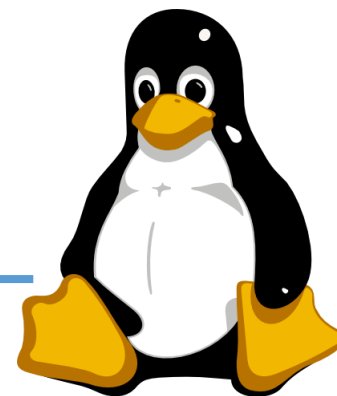


Administración de Servicios de Red

El servicio web

Ing. Denis L. Espinoza Hernández, M.Sc.
denisjev@ct.unanleon.edu.ni

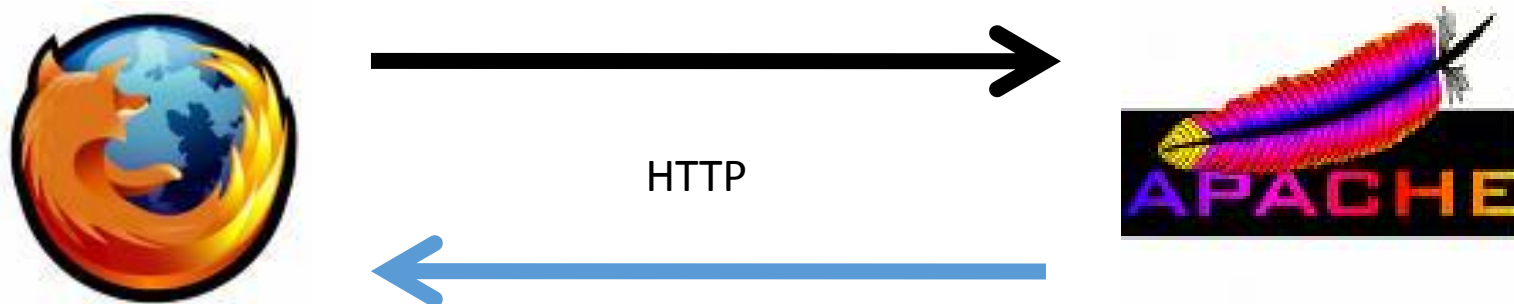




¿Qué es un servidor Web?



Un **servidor web** es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.



El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.



¿Qué es un servidor Web?



- ❑ La WEB impuso Internet sobre otras redes de datos.
- ❑ Historia:
 - ❖ 1989: inicio del servicio WEB, sólo texto.
 - ❖ 1991: primera demostración pública.
 - ❖ 1993: Mosaic, primer Navegador gráfico.
 - ❖ Posteriormente el “boom” de Internet.
- ❑ Navegadores multimedia: Netscape Navigator, MS Internet Explorer, Mozilla, etc.





Introducción al servicio WEB



WWW: World Wide Web

- ❑ Aplicación de red distribuida basada en el modelo Cliente-Servidor.
- ❑ Servicio para acceso a información, a documentos vinculados y distribuidos en muchos Host en Internet. Red de documentos vinculados.
- ❑ Posibilita, de manera sencilla:
 - Recibir información.
 - Publicar información.
 - “Navegar por un océano” de información, de sitios WEB a través de hiperenlaces y la interacción de diferentes páginas WEB.
- ❑ Funciona bajo demanda
 - Los usuarios reciben y publican “lo que quieren y cuando quieren”.

