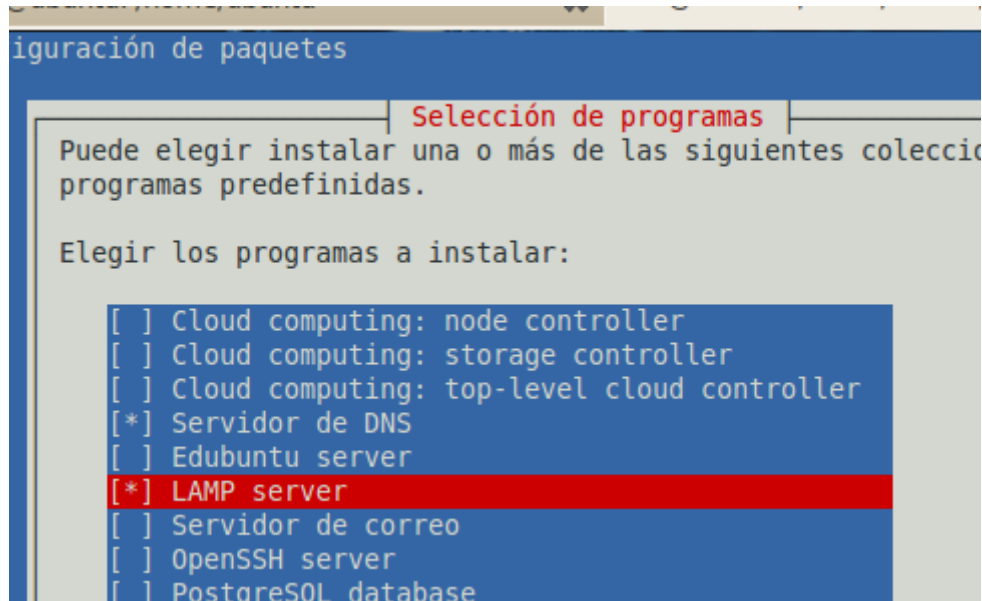
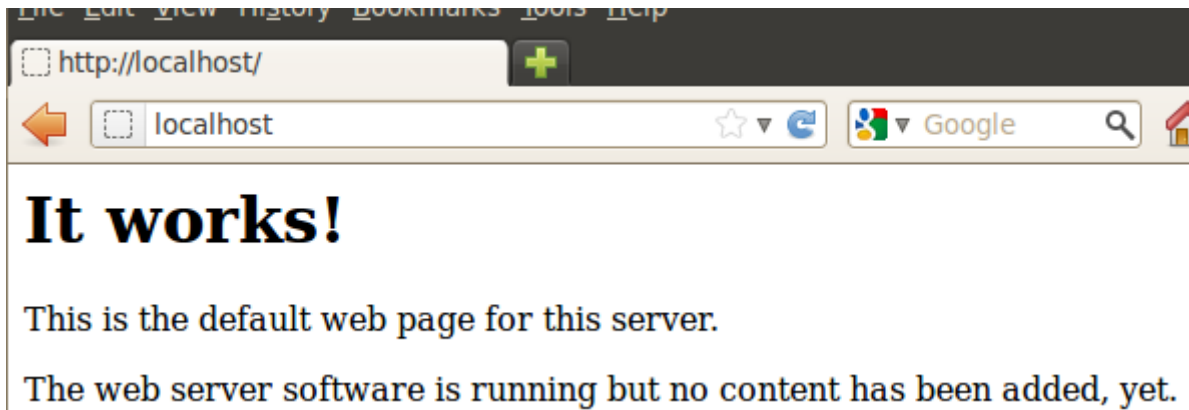


## Apache2

- Instalar el LAMP con taskel



- Hacer la prueba en un navegador del localhost.



