



Laboratorio Servidor Web Apache

El Material

Para este laboratorio solo necesitamos una computadora con sistema operativo GNU/Linux con conexión a internet.

Objetivos

- Mostrar el funcionamiento de un servicio web.
- Desarrollar habilidades para poder configurar un servidor web apache en GNU/Linux.
- Realizar capturas de pantalla de lo realizado en cada uno de los tres ejercicios.

La práctica de laboratorio consistirá en aprender a instalar y configurar el servidor web Apache 2. Cada configuración del laboratorio tiene una práctica el cual deberán realizar capturas de pantallas y ser mostrado al docente para ameritar calificación, de otra manera no se tomara en cuenta para nota.

Índice de contenido

| | |
|--|--------------------|
| El Material | 1 |
| Objetivos | 1 |
| Teoría | 2 |
| Componentes | 2 |
| Petición GET | 2 |
| Petición POST | 3 |
| Práctica | 4 |
| Instalación | 4 |
| Modos para desenvolverse en Apache2 | 4 |
| Configuración de Sitio Sencillo | 5 |
| Ejercicio #1 | 7 |
| Activación de Cifrado en el sitio (SSL, HTTPS) | 7 |
| Ejercicio #2 | 9 |
| Re dirección de Sitios | 9 |
| Ejercicio #3 | 10 |
| Temas no tocados | 10 |
| Referencias | 11 |
| Anexos | 12 |
| Script de configuración sencilla | 12 |
| Script de configuración ssl | 12 |
| Script de configuración rewrite | 12 |

Teoría

Un **servidor web** o **servidor HTTP** es un [programa informático](#) que procesa una [aplicación del lado del servidor](#) realizando conexiones bidireccionales y/o unidireccionales y [síncronas o asíncronas](#) con el cliente generando o cediendo una respuesta en cualquier lenguaje o [Aplicación del lado del cliente](#). El código recibido por el cliente suele ser [compilado](#) y ejecutado por un [navegador web](#). Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo [HTTP](#) para estas comunicaciones, perteneciente a la capa de aplicación del [modelo OSI](#). El término también se emplea para referirse al ordenador que ejecuta el programa.

La función principal de un servidor web es entregar [páginas web](#) a solicitud de clientes usando el [Protocolo de Transferencia de Hipertexto\(HTT](#)P). Esto significa que entrega documentos HTML e información adicional como imágenes, css y scripts.

Componentes

El servicio web tiene dos componentes: Cliente, agente o navegador Web ó agente y el Servidor web.

