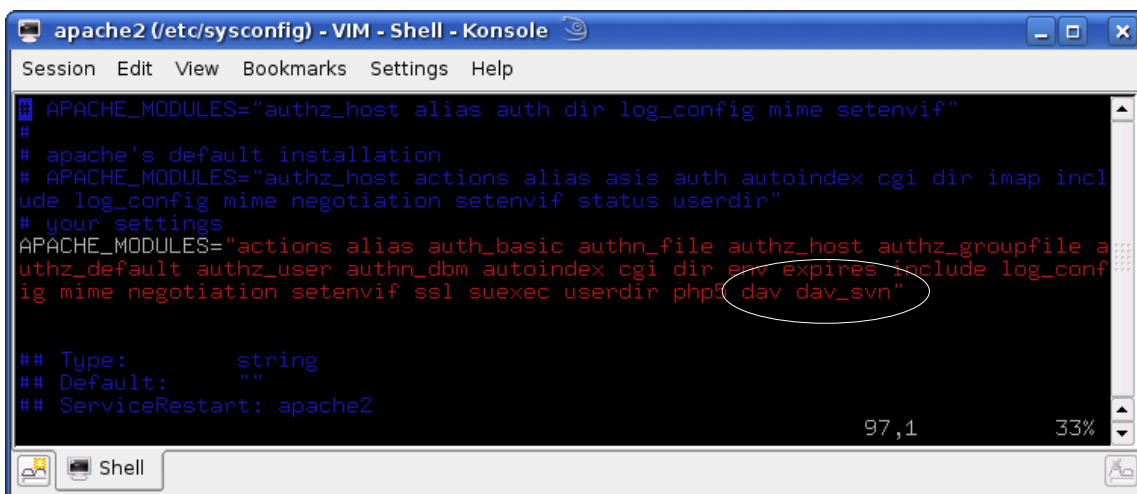
4.1 Creación de Usuario y Grupo SVN:

Luego de instalar la primera tarea que tenemos que hacer es agregar al sistema operativo un usuario de sistema llamado *svn* y cuyo grupo también deberá ser *svn*, por lo cual también tenemos que agregar un nuevo grupo de sistema llamado *svn*, este usuario es necesario para ejecutar el software por primera vez y lamentablemente no se crea automáticamente durante la instalación del paquete. Lo podemos agregar mediante los siguientes comandos:

```
superserver:/# groupadd -r svn
superserver:/# useradd -r -g svn svn
```

4.2 Configuración de Apache:

La forma más recomendable de utilizar subversion es publicando los repositorios a través del servicio http que ofrece Apache sobre el protocolo de red WebDAV/DeltaV y esto es posible gracias a los módulos que permiten su integración, así que lo primero que debemos hacer es dirigirnos al directorio */etc/sysconfig* y editar el archivo *apache2* para agregar los módulos correspondientes, una vez en el archivo nos dirigimos a la sección *APACHE_MODULES* para agregar al final el módulo *dav* y *dav_svn*:



```
apache2 (/etc/sysconfig) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help
APACHE_MODULES="authz_host alias auth_dir log_config mime setenvif"
#
# apache's default installation
# APACHE_MODULES="authz_host actions alias asis auth autoindex cgi dir imap incl
ude log_config mime negotiation setenvif status userdir"
# your settings
APACHE_MODULES="actions alias auth_basic authn_file authz_host authz_groupfile a
uthz_default authz_user authn_dbm autoindex cgi dir env expires include log_conf
ig mime negotiation setenvif ssl suexec userdir php dav dav_svn"
## Type:      string
## Default:   ""
## ServiceRestart: apache2
97,1 33%
```

El siguiente paso consiste en crear las carpetas en donde irán los repositorios de subversion mediante los siguientes comandos:

```
superserver:/# mkdir -p /srv/svn/repos
superserver:/# mkdir -p /srv/svn/user_access
superserver:/# mkdir -p /srv/svn/html
```

