

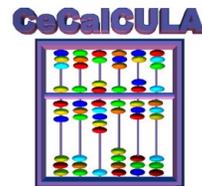
Centro de Cálculo Científico
de la Universidad de Los Andes
Mérida - Venezuela

Actividades de un

Web Master

Ing. Rodrigo Torréns H.

Noviembre 1999



www.cecalc.uia.ve

CONTENIDO

PARTE I: Papel del Web Master

PARTE II: Aspectos básicos de un servidor Web

PARTE III: Instalación de un servidor Web

PARTE IV: Seguridad. Análisis de registros de accesos

Referencias

PARTE I: Papel del Web Master

- Qué es un WebMaster**
- Habilidades que debe poseer un WebMaster**
- Tareas de un WebMaster**
- Reglas básicas a seguir**

- **Qué es un WebMaster?**

Definición más vaga:

“Persona que trabaja en contacto con el Web”.

- **Definiciones un poco más específicas:**

“Un Web Master es la persona responsable de manejar la evolución y buen funcionamiento de un sitio Web”.

Otra definición sería:

“Persona encargada de diseñar, implementar y mantener un sitio efectivo en el WWW”.

Conocimientos y habilidades necesarias (deseables) en un WebMaster

- Programación en varios lenguajes:
 - C,C++, Perl, VisualBasic, etc.
- Codificación HTML, DHTML, JavaScript, etc.
- Programación CGI.
- Configuración y administración de redes.
- Compilación e instalación de Software
- Diseño y desarrollo de Bases de Datos
- Diseño gráfico de interfaces de usuario.
- Desarrollo de software.
- Desarrollo, mercadeo y administración de proyectos.
- Tener habilidades para trabajar en ambientes UNIX y PC.

Tareas a ser realizadas en un sitio Web exitoso:

1. Recopilación / creación de contenido
2. Diseño estructural/arquitectura
3. Implementación
4. Diseño visual/gráfico
5. Administración/mantenimiento técnico

Pregunta:

*Un WebMaster debería o podría
realizar todas estas tareas?*

Respuesta:

No !

Pero lo está haciendo !

Veamos más tareas
específicas...

Tareas específicas de un WebMaster

- Recopilación y redacción de información
- Diseño de gráficos
- Coordinación de equipos interdisciplinarios
- Administración de servidores Web
- Instalación y configuración de software
- Instalación y configuración de hardware
- Actualización constante de los contenidos
- Desarrollo de "Gateways" o interfaces a otros programas (CGI)
- Contabilidad y análisis de las estadísticas de acceso al sitio
- **Identificar, conocer y entrenar a los propietarios de contenido en el uso de herramientas de publicación y conversión de documentos Web**
- **“Vigilancia”** del desarrollo, contenidos y seguridad de su sitio

(Lamentablemente - o afortunadamente - muchas de estas tareas chocan con las de un *diseñador* Web !)

Entonces, en cuales de estas tareas

se debería concentrar ?

ó

cuales debería **no** realizar ?

Existe una contradicción o ambigüedad
en cuanto al papel del WebMaster ?

(debe o no tener todos esos conocimientos y tareas asignadas ?

Se aceptan opiniones...)

En realidad muchas personas hablan acerca de que un
WebMaster

no es una sola persona

sino un

grupo de trabajo

que funciona

coordinadamente

...Papel de un WebMaster

En un sitio típico, las responsabilidades de un WebMaster pueden clasificarse en cuatro categorías:

Proveedores de contenido

Crean-editan documentos HTML, incorporan imágenes y formas electrónicas y mantienen la integridad de los enlaces.

Diseñadores

Crean el diseño arquitectónico o gráfico del sitio.

Programadores

Escriben programas CGI, Java, JavaScript, conectan el sitio con un SMDB, etc

Administradores

Se aseguran que el servidor esta funcionando apropiada y eficientemente todo el tiempo, y vigilan la seguridad e integridad de los contenidos.

