

2011-
2012

Actividad15 Administración
servidor web HTTPS (Apache2)
en Ubuntu Server -- mod_ssl,
default_ssl – certificados
digitales



José Jiménez Arias
IES Gregorio Prieto
2011-2012

En primer lugar observamos el directorio para asegurarnos que tenemos el módulo disponible/etc/apache2/mods-available

```
root@ubuntusrv04:/etc/apache2/mods-available# ls
actions.conf          cache.load           filter.load          proxy_http.l
actions.load          cern_meta.load      headers.load         proxy.load
alias.conf            cgid.conf            ident.load           proxy_scgi.l
alias.load            cgid.load            imagemap.load        reqtimeout.c
asis.load             cgi.load             include.load         reqtimeout.l
auth_basic.load       charset_lite.load    info.conf            rewrite.load
auth_digest.load     dav_fs.conf          info.load            setenvif.conf
authm_alias.load     dav_fs.load          ldap.load            setenvif.load
authm_anon.load      dav.load             log_forensic.load    snmp.load
authm_dbd.load       dav_lock.load        mem_cache.conf       ssl.conf
authm_dbm.load       dbd.load             mem_cache.load       ssl.load
authm_deflate.load   deflate.conf         mime.conf            userdir.conf
```

Posteriormente activamos el modulo con la sentencia : `a2enmod ssl` y reiniciamos el servicio para actualizar los cambios y poder trabajar con este modulo

```
root@ubuntusrv04:/etc/apache2/mods-available# a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
create self-signed certificates.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntusrv04:/etc/apache2/mods-available# /etc/init.d/apache2 reload
* Reloading web server config apache2 [ OK ]
root@ubuntusrv04:/etc/apache2/mods-available# _
```

Luego nos situamos en el directorio de los sitios disponibles /etc/apache2/sites-available observamos que contiene default-ssl y posteriormente activamos el sitio y reiniciamos.

```
root@ubuntusrv04:/etc/apache2/sites-available# ls
default default-ssl sad04 sri04
root@ubuntusrv04:/etc/apache2/sites-available# a2ensite default-ssl
Enabling site default-ssl.
Run '/etc/init.d/apache2 reload' to activate new configuration!
root@ubuntusrv04:/etc/apache2/sites-available# /etc/init.d/apache2 reload
* Reloading web server config apache2 [ OK ]
root@ubuntusrv04:/etc/apache2/sites-available# _
```

Después comprobamos que ha sido activado visualizando el directorios de los sitios activados de apache /etc/apache2/sites-enabled

```
root@ubuntusrv04:/etc/apache2/sites-enabled# ls
default-ssl sad04 sri04
root@ubuntusrv04:/etc/apache2/sites-enabled#
```

