

UT 2: Administración Servidores Web



2ºDAW – Despliegue Aplicaciones Web

Apache



<http://www.apache.org>

Fundación que proporciona soporte a la comunidad con un numeroso conjunto de proyectos “open source”.

Desarrolladores y usuarios caracterizados por ser grupo colaborativo con el deseo de crear software de alta calidad.

No solamente desarrollan Apache Http Server, también “incuban” otros proyectos muy relevantes tales como:

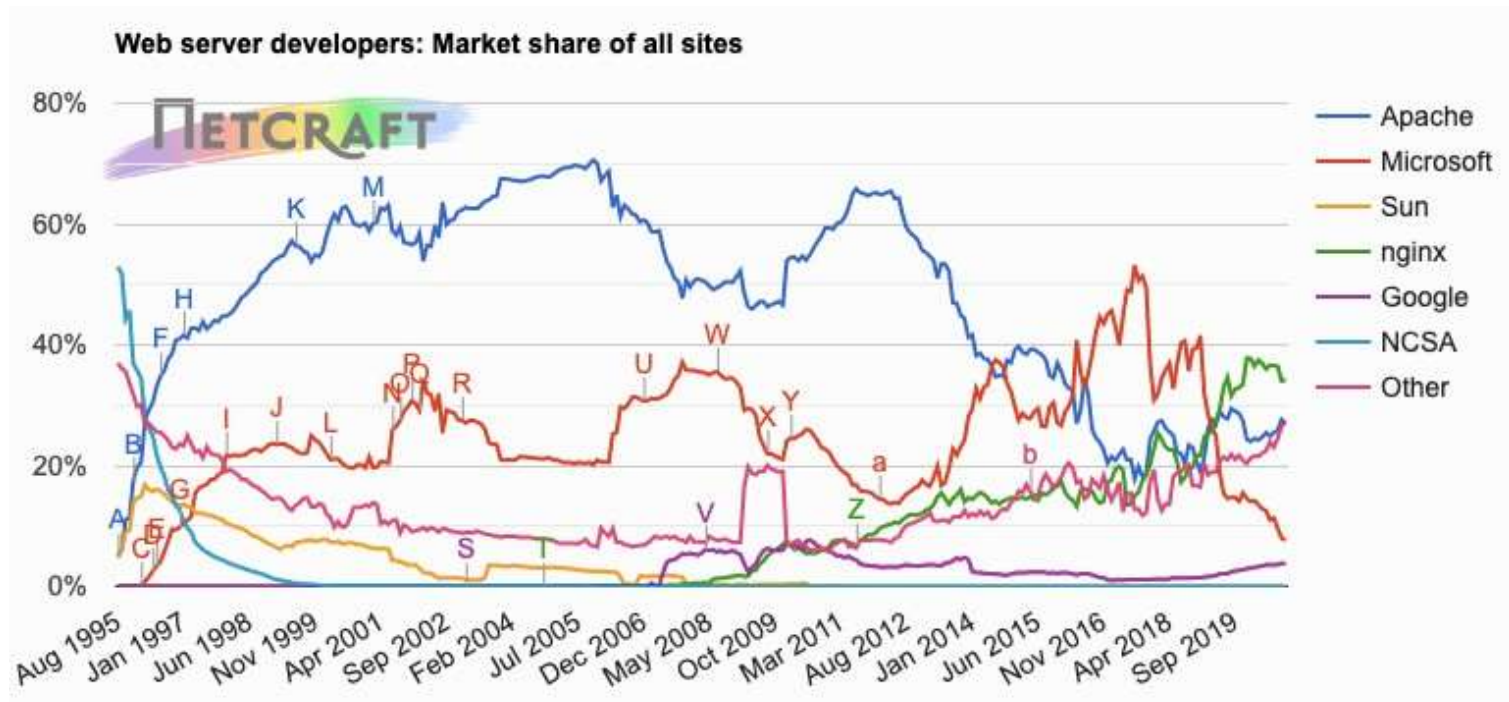
- Kafka: Software de colas.
- OpenOffice: Conocido por muchos como alternativa a Microsoft Office.
- Cordova: Permite el desarrollo de aplicaciones móviles usando HTML, CSS y JS.

<http://www.apache.org/foundation/>



Proyecto principal HTTP Server <http://httpd.apache.org>.

Apache HTTP Server Project



Developer	September 2020	Percent	October 2020	Percent	Change
Apache	50,925,159	26.46%	50,837,005	25.66%	-0.79
nginx	37,549,911	19.51%	38,606,507	19.49%	-0.02
Google	18,663,687	9.70%	19,978,491	10.08%	0.39
Microsoft	7,861,906	4.08%	7,716,026	3.89%	-0.19

Apache HTTP Server Project



Servidor Web potente, flexible y modular.

Características:

- Altamente configurable y extensible.
- Ofrece su propio API (Aplication Programming Interface).

Servidor “Open Source”.

Multiplataforma: Windows, Linux, Unix, ...

Servidor muy utilizado en Internet

Características de un servidor Web:

- Aloja y sirve páginas web cuando son solicitadas.
- Al teclear URL, se hace una solicitud al servidor web. Si se encuentra se envía, si no, transmitirá un mensaje de error.
- Puede alojar y servir páginas estáticas (que no cambian, como HTML, CSS, JS, etc).
- En el caso de Apache, puede alojar y servir páginas dinámicas que se generan mediante la ejecución de pequeños programas en otros lenguajes (PHP, CGI, Python, etc). Esto se consigue mediante la activación y uso de módulos:
 - mod_php
 - mod_cgi
 - mod_pyton
- Puede alojar varias páginas web y varios sitios web (conjunto de páginas)
- Puede alojar un dominio o varios (utilizando virtual hosts).