A mi manera de ver, un WebMaster se debería concentrar en lo siguiente:

- Actualización constante de conocimientos relativos al área
- “**Educar**” a los poseedores de la información y contenidos para que puedan publicar la información por si mismos o con muy poca ayuda siguiendo patrones de diseño previamente definidos - si fuera el caso.
- Concentrarse en las etapas tempranas de diseño y desarrollo de un sitio, en los aspectos “**arquitectónicos**” y “**navegacionales**” de la información, para que el sitio sea más fácil mantener y actualizar.
- Recopilar y analizar las estadísticas de acceso a los sitios mantenidos por él
- Asegurarse de la seguridad e integridad de los datos.
- Organizar grupos de trabajo

Algunas reglas a seguir por un WebMaster *de contenido:*

- 1- Cambiar contenidos regularmente.
- 2- Maximizar velocidad e impacto del "*Home Page*" (página inicial)
- 3- Incluir formas de interacción/comunicación con los usuarios del sitio.
- 4- Proveer al sitio de una estructura/arquitectura navegacional consistente.
- 5- Hacer que la espera por los gráficos valga la pena.
- 6- Recuerde que nada es lo suficientemente obvio.
- 7- No abuse de los enlaces externos.
- 8- Dejar a los usuarios "crear"el sitio.
- 9- Seguir una metodología de desarrollo consistente.

PARTE II: Aspectos básicos de un servidor Web

- Características básicas**
- Estructura y componentes**
- Como funciona un servidor Web?**
- Consideraciones y necesidades previas a la instalación de un servidor**
 - Hardware**
 - Software**
 - Conectividad**
- Herramientas adicionales**

SERVIDORES PARA EL WWW

Que es un Servidor Web?:

“Programa que se encuentra escuchando o esperando constantemente por una petición realizada desde la red. Cuando se recibe esta petición el servidor realiza la acción correspondiente, o responde de la manera adecuada y sigue escuchando”.

Un servidor WWW dialoga (o se comunica) con sus clientes a través del protocolo HTTP

...Características básicas

Como funciona un servidor Web

El funcionamiento de un servidor **Web** es muy sencillo y consiste básicamente en enviar al cliente los archivos que este le solicita (ver figura).

El *idioma* en que cliente y servidor se *hablan* es el protocolo **HTTP**



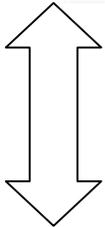
Figura 1: *Funcionamiento de un servidor Web*

Cliente Web



ACTIVIDADES DEL CLIENTE

- Solicitar archivos al servidor.
- Interpretar y desplegar código HTML.
- Interpretar Lenguajes de *Scripting* y ejecutarlos.
- Visualizar Imágenes.
- Ejecutar *Applets Java*.
- Arrancar aplicaciones externas o *plug-ins*.
- Controlar algunos aspectos de la presentación del documento (apariciencia).



Servidor



ACTIVIDADES DEL SERVIDOR

- Enviar archivos al cliente.
- Esperar por peticiones de los clientes.
- Correr programas mediante CGI y enviar respuestas a cliente.
- Establecer conexión a SMD
- Servir de "*gateways*" a otros servicios: telnet gopher, mail, B.D., ftp, etc.