Observamos el contenido del fichero default-ssl

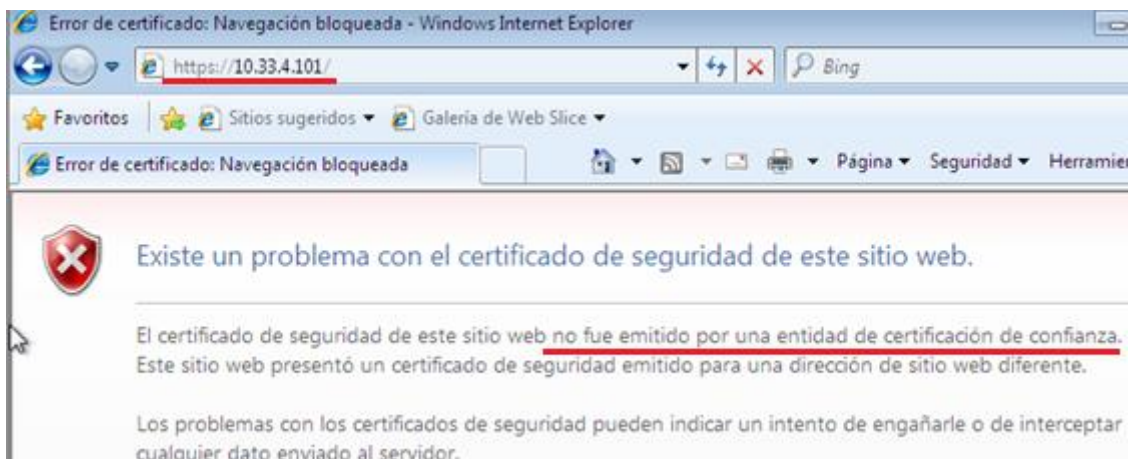
```
GNU nano 2.2.2          Archivo: default-ssl
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

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

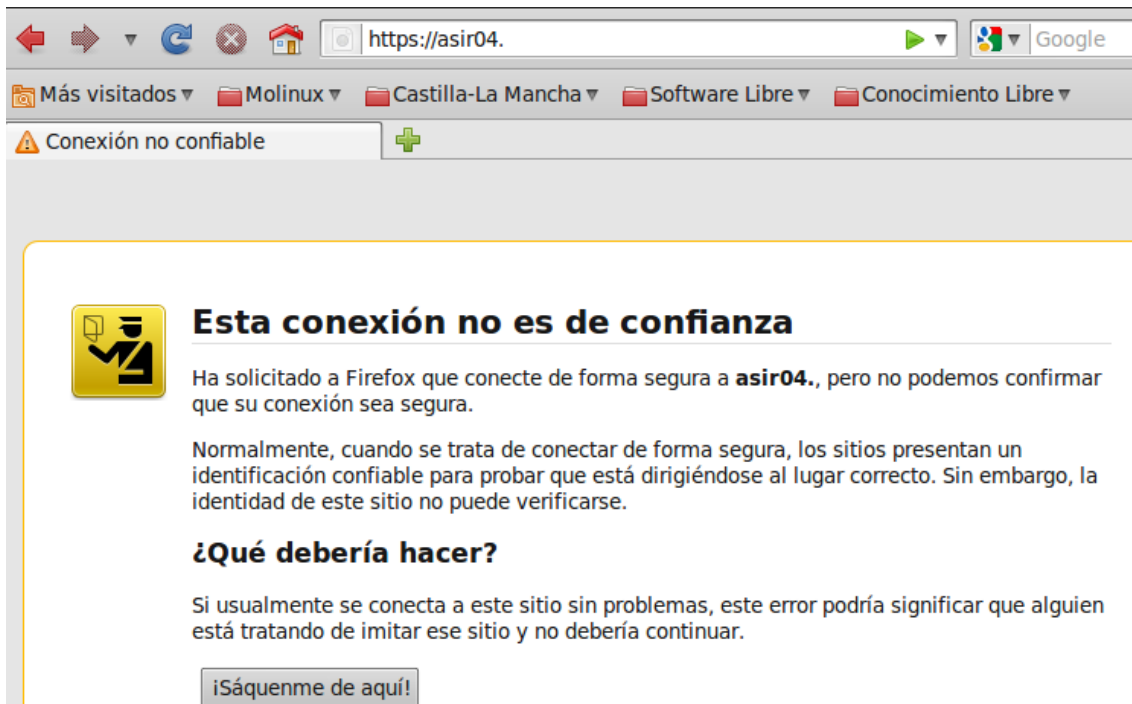
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log
```

A continuación nos situamos en un cliente W7 y entramos en <https://10.33.4.101> para probar la configuración realizada.



También podemos probar la configuración realizada desde otro sistema como por ejemplo molinux:



Luego desactivamos el sitio default-ssl para continuar con la configuración y reiniciamos el servicio apache2.

```
root@ubuntusrv04:/etc/apache2/sites-available# a2dissite default-ssl
Site default-ssl disabled.
Run '/etc/init.d/apache2 reload' to activate new configuration!
root@ubuntusrv04:/etc/apache2/sites-available# /etc/init.d/apache2 reload
* Reloading web server config apache2 [ OK ]
root@ubuntusrv04:/etc/apache2/sites-available#
```

El siguiente paso es crear un certificado digital autoafirmado con openssl para el dominio seguro04.asir04.

