



Servidores Web

Roberto Gómez Cárdenas
rogomez@itesm.mx
<http://homepage.cem.itesm.mx/rogomez>

Lámina 1 Roberto Gómez C.



Definición

- Un servidor web es un servidor conectado permanentemente a internet, que se encarga de atender las solicitudes de páginas web.
- Los Servidores Web y los navegadores se comunican mediante el Protocolo de transferencia de hipertexto (HTTP).

Lámina 2 Roberto Gómez C.



Aplicación multi-niveles

- Nivel información
 - Mantiene datos para la aplicación.
 - Almacena datos en una base datos relacional.
- Nivel intermedio
 - Implementa lógica del negocio y presentación.
 - Controla interacciones entre aplicaciones cliente y datos de la aplicación.
- Nivel cliente
 - Interfaz del cliente
 - Usuarios interactúan directamente con la aplicación a través de esta capa.

Lámina 3

Roberto Gómez C.



Client scripting vs server scripting

- Client
 - Valida entradas del usuario que serán pasadas al servidor.
 - Acceso al browser.
 - Da más fortaleza a las páginas Web con DHTML, Controles ActiveX y applets
- Server
 - Ejecutado en el servidor.
 - Genera respuestas para clientes.
 - Amplio rango de capacidades de programación.
 - Acceso al software del lado del servidor que extiende funcionalidad del servidor.

Lámina 4

Roberto Gómez C.



Páginas estáticas

- Almacenadas en el servidor en un archivo .html o html.
- No pueden ser personalizadas.



Ciente Web

1. El cliente solicita una página

→

3. El servidor Web envía el archivo al cliente

←



Servidor Web

4. El navegador recibe el archivo HTML y lo muestra.

2. El servidor Web localiza el archivo HTML

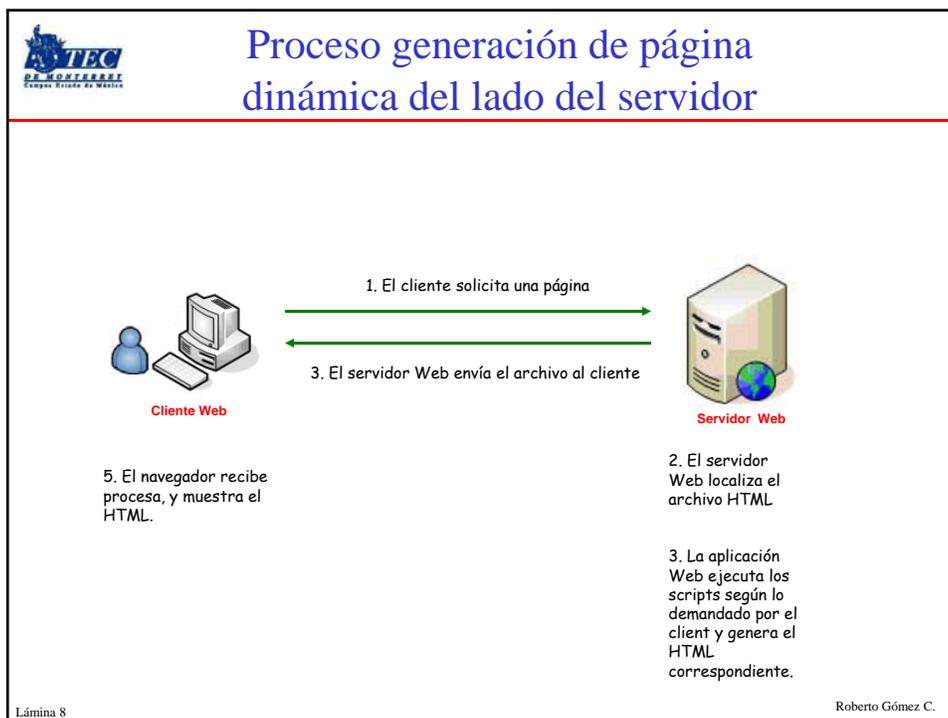
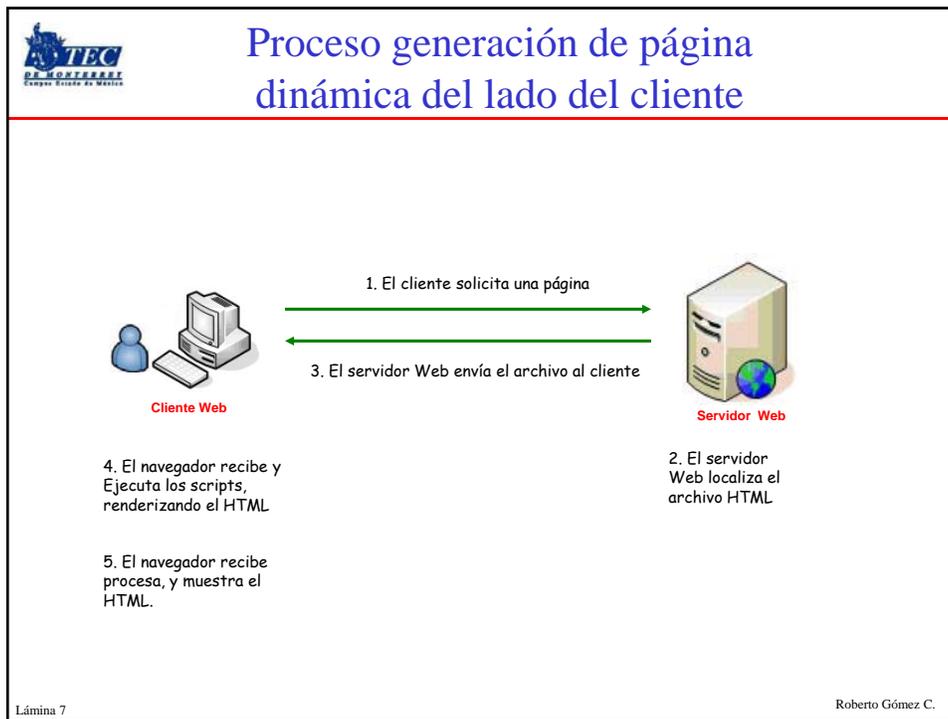
Lámina 5
Roberto Gómez C.



Páginas dinámicas

- Permiten un cierto grado de interacción.
- Esta interacción puede producirse en uno o dos de los siguientes niveles
 - Páginas dinámicas del lado del cliente
 - Son aquellas en las que la propia página contiene código para implementar dicha interactividad.
 - Se dice que dicha interactividad se realiza del lado del cliente.
 - Se implementa con lenguajes de Script
 - Páginas dinámicas del lado del servidor
 - Son aquellas generadas por una aplicación web, tal que la información contenida en ellas puede haber sido personalizada por el usuario.
 - La interactividad se realiza del lado del servidor.
 - Se implementa con diversas tecnologías.

Lámina 6
Roberto Gómez C.





Tecnologías para páginas Dinámicas del lado del cliente

- Lenguajes de scripts
 - VBScript
 - Versión de Visual Basic.
 - Competidor de Java Script.
 - Válido sólo para navegadores de Microsoft.
 - JavaScript
 - Derivado de LiveScript de Netscape
 - Basado en la familia de los lenguajes C.
 - Tiene muchos elementos de Java, pero no es Java

Lámina 9 Roberto Gómez C.



Tecnologías para páginas Dinámicas del lado del cliente

- Aplicaciones para ejecución local
 - Java Applets
 - Aplicación gráfica Java que se queda embebida en una página Web.
 - Necesita una máquina virtual Java.
 - Al ser Java es multiplataforma.
 - Active X Controls
 - Aplicación realizada en VB o en C++ basada en Tecnologías Microsoft que se embebe en HTML.
 - Sólo válido para Internet Explorer, por lo que está en desuso.
 - Animaciones Flash
 - Objetos realizados en Macromedia Flash y embebidos en la página Web.