Figura 2: Servidor y Cliente Web

Estructura y componentes de un Servidor Web

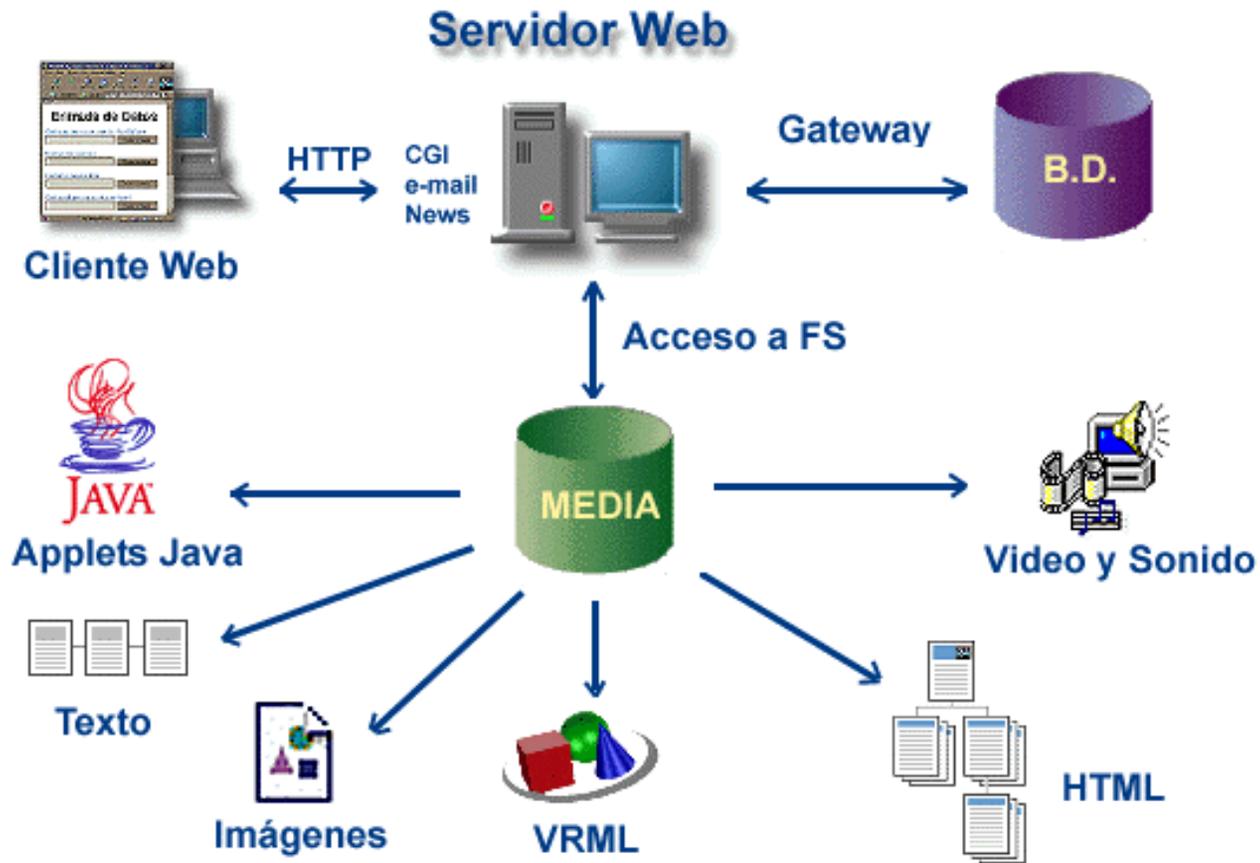


Figura 3: Estructura y componentes de un Servidor Web

Consideraciones y necesidades previas

- Consideraciones y necesidades Previas a la Instalación de un servidor.

+ **Hardware:** La mayoría de los servidores Web no impone muchas exigencias al hardware sobre el que están corriendo, por lo tanto un servidor puede funcionar perfectamente a partir de un 486 con 16MB de RAM y 10MB de espacio en disco, con tarjeta de red o modem a partir de 9600 bps (depende del tráfico esperado).

+ **Software:** Existen versiones de servidores Web para la mayoría de las plataformas de S.O existentes: Linux, SunOS, Solaris, AIX, Win95/NT, etc. Por su puesto necesitara una versión precompilada del servidor Web correspondiente a su S.O, además de un Web Browser para probar el funcionamiento del servidor.

A continuación algunos direcciones de los servidores más populares en varias plataformas:

- **Linux** : Apache Web Server : www.apache.org
- **Unix** : NCSA server : hoohoo.ncsa.uiuc.edu
- **Win 95** : Web Site : website.ora.com
- **Win 96/98/NT** : Internet Information Server : www.microsoft.com

+ **Conectividad:** El computador donde estará instalado el software servidor,deberá estar conectado a la red y corriendo el protocolo TCP/IP. Windows 95 y NT ya traen incorporados todos los protocolos necesarios, por lo que no necesita software adicional. En el caso de UNIX, es tarea de los administradores de su sistema configurar lo necesario.

Información sobre la Red y el Servidor

La información inicial que se necesita para instalar y configurar el servidor es la siguiente:

- Dirección IP de su servidor
- Nombre de dominio para su servidor, en la forma: (FQDN)
Nombre del servidor DNS
- Nombre del servidor DNS (opcional pero recomendado)
- Directorio donde van a residir sus documentos (Document Root)
- Directorio donde va residir el servidor (Server Root)
- Directorio donde residirán programas CGI (CGI-dir)

Herramientas de desarrollo adicionales:

- JAVA

 - J.D.K: *java.sun.com*

- PERL

 - Perl: *www.perl.com*

 - Perl para win 32: *www.ActiveState.com*

- Otros Lenguajes

 - Visual Basic: *www.microsoft.com*

 - GNU C/C++, Fortran: *www.gnu.org*

- S.M.B.D.

 - mSQL: *www.hughes.com*

 - Postgres: *www.postgres.com*

PARTE III: Instalación de un servidor Web: Apache Web Server

Actividades:

- 1. Escogiendo el servidor**
- 2. Obteniendo Apache**
- 3. Descomprimiendo Apache**
- 4. Instalando y configurando Apache**
- 5. Arrancando y deteniendo el servidor Web**

1. Escogiendo el servidor

Escogencia/Selección del servidor, consideraciones:

- Plataforma de Hardware/Software disponible
- Costo del Software
- Facilidad de instalación, configuración y uso
- Posibilidad de administración remota
- Robustez/confiabilidad

2. Obteniendo Apache

Coloquemos como ejemplo la instalación de Apache:

La última versión de apache se puede obtener en:

`www.apache.org`

Aquí también se encontrará información actualizada sobre la última versión de la distribución.