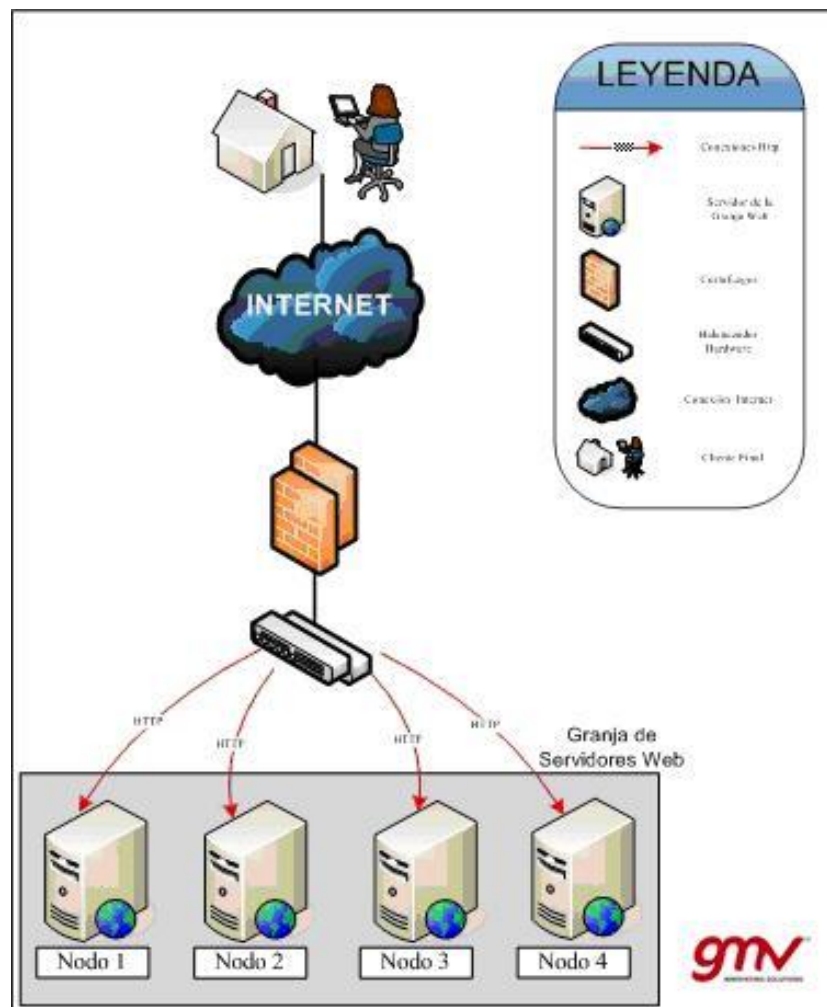




Elementos de la Arquitectura Web



- ❑ Páginas: HTML, hipertexto, hiperenlaces.
- ❑ Objetos: Pueden residir o no en el mismo Host.
- ❑ Direccionado: URL: Uniform Resource Locator.
- ❑ Entidades funcionales: Navegador, Servidor WEB.
- ❑ Protocolo HTTP.





1. Páginas WEB

- ❑ Contienen los denominados objetos WEB

- ❑ Generalmente formada por:

Fichero HTML base.

Diferentes objetos referenciados, en diferentes formatos posibles:

- ❖ Texto ordinario, hipertexto, imágenes, vídeos, audio.
- ❖ Vinculados o no a otras páginas.

2. Objetos WEB

- ❑ Archivos: HTML, JPEG, etc.

- ❑ Cada uno direccionable a través de un único URL.





3. Direccionado WEB

Se utiliza un esquema de direccionado para localizar recursos en la red en base a lo que se denomina URI (Universal Resource Identifier), según:

- ❑ Su ubicación, a través de un URL (Uniform Resource Locator).
- ❑ Sus nombres, a través de un URN (Universal Resource Name), más adecuado para servidores y contenidos replicados.

URL: Uniform Resource Locator

Elemento para el direccionamiento de recursos en la red. Apunta a un Host determinado, y a través de éste a un objeto en particular. P.e, para direccionamiento WEB.

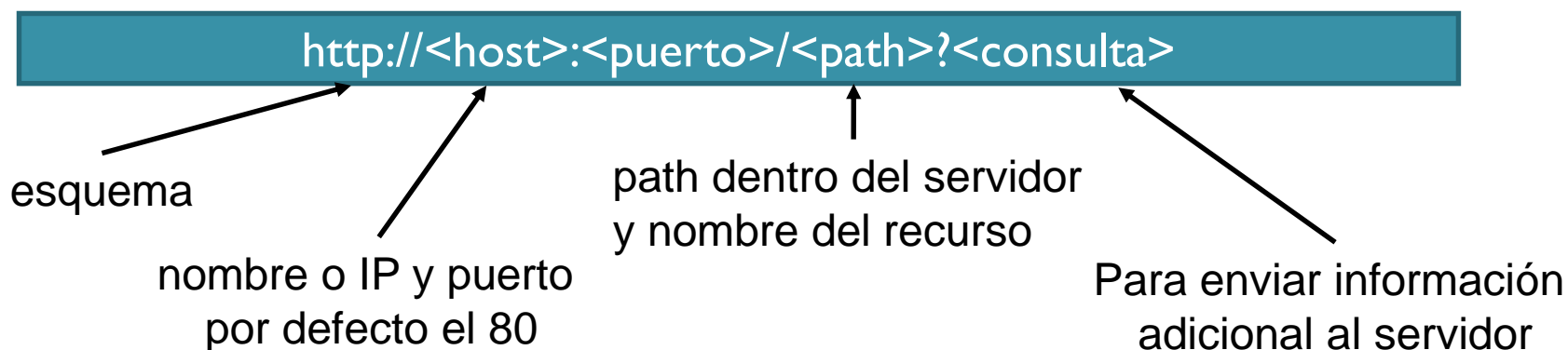




Tipos de Recursos:

- ❑ Estáticos: archivos
- ❑ Dinámicos: el resultado de la ejecución de un programa, el resultado de una consulta a una base de datos, la traducción automática de un documento, etc.

Componentes de una ULR en HTTP:





Elementos de la Arquitectura Web



Se han definido diferentes tipos de URL's según el protocolo de acceso. Posibilitan utilizar el Navegador como programa de acceso a diferentes aplicaciones de Internet:

- ❑ Acceso integrado a Internet a través del Navegador WEB.
- ❑ Se puede prescindir de programas de interfaz especializados.

Ejemplos:

Servicio	Ejemplo
FTP	ftp://ftp.cs.vv.nl/pub/readme
Telnet	telnet://www.w3.org:80
Correo	mailto:cime@gmail.com
Archivos en la máquina	file://hipertexto/html/mipagina.html





4. Navegador

Agente de Usuario

- ❑ Software que sirve de interfaz entre usuario y aplicación WEB.
- ❑ Posibilita “atrapar” y visualizar páginas WEB, y navegar por sus contenidos.
- ❑ Presentan diferentes características de configuración.

Cliente HTTP

- ❑ Software de comunicación que implementa la parte cliente del protocolo HTTP.





5. Servidor WEB

Comprende dos entidades funcionales:

Base de datos:

- ☐ Alberga los contenidos WEB (objetos).
- ☐ Cada objeto direccionable con el correspondiente URL.

Servidor HTTP:

- ☐ Software de comunicación que implementa la parte servidor del protocolo HTTP.
- ☐ Ejemplos más conocidos: Apache, quizás el más utilizado y MS Information Server.





¿Cómo es una petición Web?



Las peticiones web son realizadas por los navegadores con la siguiente sintaxis:

```
GET /index.html HTTP/1.1
```

```
Host: www.google.com.ni
```

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; es-ES; rv:1.9.0.7)
```

```
Gecko/2009021910 Firefox/3.0.7
```

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

```
Accept-Language: es-es,es;q=0.8,en-us;q=0.5,en;q=0.3
```

```
Accept-Encoding: gzip,deflate
```

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