Apache Instalación y Configuración



Rutas de interés en la instalación de apache2 (2.4 en GNU/Linux)

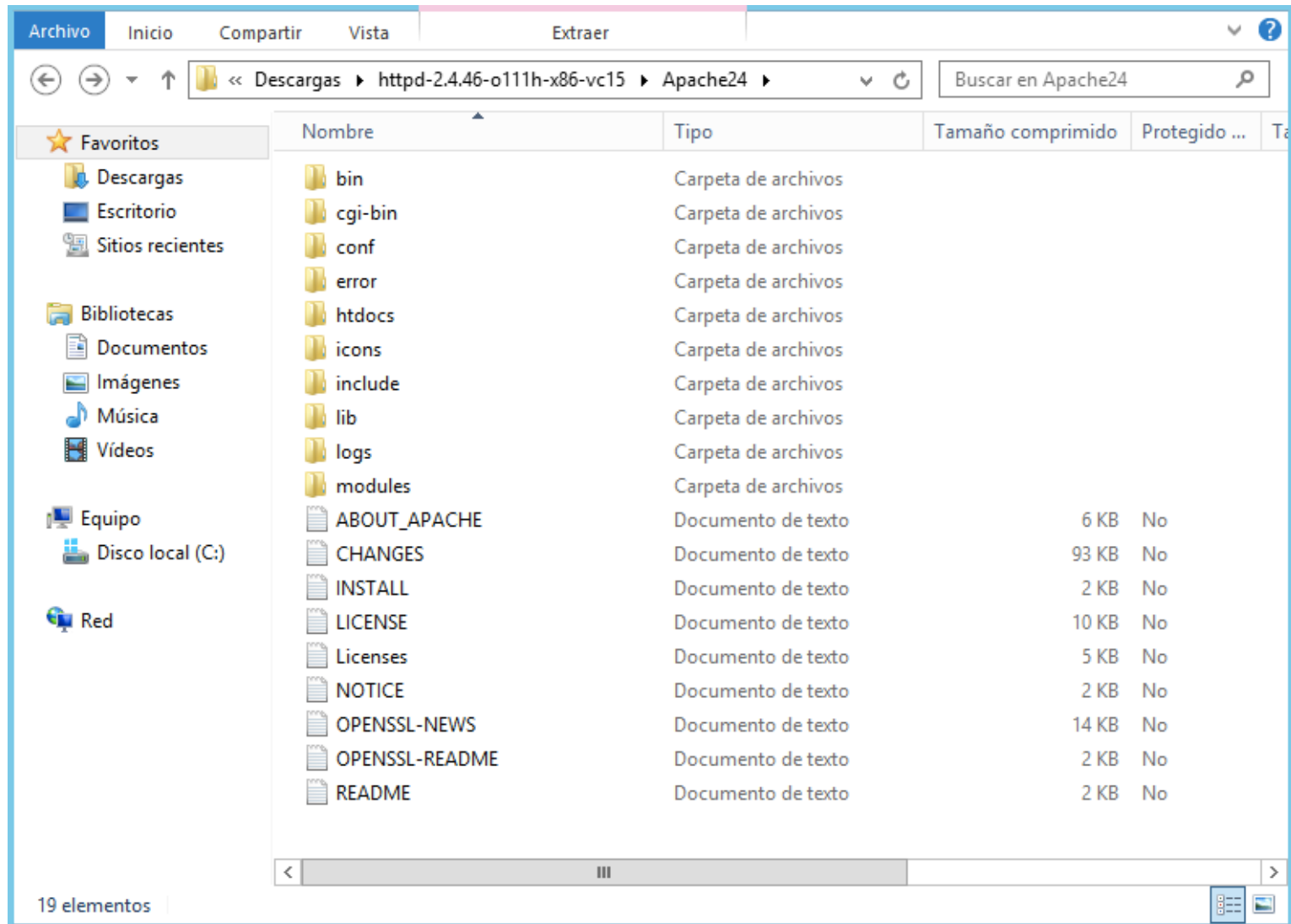
```
-rw-r--r-- 1 root root 7115 nov 26 2018 apache2.conf
drwxr-xr-x 2 root root 4096 nov 1 13:23 conf-available
drwxr-xr-x 2 root root 4096 nov 1 13:23 conf-enabled
-rw-r--r-- 1 root root 1782 nov 26 2018 envvars
-rw-r--r-- 1 root root 31063 nov 26 2018 magic
drwxr-xr-x 2 root root 12288 nov 1 13:23 mods-available
drwxr-xr-x 2 root root 4096 nov 1 13:23 mods-enabled
-rw-r--r-- 1 root root 320 nov 26 2018 ports.conf
drwxr-xr-x 2 root root 4096 nov 1 13:23 sites-available
drwxr-xr-x 2 root root 4096 nov 1 13:23 sites-enabled
```

- En otras versiones y en Windows la mayor parte de la configuración se concentra en el fichero httpd.conf (ubicado en la carpeta conf), que es el equivalente a apache2.conf
- A pesar de la distribución de la configuración en carpetas, la configuración es similar pero más modularizada.

Apache Instalación y Configuración



En Windows la instalación de apache 2.4 tiene el siguiente aspecto:



Apache Instalación y Configuración



Las diferencias entre el httpd.conf y el apache2.conf radican en cómo se construye la configuración final.