Primero instalamos openssl:

```
root@ubuntusrv04:/etc/apache2/sites-available# aptitude install openssl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Leyendo la información de estado extendido
Inicializando el estado de los paquetes... Hecho
Escribiendo información de estado extendido... Hecho
```

Con openssl instalado nos disponemos a generar llave y con estas certificados.

Para crear una llave escribimos la siguiente sentencia:

```
root@ubuntusrv04:/etc/apache2# openssl genrsa -des -out cert_joseyedu.key 4096
Generating RSA private key, 4096 bit long modulus
.....++
.....++
e is 65537 (0x10001)
Enter pass phrase for cert_joseyedu.key:
Verifying - Enter pass phrase for cert_joseyedu.key:
root@ubuntusrv04:/etc/apache2#
```

Tras escribir la sentencia y presionar *enter* nos solicitará una contraseña "inves".

La sentencia introducida se puede traducir en:

- **openssl**: comando.
- **genrsa**: genera la llave.
- **-des3**: Sistema de cifrado de la llave.
- **-out**: parámetro que indica la salida en archivo.
- **cert_joseyedu.key**: el nombre del archivo que contendrá la llave.
- **4096**: tamaño en bits de la llave.

Ahora debemos generar el certificado digital antes hemos de realizar una petición. Este certificado será firmado con la llave obtenida en el paso anterior, para ello introduciremos la siguiente sentencia:

```
root@ubuntusrv04:/etc/apache2# openssl req -new -key cert_joseyedu.key -out cert_joseyedu.csr
Enter pass phrase for cert_joseyedu.key:
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]:Ciudad Real
Locality Name (eg, city) []:Menbrilla
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Ies GP
Organizational Unit Name (eg, section) []:Zasir
Common Name (eg, YOUR name) []:joseyedu
Email Address []:joseyedu@hotmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:inves
An optional company name []:
root@ubuntusrv04:/etc/apache2#
```

Tras escribir la sentencia y presionar *enter* nos solicitará una contraseña "inves", tras introducir la contraseña podemos comenzar introducir los datos.

La sentencia introducida se puede traducir en:

- **openssl:** comando
- **req:** request para el certificado.
- **new:** parámetro que indica nueva petición.
- **key:** parámetro que indica la entrada mediante archivo.
- **cert_joseyedu.key:** nombre del archivo creado anteriormente que contiene la llave.
- **out:** parámetro que indica la salida en archivo.
- **cert_joseyedu.csr:** nombre del archivo que contendrá la información.

Ahora firmamos la petición solicitada en el paso anterior con la siguiente sentencia:

```
root@ubuntusrv04:/etc/apache2# openssl x509 -req -days 365 -in cert_joseyedu.csr  
-signkey cert_joseyedu.key -out cert_joseyedu.crt  
Signature ok  
subject=/C=ES/ST=Ciudad Real/L=Membrilla/O=Ies GP/OU=Zasir/CN=joseyedu/emailAddress=joseyedu@hotmail.com  
Getting Private key  
Enter pass phrase for cert_joseyedu.key:  
root@ubuntusrv04:/etc/apache2#
```

Tras escribir la sentencia y presionar *enter* nos solicitará una contraseña "inves".

La sentencia introducida se traduce en:

- **x509:** aplica formato de llave publica.
- **req:** request para el certificado.
- **days:** vigencia del certificado.
- **in:** indica la entrada de información mediante archivo especificado.
- **cert_joseyedu.csr:** nombre del archivo que contiene la información.
- **signkey:** indica la llave que firmará el certificado.
- **cert_joseyedu.key:** nombre del archivo creado anteriormente que contiene la llave.
- **out:** parámetro que indica la salida en archivo.
- **cert_joseyedu.crt:** archivo que contendrá el certificado.