Creadas las carpetas editamos el archivo `/etc/apache2/conf.d/subversion.conf` para definir los repositorios a los cuales se tendrá acceso mediante Apache y configurar sus métodos de autenticación respectivos. La configuración que debe ir en el archivo `subversion.conf` se detalla a continuación:

```
## Ejemplo de configuración para un repositorio de Subversion.
## Documentación completa disponible en /usr/share/doc/packages/subversion
##

<IfModule mod_dav_svn.c>
##
## project related HTML files
##
<IfModule mod_alias.c>
    Alias /repos      /srv/svn/html
</IfModule>

<Directory /srv/svn/html>
    Options          +Indexes +Multiviews -FollowSymLinks
    IndexOptions     FancyIndexing \
                    ScanHTMLTitles \
                    NameWidth=* \
                    DescriptionWidth=* \
                    SuppressLastModified \
                    SuppressSize

    order allow,deny
    allow from all
</Directory>

## Configuración para el repositorio "proyecto1" que requiere autenticación
## tanto para la lectura como para escritura en el repositorio.
<Location /repos/proyecto1>
    DAV svn
    ## Directorio en donde se encontrará el proyecto
    SVNPath /srv/svn/repos/proyecto1

    ## Configuración del tipo de autenticación de usuario
    ## y definición de la ruta de los archivos que contienen
    ## la lista de usuarios autorizados y sus respectivas claves.
    AuthType Basic
    AuthName "Authorization for proyecto1 required"
    AuthUserFile /srv/svn/user_access/proyecto1_passwdfile
    AuthGroupFile /srv/svn/user_access/proyecto1_groupfile
```

```

## Definición de los grupos a los cuales los usuarios del programa van
## asociados según sus operaciones que pueden realizar en el repositorio.
## Esto queda más claro a medida que se finaliza con la configuración.
<LimitExcept GET PROPFIND OPTIONS REPORT>
    Require group proyecto1_committers
</LimitExcept>

<Limit GET PROPFIND OPTIONS REPORT>
    Require group proyecto1_committers
    Require group proyecto1_readers
</Limit>
</Location>

## Configuración para el Repositorio "proyecto2", su característica es que
## la lectura del repositorio para usuarios anónimos está permitida, pero
## para realizar cambios en el mismo el usuario debe autenticarse.
<Location /repos/proyecto2>
    DAV svn
    ## Directorio en donde se encontrará el proyecto
    SVNPath /srv/svn/repos/proyecto2

    ## Configuración del tipo de autenticación de usuario
    ## y definición de la ruta del archivo que contiene
    ## la lista de usuarios autorizados y sus respectivas claves.
    <LimitExcept GET PROPFIND OPTIONS REPORT>
        AuthType Basic
        AuthName "Authorization for proyecto2 required"
        AuthUserFile /srv/svn/user_access/proyecto2_passwdfile
        Require valid-user
    </LimitExcept>
</Location>

</IfModule>

```

4.3 Creación de Repositorios por cada Proyecto:

El siguiente paso consiste en dirigirnos al directorio /srv/svn/repos para comenzar a crear los repositorios con el comando svnadmin y asignarles un dueño:grupo a algunas carpetas:

```

superserver:/# cd /srv/svn/repos/
superserver:/srv/svn/repos # svnadmin create proyecto1
superserver:/srv/svn/repos # chown -R wwwrun:www proyecto1/{dav,db,locks}
superserver:/srv/svn/repos # svnadmin create proyecto2
superserver:/srv/svn/repos # chown -R wwwrun:www proyecto2/{dav,db,locks}

```

Una vez culminado con la creación de los repositorios reiniciamos el servidor http con el siguiente comando:

```

superserver:/# service apache2 restart

```

4.4 Creación de Archivos de Autorización:

En esta sección procederemos a crear los archivos que contendrán los nombres de usuarios y sus respectivas contraseñas para acceder a cada uno de los repositorios que creamos en el punto anterior.