Lámina 10 Roberto Gómez C.



Tecnologías para páginas Dinámicas del lado del servidor

- Ejecutan programas o aplicaciones en el servidor que generan dinámicamente código HTML.
- Permiten que el navegador sea un cliente neutro.
- Es posible la ejecución distribuida, accediendo a distintos recursos distribuidos como bases de datos.
- Dos tecnologías
 - Independiente de la arquitectura de la página.
 - Dependiente de la arquitectura de la página.

Lámina 11

Roberto Gómez C.



Independiente de la arquitectura

- CGI: Common Gateway Interfaces
 - Tecnología usada durante mucho tiempo en los servidores que adolecían de problemas de rendimientos (lanzaba una instancia de la aplicación por cada petición del cliente independientemente del servidor web).
 - Aplicaciones escritas en C, C++ o Perl para un tratamiento adecuado de cadenas de caracteres.
 - Tecnología muy difícil de aprender y de utilizar, aunque muy difundida y utilizada.

Lámina 12

Roberto Gómez C.



Dependiente de la arquitectura

- **Active Server Pages**
 - Construida usando VBScript o JavaScript.
 - Acceden a los mismos servicios que una aplicación Windows de escritorio, incluyendo ADO, SMTP y COM.
 - Se trata de scripts interpretados cada vez que eran solicitados, por lo que son lentas.
 - Servidores Web: Internet Information Server (IIS) y otros servidores con addons.
- **Java Server Pages (JSP)**
 - Son como las páginas ASP pero implementadas en Java.
 - Destaca el concepto de Servlet
 - Es la plataforma más difundida y actualizada, aunque adolece de una serie de problemas.
 - Pueden usar EJBs (propietario) y Servicios Web.
 - Servidor Web: Tomcat, Apache y derivados e IIS

Lámina 13

Roberto Gómez C.



Dependiente de la arquitectura

- **PHP**
 - Similar a ASP, pero usando C y Perl
 - Es código libre
 - Servidor Web: Apache y derivados, IIS
- **ASP.NET**
 - Evaluación de ASP que permite usar cualquiera de los lenguajes .NET
 - Resuelve muchos de los problemas de rendimiento de ASP al ser compilado.
 - Puede utilizar Servicios Web XML
 - Servidor Web: IIS.

Lámina 14

Roberto Gómez C.



Hosting sitio web: self hosting

- Instalar un servidor web en una computadora.
- Acceso local
 - Usar dominio <localhost> o dirección IP 127.0.0.1
 - Necesario para desarrollar la parte
- Acceso global
 - Registrar un nombre dominio comprensible al ser humano.
 - Obtener una dirección IP
 - Estática: cuesta más
 - Dinámica: Necesita un sistema DNS

Lámina 15

Roberto Gómez C.



Hosting service

- Registrar un nombre de dominio
 - Asignar nombre servidor
 - El host se hace cargo del direccionamiento IP
- Desarrollo del sitio web de forma local.
- Actualización del sitio web a través de acceso remoto (ftp, ssh)

Lámina 16

Roberto Gómez C.



Implantación arquitectura

- LAMP: Más popular – totalmente open source
 - Linux como sistema operativo
 - Apache como servidor web
 - MySQL: como base de datos
 - PHP: para scripting de lado del servidor
- WAMP
 - Windows como sistema operativo
 - Apache como servidor web
 - MySQL: como base de datos
 - PHP: para scripting de lado del servidor

Lámina 17

Roberto Gómez C.

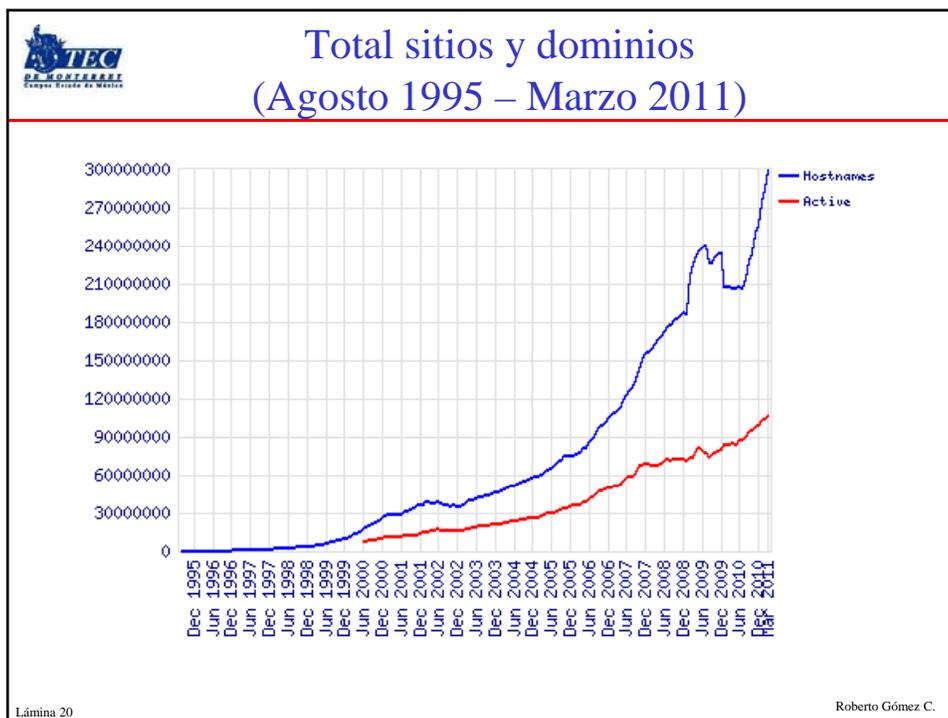
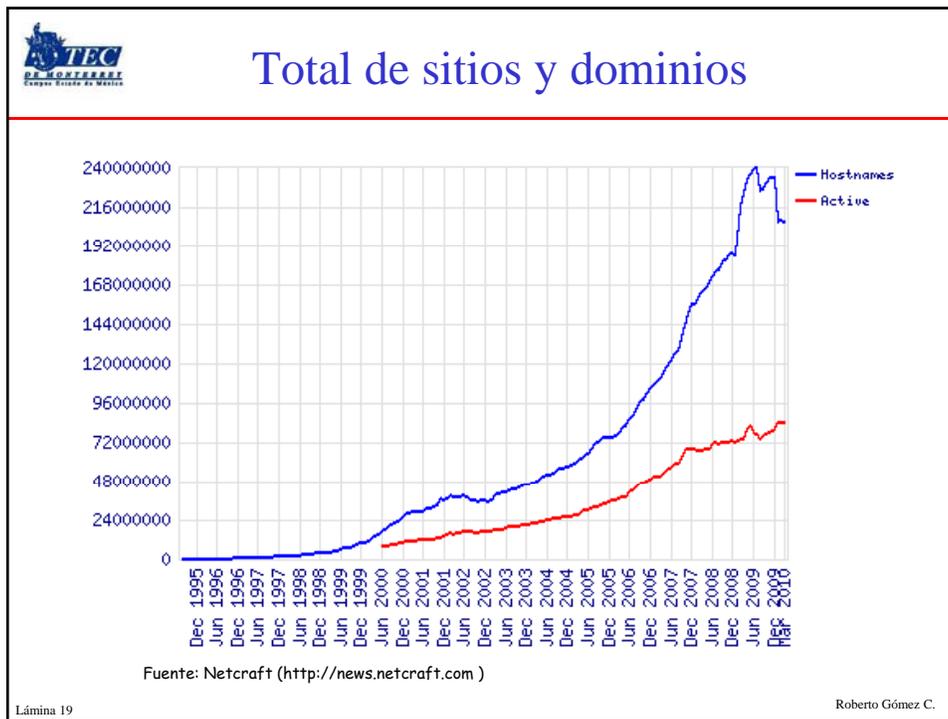