- En httpd.conf, la mayor parte de la configuración está en el propio fichero.
- En apache2.conf, se carga la configuración haciendo referencia a otros ficheros (usando Include/IncludeOptional) para componer la configuración final.

```
...  
# Include module configuration:  
IncludeOptional mods-enabled/*.load  
IncludeOptional mods-enabled/*.conf  
  
# Include list of ports to listen on  
Include ports.conf  
  
...
```

apache2.conf

```
...  
81 #LoadModule access_compat_module modules/mod_access_compat.so  
82 LoadModule actions_module modules/mod_actions.so  
83 LoadModule alias_module modules/mod_alias.so  
84 LoadModule allowmethods_module modules/mod_allowmethods.so  
85 LoadModule asis_module modules/mod_asis.so  
86 LoadModule auth_basic_module modules/mod_auth_basic.so  
87 #LoadModule auth_digest_module modules/mod_auth_digest.so  
  
...  
289 <IfModule dir_module>  
290     DirectoryIndex index.html  
291 </IfModule>  
  
...  
67 Listen 80  
  
...
```

httpd.conf

- La configuración de Apache se realiza utilizando archivos de texto, a través de directivas que permiten realizar la configuración que resulte más conveniente.
- Las directivas son las mismas para todos los sistemas operativos, pero, como hemos visto, la organización de los archivos de configuración puede variar.
- Algunas de las directivas más importantes son:
 - Listen: Indica qué puertos e IPs aceptarán peticiones.
 - ServerRoot: Directorio raíz de instalación de Apache.
 - Include: Permite incluir otros archivos de configuración.
 - DocumentRoot: Directorio desde el que se servirán archivos.
 - ErrorLog: Fichero de registro de error.
 - DirectoryIndex: fichero que se servirá por defecto para cada directorio. Por defecto, index.html
 - ErrorDocument: Acción a realizar si sucede un error.
 - Alias: Permite definir directorios virtuales. No tiene porqué estar dentro del DocumentRoot.
 - Redirect: Redirige de una página a otra.
 - LoadModule: Sirve para cargar módulos.

Con los módulos se consigue poder añadir o eliminar funcionalidades al servidor web de forma sencilla, mediante la directiva LoadModule.

Por ejemplo:

- Si necesito soporte para SSL, habilito el módulo mod_ssl.
- Necesito soporte para PHP, habilito el módulo mod_php.
- Etc.

Para habilitar o deshabilitar módulos:

- En distribuciones GNU/Linux basadas en Debian
 - apt-get para instalar o desinstalar módulos
 - Comandos a2enmod y a2dismod (enable/disable)
- En otras distribuciones
 - Se pueden hacer enlaces simbólicos de la carpeta mods-available a la carpeta mod-enabled. Ejemplo:

```
ln -s /etc/apache2/mods-available/ssl.load /etc/apache2/mods-enabled/ssl.load
ln -s /etc/apache2/mods-available/ssl.conf /etc/apache2/mods-enabled/ssl.conf
```

- En Windows
 - En el fichero httpd.conf, se descomenta la directiva LoadModule correspondiente.

Apache Host Virtuales



Alojamiento virtual de sitios web (web virtual hosting).

Simular que existen varias maquinas (hosts) con sus respectivos sitios web sobre un solo servidor web.

Por ejemplo www.daw2.iesldv.com o www.daw2.com en un mismo servidor.

Objetivos:

- ✓ Reducción del numero de maquinas físicas.
- ✓ Aprovechar mejor los recursos (uso de CPU, memoria,...) de los servidores.
- ✓ Aprovechar mejor las direcciones IP disponibles.

Apache Host Virtuales



Tipos de Virtualización en Apache:

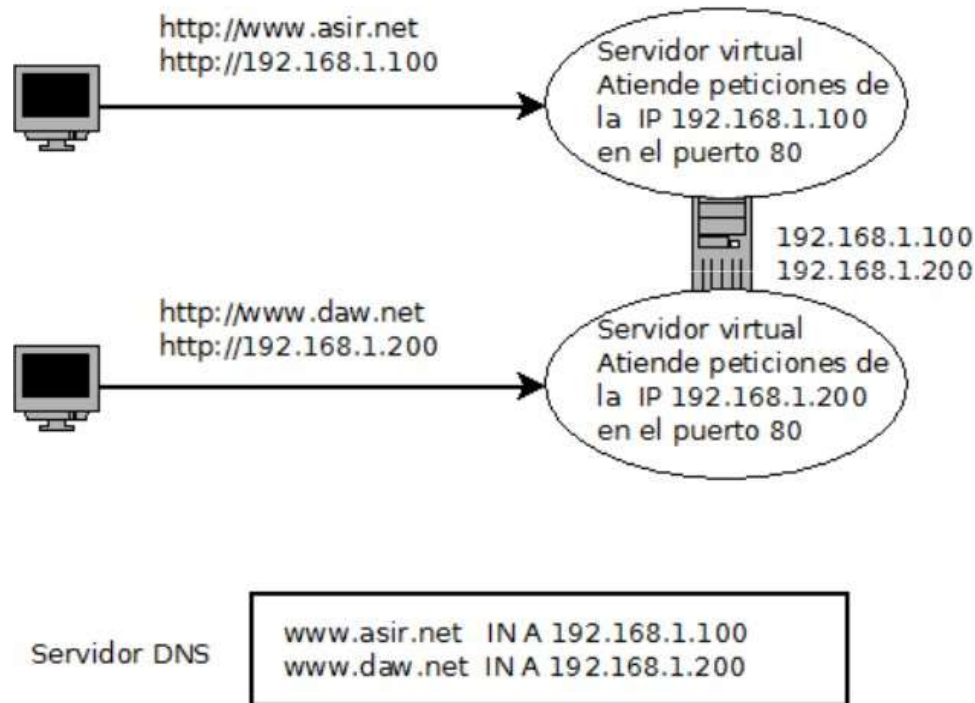
- Host Virtuales basados en IPs
- Host Virtuales basados en nombres
- Host Virtuales basados en puertos

