

Redes de Computadoras

—

Prácticas

Juana López Redondo

María Laura Da Silva Hernández

Vicente González Ruiz

2 de junio de 2011

El servicio Web

La World Wide Web (también llamada WWW o simplemente “la Web”) es un sistema de información (actualmente enorme, posiblemente el más grande de la historia) basado en millones de páginas Web distribuidas por Internet. Esta información es servida bajo demanda por los servidores Web y consultada por los usuarios a través de los clientes (también llamados navegadores) Web.

9.1. Los servidores Web

Un servidor Web es un servicio que proporciona acceso a objetos Web alojados en un host con acceso a Internet. Los servidores Web aceptan peticiones (generalmente a través del puerto 80) de los clientes Web.

Existen muchos servidores Web. Aquí mostramos una lista con algunos de ellos: CERN httpd (<http://www.w3.org/Daemon>), Apache (<http://httpd.apache.org>) y Microsoft Internet Information Server (<http://www.microsoft.com/WindowsServer2003/iis/default.mspx>).

9.2. Los navegadores Web

Un navegador es la parte cliente de la Web. Cada vez que utilizamos uno y descargamos una página estamos realizando una petición HTTP a un servidor Web, y éste nos contesta con una respuesta HTTP que contiene el objeto Web solicitado.

Inicialmente los navegadores Web sólo podían mostrar texto. Sin embargo, actualmente existen muchos tipos de objetos Web que los navegadores son capaces de procesar (como imágenes, sonidos y vídeos, programas escritos en Java y Flash, etc.). Esta es una lista de algunos de los navegadores Web más famosos: Amaya (<http://www.w3.org/Amaya>), Epiphany (<http://www.gnome.org/projects/epiphany>), Galeon (<http://galeon.sourceforge.net>), Windows Internet Explorer (<http://www.microsoft.com/spain/windows/ie/default.mspx>), Konqueror

9.3 El HTTP (HyperText Transfer Protocol)

(<http://konqueror.org>), Lynx (<http://lynx.browser.org>), Mozilla Firefox (<http://www.mozilla-europe.org/es/products/firefox>), Netscape Navigator (<http://browser.netscape.com>), Opera (<http://www.opera.com>), Safari (<http://www.apple.com/es/macosx/features/safari>) y Shiira (<http://hmdt-web.net/shiira/en>).

9.3. El HTTP (HyperText Transfer Protocol)

El HTTP (<http://www.w3.org/Protocols>) es el protocolo de transferencia de información básico¹ en la Web. Fue ideado a principios de los 90 en el CERN (<http://www.cern.ch>) por Tim Berners-Lee (<http://www.w3.org/People/Berners-Lee>).

9.4. Los objetos Web y las URL's

El nombre común para todo lo que es posible transferir a través de la Web se conoce como *objeto Web*. Por definición cada objeto puede ser referenciado a través de una URL (Uniform Resource Locator). En los navegadores Web, las URLs se muestran en la entrada que se sitúa normalmente en la parte superior de la ventana del navegador.

9.5. El HTML (HyperText Markup Language)

El HTML (<http://www.w3.org/MarkUp>) es el lenguaje en el que están escritas las páginas Web. En su versión más básica, una página Web es un conjunto de instrucciones HTML que referencian a otros objetos Web.

9.6. La World Wide Web Consortium (W3C)

La World Wide Web Consortium (W3C) es el organismo internacional encargado de regular el desarrollo de la Web. La Web es el sistema de información más importante de Internet y muchas compañías y grupos de investigación (entre otros) añaden constantemente nuevas funcionalidades. La W3C se encarga (básicamente) de que cualquier cliente Web sea capaz de comunicarse con cualquier servidor Web.

9.7. Los proxys Web

La Web consume un gran porcentaje del ancho de banda de Internet. Un proxy Web que es un tipo especial de servidor Web que funciona como una caché de objetos Web y por tanto, instalados en los puntos adecuados, pueden reducir significativamente el consumo de ancho de banda.

¹Ahora existen muchos otros que también se utilizan como el File Transfer Protocol (FTP).

9.8 La caché de los navegadores Web

El funcionamiento de un proxy Web es bastante simple de entender. Un proxy típicamente es utilizado por los usuarios de una red con suficientemente ancho de banda. Si suponemos que generalmente dichos usuarios van a acceder a un conjunto de objetos Web más de una vez en un intervalo de tiempo dado, es más eficiente colocar dichos objetos en el proxy Web y acceder a él.

En realidad un proxy Web funciona como servidor y como cliente. Cuando un usuario configura su navegador Web para utilizar un proxy Web y accede a un objeto Web a través de su URL original (que referencia el objeto en el servidor Web, no en el proxy Web), en realidad accede (siempre) al proxy Web. En el caso más simple, el proxy Web entonces busca el objeto en su caché y si lo encuentra “fresco”², se lo sirve al navegador Web cliente. Nótese que en este contexto el proxy Web está funcionando como un servidor Web.