```
Keep-Alive: 300
```

La estructura de la petición es *atributo: valor*. Para cada uno de los elementos.



¿Cómo es una respuesta web?



Si el servidor posee la página, responde con una primera línea con la sentencia 200 ok más otras cabeceras y finalmente después de dos líneas en blanco, el código html.

HTTP/1.1 200 OK

Date: Mon, 23 Mar 2009 05:23:21 GMT

Server: Stronghold/2.2 Apache/1.2.5 C2NetUS/2005

Connection: close

Transfer-Encoding: chunked

Content-Type: text/html

<html>

 <head>

 <title>name.space nslookup results</title>

</html>



Cache Web



Procedimiento para mejorar las prestaciones del servicio WEB, posibilitando:

- ☐ Reducir el tiempo de obtención de un objeto, en particular si su tamaño es grande y se dispone de poco ancho de banda de acceso.
- ☐ Reducir el tráfico WEB y la carga de trabajo de los servidores WEB.

Se basa en el almacenaje en caché de copias de objetos obtenidos:

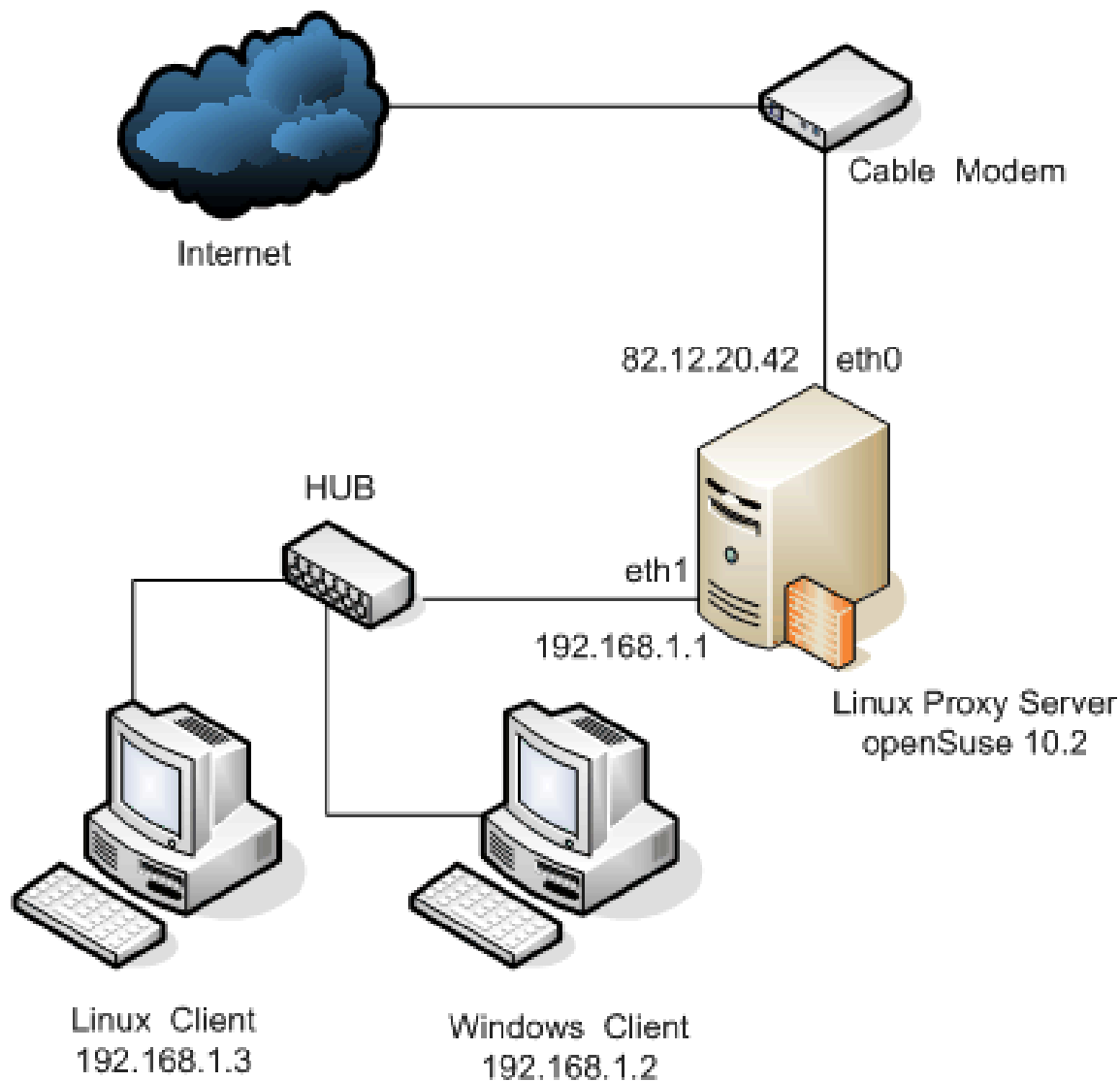
- ☐ En el cliente WEB, y lo gestiona el Navegador
- ☐ En un servidor de red actuando de “intermediario”, un Proxy WEB

El problema es como saber si el objeto que se tiene en caché del no ha sido actualizado en el servidor WEB.





Servidor Proxy Web





GET Condicional



Garantiza la actualidad de los objetos recibidos en el Navegador mediante caché.

Al solicitar la página la primera vez se solicita y se guarda la página en la cache, almacenando el valor Last-Modified.

```
HTTP/1.1 200 OK
Date: Sun, 19 Jan 2003 18:55:49 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Wed, 18 Dec 2002 12:11:06 GMT
ETag: "27ae90-44-9221aa80"
Content-Length: 68
Connection: close
Content-Type: text/html
<html> <!-- codigo de la página -->
```





GET Condicional



Cuando la página es solicitada nuevamente: El GET incluye la cabecera: If-Modified-Since.

- ❑ Si la respuesta ha sido modificada se transmite la entidad con código de estado 200.
- ❑ Si la respuesta no ha sido modificada no se transmite la entidad y se envía un código de estado 304.