Referencias:

<http://httpd.apache.org/docs/2.4/es/vhosts>

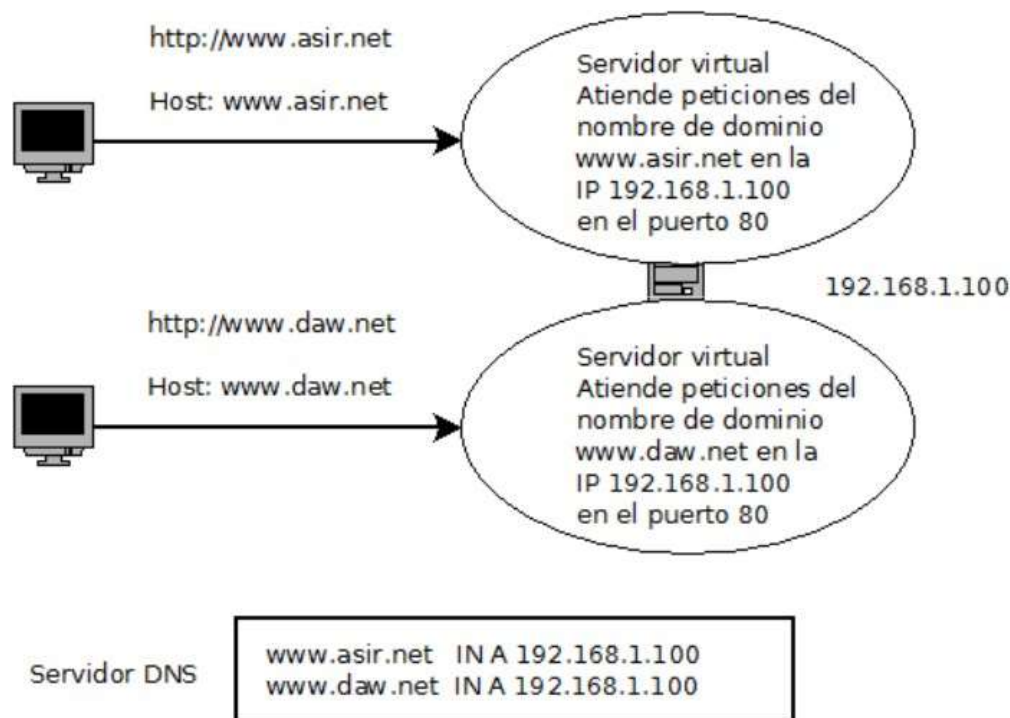
<http://httpd.apache.org/docs/2.4/es/vhosts/examples.html>

Virtualización basada en IPs



- El servidor tendrá diferentes direcciones IP por cada servidor web virtual.
- Cada servidor virtual atenderá peticiones en una dirección IP diferente.
- A efectos de los usuarios es como si existiesen varios servidores web, uno en cada dirección IP.

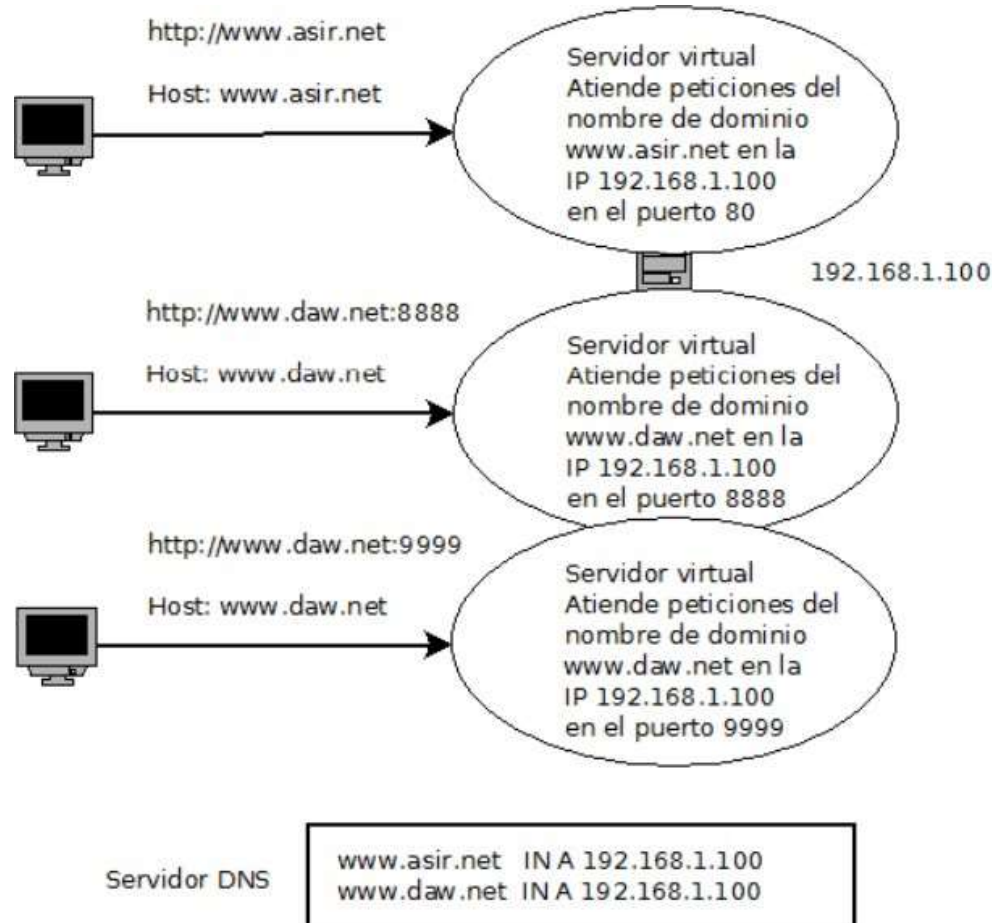
Virtualización basada en nombres



- El servidor permite alojar varios nombres de dominio sobre la misma dirección IP
- Cada servidor virtual atiende las peticiones de un nombre de dominio.
- Hay que configurar un servidor DNS que asocie los nombres de dominio con la misma dirección IP (CNAME)