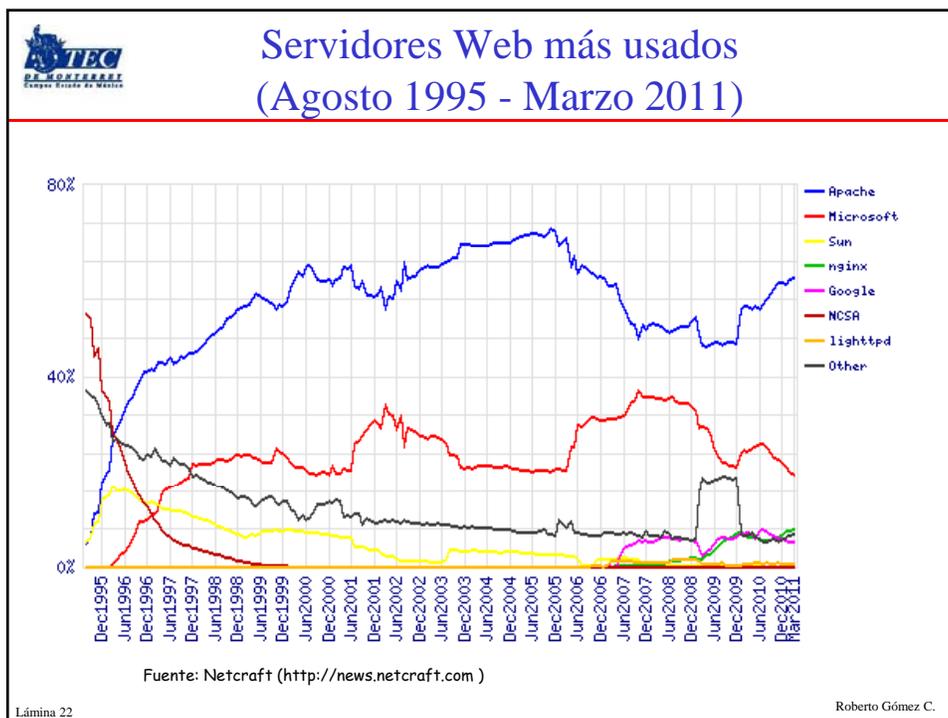
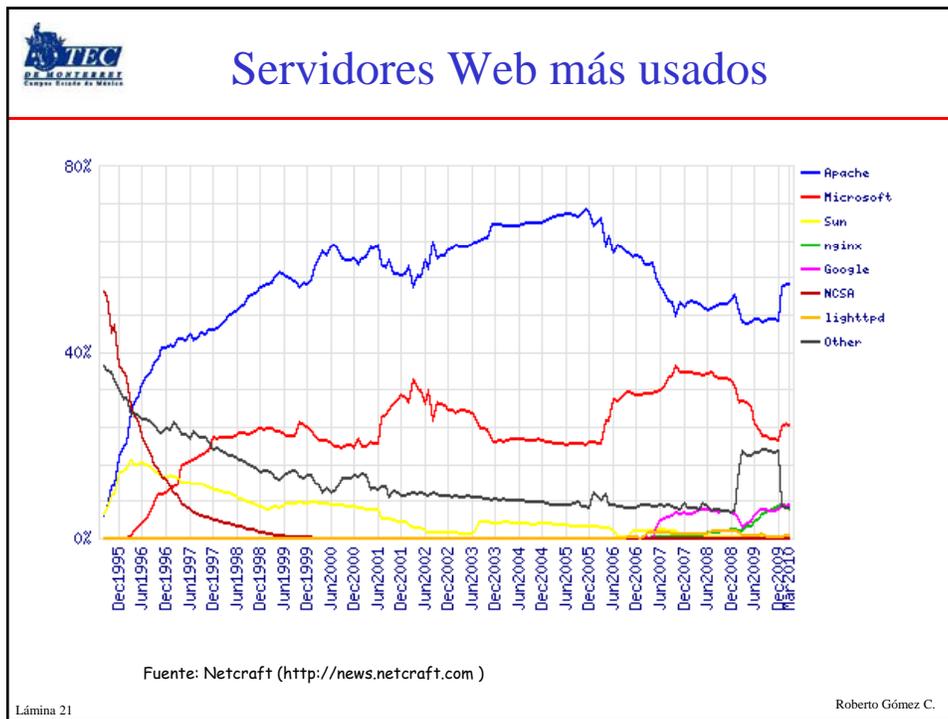
Implantación arquitectura

- WISA: todo el paquete Microsoft
 - Windows como sistema operativo
 - ISS como servidor web
 - SQL Server como base de datos
 - ASP o APS.NET: para scripting de lado del servidor

Lámina 18

Roberto Gómez C.







Servidores Web más usados

Developer	February 2010	Percent	March 2010	Percent	Change
Apache	112,903,926	54.46%	112,747,166	54.55%	0.09
Microsoft	50,928,226	24.57%	50,572,540	24.47%	-0.10
Google	14,315,464	6.91%	14,592,133	7.06%	0.16
nginx	13,978,719	6.74%	12,673,962	6.13%	-0.61
lighttpd	1,097,685	0.53%	1,657,584	0.80%	0.27

Developer	February 2011	Percent	March 2011	Percent	Change
Apache	171,195,554	60.10%	179,720,332	60.31%	0.21
Microsoft	57,084,126	20.04%	57,644,692	19.34%	-0.70
nginx	21,570,463	7.57%	22,806,060	7.65%	0.08
Google	14,454,484	5.07%	15,161,530	5.09%	0.01
lighttpd	1,953,966	0.69%	1,796,471	0.60%	-0.08

Fuente: Netcraft (<http://news.netcraft.com>)

Lámina 23 Roberto Gómez C.



Hosts más confiables (Feb 2010)

Rank	Company site	OS	Outage hh:mm:ss	Failed Req%	DNS	Connect	First byte	Total
1	www.navisite.com	Linux		0.000	0.779	0.033	0.552	0.656
2	DataPipe	FreeBSD	0:00:00	0.005	0.397	0.035	0.061	0.091
3	INetU	unknown	0:00:00	0.005	0.524	0.050	0.106	0.186
4	Hosting 4 Less	Linux	0:00:00	0.011	0.428	0.105	0.220	0.560
5	www.singlehop.com	Linux	0:00:00	0.016	0.205	0.052	0.342	0.570
6	www.dinahosting.com	Linux	0:00:00	0.016	0.115	0.089	0.182	0.182
7	New York Internet	FreeBSD	0:00:00	0.021	0.054	0.031	0.070	0.195
8	Virtual Internet	Linux	0:00:00	0.021	0.617	0.078	0.210	0.443
9	www.memset.com	Linux	0:00:00	0.021	0.616	0.080	0.160	0.160
10	Hostbasket	Windows Server 2008	0:00:00	0.021	0.377	0.083	0.177	0.177

Fuente: <http://uptime.netcraft.com/perf/reports/Hosters>

Lámina 24 Roberto Gómez C.