Puede obtener tanto los fuentes, como versiones precompiladas.

Aquí explicaremos brevemente la instalación de una versión de Apache precompilada para Linux, aunque el proceso es parecido en todas las versiones de Apache

3. Descomprimiendo Apache

La distribución de Apache obtenida se puede descomprimir con la siguiente orden:

```
tar -xvtz apache_1.2.0.i386-unknown-linux-ELF.tar.gz
```

Este comando generará una estructura de directorios parecida a la siguiente:

ABOUT_APACHE_LICENCE	conf/	logs/
CHANGES	htdocs/	scr/
README	cgi-bin/	support/
KEYS	icons/	

Archivos del servidor

El contenido y propósito de los archivos principales es el siguiente:

`conf/` : contiene los archivos de configuración principales del servidor WEB

`logs/` : aquí se guardan los archivos de estadísticas de accesos y errores del servidor

`htdocs/` : por omisión, directorio que va a contener archivos gráficos y HTML del servidor

`cgi-bin/` : directorio donde por omisión deben estar colocados los programas CGI

`support/` : contiene archivos y programas utilitarios (no en todas las versiones)

`icons/` : archivos gráficos que usa el servidor

4. Instalando y configurando Apache

El siguiente paso es editar los archivos de configuración para el servidor, tarea que consiste en *entonar* o configurar varias *directivas* en 3 archivos principales ubicados en el directorio `conf`:

- `srm.conf`
- `access.conf`
- `httpd.conf`.

Adicionalmente se tendría que editar el archivo `mime.types` cada vez que se quiera agregar otros tipos de datos.

Tenga mucho cuidado con los parámetros de estos archivos, ya que cualquier error implica un funcionamiento incorrecto del servidor.

(Nota: en las últimas versiones de Apache todas las directivas vienen incluidas en el archivo `httpd.conf`)

Archivos de configuración del servidor

En la siguiente tabla se da una descripción general de las funciones generales de cada uno de estos archivos.

Nombre del archivo	FUNCION
<code>httpd.conf</code>	En este archivo se indica la configuración principal del servidor. Aquí se configuran atributos como la asignación del puerto para el servidor, el dueño bajo el que se ejecuta el servidor, etc. También se indican aquí, parámetros que controlan la ejecución de servidores httpd paralelos.
<code>srm.conf</code>	Aquí se ajustan parámetros como la raíz del árbol de documentos, funciones especiales como SSI, manejo de los mapas sensitivos, etc.
<code>access.conf</code>	Gestiona restricciones de acceso al servidor.
<code>mime.conf</code>	Especifica asociaciones entre tipos MIME conocidos y extensiones de archivo.

Directivas de configuración

Directivas contenedoras (en `access.conf`):

se refieren a un determinado directorio o conjunto de archivos, y se usan para englobar o incluir otras. Por Ej.:

```
<Directory /home/usuarios/pguillen>  
...  
</Directory>
```

especifica que todas las directivas encerradas hacen referencia solo al directorio `/home/usuarios/pguillen`

Directivas para restricciones de acceso (en `access.conf`):

las más comunes (dentro de una directiva contenedora) son:

- `Options`: permite indicar opciones disponibles en un directorio, como la posibilidad de ejecutar CGI's, incorporar SSI, etc.
- `order`: indica el orden en que se evaluarán las directivas `allow` y `deny`
- `allow from`, `deny from`: especifican una máscara de máquinas a las que se permitirá o denegará el acceso al directorio.

4. ...Instalando y configurando Apache

Directivas para CGI (en `srm.conf`):

permite designar un directorio para que ejecute programas CGI, con la directiva `ScriptAlias`. Por ejemplo:

```
ScriptAlias /cgi-bin/ /home/usuarios/web/programas-cgi/
```

redirige las peticiones a `/cgi-bin/` al directorio `/home/usuarios/web/programas-cgi`, y lo considera como directorio de CGIs.