Con el fin de tener organizados nuestros certificados creamos la carpeta ssl dentro de /etc/apache2 y movemos los certificados recién creados a este directorio.

```
root@ubuntusrv04:/etc/apache2# cp cert_joseyedu.key ssl/cert_joseyedu.key
root@ubuntusrv04:/etc/apache2# cp cert_joseyedu.csr ssl/cert_joseyedu.csr
root@ubuntusrv04:/etc/apache2# cp cert_joseyedu.crt ssl/cert_joseyedu.crt
root@ubuntusrv04:/etc/apache2# cd ssl
root@ubuntusrv04:/etc/apache2/ssl# ls
cert_joseyedu.crt cert_joseyedu.csr cert_joseyedu.key
root@ubuntusrv04:/etc/apache2/ssl#
```

A continuación nos situamos en el directorio /var/www y creamos la carpeta seguro un archivo de nombre seguro.html

```
GNU nano 2.2.2 Archivo: seguro.html
<html>
<body>
<h1> CONEXION SEGURA DE JOSE Y EDU... 15</h1>
</body>
</html>
```

Realizamos un ls para comprobar que todo está en orden, el directorio y el archivo.

```
root@ubuntusrv04:/var/www/seguro# ls
seguro.html
root@ubuntusrv04:/var/www/seguro#
```

A continuación para no manipular el fichero default, hacemos una copia de default con el nombre seg-default.

```
root@ubuntusrv04:/etc/apache2/sites-available# cp default seg-default
root@ubuntusrv04:/etc/apache2/sites-available# ls
default default-ssl sad04 seg-default sri04
root@ubuntusrv04:/etc/apache2/sites-available#
```

Después editamos el nuevo sitio seg-default:

```
GNU nano 2.2.2 Archivo: seg-default
<VirtualHost *:443>
  ServerAdmin webmaster@localhost
  ServerName seguro.asir04
  DocumentRoot /var/www/seguro

  <Directory />
    Options FollowSymLinks
    AllowOverride None
    Options FollowSymLinks MultiViews
  </Directory>

  <Directory /var/www/seguro>
    DirectoryIndex seguro.html
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
  </Directory>
[ 48 líneas escritas ]
```

Posteriormente editamos nuestro archivo de configuración dns, para que resuelva el nuevo sitio:

```
GNU nano 2.2.2 Archivo: /etc/bind/db.asir04.net
;
; BIND data file for local loopback interface
;
$ORIGIN asir04.
$TTL 604800
asir04.      IN      SOA      asir04. root.asir04. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
asir04.      IN      NS       ubuntu04.asir04.
ubuntu04.asir04.  IN    A        10.33.4.101
windows7.asir04.  IN    A        13.33.4.10
debian04.asir04.  IN    A        10.33.4.30
molinux04.asir04.  IN    A        10.33.4.60
windowsxp04.asir04.  IN   A        10.33.4.20
seguro.asir04.   IN    CNAME    ubuntu04.asir04.
```

Y reiniciamos el servicio bind9 con la sentencia: *service bind9 restart*

El siguiente paso es activar el sitio recientemente configurado seg-default:

```
root@ubuntusrv04:/etc/apache2/sites-available# a2ensite seg-default
Enabling site seg-default.
Run '/etc/init.d/apache2 reload' to activate new configuration!
root@ubuntusrv04:/etc/apache2/sites-available# service apache2 reload
* Reloading web server config apache2
root@ubuntusrv04:/etc/apache2/sites-available#
```

A continuación añadimos las líneas para leer los certificados en el fichero seg-default:

```
GNU nano 2.2.2 Archivo: seg-default
<Directory "/usr/share/doc/">
  Options Indexes MultiViews FollowSymLinks
  AllowOverride None
  Order deny,allow
  Deny from all
  Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/cert_joseyedu.crt
SSLCertificateKeyFile /etc/apache2/ssl/cert_joseyedu.key
</VirtualHost>
```