Pero, ¿qué ocurre cuando el proxy Web no contiene el objeto Web solicitado? Entonces el proxy Web actúa como un cliente Web e intenta acceder al objeto Web a través de su URL original. Cuando lo consigue, actualiza su caché y lo envía al navegador Web del usuario. Si lo tiene entonces simplemente pregunta al siguiente servidor de la jerarquía Web (que podría ser otro proxy Web) si el objeto que contiene está realmente fresco y puede servirlo o si por lo contrario está obsoleto y debe reclamarlo antes de servirlo.

Finalmente indicar que, aunque no configuremos nuestro navegador Web para utilizar un proxy Web, es probable que lo estemos utilizando. Muchos ISP colocan **proxys Web transparentes** para reducir el tráfico Web a través de su red.

Existen decenas de implementaciones diferentes de proxys Web. Sin embargo el más utilizado es, con diferencia, Squid (<http://www.squid-cache.org>).

9.8. La caché de los navegadores Web

El tema del ahorro de ancho de banda es una cuestión fundamental, y no sólo para los administradores de las redes. Por este motivo, la mayoría de los navegadores Web implementan su propia caché con el objeto de reducir el tráfico. Así, cuando accedemos a un objeto, el navegador primero comprueba si tiene una copia del mismo en su caché y si este está “fresco”³, entonces el navegador visualiza el objeto sin necesidad de descargarlo de Internet.

9.9. El cliente Web Mozilla Firefox

Mozilla Firefox es un navegador Web que ha sido desarrollado por Mozilla Corporation (<http://www.mozilla.org/reorganization>) y cientos de voluntarios, siendo, junto con Microsoft Internet Explorer y Apple Safari, uno de los navegadores más utilizados. Además, está disponible prácticamente para todos los sistemas

²Para ello: (1) o el proxy le pregunta al servidor Web que contiene el objeto si su copia es tan nueva como la que él tiene.

³Recuérdese que ésto implica en cualquier caso realizar un GET condicional.

9.10 Apache

operativos con entorno gráfico y gracias a que se distribuye en código fuente es considerado, por muchos (y entre ellos el autor de este texto) como el navegador más seguro que existe.

9.9.1. Instalación

Debian's:

```
root# apt-get install firefox
```

Red Hat's:

```
root# yum install firefox
```

Gentoo's:

```
root# emerge firefox
```

9.9.2. Ejecución

```
usuario$ firefox &
```

9.10. Apache

Apache es un servidor Web. De hecho, es el servidor Web más difundido actualmente (http://news.netcraft.com/archives/web_server_survey.html). Si a esto sumamos que la Web es la aplicación más difundida en Internet, podemos asegurar que Apache es el servidor más extendido de la historia.

9.10.1. Instalación

Debian's:

```
root# apt-get install apache2
```

Red Hat's:

```
root# yum install httpd
```

Gentoo's:

```
root# emerge apache
```

9.10 Apache

9.10.2. ¿Está corriendo el servicio?

Si tras lanzar:

```
usuario$ firefox http://localhost &
```

se carga un objeto Web, entonces la instalación ha sido un éxito.

9.10.3. Configuración

El directorio:

Debian's:

```
/etc/apache2/
```

Red Hat's:

```
/etc/httpd/
```

Gentoo's:

```
/etc/apache2/
```

contiene una serie de ficheros y directorios. El fichero de configuración principal es **apache2.conf**. El resto de ficheros de configuración son cargados desde éste. Entre los directorios encontramos los que indican qué módulos hay disponibles y qué módulos hay habilitados (veremos más tarde qué son los módulos), así como qué *sites* hay disponibles y habilitados (esto está relacionado con los *virtual hosts*, más tarde también hablaremos de esto).

Cada vez que Apache es ejecutado (o re-ejecutado) se leen los ficheros de configuración que están bastante bien documentados. Dichos ficheros se organizan en secciones que contienen un conjunto de directivas. Cada directiva debe escribirse en una única línea. Veamos cómo controlar algunas de las posibilidades de Apache modificando dichas directivas y el contenido de los directorios de configuración.