- Es la forma de alojamiento virtual más utilizada. Permiten alojar múltiples dominios en una única máquina (ahorro equipos e Ips, además se facilita administración centralizada)

Virtualización basada en puertos



- Cada servidor virtual atiende peticiones en una **dirección IP** y/o **dominio:puerto** diferentes
- Consiste en combinar el alojamiento basado en IP y/o en nombres con el uso de varios puertos de escucha.

Directivas para Virtualización en Apache:

`<VirtualHost> ... </VirtualHost>`

`NameVirtualHost`

`ServerName`

`ServerAlias`

`ServerPath`

Server configuration

```
# Ensure that Apache listens on port 80
Listen 80

# Listen for virtual host requests on all IP addresses
NameVirtualHost *:80

<VirtualHost *:80>
    DocumentRoot /www/example1
    ServerName www.example.com

    # Other directives here
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot /www/example2
    ServerName www.example.org

    # Other directives here
</VirtualHost>
```


Apache Host Virtuales Ubuntu/Debian



La versión precompilada (.deb) para Debian/Ubuntu tiene habilitados servidores virtuales.

Tiene creado y habilitado un servidor virtual por defecto:

- Carpeta **/etc/apache2/sites-available/** contiene los ficheros de configuración de servidores virtuales disponibles.
- Fichero **000-default.conf** con la configuración servidor virtual por defecto y el fichero **000-defaultssl.conf** con la configuración de un servidor por defecto seguro.
- Carpeta **/etc/apache2/sites-enabled/** contiene enlaces simbólicos a los ficheros de **sites-available**. Los enlaces serán los servidores virtuales habilitados.
- Por defecto, está creado el fichero **000-default.conf** que es un enlace al fichero **000-default.conf** del directorio **sites-available**

Apache Control de acceso



Control de acceso a recursos: ficheros, directorios, URL...

- Control de acceso por host (IP/nombre_dominio)
 - Módulo mod_authz_host
(directiva Require)

```
<Directory /var/www/html/profesor>  
    Options Indexes FollowSymLinks  
    AllowOverride None  
    Require ip 127.0.0.1  
    Require ip 192.168.1.16  
</Directory>
```

- Control de acceso por variables de entorno
 - Módulo mod_authz_host.
 - Módulo mod_setenvif.

```
<If "%{HTTP_USER_AGENT} == 'BadBot'">  
    Require all denied  
</If>
```

- Control de acceso usando el módulo rewrite
 - Módulo mod_rewrite.

```
RewriteEngine On  
RewriteCond "%{TIME_HOUR}" ">=20" [OR]  
RewriteCond "%{TIME_HOUR}" "<07"  
RewriteRule "^/fridge" "-" [F]
```

Referencia

<http://httpd.apache.org/docs/2.4/es/howto/access.html>

Apache Autenticación y Autorización



Dos tipos de autenticación: Basic y Digest.

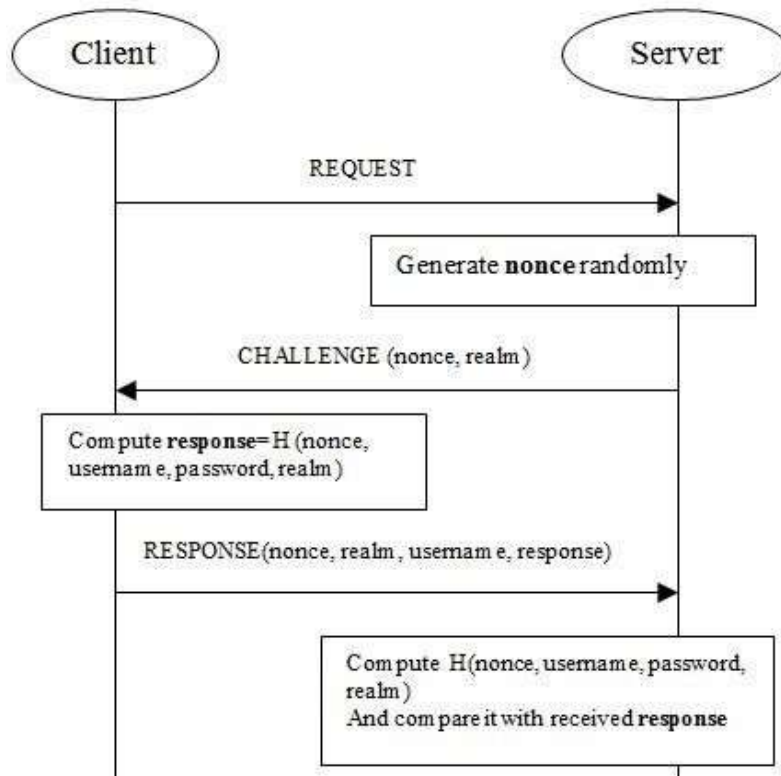
- Basic (módulo mod_auth_basic)
 - La contraseña es enviada por el cliente en texto plano.
 - Autenticación y autorización sobre fichero de texto (htpasswd).

```
> Transmission Control Protocol, Src Port: 51487, Dst Port: 80, Seq: 452, Ack: 1858, Len: 534
▼ Hypertext Transfer Protocol
  > POST /lab/webapp/basicauth HTTP/1.1\r\n
    Host:
    User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  > Content-Length: 0\r\n
    Referer: http://          appspot.com/lab/webapp/basicauth\r\n
  ▼ Authorization: Basic YWRtaW46YWFKZGQ=\r\n
    Credentials: admin:aadd
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
```

Apache Autenticación y Autorización



- Digest (módulo mod_auth_digest)



- La contraseña se envía cifrada (cifrado débil, no es seguro) por el cliente.
- Autenticación y autorización sobre fichero de texto (htdigest).