Por último visualizamos el fichero default, puesto que hicimos una copia, este fichero es perfectamente funcional, para los demás ficheros y directorio de nuestro servidor apache:

```
GNU nano 2.2.2 Archivo: default
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName ubuntusrv04.asir04
  DocumentRoot /var/www
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
```

A continuación paramos y arrancamos el servicio, observamos que nos pide la contraseña para iniciar el servicio. También observamos 1 warning, este se debe a que aún no hemos configurado los NameVirtualHost.

```
root@ubuntusrv04:/etc/bind# service apache2 start
* Starting web server apache2
[Fri Jan 20 04:55:27 2012] [warn] NameVirtualHost 10.33.4.101:80 has no VirtualHosts
Apache/2.2.14 mod_ssl/2.2.14 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server seguro.asir04:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
[ OK ]
root@ubuntusrv04:/etc/bind#
```

Antes de continuar, añadimos los NameVirtualHost

```
Listen 80
NameVirtualHost *:80
NameVirtualHost *:443
<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will
    # the VirtualHost statement in /etc/apache2/sites-
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hos
    # supported by MSIE on Windows XP.
    Listen 443
</IfModule>
```

Tras configurar y añadir los NameVirtualHost, reiniciamos el servicio apache y observamos como los warning anteriores desaparecen.

```
root@ubuntusrv04:/var/log# service apache2 start
* Starting web server apache2
Apache/2.2.14 mod_ssl/2.2.14 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server seguro.asir04:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
[ OK ]
```

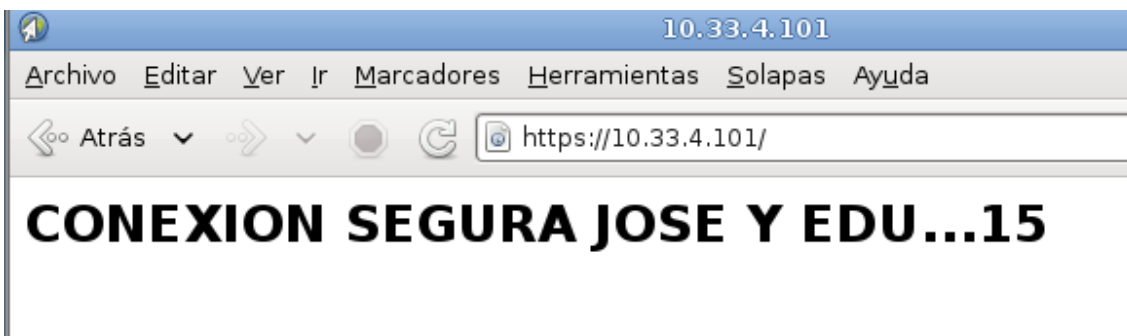
Observamos que podemos acceder sin seguridad es decir con http al sitio.



Name	Last modified	Size	Description
datos/	09-Jan-2012 11:45	-	
indice.html	09-Jan-2012 09:54	177	
log/	09-Jan-2012 16:36	-	
no_encontrada.html	09-Jan-2012 15:59	87	
red.html	09-Jan-2012 11:29	82	
sad04/	18-Jan-2012 17:02	-	
seguro/	20-Jan-2012 04:19	-	
sri04/	18-Jan-2012 17:02	-	
webalizer/	15-Jan-2012 20:51	-	

Apache/2.2.14 (Ubuntu) Server at asir04 Port 80

Y que también podemos acceder de forma segura mediante certificado.