Hosts mas confiables (marzo 2011)

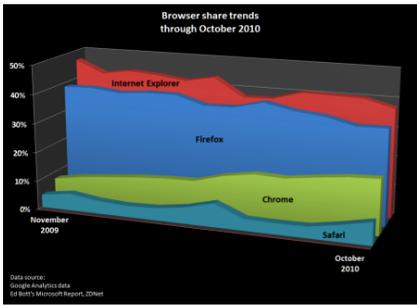
Rank	Performance graph	Company site	OS	Outage hh:mm:ss	Failed Req%	DNS	Connect	First byte	Total	Kb/s	size(K)
1	www.serverbeach.com	www.serverbeach.com	Linux		0.000	0.091	0.013	0.031	0.109	694	46
2	www.datapipe.net	Datapipe	FreeBSD		0.000	0.051	0.016	0.032	0.034	998	5
3	www.rackspace.com	Rackspace	F5 Big-IP		0.000	0.616	0.038	0.077	0.077	-	0
4	www.swishmail.com	Swishmail	FreeBSD		0.000	0.173	0.038	0.078	0.204	86	17
5	www.logicworks.net	www.logicworks.net	Linux		0.000	0.108	0.039	0.513	0.639	183	32
6	www.servint.net	Servint	Linux		0.000	1.217	0.040	0.084	0.218	129	25
7	www.reliableServers.com	ReliableServers.com	FreeBSD		0.000	0.130	0.041	0.087	0.185	63	8
8	www.qubenet.net	www.qubenet.net	Linux		0.000	0.116	0.051	0.103	0.103	-	0
9	www.singlehop.com	www.singlehop.com	Linux		0.000	0.202	0.052	0.608	0.870	132	39
10	www.aruba.it	www.aruba.it	Windows Server 2003		0.000	0.256	0.053	0.108	0.334	102	49
11	www.othellotech.net	www.othellotech.net	Linux		0.000	0.103	0.059	0.260	0.400	113	20
12	www.vi.net	Virtual Internet	Linux		0.000	0.207	0.060	0.124	0.141	159	24
13	www.netcetera.co.uk	www.netcetera.co.uk	unknown		0.000	0.034	0.061	0.124	0.250	108	19
14	www.memset.com	www.memset.com	Linux		0.000	0.167	0.064	0.253	0.446	116	25
15	www.uk2.net	www.uk2.net	Linux		0.000	0.202	0.070	0.145	0.158	170	15

Fuente: <http://uptime.netcraft.com/perf/reports/Hosters>

Lámina 25
Roberto Gómez C.

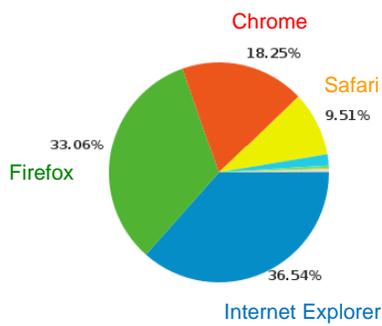


¿Y el navegador?



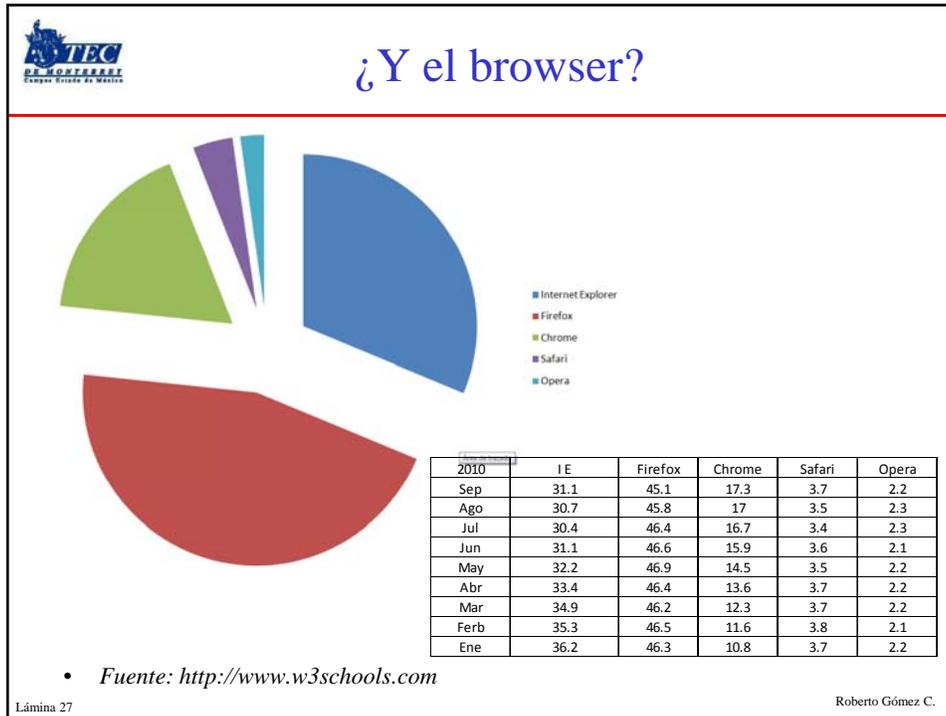
Data source: Google Analytics data, ISI Bell's Microsoft Report, 22000

- Fuente: ZDNet



- Fuente: Blog de Ed Bott

Lámina 26
Roberto Gómez C.



 El servidor Web Apache

- Actualmente es el servidor Web más popular.
- Estabilidad
- Eficiencia
- Portabilidad
- Open-source

Lámina 28 Roberto Gómez C.



Paquetes todo-en-uno

- EasyPHP
 - Incluye PHPMyAdmin para administrar base de datos MySQL
 - Instalación y configuración.
- AbriaSoft Melin Desktop Edition
 - Incluy PHP MyAdmin
- WAMP Server
- PHP Triad

Lámina 29

Roberto Gómez C.



Arquitectura Apache

- Estructurado en módulos.
- Cada módulo contiene un conjunto de funciones relativas a un aspecto concreto del servidor.
- El archivo binario httpd contiene un conjunto de módulos que han sido compilados.
- La funcionalidad de estos módulos puede ser activada o desactivada al arrancar el servidor.
- Los módulos de Apache se pueden clasificar en tres categorías:
 - Módulos Base
 - Módulo con las funciones básicas de Apache.
 - Módulos Multiproceso
 - Responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender las peticiones.
 - Módulos Adicionales
 - Cualquier otro módulo que le añada una funcionalidad al servidor.

Lámina 30

Roberto Gómez C.



Alta/Baja servicios OpenSuse 11

- Modo gráfico I:

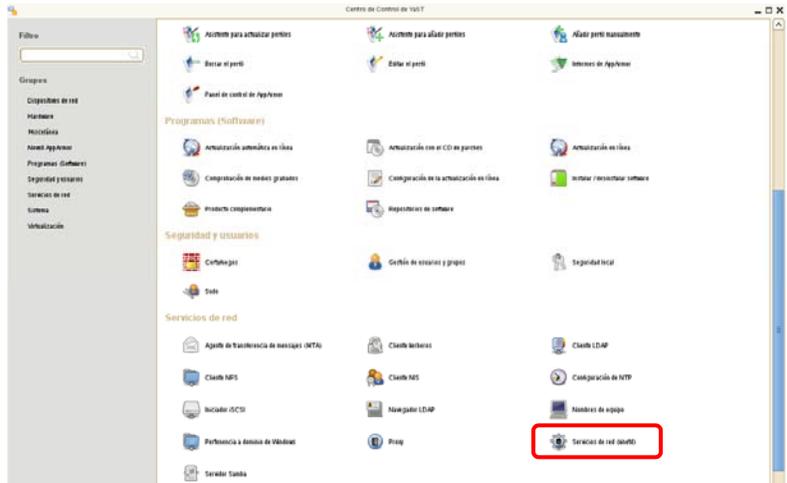


Lámina 31
Roberto Gómez C.