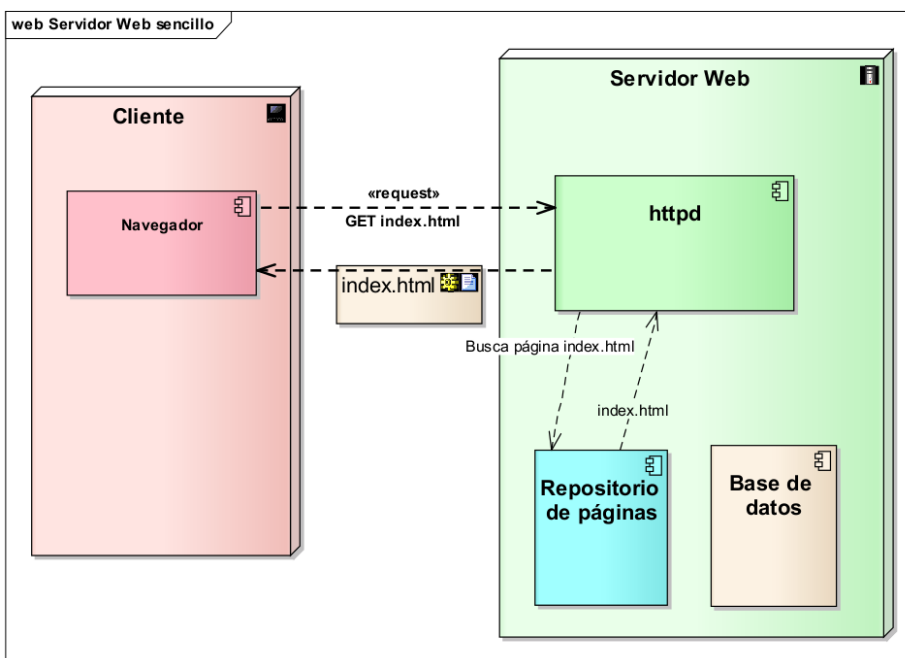


Ilustración 1: Servidor Web Sencillo

Navegadores Web

Existe variedad en los clientes

web, ya sea por su diseño o su funcionamiento, pero todos deben seguir un estandar para el interprete HTML. La diferencia marcada entre los navegadores son los motores que utilizan entre los que estan: WebKit, IE y Gecko puede ver una lista en internet¹.

Servidores Web

Para los servidores web, también existen variedad de servidores, pero los principales dominantes del WWW son Apache y IIS. Pueden ver una lista comparativa de servidores en la wikipedia².

Petición GET

Un servidor web opera mediante el protocolo [HTTP](#), de la capa de aplicación del [Modelo OSI](#). Al protocolo HTTP se le asigna habitualmente el puerto [TCP 80](#). Las peticiones al servidor suelen realizarse mediante HTTP utilizando el método de petición [GET](#) en el que el recurso se solicita a través de la url al servidor web.

GET /index.html HTTP/1.1 HOST: www.host.com

¹Lista de navegadores web http://es.wikipedia.org/wiki/Navegador_web

² Comparativa de servidores web http://en.wikipedia.org/wiki/Comparison_of_web_server_software

En la barra de URL de un navegador cualquiera la petición anterior sería análoga a la siguiente dirección Web:

`www.host.com/index.html`

Petición POST

Es el segundo tipo de petición [HTTP](#) más utilizado. Los datos a enviar al servidor se incluyen en el cuerpo de la misma petición con las cabeceras HTTP asignadas correspondientemente respecto al tipo de petición. Generalmente se asocia con los [formularios web](#) en el que los datos suelen ser cifrados para enviarlos de manera segura al servidor.

Por motivos de convención se incluye en la petición la cabecera `application/x-www-form-urlencoded` que indica el formato o codificación de los datos a enviar; esta es *variable->valor* en el formato: `variable=valor` separada cada par `variable->valor` por `&`. Esta cabecera, en los formularios [HTML](#) se envía automáticamente, pero en otras tecnologías web tal como [AJAX](#), si se desea hacer correctamente una petición POST debe ser especificado o instanciado el objeto: `setRequestHeader("Content-type:application/x-www-form-urlencoded");`
`ajax.send(data);`

Si se utilizase el método GET los datos deberían de ser añadidos a la URL, lo que los expondría a ser vistos de forma directa.