4.4.1 Accesos para el repositorio Proyecto1:

El repositorio proyecto1, como se había especificado en el archivo *subversion.conf*, exigirá autenticación tanto para la lectura del repositorio como también para la escritura en el mismo, así que primeramente creamos un archivo que contendrá a los usuarios que tendrán acceso al repositorio proyecto1 y sus respectivos passwords, a este archivo se le asignará un dueño:grupo y permisos seguros para que no cualquiera tenga acceso:

```
superserver:/# touch /srv/svn/user_access/proyecto1_passwdfile
superserver:/# chown root:www /srv/svn/user_access/proyecto1_passwdfile
superserver:/# chmod 640 /srv/svn/user_access/proyecto1_passwdfile
```

Una vez listo con el paso anterior le introducimos los usuario y respectivas contraseñas al archivo mediante el comando htpasswd2:

```
superserver:/# htpasswd2 /srv/svn/user_access/proyecto1_passwdfile gabriel
superserver:/# htpasswd2 /srv/svn/user_access/proyecto1_passwdfile jorge
```

Pero esto aún no termina para el repositorio proyecto1, creamos un archivo que ya contiene los usuarios y ahora aún nos falta crear un nuevo archivo donde vamos a agrupar a los usuarios según las operaciones que tendrán autorizadas a realizar:

```
superserver:/# touch /srv/svn/user_access/proyecto1_groupfile
superserver:/# chown root:www /srv/svn/user_access/proyecto1_groupfile
superserver:/# chmod 640 /srv/svn/user_access/proyecto1_groupfile
```

editamos el archivo y le agregamos las 2 líneas que siguen a continuación:

```
proyecto1_committers: gabriel
proyecto1_readers: gabriel jorge
```

lo que significa que el usuario gabriel tendrá tanto autorización para leer el repositorio como también para escribir en él, sin embargo, el usuario jorge solo podrá leer el repositorio pero no escribir en él.

4.4.2 Accesos para el repositorio Proyecto2:

El repositorio proyecto2 permitirá a cualquier usuario anónimos la lectura del contenido del repositorio pero si se desea escribir en el repositorios solo permitirá a los usuarios definidos en el archivo de accesos del mencionado repositorio, así que en este caso también creamos un archivo para los usuarios que tendrán autorización, se le asigna el dueño:grupo respectivo y sus permisos correspondientes y por último se le agrega al archivo los usuarios a los cuales les queremos dar

permiso de escritura para el repositorio.

```
superserver:/# touch /srv/svn/user_access/proyecto2_passwdfile
superserver:/# chown root:www /srv/svn/user_access/proyecto2_passwdfile
superserver:/# chmod 640 /srv/svn/user_access/proyecto2_passwdfile
superserver:/# htpasswd2 /srv/svn/user_access/proyecto2_passwdfile gabriel
```

Llegando a este punto ya podemos acceder e importar nuestros archivos a los 2 repositorios con cualquier cliente para subversion, entre los cuales he probado los clientes gráficos [Tortoisesvn](#) para Windows (muy bueno) y eSvn o kdesvn para GNU/Linux, estos 2 últimos se encuentran disponibles en los repositorios de paquetes de openSUSE. Quisiera recordar también que no hay que olvidarse de **“abrir el puerto HTTP (80) en el firewall del servidor”** para no tener problemas.