Transmission Control Protocol, Src Port: 51720, Dst Port: 80, Seq: 1, Ack: 1, Len: 627

Hypertext Transfer Protocol

> GET /lab/webapp/digest/1 HTTP/1.1\r\n

Host: .appspot.com\r\n

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

Referer: http://.b.appspot.com//lab/webapp/digest\r\n

Authorization: Digest username="admin", realm=" ", nonce="943b7337161e5baf85f7af36ccf20b85",

Connection: keep-alive\r\n

La configuración de autenticación y autorización suele encontrarse en los ficheros .htaccess

- Son ficheros que permiten la configuración personalizada de directorios.
- Cada vez que se produce una petición:
 - El servidor busca en la ruta del recurso que ha solicitado el cliente un fichero con el nombre .htaccess.
 - Aplica sobre el directorio las directivas definidas.
- En la configuración del servidor hay que permitir el uso de estos ficheros.

HTTP no es un protocolo seguro.

- Intercambio de información en texto plano (susceptible a sniffing – p.ej. se podrían ver las tramas con Wireshark).
- Basic y Digest no son seguros.
- No se garantiza que los equipos involucrados en la transferencia son (spoofing y man-in-the-middle).
- Robo o falsificación de cookies y/o parámetros (robo de identidad y suplantación de webs)

Debemos tener en cuenta:

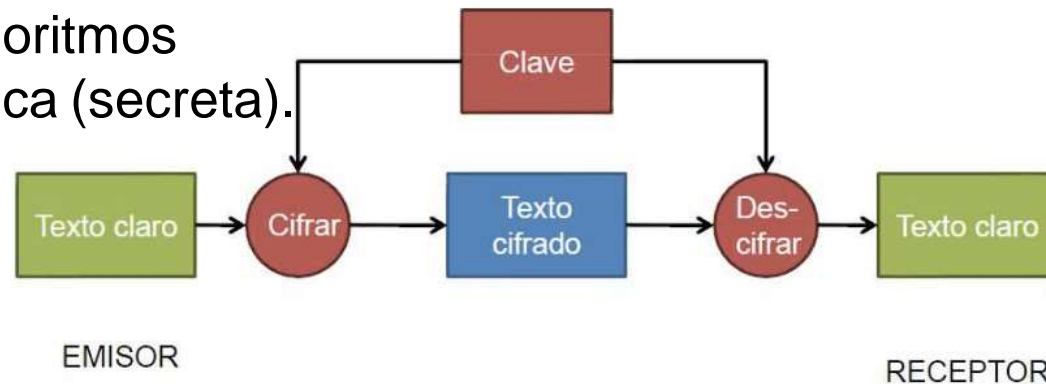
- Vulnerabilidades en clientes y servidores.
- Vulnerabilidades en las aplicaciones.
 - XSS, SQL Injection, ...

Alternativa: Cifrado de la comunicación.

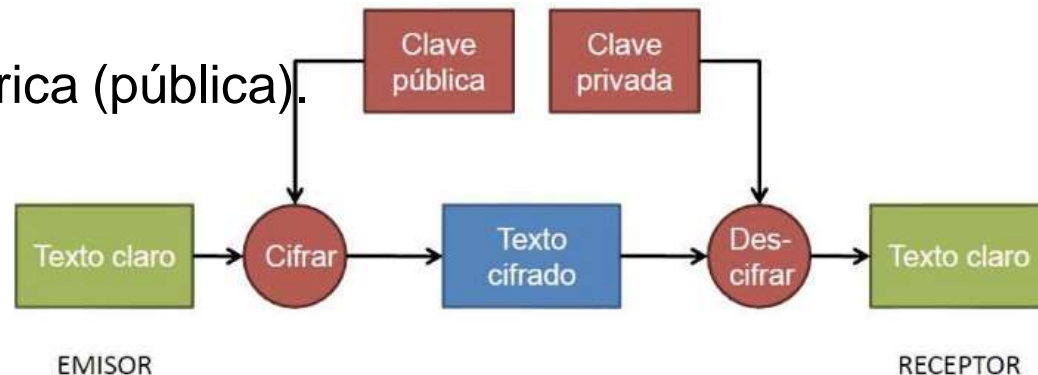
Cifrar información consiste en transformar un mensaje en claro en un mensaje ininteligible que solo puede ser descifrado por alguien autorizado. Se basa en la utilización algoritmos (públicos) y claves de cifrado.

Dos tipos de algoritmos

- Clave simétrica (secreta).



- Clave asimétrica (pública).



Una forma de cifrar es utilizar un certificado digital.

Un certificado digital es un documento/archivo que contiene:

- Información sobre una persona, entidad, empresa, organización, ... (nombre, dirección, email, ...)
- La clave pública del propietario (persona, entidad, ...).
- La firma digital de un organismo de confianza, una autoridad de certificación (CA, Certificate Authority) que garantiza que la clave pública que contiene el certificado se corresponde con el propietario del mismo.