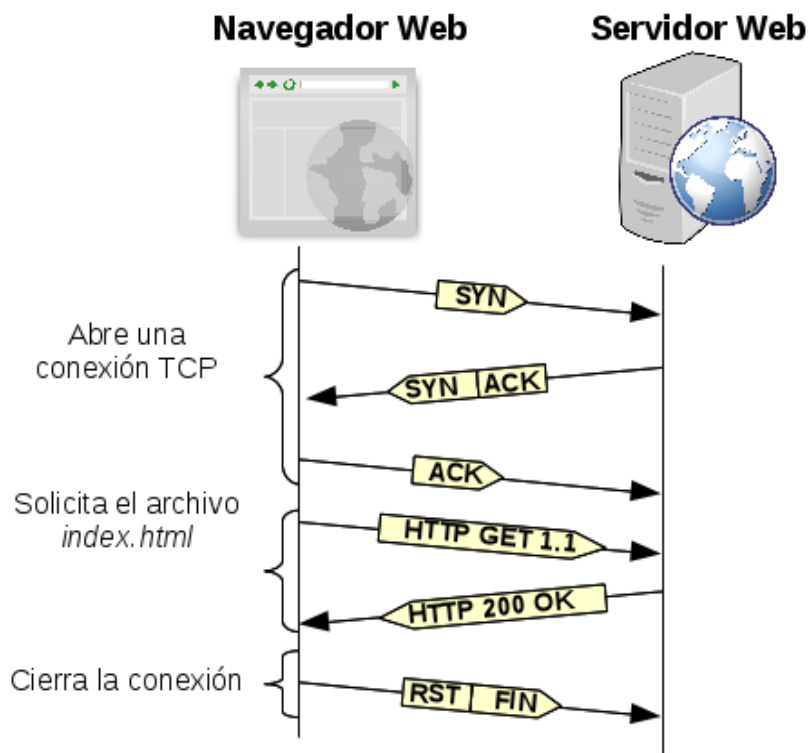


Ilustración 2: Comunicación HTTP

Por cada archivo `.html`, `.css`, `.js`, y de imagen (`jpg`, `png`, etc.), realiza esta comunicación (ver Ilustración 2: Comunicación HTTP), por lo tanto si carga 7 archivos, son siete veces que hace el `syn`, `syn ack`, `ack`, `http get`, `http ok`, `rst fin`. Ejemplo, ejecutar en la terminal:

```
$ wget --page-requisites -vv http://www.w3schools.com/html/default.asp
```

Práctica

Para la práctica se utilizará el servidor web Apache 2 , de la Fundación Apache³



Instalación

Para instalar los archivos necesarios de nuestro servidor Apache podemos utilizar aptitude/apt-get desde una consola como usuario root⁴:

```
// Instalación del servidor Web Apache
# aptitude update

# aptitude install apache2-mpm-prefork apache2 apache2.2-common
```

Instalamos el `apache2-mpm-prefork` explícitamente por ser el módulo de multiprocesado de apache que se complementa con php5.

Para verificar que nuestra instalación fué exitosa, simplemente abrimos nuestro navegador con la url <http://localhost/> ó <http://127.0.0.1> y tendríamos que ver la siguiente página.

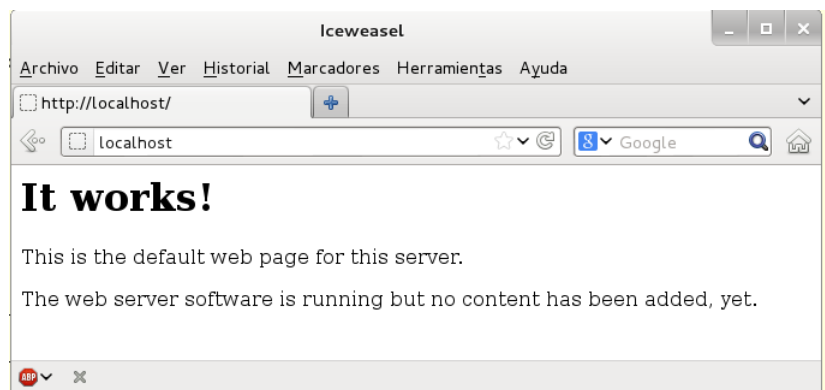


Ilustración 4: Pantalla por defecto de apache

El servidor de apache nombra cada sitio que se configure `VirtualHost` (host virtual), este nombre denota la configuración para cada uno de los sitios que estén publicados por apache. Cada sitio comienza y termina con la etiqueta `VirtualHost` .

Modos para desenvolverse en Apache2

Carpeta de Configuración

Existen tres carpetas que debemos conocer sobre nuestro servidor web, y estas son:

1. `/etc/apache2/` : Donde van la configuraciones de nuestro sitios web y del servidor web.
2. `/var/log/apache2/` : Donde van los logs⁵ de nuestros sitios web
3. `/var/www/` : Donde van los archivos de nuestro sitio web.

³ Sitio oficial de la Fundación Apache <http://www.apache.org/>

⁴ Se asume que los repositorios de software ya se tienen configurados en `/etc/apt/sources.list`

⁵ Log: La palabra **log** es un término anglosajón, equivalente a la palabra [bitácora](#) en [español](#), para este documento se utiliza para guardar registros de actividad del servidor apache.