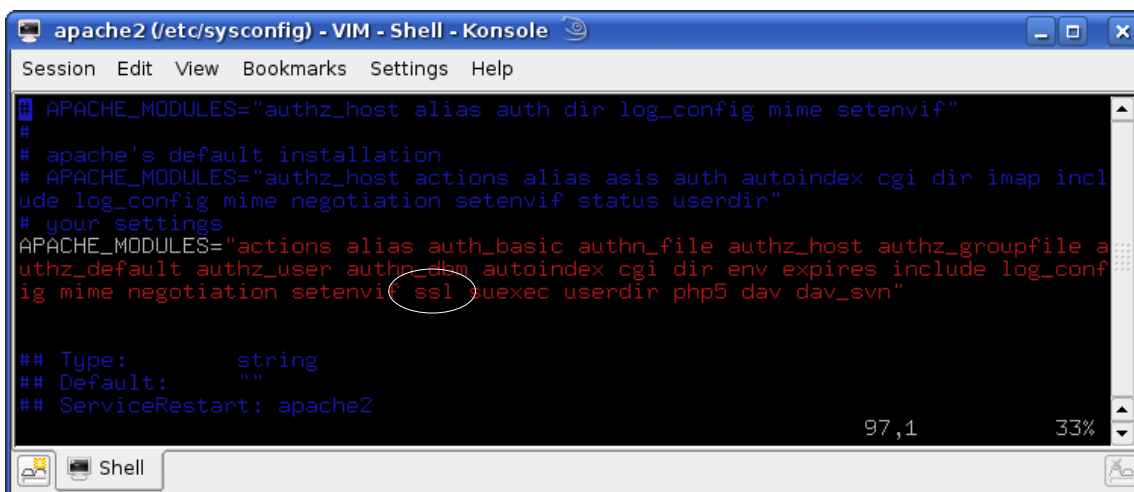
OBS: Cuando se trata de acceder a un repositorio de subversion a través de un servidor proxy es posible que se presenten inconvenientes, esto es a causa de que los servidores proxy no suelen interpretar o soportar todos los métodos que ofrece WebDAV, sobre este caso, en la página oficial de subversion tratan este tema en el siguiente enlace: <http://subversion.tigris.org/faq.html#proxy>

4.5 Conexiones seguras mediante APACHE+SSL:

Subversion puede aprovechar los beneficios que ofrece Apache y uno de ellos es el acceso seguro mediante el protocolo https que implementa SSL para establecer las conexiones.

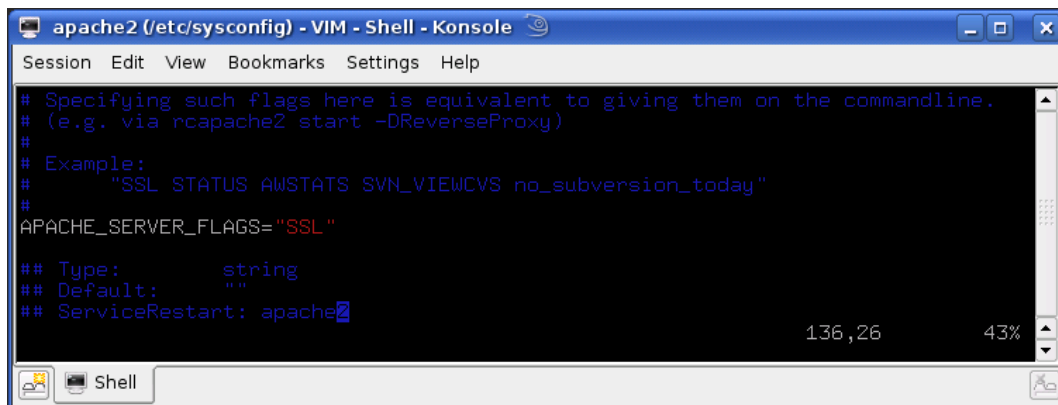
4.5.1 Requisitos:

Lo primero que tendremos que hacer es dirigirnos nuevamente al directorio `/etc/sysconfig` y abrir el archivo **“apache2”** nuevamente, donde tendremos que verificar que el módulo **“ssl”** esté definido en la directiva **APACHE_MODULES**:



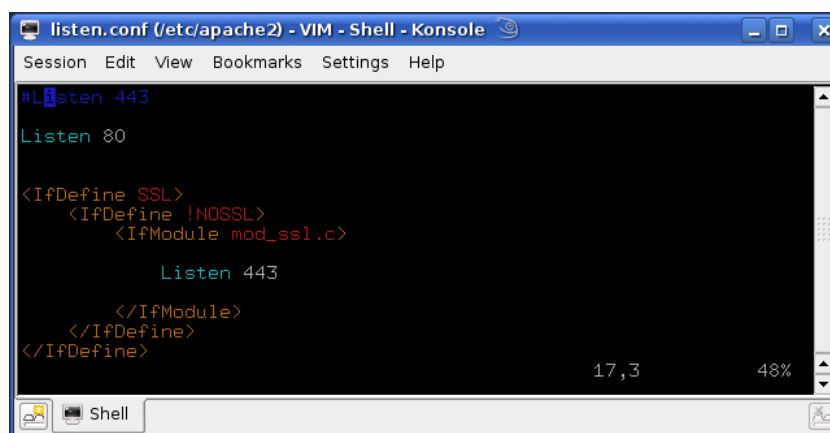
```
apache2 (/etc/sysconfig) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help
# APACHE_MODULES="authz_host alias auth dir log_config mime setenvif"
#
# apache's default installation
# APACHE_MODULES="authz_host actions alias asis auth autoindex cgi dir imap include log_config mime negotiation setenvif status userdir"
# your settings
APACHE_MODULES="actions alias auth_basic authn_file authz_host authz_groupfile authz_default authz_user authz_dbm autoindex cgi dir env expires include log_config mime negotiation setenvif ssl suexec userdir php5 dav dav_svn"
## Type:          string
## Default:       ""
## ServiceRestart: apache2
97,1 33%
```

También, en el mismo archivo tenemos que verificar que la directiva **APACHE_SERVER_FLAGS** sea igual a **“SSL”** como se muestra a continuación:



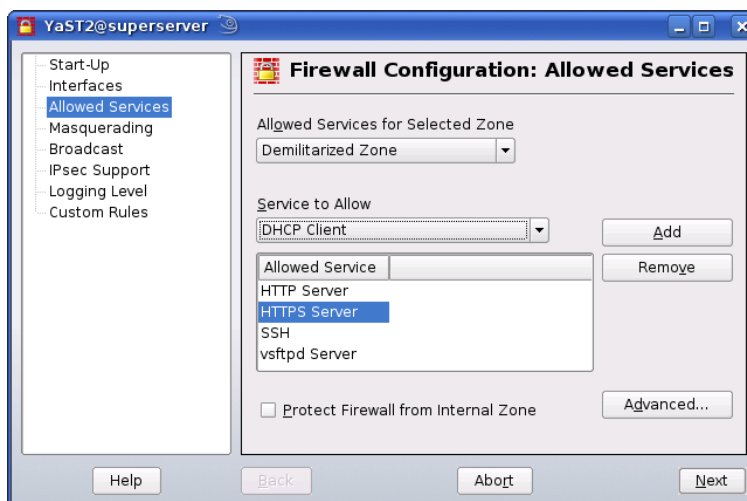
```
apache2 (/etc/sysconfig) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help
# Specifying such flags here is equivalent to giving them on the commandline.
# (e.g. via rcapache2 start -DReverseProxy)
#
# Example:
# "SSL STATUS AWSTATS SVN_VIEWCVS no_subversion_today"
#
APACHE_SERVER_FLAGS="SSL"
## Type:      string
## Default:   ""
## ServiceRestart: apache2
136,26 43%
```

Tenemos que verificar que Apache esté correctamente configurado para que escuche en el puerto indicado para las conexiones seguras, para ello nos vamos al directorio `/etc/apache2/` y ahí abrimos el archivo llamado `listen.conf` que debería tener el siguiente contenido:



```
listen.conf (/etc/apache2) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help
#Listen 443
Listen 80
<IfDefine SSL>
  <IfDefine !NOSSL>
    <IfModule mod_ssl.c>
      Listen 443
    </IfModule>
  </IfDefine>
</IfDefine>
17,3 48%
```

No hay que olvidarse de abrir el puerto 443 (puerto que utiliza el protocolo https) en el firewall de nuestro servidor, esto lo podemos hacer fácilmente desde **YaST2 --> Security and Users --> Firewall --> Allowed Services** agregando a la lista de la zona que está asociada a nuestra placa de red el nombre del servicio a autorizar, en este caso **"HTTPS Server"**