Certificados raíz:

- Emitidos por las autoridades de certificación para si mismas con su clave pública.
- Son necesarios para verificar la autenticidad de los certificados emitidos por ellas.

Certificados autofirmados:

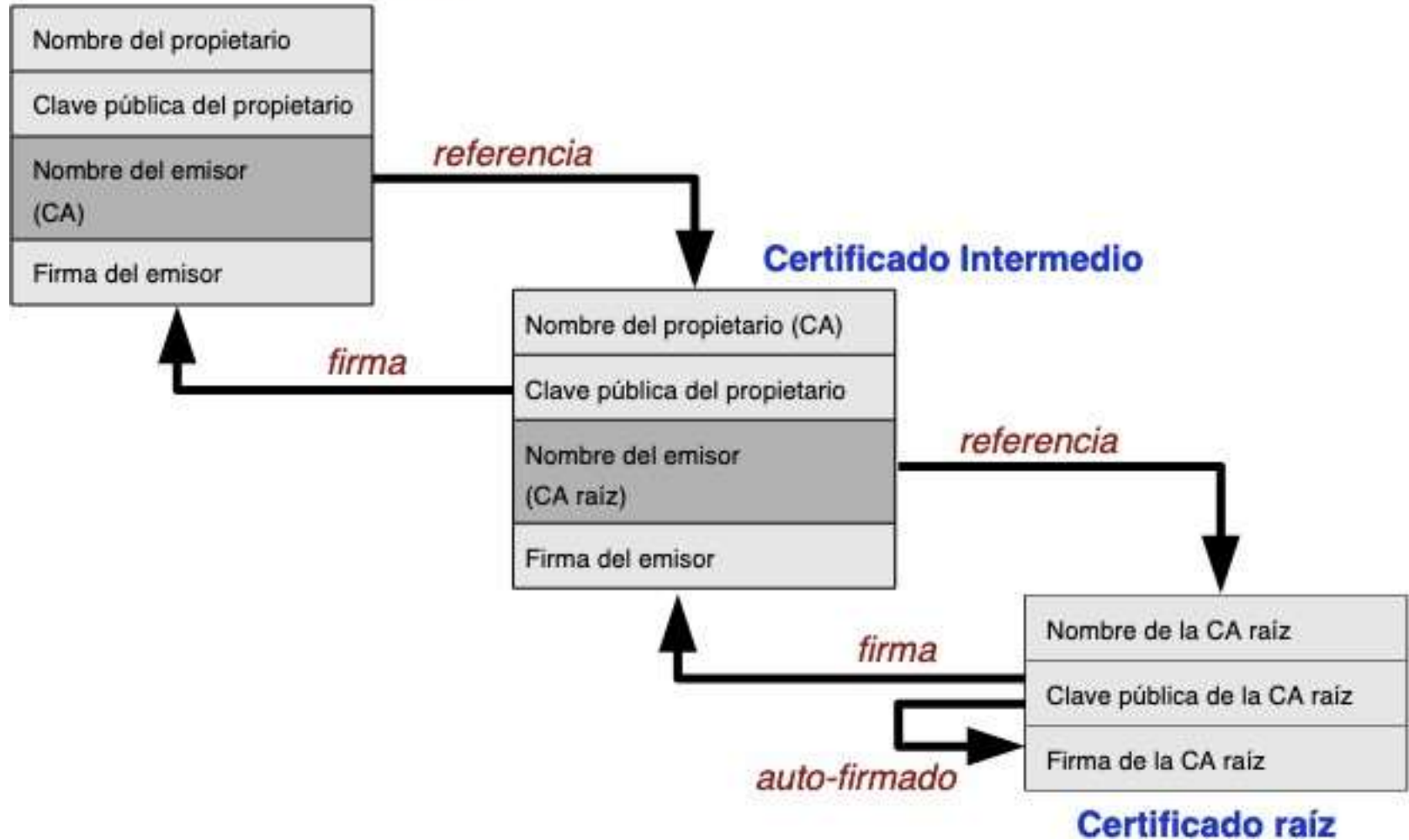
- Se realiza sin la intervención de una autoridad certificadora.
- No existe ningún mecanismo automático que garantice la autenticidad del certificado.

Certificados intermedios y finales:

- Deben estar firmados por una entidad certificadora

Apache HTTPS y SSL

Certificado de entidad final



HTTPS (Hyper Text Transfer Protocol Secure).

- Protocolo que utiliza SSL/TLS para encapsular mensajes HTTP.
- Clientes.
 - Utilizan https:// en las URIs (o URLs).
- Servidores.
 - Por defecto escuchan peticiones HTTPS en el puerto 443/TCP.

SSL/TLS Se basa en el uso

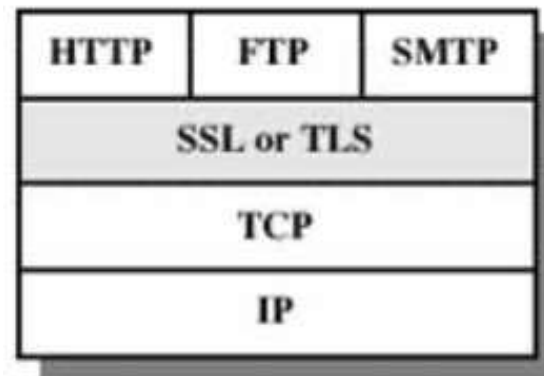
- Algoritmos criptográficos.
 - Clave privada (simétrica) (3DES, AES, RC, ...).
 - Clave publica (asimétrica) (RSA, DSA, ...).
- Certificados digitales (X.509)
- Infraestructura de clave pública (PKI)
- Autoridades de certificación.