Además están las sub carpetas y archivos que es donde nosotros realizaremos las configuraciones, estas son:

1. `/etc/apache2/sites-available/` : Configuraciones de los VirtualHosts disponibles.
2. `/etc/apache2/sites-enabled/` : VirtualHosts Activados (Que se pueden ver desde un navegador web)
3. `/etc/apache2/mods-available/` : Configuraciones de los modulos disponibles de apache.
4. `/etc/apache2/mods-enabled/` : Configuraciones de los modulos habilitados de apache.
5. `/etc/apache2/ports.conf` : Configuraciones de los puertos donde escucha nuestro apache.

Iniciar, detener y recargar el servidor web

Para poder iniciar el servidor apache ejecutar:

```
# /etc/init.d/apache2 start
```

ó

```
# service apache2 start
```

Para poder detener el servidor apache ejecutar:

```
# /etc/init.d/apache2 stop
```

ó

```
# service apache2 stop
```

Para poder recargar la configuración del servidor apache ejecutar:

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

ó

```
# service apache2 reload
```

Activar y desactivar sitios web

Para poder activar el sitio (VirtualHost):

```
# a2ensite <nombre del archivo de la configuración>
```

Para poder desactivar el sitio web (VirtualHost):

```
# a2dissite <nombre del archivo de la configuración>
```

Activar y desactivar modulos web

Para poder activar el módulo:

```
# a2enmod <nombre del módulo>
```

Para poder desactivar el módulo:

```
# a2dismod <nombre del módulo>
```

Ver los logs de nuestro servidor en modo interactivo

Ejecutar:

```
# tail -f /var/log/apache2/*.log
```

Para detenerlo, presionar Ctrl+C

Configuración de Sitio Sencillo

Creación de nuestro sitio

Creamos una carpeta llamada `redes1` en `/var/www/` y allí descomprimos el archivo de nuestro

sitio de ejemplo (<http://www.freecsstemplates.org/download/zip/crosswalk/>).

Crear entrada en el dns local

Editar el archivo `/etc/hosts` y agregar la siguiente línea:

```
127.0.0.1    uls.redes1
```

En este paso estamos configurando un dns local, para que al digitar el dominio `uls.redes1` nos muestre nuestro servidor web.

Configuración del VirtualHost

Crear el archivo `/etc/apache2/sites-available/uls.redes1` con el siguiente contenido:

```
<VirtualHost *:80>
    DocumentRoot /var/www/redes1/
    ServerName uls.redes1
    LogLevel warn
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

La configuración del virtual contiene varias directrices, estas son:

1. `<VirtualHost *:80>` : Donde le indicamos el inicio de la configuración del sitio, que puerto y que en que IPs (en caso de que nuestro servidor tenga más de una IP) del servidor queremos que publique nuestro sitio.
2. `DocumentRoot`: Que nos muestra la carpeta donde está almacenado nuestro sitio. Alternativamente podemos poner nuestro sitio en una carpeta del usuario para que podamos modificar archivos de nuestro sin tener que cambiar permisos a la carpeta `/var/www/` .
3. `ServerName`: Es el nombre que responderá nuestro sitio. Si se requiere que responda a más de un nombre se agrega la directriz `ServerAlias`, por ejemplo `ServerAlias www.uls.redes1`

Activación del VirtualHost

Se ejecuta el comando:

```
# a2ensite uls.redes1
```

Recargar la configuración de apache

Hacer la recarga de la configuración de apache de acuerdo al comando que nos muestra.

Verificación del sitio

Abrir en el navegador la siguiente dirección: <http://uls.redes1> y deberá aparecer algo como lo siguiente.



Ilustración 5: Pantalla de Sitio de Ejemplo

Verificación de Registro

Para ver los registros de actividad del sitio pueden hacerlo a través de los logs del apache:

```
# tail -f /var/log/apache2/*.log
```

ó a través de Wireshark. Si queremos cancelar la lectura de logs, ejecutaremos `Ctrl+C`.

En el anexo pueden encontrar los comandos sugeridos a usar en consola para la configuración del sitio web sencillo.

Ejercicio #1

Configurar un `VirtualHost` adicional que responda al nombre `redes1.<carnet>` donde `carnet` es su número de carnet del estudiante que realiza la práctica. Puede descargar un sitio de ejemplo desde <http://www.freecsstemplates.org/>

Después de haber terminado el ejercicio, seguir con lo siguiente. No olvidarse de realizar **capturas de pantalla**.

Activación de Cifrado en el sitio (SSL, HTTPS)

Los certificados de seguridad permiten enviar información a través de la red de forma segura imposibilitando la el robo de información por un agente intermedio. Para habilitar los certificados de seguridad que vienen por defecto con apache, hacemos:

Habilitar el modulo de apache ssl

Para ello utilizamos el habilitador de módulos de apache