```
GET //css/contenidos.css HTTP/1.1
```

```
....
```

```
If-Modified-Since: Fri, 23 Jun 2006 08:45:05 GMT
```

```
HTTP/1.1 304 Not Modified
```

```
Date: Fri, 25 Aug 2006 13:49:41 GMT
```

```
Server: Apache/2.0.54 (Ubuntu) PHP/5.0.5-2ubuntu1.3
```

```
...
```

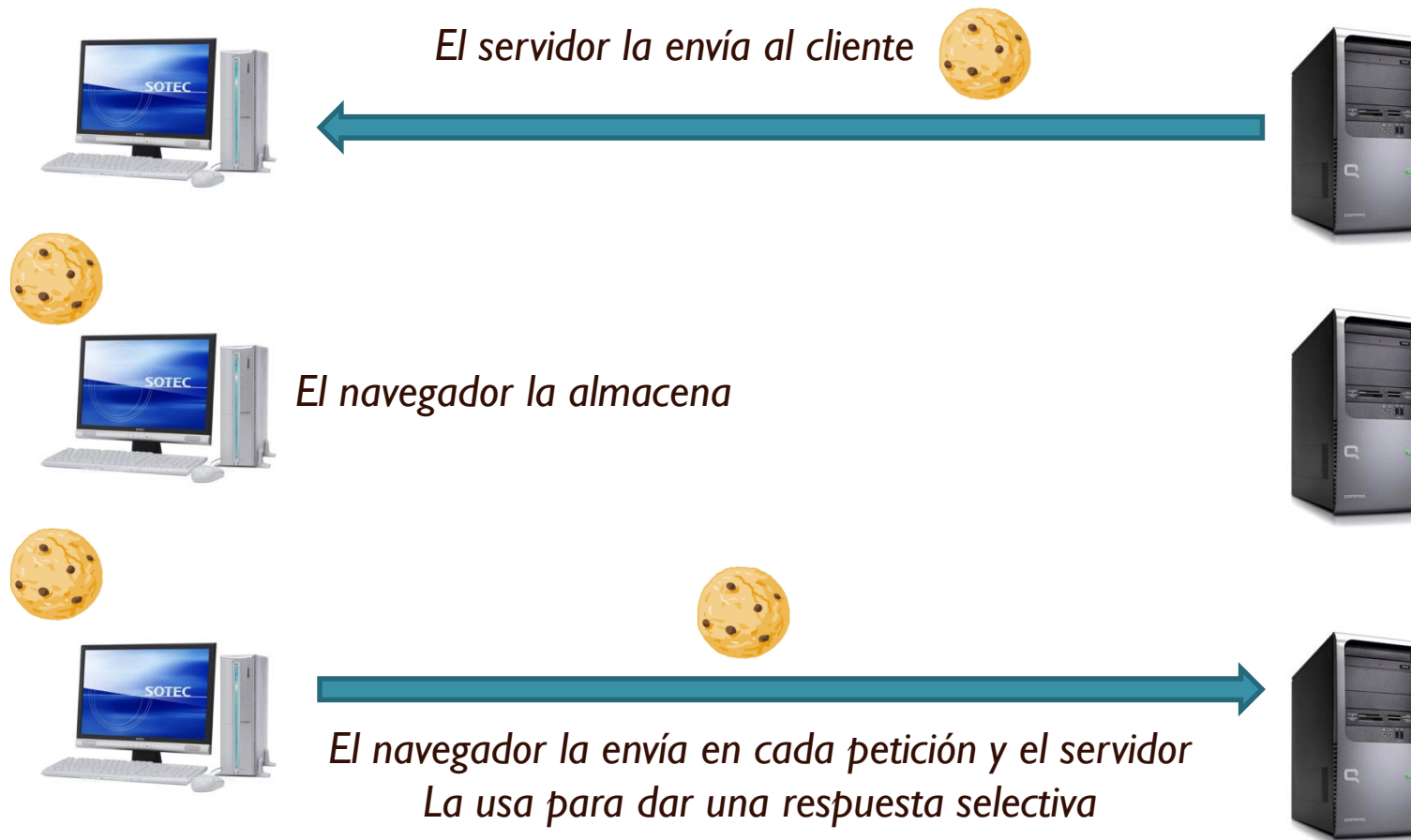




Las Cookies



Son pequeños trozos de información que:





Características de las Cookies



Las principales características de las Cookies son:

- ☐ Su principal utilidad es la de solventar la falta de estado en las comunicaciones web.
- ☐ Identificarnos
- ☐ Definir carritos de compra
- ☐ Pueden ser permanentes o tener un tiempo de vida máximo
- ☐ Tienen restricciones de seguridad: Solo el servidor que las generó puede acceder a ellas.





Ejemplo de Cookies



El Servidor crea la Cookies:

```
HTTP/1.1 200 OK
Date: Fri, 25 Aug 2006 08:10:54 GMT
Server: Apache
Set-Cookie: JSESSIONID=82818520216567447A4FFA18D02F8A9D;Path=/
```

El Cliente envía la Cookies en cada petición:

```
GET /nada.jsp HTTP/1.1\r\n
Host: www.lne.es\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Cookie:
__utma=184110843.1800093202.1155130588.1156405623.1156489485.10;
JSESSIONID=82818520216567
```





Qué es Apache



Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras. Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente.





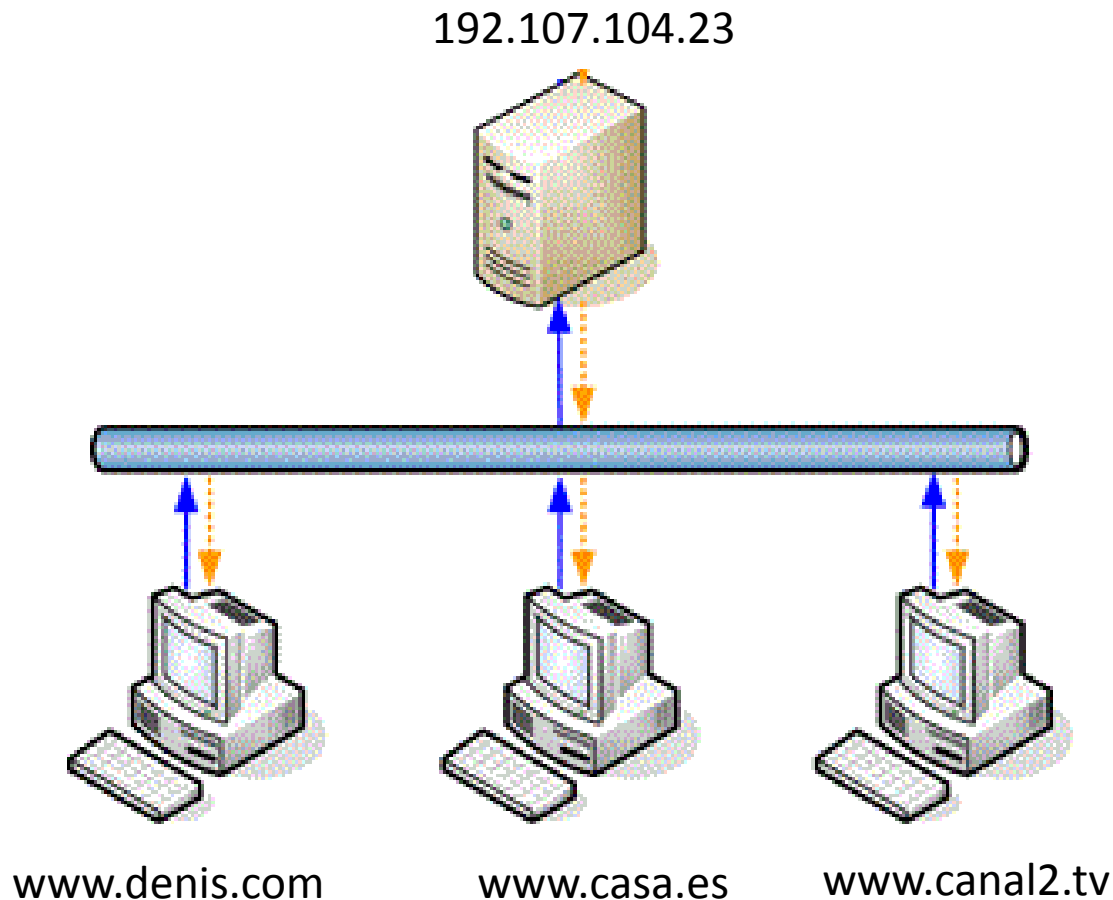
- ❖ **apache2.conf**: Es el fichero de configuración general.
- ❖ **httpd.conf**: Anterior fichero de configuración hoy esta vacío.
- ❖ **envvars**: Especifica las variables de entorno que serán exportadas. Contiene el usuario y el grupo con el que se ejecutará apache y la ruta del fichero del pid.
- ❖ **ports.conf**: Especifica el puerto en el que escucha el servidor web. Además se especifica que si el módulo ssl está activo, debe de ocuparse el puerto 443.
- ❖ **conf.d**: En este directorio se agregan algunas configuraciones extras.
- ❖ **mods-available**: Contiene todos los módulos que se instalan por defecto.
- ❖ **mods-enabled**: Contiene los módulos que están activos en este momento. Estos son un enlace blando a los modulo que se encuentran en mods-available.
- ❖ **sites-available**: Contiene los ficheros de configuración los sitios que se pueden tener en el servidor.
- ❖ **sites-enabled**: Contiene los ficheros de configuración de los sitios habilitados en el servidor. Son enlaces blandos a los ficheros de sites-available.



Servidores Virtuales en Apache



Apache soporta servir diversos sitios web con un sólo servidor. Para ello proporciona facilidades de creación de dominios virtuales en función de diversas direcciones IP o diversos nombres por IP.





Directivas para los servidores virtuales



- ❖ **ServerAdmin:** Dirección de correo electrónico del administrador.
- ❖ **ServerName:** Nombre que el servidor utiliza para identificarse.
- ❖ **DocumentRoot:** El directorio raíz desde el que se servirán los documentos. Por defecto es el directorio.
- ❖ **DirectoryIndex:** Especifica el fichero que se servirá por defecto.
- ❖ **AccessFileName:** Especifica el nombre de fichero de configuración en caso de que éste sea diferente de `.htaccess`. Para que esta configuración funcione, la directiva `AllowOverride` tiene que tener un valor adecuado.
- ❖ **Alias:** las directivas `Alias` permiten definir accesos a directorios que se encuentran fuera del `DocumentRoot`.

La sintaxis es la siguiente: `Alias url directorio`.

Por ejemplo:

`Alias /docs /home/documentos`



Directivas de control de acceso



La mayor parte de secciones de localización (Directory, Location, etc.) incluyen en su configuración una serie de directivas de control de acceso:

- ❖ Allow: Nos permite especificar quién está autorizado a acceder al recurso. Podemos especificar direcciones IP, nombres de máquina, fragmentos del nombre o dirección. La palabra clave all para indicar todos los clientes.
- ❖ Deny: Nos permite especificar a quién no dejamos acceder al recurso. Disponemos de las mismas opciones que con Allow.
- ❖ Order: Nos permite afinar el funcionamiento de las directivas Allow y Deny. Disponemos de dos opciones:
 - ☐ Allow,Deny. El acceso está denegado por defecto y sólo podrán entrar los clientes que cumplan las especificaciones de Allow y no cumplan las de Deny.
 - ☐ Deny,Allow. El acceso está permitido por defecto y sólo podrán entrar los clientes que no cumplan las especificaciones de Deny o cumplan las de Allow.



Arranque y configuración