Alta/Baja servicios OpenSuse 11

- Modo gráfico II:



Lámina 32
Roberto Gómez C.



Alta/Baja servicios OpenSuse 11

- Modo comando
 - Directorio: `/etc/init.d`
 - Elegir servicio:
 - `./<servicio> start`
 - `./<servicio> stop`
 - `./<servicio> restart`
 - En algunas versiones es posible usar la utilería `service`:
 - `service <servicio> start`
 - `service <servicio> stop`
 - `service <servicio> restart`

Lámina 33

Roberto Gómez C.



Alza de servicios desde arranque sistema operativo

- Comando `chkconfig`
 - Permite configurar que servicios arrancan/detienen en cada nivel de ejecución.
 - No inicia ni detiene servicios a tiempo real, tan solo crea o elimina los enlaces necesarios.
- Ejemplos ejecución
 - `chkconfig --list`
 - Proporciona una lista completa de todos los servicios instalados y para cada nivel si arrancará (on) al entrar a ese nivel o se detendrá (off) o simplemente no se iniciara.
 - Al final de la lista vienen los servicios que dependen del superservidor `xinetd`.

Lámina 34

Roberto Gómez C.



Ejemplo salida chkconfig --list

```

#> chkconfig --list
apache2          0:off 1:off 2:off 3:off 4:off 5:off 6:off
bluetooth        0:off 1:off 2:off 3:off 4:off 5:off 6:off
cron              0:off 1:off 2:on  3:on  4:off 5:on  6:off
cups              0:off 1:off 2:on  3:on  4:off 5:on  6:off
dhcpcd            0:off 1:off 2:off 3:off 4:off 5:off 6:off
ldap              0:off 1:off 2:off 3:off 4:off 5:off 6:off
mysql             0:off 1:off 2:off 3:off 4:off 5:off 6:off
named             0:off 1:off 2:off 3:off 4:off 5:off 6:off
network          0:off 1:off 2:on  3:on  4:off 5:on  6:off
nfs               0:off 1:off 2:off 3:on  4:off 5:on  6:off
quotad           0:off 1:off 2:off 3:off 4:off 5:off 6:off
smb               0:off 1:off 2:off 3:on  4:off 5:on  6:off
vmware            0:off 1:off 2:on  3:on  4:off 5:on  6:off
xinetd            0:off 1:off 2:off 3:on  4:off 5:on  6:off
zebra             0:off 1:off 2:off 3:off 4:off 5:off 6:off
xinetd based services:
  rsync:          off
  sane-port:     off
  servers:       off
  services:      off
  swat:           on
  sysstat:       off
  tftp:           off
  time:          off
  time-udp:      off
  vnc:           off
  vsftpd:        off

```

Lámina 35 Roberto Gómez C.



Otras opciones

- Si se indica como argumento el nombre de algún servicio nos regresará su estatus:


```
# chkconfig smb
smb    on
```
- Sin ningún argumento nos dará una lista con el status de todos los servicios:


```
# chkconfig smb
httpd   off
mysql  on
smb     on
```
- Se desea que el servidor web apache (httpd) inicie cuando se entra en el nivel 5, entonces usamos la opción --level:


```
# chkconfig - -level 5 httpd on
```
- La base de datos MySQL no se desea que inicie en los niveles 3 y 5, solo hasta que el administrador decida arrancarla:


```
# chkconfig - -level 35 mysql off
```

Lámina 36 Roberto Gómez C.



Configuración de Módulos

- Cada módulo tiene asignadas **sus propias directivas de configuración**.
- Para cargar un módulo se utiliza la directiva LoadModule.
- Para configurar los módulos dependiendo de su presencia se utiliza la directiva <IfModule> que permite interpretar su contenido si un módulo está instalado y activado.
- Ejemplo

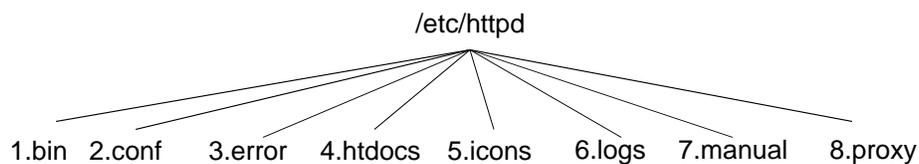

```
<IfModule mod_mime_magic.c>
MIMEMagicFile conf/magic
</IfModule>
```
- La lista de módulos de los que consta el servidor se puede ver ejecutando `httpd -l`

Lámina 37

Roberto Gómez C.



Directorios Apache



1. Archivos ejecutables
2. Archivos configuración
3. Archivos con mensajes de error
4. Directorio raíz por default del servidor
5. Directorio donde se encuentran los iconos usados por el servidor
6. Directorio donde se encuentran las bitácoras
7. Directorio donde se encuentra el manual
8. Directorio con los archivos cache del servidor

Lámina 38

Roberto Gómez C.



Apache 2

- /etc/apache2/
 - httpd.conf
 - listen.conf: Listen
 - uid.conf: user, group
 - default-server.conf
 - errors.conf

```

/etc/apache2/
|
|- charset.conv
|- conf.d/
|  |
|  |- *.conf
|
|- default-server.conf
|- errors.conf
|- httpd.conf
|- listen.conf
|- magic
|- mime.types
|- mod_*.conf
|- server-tuning.conf
|- ssl-global.conf
|- ssl.*
|- sysconfig.d
|  |
|  |- global.conf
|  |- include.conf
|  |- loadmodule.conf . .
|
|- uid.conf
|- vhosts.d
|  |- *.conf

```

Lámina 39
Roberto Gómez C.



Archivos configuración Apache 2

- httpd.conf
 - Archivo de configuración del servidor Apache principal. Evite modificar este archivo. Está integrado principalmente por declaraciones y ajustes globales.
- charset.conv
 - Especifica los conjuntos de caracteres que se deben utilizar para los distintos idiomas. No se debe editar.
- conf.d/*.conf
 - Archivos de configuración añadidos por otros módulos
- Magic
 - Datos para el módulo mime_magic que permiten que Apache determine automáticamente el tipo MIME de los archivos desconocidos. No se debe modificar.

Lámina 40
Roberto Gómez C.



Archivos configuración Apache 2

- mime.types
 - Tipos MIME reconocidos por el sistema (en realidad, se trata de un enlace a /etc/mime.types). No se debe editar.
- mod_*.conf
 - Archivos de configuración para los módulos que se instalan por defecto.
- errors.conf
 - Define el modo en que Apache responde a los errores.
- server-tuning.conf
 - Incluye directivas de configuración para los distintos módulos de multiprocesamiento o MPM
- ssl-global.conf y ssl.*
 - Configuración de SSL global y datos de certificado SSL
- vhosts.d/*.conf
 - Aquí debe encontrarse la configuración de hosts virtuales

Lámina 41

Roberto Gómez C.



Archivos default-server.conf, uid.conf y listen.conf

- Directivas default-server.conf
 - DocumentRoot
 - DirectoryIndex
- Directivas uid.conf
 - User (con valor wwwrun)
 - Group (con valor www)
- Directivas listen.conf
 - Listen (con valor 80)
 - Valores de hosts virtuales

Lámina 42

Roberto Gómez C.



Detalles DocumentRoot

- El acceso al sistema de archivos completo está prohibido por defecto por razones de seguridad.
- Por lo tanto, se debe desbloquear explícitamente el directorio DocumentRoot en el que haya colocado los archivos que debe proporcionar Apache.

```
<Directory "/srv/www/example.com_htdocs">  
  Order allow,deny  
  Allow from all  
</Directory>
```

- Si se cambia el path, es necesario que toda la ruta cuente con los permisos para que cualquiera puede acceder a los directorios que la componen.