- a2enmod userdir para actualizar el directorio del usuario
- Reiniciar el servicio con `/etc/init.d/apache2 restart`

```

root@ubuntu:/home/ubuntu# a2enmod userdir
Enabling module userdir.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntu:/home/ubuntu# /etc/init.d/apache2 restart
 * Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.1.1 for ServerName
... waiting apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
[ OK ]
root@ubuntu:/home/ubuntu#

```

- aptitude install libapache2-mod-perl2 --> Instalar los módulos del perl

Para quitar la virguirilla

## Quitando la ~ o virguirilla del home de los usuarios

A la hora de instalar un servidor casero Apache, existen un modulo de apache, userdir, que permite que los distintos usuarios del sistema tengan una carpeta en su directorio home donde pueden alojar sus webs y que estas se publiquen automáticamente en el servidor.

La carpeta por defecto es /home/usuario/public\_html/. Para acceder desde inet a esta carpeta la url es http://localhost/~usuario/. A mi modo de ver bastante incomodo.

Ejecutamos el siguiente comando

```
sudo aptitude install libapache2-mod-perl2
```

Que nos instalara un nuevo mod de apache, que nos permitirá a su vez ejecutar el siguiente código, que quitara la virguirilla (~) de las urls de los usuarios.

Una vez instalado ejecutamos el comando

```
gksu gedit /etc/apache2/httpd.conf
```

Y añadimos el siguiente código:

```

opendir H, "/home/";
my @dir = readdir(H);
closedir H;
foreach my $u (@dir) {
    next if $u =~ m/^\./;
    if (-e "/home/$u/public_html"){
        push @Alias, ["$u", "/home/$u/public_html/"];
    }
}

```

[Clica aqui para que veas bien el código anterior con caracteres especiales](#)

- Hacer la prueba con -->
  - mkdir public\_html
    - chmod 755 pu[TAB]
    - gedit public\_html/index.html
    - Introducir texto HTML.
    - Entonces tendremos algo similar a esto



La carpeta de apache2 es `/var/www/`

HTTPS --> Websegura

### ACTIVAR SSL en APACHE2 (Opción Google)

1. Habilitamos el módulo ssl

```
sudo a2enmod ssl
```

2. Habilitamos la conf. SSL por default

```
sudo a2ensite default-ssl
```

3. Reiniciamos

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

4. Ya con esto queda habilitado, puedes probarlo entrando al localhost con `https://127.0.0.1`

Para crear nuestro certificado, debemos seguir los siguientes pasos:

#Creamos las llaves

```
cd /etc/apache2
```

```
sudo openssl genrsa -des3 -out server.key 1024
```

5. Creamos el certificado con la llave

```
sudo openssl req -new -key server.key -out server.csr
```

6. Creamos nuestro certificado contestando a las preguntas que se nos indique.

```
sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

7. Cambiamos de carpeta la llave y certificado

```
sudo cp server.crt /etc/ssl/certs
```

8. Abrimos el archivo de configuración default-ssl

```
sudo cp server.key /etc/ssl/private
```

9. Reemplazamos y habilitamos la siguiente configuración.