```
NameVirtualHost *:80          #Fichero 000-default
<VirtualHost *:80>
    ServerAdmin webmaster@uoc.edu
    DocumentRoot /var/www
    ErrorLog /usr/logs/uoc_error_log
    TransferLog /usr/logs/uoc_access_log
</VirtualHost>
```

```
<VirtualHost *:80>          #Fichero denis.site
    ServerAdmin webmaster@localhost
    ServerName www.denis.com
    DocumentRoot /home/denis/sitio
</VirtualHost>.
```

Para probarlo en la misma máquina, debemos de añadir los nombres de nuestro sitio al fichero /etc/hosts para que se pueda resolver la petición:

```
127.0.0.1 localhost
127.0.0.1 www.denis.com
```



Utilización de los ficheros .htaccess



Los ficheros .htaccess, almacenan un conjunto de pares de valores (usuario - clave) que permiten agregar ciertas restricciones de acceso a un directorio

Pasos para su utilización:

```
<Directory "/var/www/privado">  
    Options Includes  
    AllowOverride All  
    Order allow,deny  
    Allow from all  
</Directory>
```

Crear el directorio /var/www/privado/ realizando lo siguiente:
mkdir /var/www/privado

Genere el fichero /var/www/privado/.htaccess realizando lo siguiente:
touch /var/www/privado/.htaccess