```
# a2enmod ssl
```

Configuración del VirtualHost SSL

Para la configuración del VirtualHost con ssl, utilizaremos la configuración del sitio `uls.redes1`, la copiaremos y el nuevo archivo se llamara `uls.redes1-ssl`. El contenido del archivo será el siguiente:

```
<VirtualHost *:443>
    DocumentRoot /var/www/redes1/
    ServerName uls.redes1

    SSLEngine on

    SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
    SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

    LogLevel warn

    ErrorLog ${APACHE_LOG_DIR}/error-ssl.log
    CustomLog ${APACHE_LOG_DIR}/access-ssl.log combined
</VirtualHost>
```

Los cambios que se hicieron con respecto a la configuración del sitio sin SSL son:

1. En vez de utilizar el puerto 80, se utiliza el puerto 443.
2. `SSLEngine on`: Le decimos que vamos a utilizar el motor de ssl.
3. `SSLCertificateFile`: Le decimos qué certificado utilizaremos para el sitio.
4. `SSLCertificateKeyFile` : Le decimos que llave privada utilizaremos para el certificado.

Para obtener más información sobre cómo generar los certificados de seguridad para el apache, se recomienda leer la información del siguiente enlace <http://www.certsuperior.com/SSLApache.aspx>

Activación del VirtualHost

Se ejecuta el comando:

```
# a2ensite uls.redes1-ssl
```

Recargar la configuración de apache

Hacer la recarga de la configuración de apache de acuerdo al comando que nos muestra.

Verificación del sitio SSL

Abrir en el navegador la siguiente dirección: <https://uls.redes1>

Verificación de Registro

Para ver los registros de actividad del sitio pueden hacerlo a través de los logs del apache:

```
# tail -f /var/log/apache2/*-ssl.log
```

ó a travez de wireshark

En el anexo pueden encontrar los comandos sugeridos a usar en consola para la configuración del sitio web con ssl.

Ejercicio #2

Crear un sitio VirtualHost copiando la configuración del ejercicio #1 con certificados de seguridad, de modo que al abrir el navegador se puede acceder a la dirección `https://redes1.<carnet>` donde `<carnet>` es el carnet del estudiante que esta desarrollando la práctica.

Después de haber terminado el ejercicio, seguir con lo siguiente. No olvidarse de realizar **capturas de pantalla**.

Re dirección de Sitios

La redirección nos sirve para poder reescribir al vuelo las solicitudes que llegan a nuestro servidor. Para realizar esto utilizamos el módulo `rewrite` de apache. En ejemplo de una redirección puede ser que necesitemos redirigir el tráfico de un dominio viejo al nuevo. Aunque la redirección puede ser mas complicada utilizando expresiones regulares⁶, en este documento solo realizaremos una redirección sencilla redirigiendo el tráfico de <http://uls.redes1> a <http://www.google.com> haciendo una búsqueda con unas palabras. Para más información de como se usa el módulo `rewrite` ver http://httpd.apache.org/docs/current/mod/mod_rewrite.html.

Habilitar el modulo de apache rewrite

Para ello utilizamos el habilitador de módulos de apache

```
# a2enmod rewrite
```

Configuración del VirtualHost Rewrite

Para la configuración del VirtualHost con `rewrite`, utilizaremos la configuración del sitio `uls.redes1`, y le agregaremos las siguientes líneas después de la directiva `ServerName`.