```
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
```

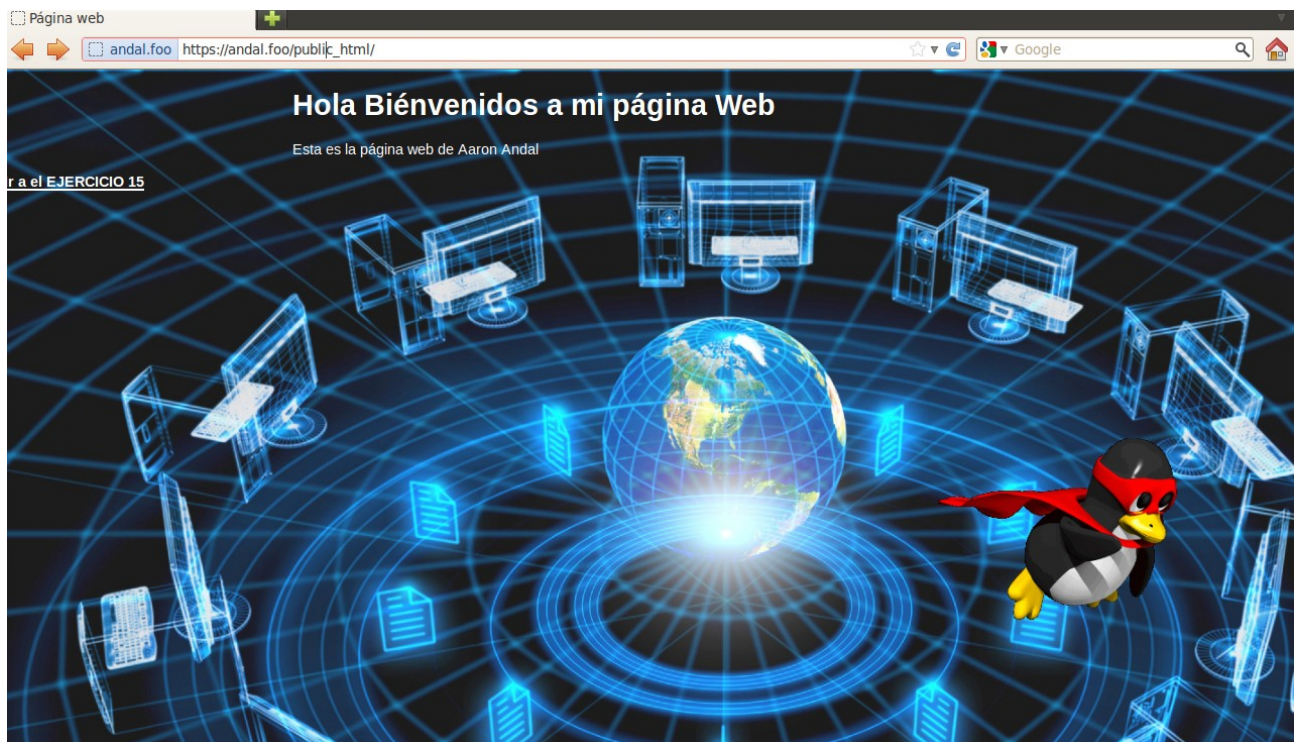
```
SSLCertificateFile /etc/ssl/certs/server.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/server.key
```

10. Habilitamos el archivo

```
sudo a2ensite default-ssl
```

11. REINICIAMOS





```
root@ubuntu:/etc/apache2/sites-available# service apache2 restart
* Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
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 127.0.1.1:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
[ OK ]
root@ubuntu:/etc/apache2/sites-available#
```

## Servidores VIRTUALES

Configurar los dominios con /etc/hosts o bind9

/etc/apache2/sites-available/default

NameVirtualHost \*:80

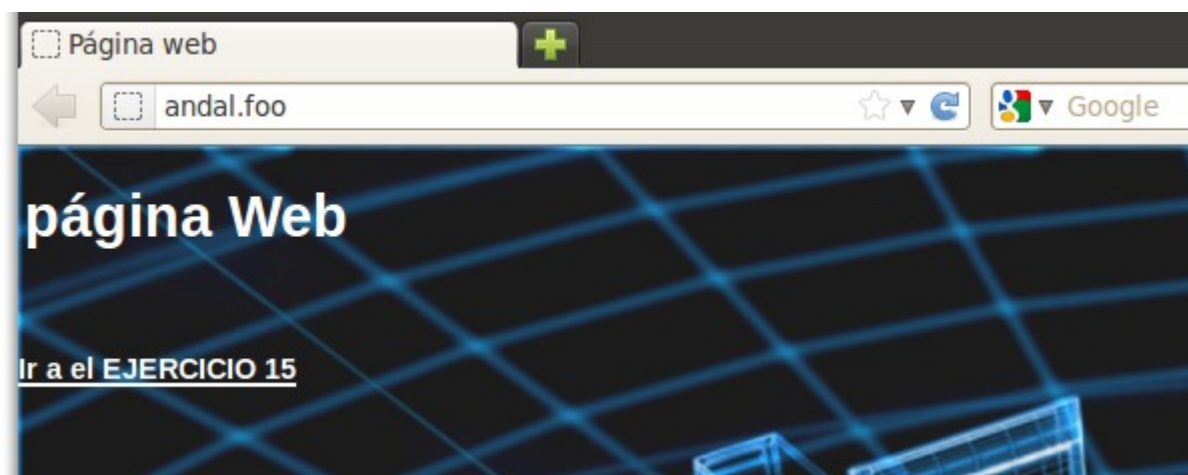
<VirtualHost \*:80>

ServerName www.NOM.foo

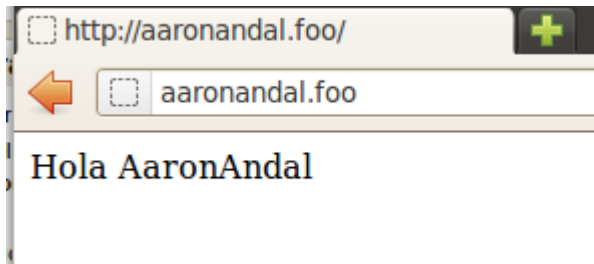
DocumentRoot /var/www/NOM

</VirtualHost>

Servidor virtual: andal.foo



Servidor virtual: aaronandal.foo



**Al iniciar la máquina teníamos un error al reiniciar apache2:**

**(98)address already in use make\_sock could not bind to address**

**Para solucionarlo**

**kill -9 (PID\_apache2)**

**service apache2 restart**



## Instal·lació de ssl para Apache

1.- Hem de tindre un certificat, el creem