Lámina 43

Roberto Gómez C.



Archivo server-tuning.conf

- **StartServers**
 - Número inicial de procesos servidor a arrancar.
 - Valor por defecto: 5
- **MaxClients**
 - Máximo número de conexiones simultaneas de clientes.
 - Valor limitado por la cantidad de RAM con que se cuente.
 - Valor por defecto: 150
- **MinSpareThreads**
 - Mínimo número de threads de trabajo que se mantienen en reserva.
 - Valor por defecto: 5

Lámina 44

Roberto Gómez C.



Archivo server-tuning.conf

- **MaxSpareThreads**
 - Máximo número de threads de trabajo que se mantienen en reserva
 - Valor por defecto: 10
- **ThreadsPerChild**
 - Número constante de threads de trabajo en cada proceso.
 - Valor por defecto: 25
- **MaxRequestPerChild**
 - Número máximo de peticiones que un proceso del servidor atiende.
 - Valor por defecto: 10000

Lámina 45

Roberto Gómez C.



CGIs

- CGI: Common Gateway Interface
- Especificación para la comunicación de datos genéricos entre el servidor Web y una aplicación genérica que se ejecuta en la misma máquina donde reside el servidor.
- El objetivo es permitir que el servidor responda con contenido no previamente creado, sino obtenido dinámicamente.
- La generación de contenidos dinámicos y en general el procesamiento de datos tiene serias repercusiones en la seguridad del servidor.
- Las directivas de configuración de CGI están diseñadas para reducir el ámbito de ejecución de programas lo máximo posible.

Lámina 46

Roberto Gómez C.



Configurando Apache para permitir CGIs

- La directiva ScriptAlias le indica a Apache que un directorio concreto es asignado para programas CGI.
- Apache asume que cada archivo dentro de este directorio es un programa CGI e intentara ejecutarlo, cuando sea solicitado por un cliente
- Ejemplo

```
ScriptAlias /cgi-bin/ /usr/local/apache/cgi-bin/
```
- Toda URL que contenga el texto **/cgi-bin/** se redirige a los programas contenidos en el directorio dado.

Lámina 47

Roberto Gómez C.



CGI fuera de los directorios ScriptAlias

- Los programas CGI son restringidos a menudo a los directorios ScriptAlias por razones de seguridad.
- De este modo, los Administradores pueden controlar estrictamente a quien se permite usar programas CGI.
- Sin embargo, con las precauciones de seguridad apropiadas, no hay razón por la que los programas CGI no puedan ser ejecutados desde directorios arbitrarios.
- Se puede usar la directiva Options, dentro de un archivo de configuración principal del servidor, para especificar que la ejecución CGI esta permitida en un directorio concreto:

```
<Directory /usr/local/apache/htdocs/directoriocualquiera>  
Options +ExecCGI  
</Directory>
```

Lámina 48

Roberto Gómez C.



La directiva AddHandler

- Es necesario indicarle al servidor que archivos son CGI.
- La directiva AddHandler le indica al servidor que trate todos los ficheros con extensión cgi o pl como programas CGI:

```
AddHandler cgi-script cgi pl
```

- Otra opción es el archivo .htaccess, añadiendo la directiva siguiente:

```
Options +ExecCGI
```

- No olvidar que para usar un archivo .htaccess, es necesario añadir la directiva siguiente en el archivo de configuración de Apache

```
AllowOverride Options
```

Lámina 49

Roberto Gómez C.



Escribiendo programas CGI

- Dos diferencias entre programación regular y programación CGI
 1. Toda la salida de un programa CGI debe estar precedida por una cabecera tipo-MIME.
 - Esta es una cabecera HTTP que le dice al cliente que orden está recibiendo de sus contenidos.
 - La mayoría de las veces, esto sería algo así:


```
Content-type: text/html
```
 2. La salida necesita estar en HTML, o algún otro formato que un navegador pueda ser capaz de visualizar.
 - Normalmente, será HTML, pero en ocasiones puedes querer escribir un programa CGI que genere una imagen gif, u otro contenido que no fuera HTML.

Lámina 50

Roberto Gómez C.



Primer ejemplo programa CGI

- Escrito en Perl

```
#!/usr/bin/perl  
print "Content-type: text/html\r\n\r\n";  
print "Hola Mundo.";
```

- El programa se almacena en archivo de nombre hola.pl
- Desde el navegador llamarlo de la siguiente forma:

`http://www.example.com/cgi-bin/hola.pl`

- Si esta trabajando a nivel local (sin salida internet)

`http://127.0.0.1/cgi-bin/hola.pl`

Lámina 51

Roberto Gómez C.



Posibles errores

- El código fuente del programa o un mensaje "POST Method Not Allowed"
 - Significa que Apache no esta correctamente configurado para procesar programas CGI.
 - Verificar configuración Apache.
- Un mensaje que empieza con "Forbidden"
 - Quiere decir que hay un problema de permisos.
- Un mensaje diciendo: "Internal Server Error"
 - Permisos archivos.
 - Información de rutas.
 - Errores de sintáxis.
 - Verificar los errores en las bitácoras de error de Apache.

Lámina 52

Roberto Gómez C.



Segundo ejemplo de CGI

```
#!/bin/sh
echo Content-type: text/html
echo
echo
echo "<HTML>"
echo "<HEAD>"
echo "</HEAD>"
echo "<BODY>"
echo "<H2>Users logged in are:</H2>"
echo "<PRE>"
who
echo "</PRE>"
echo "</BODY>"
echo "</HTML>"
```

Lámina 53

Roberto Gómez C.



Bitácoras Apache

- No solo son útiles en el caso de anomalías sino también para obtener información sobre lo procesado por el servidor.
- El acceso a las bitácoras debe estar restringido a los administradores.
- Dos tipos de información
 - Las bitácoras de acceso: se registra cualquier petición enviada al servidor.
 - Las bitácoras de error
- Apache cuenta con directivas para la administración de eventos relacionados con los dos tipos de información anteriores.
 - <ErrorLog>
 - <LogLevel>

Lámina 54

Roberto Gómez C.



La directiva CustomLog

- Sintaxis:
 - CustomLog archivo_bitacoras formato_bitacoras
- Primer argumento
 - Nombre del archivo donde los registros de las bitácoras deben escribirse.
 - Puede ser un nombre completo, una ruta relativa o un pipe.
- Segundo argumento especifica un formato para línea del archivo de bitácora.
- A definir en el archivo default-server.conf

Lámina 55

Roberto Gómez C.



Bitácoras de acceso

- Almacena cada petición enviada al servidor web.
- Una configuración típica es


```
LogFormat "%h %l %u %t \"%r\" %<s %b" common
CustomLog logs/access_log common
```

 - lo anterior define el nickname common y lo asocia con un formato particular de cadena de caracteres.
- Caracteres alfanuméricos pueden ser usados dentro del formato.
- Formato consta de caracteres especiales, precedidos de un carácter % que le indica al servidor que información almacenar.
- Si es necesario incluir las comillas hay que precederlas de un carácter “\”.
- El formato puede contener caracteres especiales de control como “\n” (salto de línea) y “\t” (tabulador).
- La directiva CustomLog define el nombre del archivo donde se almacenará información con el formato definido.
 - Por default esta en /etc/httpd/logs/access_log

Lámina 56

Roberto Gómez C.