```
RewriteEngine on
```

```
RewriteRule ^/buscar/(.*) http://www.google.com/search?as_q=$1 [R,L]
```

Los cambios que se agregaron son:

1. `RewriteEngine on`: Le decimos que vamos a utilizar el motor `rewrite`.

⁶ Puede ver mas información sobre expresiones regulares en http://es.wikipedia.org/wiki/Expresi%C3%B3n_regular

2. RewriteRule: Le decimos qué queremos hacer una redirección
 - a) `^/buscar/`: Todo lo que comienza con `"/buscar/"`.
 - b) `(.*)`: Argumentos enviarlos a la variable `$1`.
 - c) `R`: (redirect) bandera para forzar una redirección HTTP.
 - d) `L`: (last) bandera para detener el procesamiento de redireccionado y no se quede en un bucle infinito.

Para obtener más información sobre Re escritura de URLs, se recomienda leer la información del siguiente enlace <http://dns.bdat.net/documentos/apache/x936.html>

Recargar la configuración de apache

Hacer la recarga de la configuración de apache.

Verificación de la redirección

Abrir en el navegador la siguiente dirección: <https://uls.redes1/buscar/luterana> y verificar que sucede después.

Verificación de Registro

Para ver los registros de actividad del sitio pueden hacerlo a través de los logs del apache:

```
# tail -f /var/log/apache2/*.log
```

ó a travez de wireshark

En el anexo pueden encontrar los comandos sugeridos a usar en consola para la configuración de Re dirección de sitio.

Ejercicio #3

Configurar el `VirtualHost` del del ejercicio #1 para que redirija el tráfico hacia <http://proyectos.uls.edu.sv>

No olvidarse de realizar **capturas de pantalla**.

Temas no tocados

Aunque hemos visto una lista de configuraciones de apache amplia e interesante, siempre se quedan temas que no se verán por cuestión de tiempo. Dentro de esos temas están:

Control de accesos⁷, expresiones regulares⁸, proxy inverso⁹, webDav¹⁰, directorios con claves¹¹, etc.

Índice de ilustraciones

| | |
|--|---|
| Ilustración 1: Servidor Web Sencillo..... | 2 |
| Ilustración 2: Comunicación HTTP..... | 3 |
| Ilustración 3: Logo Apache..... | 4 |
| Ilustración 4: Pantalla por defecto de apache..... | 4 |
| Ilustración 5: Pantalla de Sitio de Ejemplo..... | 7 |

Referencias:

- **Presentación:** Servidores Web – Apache
Autor: Juan Pavón Mestras
Editorial: Universidad Complutense Madrid
URL: <http://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>
- **Networking Fundamentals**
Autor: Silvano Da Ros
Editorial: Cisco Press
URL: <http://fengnet.com/book/CNF/ch08lev1sec1.html>
- **Página:** Servidor Web, Navegador Web
Autor: Fundación Wikimedia Inc
URL: http://es.wikipedia.org/wiki/Servidor_web , http://es.wikipedia.org/wiki/Navegador_web

7 Control de Accesos: <http://httpd.apache.org/docs/2.0/es/howto/auth.html>

8 Expresiones Regulares: <http://iie.fing.edu.uy/~vagonbar/unixbas/expreg.htm>

9 Proxy Inverso: <http://rm-rf.es/configurar-apache-como-reverse-proxy-proxy-inverso/>

10 WebDav: <http://ubuntuforums.org/archive/index.php/t-157415.html>

11 Directorios con claves: <http://www.juanfelipe.net/node/23>

Anexos

Script de configuración sencilla

```
mkdir /var/www/redes1
cd /var/www/redes1
wget http://www.freecsstemplates.org/download/zip/crosswalk/
mv index.html crosswalk.zip
unzip crosswalk.zip
nano /etc/hosts
nano /etc/apache2/sites-available/uls.redes1
a2ensite uls.redes1
/etc/init.d/apache2 restart
links2 http://uls.redes1
```

Script de configuración ssl

```
a2enmod ssl
cd /etc/apache2/sites-available/
cp uls.redes1 uls.redes1-ssl
nano uls.redes1-ssl
/etc/init.d/apache2 restart
links2 https://uls.redes1
```

Script de configuración rewrite

```
a2enmod rewrite
nano /etc/apache2/sites-available/uls.redes1
nano uls.redes1
/etc/init.d/apache2 restart
links2 https://uls.redes1/buscar/pricensa
```