```
make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/certs/apache2.pem
```

2.- Ens hem d'assegurar que dins de /etc/apache2/ports.conf està descomentada la línia que posa Listen 443.

3.- Hem de configurar dins /etc/apache2/sites-available/default

Afegim:

```
NameVirtualHost websegura.ubuntu.foo:443
<VirtualHost websegura.ubuntu.foo:443 >
  ServerName websegura.ubuntu.foo
  DocumentRoot /var/www/websegura
  SSLEngine On
  SSLCertificateFile /etc/ssl/certs/apache2.pem
  ErrorLog /var/log/apache2/error.log
  CustomLog /var/log/apache2/access.log combined
</VirtualHost>
<Directory "/var/www/websegura">
  Options Indexes FollowSymlinks MultiViews
  AllowOverride None
  Order allow,deny
  Allow from all
  SSLRequireSSL
</Directory>
```

4.- Hem d'activar el mòdul ssl

```
a2enmod ssl
```

5.- Editem el fitxer hosts per a que sàpiga la IP de nuestra web.

6.- Hem de crear la carpeta de la nostra web segura

```
/var/www websegura
```





Este es el Servidor websegura

```
NameVirtualHost aaronandal.foo:443
<VirtualHost aaronandal.foo:443>
    ServerName aaronandal.foo
    DocumentRoot /var/www/aaronandal
    SSLEngine On
    SSLCertificateFile /etc/ssl/certs/apache2.pem
    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/error.log combined
    <Directory "/var/www/aaronandal">
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        SSLRequireSSL
    </Directory>
</VirtualHost>
```

---

```
NameVirtualHost andal.foo:443
<VirtualHost andal.foo:443>
    ServerName andal.foo
    DocumentRoot /var/www/andal
    SSLEngine On
    SSLCertificateFile /etc/ssl/certs/apache2.pem
    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/error.log combined
    <Directory "/var/www/andal">
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        SSLRequireSSL
    </Directory>
</VirtualHost>

NameVirtualHost aaronandal.foo:443
<VirtualHost aaronandal.foo:443>
    ServerName aaronandal.foo
    DocumentRoot /var/www/aaronandal
    SSLEngine On
    SSLCertificateFile /etc/ssl/certs/apache2.pem
    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/error.log combined
    <Directory "/var/www/aaronandal">
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        SSLRequireSSL
    </Directory>
</VirtualHost>
```

---



## Opción 3 Barrantes

### Configuracion de servicio https

1.- Instalamos Apache o en mi caso el LAMP (con tasksel) o

```
sudo aptitude install apache2
```

2.- Habilitamos el modulo SSL

```
sudo a2enmod ssl
```

3.- Habilitamos la configuracion SSL por default

```
sudo a2ensite default-ssl
```

Reiniciamos el servidor

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

4.- Creamos las llaves

```
cd /etc/apache2
```

```
sudo openssl genrsa -des3 -out server.key 1024
```

5.- Creamos el certificado con la llave