10.33.4.101

Archivo Editar Ver Ir Marcadores Herramientas Solapas Ayuda

https://10.33.4.101/

CONEXION SEGURA JOSE Y EDU...15

Volvemos al fichero seg-default y lo editamos de tal forma que eliminemos el Indexes para que no muestre el contenido del directorio en el caso de que no encuentre seguro.html

```
<Directory /var/www/seguro>
  DirectoryIndex seguro.html
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  allow from all
</Directory>
```

En la parte inferior del archivo seg-default editamos 3 directivas más, ErrorLog, CustomLog y ErrorDocument.

```
ErrorLog /var/log/apache2/seguro.error.log
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn
CustomLog /var/log/apache2/seguro.access.log combined
ErrorDocument 404 /sinseguridad.html
```

Los archivos seguro.error.log y seguro.access.log los crearemos en el directorio /var/log/apache2 y estarán vacíos, sin embargo, el fichero sinseguridad.html cuyo contenido mostraremos a continuación lo crearemos en el directorio /var/www/seguro

```
GNU nano 2.2.2      Archivo: sinseguridad.html
<html>
<body>
<h1>SORRY.. LA PAGINA ASIR04 NO HA SIDO ENCONTRADA</h1>
</body>
</html>
```

Observamos el contenido del directorio /var/www/seguro

```
root@ubuntusrv04:/var/www/seguro# ls
seguro.html  sinseguridad.html
root@ubuntusrv04:/var/www/seguro#
```

Observamos el contenido del directorio /var/log/apache2

```
root@ubuntusrv04:/var/log/apache2# ls
access.log  other_vhosts_access.log  seguro.error.log
error.log  seguro.access.log      ssl_access.log
root@ubuntusrv04:/var/log/apache2#
```

Por último reiniciamos el servidor y por curiosidad, vamos a error a proposito introduciendo la contraseña, comprobamos que solo con "inves" inicia.

```
root@ubuntusrv04:/etc/apache2# service apache2 restart
* Restarting web server apache2
Apache/2.2.14 mod_ssl/2.2.14 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server seguro.asir04:443 (RSA)
Enter pass phrase:
Apache:mod_ssl:Error: Pass phrase incorrect (5 more retries permitted).
Enter pass phrase:
Apache:mod_ssl:Error: Pass phrase incorrect (4 more retries permitted).
Enter pass phrase:

OK: Pass Phrase Dialog successful.
root@ubuntusrv04:/etc/apache2#
```

COMPROBAMOS EL FUNCIONAMIENTO :

Nos situamos en un cliente molinux y en la barra del navegador escribimos:

Pulsamos en **Agregamos excepción...**

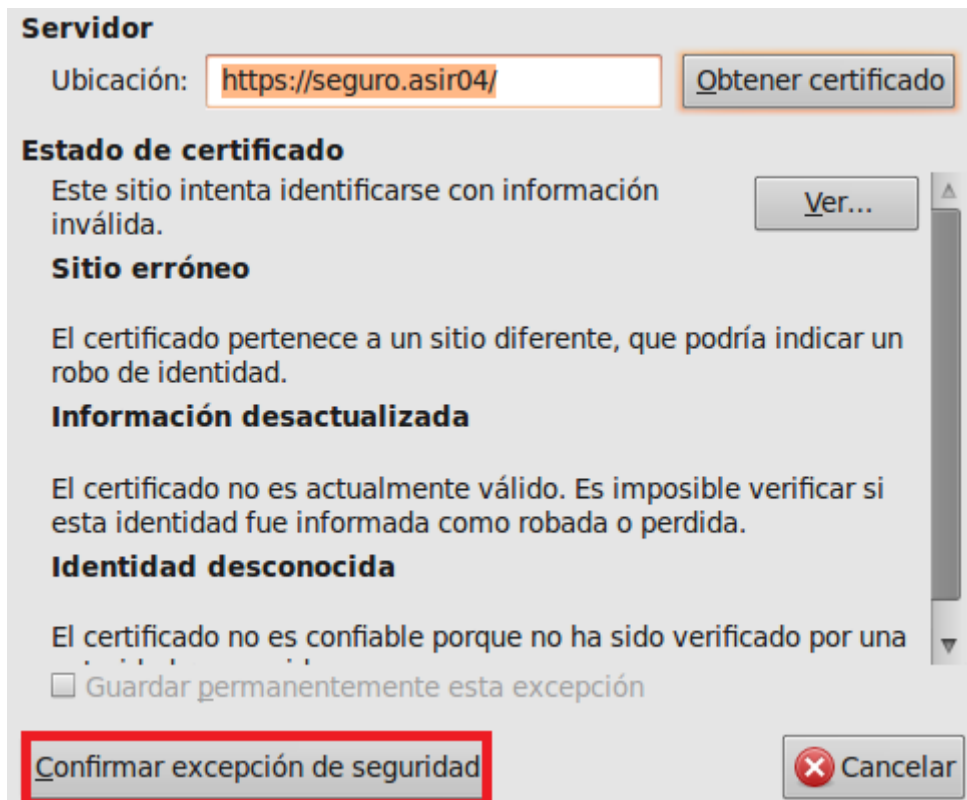


Si entiende lo que está pasando, puede decirle a Firefox que comience a confiar en la identificación de este sitio. **Aunque confíe en el sitio, este error podría significar que alguien está alterando su conexión.**

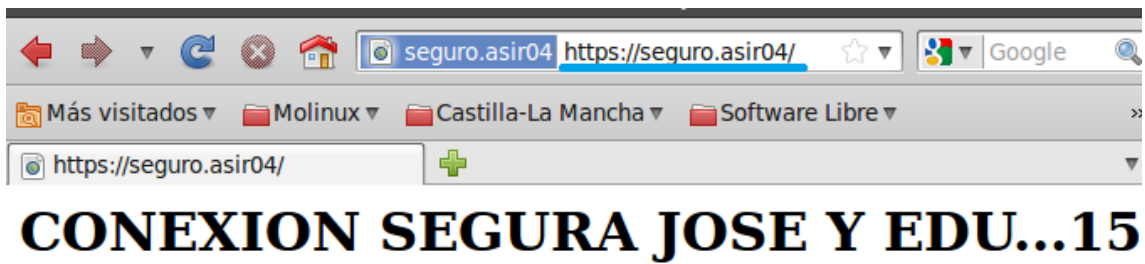
No agregue una excepción a menos que conozca que hay una buena razón para que este sitio no use una identificación confiable.

Agregar excepción...

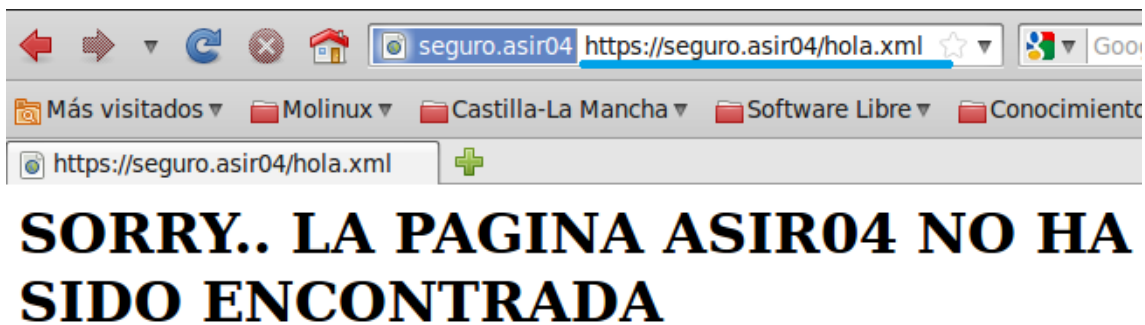
A continuación pulsamos en **Confirmar excepción de seguridad:**



Tras pulsa en confirmar excepción de seguridad nos lleva la página siguiente.



Por último recordamos que en pasos quitamos el Indexes del directorio, y modificamos para que cuando se produjera un error 404 mostrara la pagina "sinseguridad.html" pues bien aquí la tenemos.



Como conclusión a esta práctica y a la ud04 decir que me ha parecido muy didáctica a la par que entretenida.