Formato bitácora Apache

Símbolo	Información almacenada
%h	El host remoto.
%l	Almacena el logname, si este es proporcionado.
%u	Usuario remoto (muy útil si se accede al servidor a través de autenticación).
%t	La fecha y hora de la petición.
%r	La petición hecha al sitio web.
%s	El status de la petición (201, 301, 404, 500) el carácter ">" en frente de la "s" asegura que solo el último status sea almacenado.
%b	Bytes enviados por la petición (seguimiento del ancho de banda usado).
%i	Da seguimiento a los enviados en el encabezado HTML. Añadiendo (Referer) y (User Agent) se captura el URL de referencia y tipo de browser.

Lámina 57 Roberto Gómez C.



Bitácoras de errores.

- Los mensajes de error son almacenados de forma separada, usando un archivo diferente.
- Dentro de la directiva se definen el nivel de error y el lugar donde se almacenarán los mensajes.


```
ErrorLog logs/error_log
LogLevel warn
```
- El nivel de error define el tipo de mensaje a ser almacenado.
 - Mensajes de menor nivel no son almacenados.
- El formato de error es relativamente libre y descriptivo.
- Por default el archivo de almacenamiento se encuentra en


```
/var/log/apache2/error_log
```
- Ejemplo mensaje


```
[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client
denied by server configuration: /export/home/live/ap/htdocs/test
```

Lámina 58 Roberto Gómez C.



Niveles de error

Nivel	Descripción
Emerg	Emergencia – el sistema no se puede usar.
alert	Alguna acción se debe llevar a cabo inmediatamente.
Crit	Condiciones críticas.
Error	Condiciones de error.
Warn	Condiciones de advertencia.
Notice	Condición normal pero significativa.
Info	Información
Debug	Mensajes de depuración

- Si se asigna el nivel a crit, los mensajes de niveles más importantes también serán almacenados.
 - Mensajes alert y emerg en este caso.
- Los mensajes de nivel notice siempre son almacenados.
- Nota: hay una liga entre este directorio y /var/log/http

Lámina 59
Roberto Gómez C.



Redirección de páginas

- Parte del modulo `mod_alias`.
 - Módulo que proporciona medios para mapear diferentes parte del sistema de archivos del host en el árbol del documento y en la redirección del URL.
- Directiva `Redirect`
 - Mapea un viejo URL en uno nuevo.
 - El URL nuevo es regresado al cliente.
 - A definir en el archivo `default-server.conf`
- Ejemplo
 - `Redirect /service http://foo2.bar.com/service`
 - Si el cliente solicita la página `http://myserver/service/foo.txt`, este será redirigido a la página `http://foo2.bar.com/service/foo.tx`

Lámina 60
Roberto Gómez C.



Configuración descentralizada

- Conforme el tamaño de los datos de un servidor es más grande, las tareas de configuración se complican.
- Si la configuración reside en un único fichero su tamaño puede llegar a ser inmenso.
- Apache permite la configuración descentralizada mediante la presencia de ficheros de configuración en los directorios de documentos.
- Las directivas locales se almacenan en el archivo `.htaccess` (se puede cambiar este nombre) en el propio directorio de datos.
- La sintaxis de estos ficheros es la misma que para `httpd.conf`.

Lámina 61

Roberto Gómez C.



Configuración descentralizada

- Sólo un subconjunto de directivas pueden incluirse en el archivo `.htaccess`:
 - La directiva `AllowOverride` permite especificar en la configuración del `httpd.conf` qué directivas pueden aparecer en el `.htaccess`
 - Ej: `AllowOverride Options`
 - que permite poner en el `.htaccess`:
 - `Options +ExecCGI`
 - `AllowOverride` dentro de un ámbito de directorio:


```
<Directory "C:/Program Files/Apache Group/Apache/htdocs">
AllowOverride None
Options None
</Directory>
```

Lámina 62

Roberto Gómez C.



Configuración descentralizada

- AllowOverride permite activar o desactivar 5 categorías de directivas:
 - AuthConfig
 - Permite directivas de seguridad como AuthName, Satisfy, y Require
 - FileInfo
 - Controlan como procesar los ficheros
 - Indexes
 - Afecta a los listados e incluye directivas como IndexOptions, AddDescription, y DirectoryIndex.
 - Limit
 - Similar a AuthConfig pues las directivas que cubre son también de seguridad. Incluyen directivas Order, Allow, y Deny.
 - Options
 - Para permitir opciones como ContentDigest, XBitHack, y Options..

Lámina 63

Roberto Gómez C.



El archivo .htaccess

- Ejemplo (si AllowOverride AuthConfig)

```
AuthUserFile /srv/www/htdocs/algo/.htpasswd
AuthName "Autenticación de Usuario"
AuthType Basic
Require valid-user
```

- Se protege el contenido del directorio con un mecanismo de autenticación.
- El archivo que contiene los usuarios y sus claves se especifica por la directiva AuthUserFile.
- Se permite acceder únicamente a usuarios debidamente autenticados y cuyas peticiones procedan de máquinas en la red 163.117.139.*

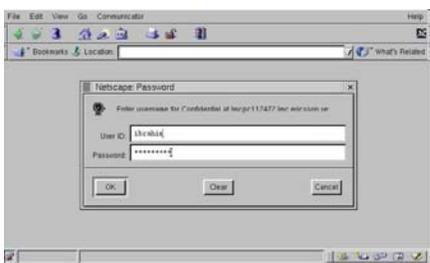
Lámina 64

Roberto Gómez C.



Ejemplo de acceso con autenticación

- Cuando .htaccess requiere autenticación, el cliente se encarga de ofrecer al usuario el interfaz para obtenerla.



- Una vez obtenidos los datos por el navegador, se envían al servidor para su comprobación, y en caso de ser correctos se envía el documento requerido.
- Los archivos de autenticación es particular para ese directorio y pueden estar en cualquier lugar del sistema de ficheros

Lámina 65
Roberto Gómez C.



Definiendo contraseñas

- Comando htpasswd
- Ejemplo uso

```
htpasswd -c /home/httpd/bin/.htpasswd username
```

- Donde
 - /home/httpd/bin/.htpasswd: directorio donde van a residir la contraseña.
 - username: nombre del usuario
- Después se preguntara la contraseña.
- Para apache2 es necesario usar: htpasswd2

Lámina 66
Roberto Gómez C.



Servidores virtuales

- El servidor web responde a peticiones mediante el envío de información interna.
- ¿Se pueden servir peticiones de más de una máquina?
 - Ejemplo: Las direcciones `www.example.com` y `mail.example.com` deben ser gestionadas por el mismo servidor.
- La solución que ofrece Apache está basada en el concepto de host virtual.
- El servidor responde a las peticiones recibidas no sólo por el servidor por defecto sino también por cada uno de los hosts virtuales.

Lámina 67

Roberto Gómez C.



Directivas hosts virtuales

- A efectos de configuración, un host virtual se configura exactamente igual que uno normal.
- Las directivas de configuración se incluyen en el elemento `<VirtualHost>`
- La mayoría de directivas de configuración global se pueden incluir en el contexto de un host virtual.
- Ejemplo:

```
<VirtualHost *>
  ServerName www.domain.tld
  ServerAlias domain.tld *.domain.tld
  DocumentRoot /www/domain
</VirtualHost>
```

- Apache permite dos modalidades de host virtuales: basados en nombre y basados en dirección IP.

Lámina 68

Roberto Gómez C.



Host Virtuales Basados en Nombre

- Dos nombres de dominio tienen idéntica dirección de IP, pero deben ser tratados como dos hosts virtuales.
- Apache canaliza las peticiones dependiendo del nombre de la máquina en la URL.
- Cuando se utilizan hosts virtuales, el propio host por defecto de apache pasa a ser un host virtual.
 - Ventajas: Se ahorran direcciones IP para hosts que únicamente se utilizan para referenciar documentos en URIs.
 - Inconveniente: Ciertas aplicaciones de seguridad, identifican los recursos por su dirección de IP y no por su nombre.