Ofrece

- Confidencialidad.
- Autenticación.
- Integridad.
- No repudio.

Se ejecuta en una capa entre los protocolos de aplicación (HTTP, SMTP o FTP) y el protocolo de transporte TCP.

- HTTPS, FTPS, SMTPS, POPS, IMAPS, ... se basan en SSL/TLS.



- También es posible implementarlo sobre UDP.

Los ficheros de acceso y error de Apache son:

- access.log
- error.log

Y se encuentran en:

<ruta de instalación de apache>[/log], dependiendo del sistema operativo.

Existen múltiples directivas para personalizar logs.

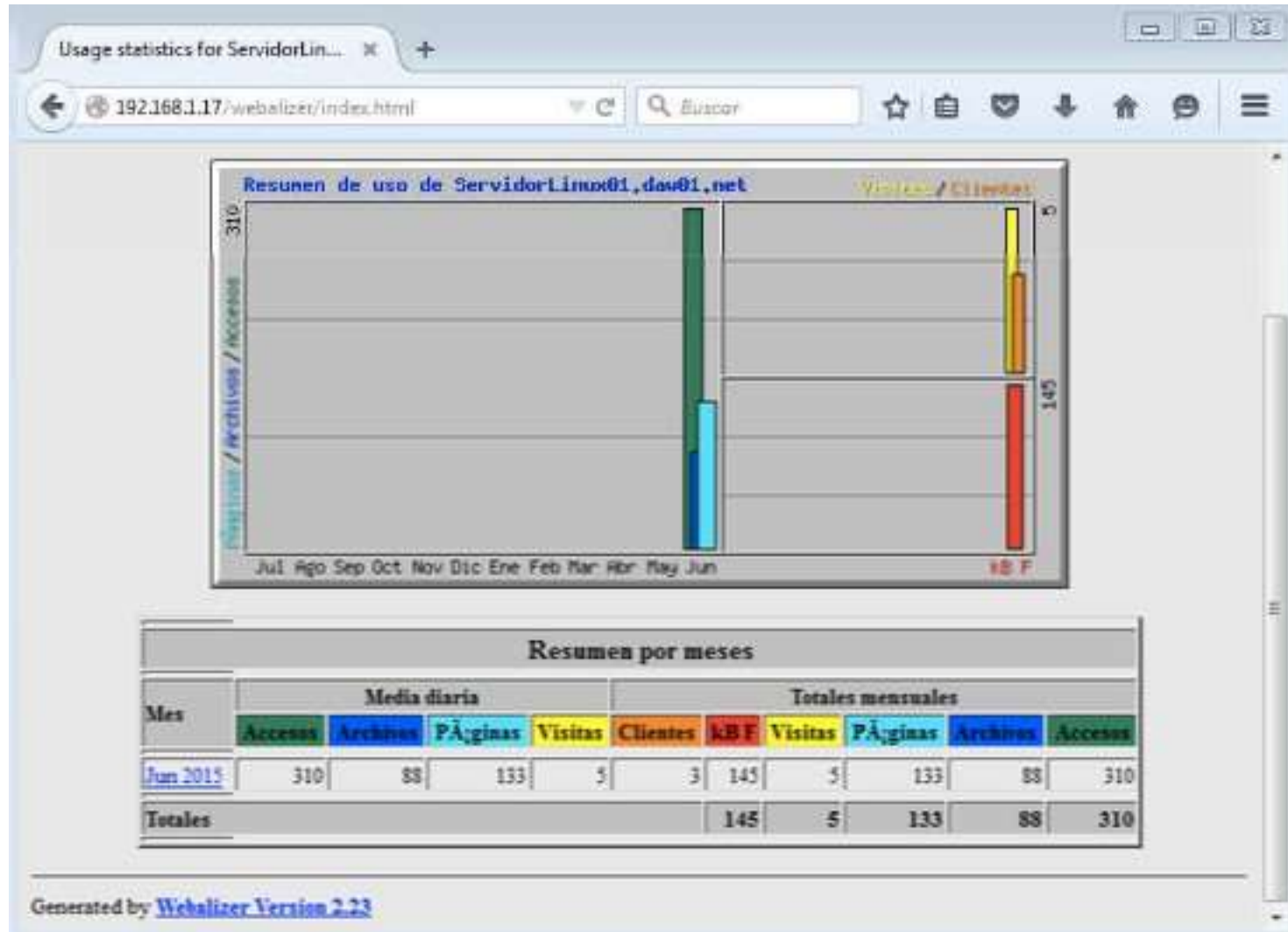
- ErrorLog
- LogLevel
- CustomLog
- LogFormat
- ...

Apache tiene que tener permisos para escribir en el directorio donde se guardan los ficheros de logs.

Para analizar los logs y realizar estadísticas se pueden utilizar herramientas de monitorización. Una de ellas es Webalizer, que suele venir de serie con XAMPP.

- Es una herramienta de análisis de logs
- Genera páginas con analíticas
 - Estadísticas de acceso
 - Estadísticas de uso
- Las estadísticas generadas incluyen:
 - Hits
 - Visitas
 - Referrers (página previa)
 - País de los usuarios
 - Cantidad de datos descargados
- Las estadísticas pueden consultarse por distintos periodos de tiempo (hora, día, mes, año).

Apache Logs y Webalizer



Bibliografía

- ✓ Servicios de Red e Internet. Álvaro García Sánchez, Luis Enamorado Sarmiento, Javier Sanz Rodríguez. Editorial Garceta.