9.10.4. Localización de los objetos Web en el servidor

La directiva **DocumentRoot** define el directorio que contiene los objetos Web del servidor. Dicha directiva cuando no se declara, tiene el siguiente valor por defecto:

Debian's:

```
/var/www
```

Red Hat's:

9.10 Apache

`/var/www/html`

Gentoo's:

`/var/www/localhost/htdocs`

Apache además puede servir los objetos que se colocan en el directorio `public_html` de los usuarios. Por ejemplo, para acceder al directorio `public_html` del usuario "alumno" en la máquina local deberíamos utilizar la URL:

`http://localhost/~alumno`

Para conseguir esto debemos hacer lo siguiente:

1. **Cargar el módulo "userdir":**⁴ En este caso necesitamos cargar el módulo "userdir" (si es que no está ya cargado).

```
# Comprobamos si el m'odulo "userdir" est'a cargado
```

```
root# apache2ctl -M
```

```
Loaded Modules: core_module
```

```
(static) log_config_module
```

```
(static)
```

```
:
```

```
status_module (shared)
```

```
Syntax OK
```

```
# Esto tambi'en puede hacerse mostrando el contenido del directorio:
```

```
root# ls -l /etc/apache2/mods-enabled/
```

```
total 0
```

```
lrwxrwxrwx 1 root root 40 2007-01-08 18:51 actions.load ->\n/etc/apache2/mods-available/actions.load
```

```
lrwxrwxrwx 1 root root 28 2007-01-03 16:03 alias.load ->\n../mods-available/alias.load
```

```
:
```

```
lrwxrwxrwx 1 root root 29 2007-01-03 16:03 status.load ->\n../mods-available/status.load
```

```
# Aunque aqu'i s'olo aparecer'ian los m'odulos de tipo "shared",\n# que no se instalan por defecto en el servidor
```

```
# Si no lo est'a, configuramos Apache para que lo cargue
```

```
root# a2enmod userdir
```

⁴Apache es capaz de realizar las más diversas tareas que se le pueden encargar a un servidor Web. Como hay muchos usuarios que no utilizan a la vez todas las posibilidades, el servidor se ha diseñado de forma modular. Por tanto, dependiendo de la tarea que deseamos realizar habrá que cargar el módulo correspondiente.

9.10 Apache

```
# Forzamos a Apache a leer todos los m'odulos
root# echo "ServerName localhost" >> /etc/apache2/apache2.conf
root# /etc/init.d/apache2 force-reload
```

2. Crear el directorio que almacena los objetos Web:

```
# Creamos el directorio
alumno$ mkdir public_html

# Comprobamos que tenga permisos de lectura y de acceso
alumno$ ls -la public_html/
total 8
drwxr-xr-x  2 alumno alumno 4096 2007-01-13 13:27 .
drwxr-xr-x 14 alumno alumno 4096 2007-01-13 13:27 ..

# Copiamos un fichero con un objeto Web
alumno$ cp /var/www/index.html public_html

# Comprobamos que tenemos acceso a dicho fichero
alumno$ firefox http://localhost/~alumno &
```

Ejercicio 13: Habilite Apache para que los usuarios puedan publicar objetos Web en sus directorios `public_html`. Compruebe que el servicio funciona correctamente.

9.10.5. Binding

Apache por defecto escucha en el puerto 80 a través de todos los interfaces de red de los que dispone el host. Este comportamiento puede modificarse para:

Escuchar en un puerto distinto del 80:

Esto se hace a través de la directiva `Listen`. Ejemplo:

```
# Usando un editor de ficheros ASCII, buscar en el fichero de
# configuraci'on de Apache la directiva "Listen 80" y cambiarla por
# "Listen 8080". Si dicha directiva aparece en el fichero
# "/etc/apache2/ports.conf", esto puede hacerse tambi'en con los
# siguientes comandos:
root# sed "s/80/8080/g;" /etc/apache2/ports.conf >\
/etc/apache2/ports.conf.new
root# mv /etc/apache2/ports.conf.new /etc/apache2/ports.conf
# Ojo que es posible que adem'as haya que modificar alg'un fichero extra!
# L'eanse detenidamente los comentarios en "/etc/apache2/ports.conf"
# acerca de esta posibilidad.

# Re-iniciamos Apache.
```


9.10 Apache

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

```
# Comprobamos el cambio.
```

```
alumno$ firefox http://localhost:8080 &
```

```
# Deshacemos el cambio.
```

```
root# sed "s/8080/80/g;" /etc/apache2/ports.conf >\
/etc/apache2/ports.conf.new
```

```
root# mv /etc/apache2/ports.conf.new /etc/apache2/ports.conf
```

```
# Reiniciamos Apache
```

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

Ejercicio 14: Realice las modificaciones oportunas para que Apache escuche en el puerto 8080 y compruebe que realmente es así. Finalmente deshaga dichas modificaciones y compruebe que Apache vuelve a escuchar en el puerto 80.

Escuchar en más de un puerto:

Ejemplo:

```
root# cat >> /etc/apache2/ports.conf << EOF
Listen 8080
EOF
```

```
# Reiniciamos Apache
```

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

```
# Comprobaci'ón
```

```
alumno$ firefox http://localhost:80 &
```

```
alumno$ firefox http://localhost:8080
```

```
# Desahacemos el cambio.
```

```
root# sed "s/Listen 8080//g;" /etc/apache2/ports.conf >\
/etc/apache2/ports.conf.new
```

```
root# mv /etc/apache2/ports.conf.new /etc/apache2/ports.conf
```

```
# Reiniciamos Apache
```

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