Lámina 69

Roberto Gómez C.



- En OpenSuse se cuentan con archivos de tipo esqueleto que se encuentran en vhosts.d, dentro del directorio /etc/apache2.
- Copiar vhost.template al archivo donde se va a llevar a cabo la configuración.
- Es importante que el nombre del archivo termine en .conf, para que sea cargado al reiniciar el servidor.
- Importante: modificar el archivo /etc/hosts para que redireccione el nombre del servidor a la dirección correcta.

Lámina 70

Roberto Gómez C.



Ejemplo

```
<VirtualHost *>
  ServerAdmin toto@acme.com
  ServerName toto.com
  ServerAlias toto
  DocumentRoot /srv/www/vhosts/toto
  ErrorLog /var/log/apache2/toto-error_log
  CustomLog /var/log/apache2/toto-access_log combined
  ServerSignature On
  ScriptAlias /cgi-bin/ "/srv/www/vhosts/toto/cgi-bin/"
  <Directory "/srv/www/vhosts/toto/cgi-bin">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
  </Directory>
```

Lámina 71

Roberto Gómez C.



Cotinuación ejemplo

```
<Directory "/srv/www/vhosts/toto">
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
</VirtualHost>
```

Lámina 72

Roberto Gómez C.



Servidores Web

Roberto Gómez Cárdenas
rogomez@itesm.mx
<http://homepage.cem.itesm.mx/rogomez>

Lámina 73 Roberto Gómez C.



Configuración de un Host Virtual Basado en Nombre

- Para utilizar hosts virtuales basados en nombre es preciso especificar la dirección de IP y puerto por el que se reciben las peticiones.
- La directiva NameVirtualHost ip se utiliza para definir este parámetro.
- A continuación se debe crear un bloque con la directiva <VirtualHost> para cada uno de los hosts que se quieran albergar.
- El argumento de <VirtualHost> debe ser idéntico al incluido en la directiva NameVirtualHost.
- La definición del host virtual debe incluir como mínimo las directivas ServerName y DocumentRoot especificando, respectivamente el nombre y localización de los documentos del servidor.
- Como el host por defecto desaparece, la creación de un host virtual requiere la definición de dos elementos VirtualHost

Lámina 74 Roberto Gómez C.



Ejemplo

```

NameVirtualHost *:80
<VirtualHost *:80>
  ServerName toto
  ServerAlias www.toto.com *.toto.com
  DocumentRoot /srv/www/toto
  <Directory "/srv/www/toto">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order Allow,Deny
    Allow from all
  </Directory>
</VirtualHost>

```

Lámina 75 Roberto Gómez C.



Ejemplo de Virtual Host Basado en Nombre

- Supongamos que se quieren servir documentos para los hosts: www.dominio.com y www.otrodominio.com.

```

NameVirtualHost *
<VirtualHost *>
  ServerName www.domain.com
  ServerAlias domain.com *.domain.com
  DocumentRoot /www/domain
</VirtualHost>
<VirtualHost *>
  ServerName www.otherdomain.com
  DocumentRoot /www/otherdomain
</VirtualHost>

```

Lámina 76 Roberto Gómez C.



Ejemplo de Virtual Host Basado en Nombre

```
NameVirtualHost *  
<VirtualHost *>  
  ServerName www.domain.com  
  ServerAlias domain.com *.domain.com  
  DocumentRoot /www/domain  
</VirtualHost>  
<VirtualHost *>  
  ServerName www.otherdomain.com  
  DocumentRoot /www/otherdomain  
</VirtualHost>
```

- El parámetro "*" en la primera directiva indica que se deben procesar todas las peticiones recibidas en el servidor (puede tener más de una dirección de IP).
- El primer host virtual sirve todas las peticiones recibidas en los dominios acabados en **domain.com**.
- El segundo host virtual procesa sólo aquellas peticiones dirigidas a **www.otherdomain.com**.
- La directiva **ServerAlias** permite que el servidor sea referido mediante más de un nombre.
- Esta configuración depende de que el **DNS tenga la información correcta**.

Lámina 77

Roberto Gómez C.



Múltiples sitios en diferentes puertos

- Supongamos que tenemos múltiples dominios redirigidos a la misma IP.
- Se quiere servir diferente información por diferentes puertos.
- La directiva Listen permite considerar más de un puerto.

Lámina 78

Roberto Gómez C.



Ejemplo

```

Listen 80
Listen 8080
NameVirtualHost 167.10.20.30:80
NameVirtualHost 167.10.20.30:8080
<VirtualHost 167.10.20.30:80>
  ServerName www.servidor1.com
  DocumentRoot /home/servidor1/html
  ErrorLog /logs/httpd/error_log1
  TransferLog /logs/httpd/access_log1
</VirtualHost>
<VirtualHost 167.10.20.30:8080>
  ServerName www.servidor2.com
  DocumentRoot /home/servidor2/html
  ErrorLog /logs/httpd/error_log2
  TransferLog /logs/httpd/access_log2
</VirtualHost>
<VirtualHost 167.10.20.30:80>
  ServerName www.servidor3.com
  DocumentRoot /home/servidor3/html
  ErrorLog /logs/httpd/error_log3
  TransferLog /logs/httpd/access_log3
</VirtualHost>
<VirtualHost 167.10.20.30:8080>
  ServerName www.servidor4.com
  DocumentRoot /home/servidor4/html
  ErrorLog /logs/httpd/error_log4
  TransferLog /logs/httpd/access_log4
</VirtualHost>

```

Lámina 79

Roberto Gómez C.



Host virtuales basados en direcciones IP

- Un servidor dispone de más de una dirección IP.
- Esto implica que se dispone de más de una tarjeta de red o que el sistema operativo administra más de una interfaz de red.
- En el caso de Linux/Unix es posible usar utilerías como ifconfig o técnicas como ip aliases permiten implementar esta configuración.
- Los paquetes se dirigen a una de las posibles direcciones IP, pero se procesan por la misma máquina.
- Ventajas
 - A todos los efectos, al exterior se ofrecen varias direcciones de IP.
- Desventajas
 - Si el número de hosts es muy elevando, se puede tener problemas con el número de IPs disponibles.

Lámina 80

Roberto Gómez C.



Ejemplo de configuración virtuales basados en IP

```
<VirtualHost 10.9.8.3>
  ServerAdmin wb@maquina.com
  DocumentRoot /home/toto/www
  ServerName www.toto.com
  ErrorLog /logs/httpd/erro.toto
  TransferLog /logs/httpd/acceso.toto
</VirtualHost>
```

```
<VirtualHost 10.9.8.4>
  ServerAdmin wb@maquina.com
  DocumentRoot /home/cachafas/www
  ServerName www.cachafas.com
  ErrorLog /logs/httpd/erro.cachafas
  TransferLog /logs/httpd/acceso.cahafas
</VirtualHost>
```

- No es preciso usar la directiva NameVirtualHost.
- La directiva <VirtualHost> puede especificar el nombre de un host en lugar de una dirección IP.
- Se recomienda utilizar una dirección IP e incluir la directiva ServerName para garantizar el funcionamiento independientemente del servicio DNS.
- Apache permite combinar de forma arbitraria hosts virtuales basados en nombres, basados en IPs y discriminación por número de puerto.

Lámina 81 Roberto Gómez C.



Servidores Web

Roberto Gómez Cárdenas
rogomez@itesm.mx
<http://homepage.cem.itesm.mx/rogomez>

Lámina 82 Roberto Gómez C.