4.5.2 Generación del Certificado

Varias formas existen para generar un certificado, una de ellas y el método más simple es generar uno automáticamente que creará claves tontas, que es un certificado básico y sin mucho sentido, no posee información interesante adjunta, para ello hay que ejecutar el siguiente comando:

```
superserver: /# /usr/bin/gensslcert
```

Este comando sobre escribirá los siguientes archivos:

- /etc/apache2/ssl.crt/ca.crt
- /etc/apache2/ssl.key/server.key
- /etc/apache2/ssl.crt/server.crt
- /etc/apache2/ssl.csr/server.csr

También creará un nuevo archivo **CA.crt** en el directorio `/srv/www/htdocs/` para ser descargado por los clientes que se conectan bajo el protocolo https. Luego hay que crear una copia del archivo `/etc/apache2/vhosts.d/vhost-ssl.template` y renombrarlo a `/etc/apache2/vhosts.d/vhost-ssl.conf`.

```
superserver: /# cd /etc/apache2/vhosts.d
superserver: /etc/apache2/vhosts.d # cp vhost-ssl.template vhost-ssl.conf
```

Con esto hemos creado un host virtual exclusivo para las conexiones seguras, reiniciamos el servidor apache con el comando `service apache2 restart` y verificamos la configuración de nuestro virtual host con el comando:

```
superserver: /# httpd2 -S -DSSL
```

Si queremos generar un certificado con información relevante y con un carácter más responsable podemos ir indicándole algunos datos complementarios al comando **“gensslcert”** para enriquecer el contenido del certificado. La siguiente tabla muestra las opciones disponibles para anexar información al certificado:

Parámetro	Descripción
-C	Nombre Común
-N	Comentario
-c	País. Solo expresado en 2 letras Ej: PY, AR, BR, etc.
-s	Estado, Provincia o Departamento, como más les guste
-l	Ciudad
-o	Organización
-u	Unidad Organizacional
-n	Nombre de Dominio Totalmente Expresado
-e	Dirección de Correo Electrónico del Administrador
-y	Días de Valides del Certificado del Servidor (Certificado Privado)

4.5.3 Generación de Certificado con Información Adicional:

```
superserver:/# gensslcert -C "labitacoradegabriel" -N "Este es un Certificado de Ejemplo" -c PY -s Itapua -l "Bella Vista" -o "Gabriel-Corp." -u "Unidad Informatica" -n "labitacoradegabriel.wordpress.com" -e "labitacoradegabriel@gmail.com" -y 60
```

Al ejecutar el comando se crearán una serie de archivos inicialmente llamados con el nombre **labitacoradegabriel** en los siguientes directorios:

- /etc/apache2/ssl.crt/labitacoradegabriel-ca.crt
- /etc/apache2/ssl.crt/labitacoradegabriel-server.crt
- /etc/apache2/ssl.csr/labitacoradegabriel-server.csr
- /etc/apache2/ssl.key/labitacoradegabriel-ca.key
- /etc/apache2/ssl.key/labitacoradegabriel-server.key
- /srv/www/htdocs/LABITACORADEGABRIEL-CA.crt

Si queremos que éste sea el certificado utilizado para la autenticación tendremos que editar el archivo **vhost-ssl.conf** en el directorio **/etc/apache2/vhosts.d/** y modificar las directivas que se citan a continuación con los nuevos nombres de mencionados anteriormente:

```
SSLCertificateFile /etc/apache2/ssl.crt/labitacoradegabriel-server.crt  
SSLCertificateKeyFile /etc/apache2/ssl.key/labitacoradegabriel-server.key
```

Por último debemos reiniciar el servidor Apache para que los cambios tengan efecto y el nuevo certificado entre en funcionamiento, si tratamos de acceder a cualquiera de nuestros repositorios vía https en, por ejemplo Firefox, el servidor emitirá el certificado con la información que hemos introducido.