```
sudo openssl req -new -key server.key -out server.csr
```

6.- Creamos nuestro certificado, contestando las preguntas que nos indique recordando la contraseña

```
sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

7.- Cambiamos de carpeta la llave y certificado

```
sudo cp server.crt /etc/ssl/certs/
```

```
sudo cp server.key /etc/ssl/private/
```

Abrimos el archivo de configuracion default-ssl

```
cd /etc/apache2/sites-available
```

```
sudo vim default-ssl
```

reemplazamos y habilitamos la siguiente configuracion

```
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
```

```
SSLCertificateFile /etc/ssl/certs/server.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/server.key
```

8.- Habilitamos el archivo

```
sudo a2ensite default-ssl
```

Reiniciamos apache y listo!!

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

9.-abrimos la terminal y ejecutamos

```
make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/certs/apache2.pem
```

10.- Luego luego habilitamos el puerto Listen 443

```
sudo gedit /etc/apache2/ports.conf
```

10A.- Tenemos que crear en el /var/www el directorio que queramos que sea https, en este caso serÃa /var/www/caixard

11.- Luego agregamos el directorio donde queremos poner el https en el archivo sudo gedit /etc/apache2/sites-available/default-ssl

```
<VirtualHost _default_:443>
```

```
    ServerAdmin webmaster@localhost
```

```
    DocumentRoot /var/www/chasekid
```

```
    <Directory />
```

```
        Options FollowSymLinks
```

```
        AllowOverride None
```

```
    </Directory>
```

```
    <Directory /var/www/chasekid>
```

```
        Options Indexes FollowSymLinks MultiViews
```

```
        AllowOverride None
```

```
        Order allow,deny
```

```
        allow from all
```

```
    </Directory>
```

12.- Entramos el el siguiente archivo sudo gedit /etc/apache2/sites-available/default

**NameVirtualHost \*:80**

**<VirtualHost \*:80>**

**ServerName www.chasekid.foo**

**DocumentRoot /var/www/chasekid**

**ServerAlias chasekid.foo**

**</VirtualHost>**

**<VirtualHost \*:443>**

**SSLEngine on**

**SSLCertificateFile /etc/ssl/certs/server.crt**

**SSLCertificateKeyFile /etc/ssl/private/server.key**

**ServerName www.chasekid.foo**

**DocumentRoot /var/www/chasekid**

**ServerAlias chasekid.foo \*.chasekid.foo**

**</VirtualHost>**

**13.- Tenemos que añadir al fichero gedit /etc/hosts la línea correspondiente.**

**192.168.1.203chasekid.foo**

**Tenemos que tener en cuenta que este añadido anterior ha de estar antes de la instrucción en perl.**

**Luego reiniciamos el servicio apache**

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

Redirigir el HTTPS/SSL para que siempre cuando introduzcamos el dominio.foo sin https (http) nos redirija en el seguro.

Añadir lo siguiente en el archivo html

```
<META HTTP-EQUIV="REFRESH" CONTENT="0;URL=https://www.chasekid.foo">
<title>P&acute;gina web</title>
```

Permitir a usuarios y su password en el acceso web.

En /var/www/usuarios

touch .usuarios

htpasswd -c .usuarios aaron

pass: ....

En /etc/apache2/sites-available/default

```
<Directory /var/www/usuarios>
    AuthType Basic
    AuthName "PPE: &Aacute;rea de Aaron"
    AuthUserFile /var/www/usuarios/.usuarios
    require valid-user
</Directory>
```





## En HTTPS

```
<VirtualHost aaronandal.foo:443>
    ServerName aaronandal.foo
    DocumentRoot /var/www/aaronandal
    SSLEngine On
    SSLCertificateFile /etc/ssl/certs/apache2.pem
    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/error.log combined
    <Directory "/var/www/aaronandal">
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        SSLRequireSSL
    </Directory>
    <Directory "/var/www/aaronandal">
        AuthType Basic
        AuthName "PPE: área de administración"
        AuthUserFile /var/www/aaronandal/.usuarios
        require valid-user
    </Directory>
</VirtualHost>
```