Asociación de tipos MIME (archivo `mime.types`):

Los tipos **MIME** conocidos y sus extensiones asociadas se encuentran en el archivo `mime.types`. Un ejemplo sería:

```
x-world/w-vrml      wr1 vrml
```

```
text/html          html htm
```

Restricciones de acceso (archivo `access.conf`):

Es posible restringir el acceso a un determinados directorio en función del origen de la petición y también se pueden emplear identificaciones mediante *login* y *passwords*. En la parte 4 de este curso explicaremos en detalle todo lo necesario para configurar restricciones de acceso.

5. Arrancando y deteniendo el servidor

Arrancando el servidor Apache

El servidor inicia su operación ejecutando el siguiente comando:

```
httpd
```

...ubicado en el directorio `src`. Al hacerlo se buscará el archivo `httpd.conf` en la ubicación con que fue compilado el código (por omisión: `/usr/local/apache/conf/httpd.conf`). Si este archivo está en otro sitio, se debe indicar su localización con el argumento `-f`. Por ejemplo el comando:

```
/usr/local/apache/httpd -f /usr/utiles/apache/conf/httpd.conf
```

...arrancará el servidor Apache, utilizando el archivo de configuración `httpd.conf` ubicado en el directorio `/usr/utiles/apache/conf`

Si todo funciona correctamente, inmediatamente se debe retornar al *prompt* de comandos, lo que indicará que el servidor está levantado y esperando por peticiones de clientes. Si algo va mal, en este momento se recibirá un mensaje de error. Si el servidor arrancó bien, en este momento puede usar su *browser* y obtener los documentos Web o archivos que allí se encuentren. La dirección con que consultaría los documentos sería:

```
http://nombre_de_su_host/camino_al_documento
```

Si por alguna razón el servidor no está corriendo, leer el mensaje mostrado en pantalla al intentar correr el comando `httpd`. También se puede revisar el registro de errores del servidor, si se desea información adicional. Por omisión este archivo se llama `"error_log"` y está ubicado en el directorio `logs` bajo el directorio `ServerRoot`.

Deteniendo *Apache*

Al arrancar el servidor Ud. habrá notado que existen varios ejecutables (`httpd`) corriendo en su sistema.

Para detenerlos, no bastará con “matar” uno por uno estos procesos, ya que estos serán rearrancados por el "proceso padre" que los originó.

Para detener Apache hay que enviar una señal `TERM` a dicho proceso padre. El PID (número identificador) de este proceso está escrito en el archivo `httpd.pid` ubicado regularmente en el directorio `logs` bajo el directorio `ServerRoot`.

La señal `TERM` significa que se debe detener el proceso padre, indicado por su PID, junto con todos sus hijos. Todos los solicitados son cancelados y no se acepta ninguna otra.

El comando que detiene al servidor sería el siguiente:

```
kill -TERM numero_de_PID
```

Fin de la instalación

Con estas 5 sencillas actividades,
termina la instalación y configuración
del servidor Apache.

PARTE IV: Aspectos de configuración y seguridad adicionales

Finalmente solo mencionaremos aquí algunos detalles y posibilidades de restricción de acceso a un servidor Web.

- Configuración de restricciones de acceso
 - Usando autenticación de usuarios

 - Usando archivos de restricción de acceso (global y por directorio)

 - Restricción de Acceso basada en direcciones IP, nombres de *host* y Dominios:
 - Inclusión
 - Exclusión

Consideraciones de seguridad Generales

- Permisología de acceso a los archivos
- Limitaciones a los programas CGI
- Permitir a usuarios sobrescribir archivos de configuración global
- SSI (Server Side Include)
- Limitando directorios para CGI
- Protección de acceso a archivos y directorios
- Ejemplos de *Scripts* CGI inseguros

Generación de estadísticas de acceso

Es perfectamente posible que cualquier aplicación en el servidor tenga acceso a los archivos "log" del servidor Web; por lo tanto se puede extraer de estos archivos valiosa información sobre los clientes que navegan por nuestro sitio Web, sobre los archivos y documentos consultados, y sobre los errores que pueden haber ocurrido en la transmisión de documentos y la ejecución de programas CGI.

Existen muchas aplicaciones comerciales y gratuitas que realizan este trabajo. A continuación algunas direcciones útiles:

Gratuitas:

- WebLog: <http://awsd.com/scripts/weblog/index.shtml>
- access_log analyzer: <http://www-ece.engr.ucf.edu/~mav/Projects/wtools/waccess.html>
- OpenWebScope: www.openwebscope.com

Comerciales:

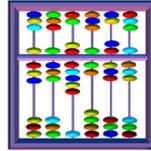
- eXTReMe Tracker: <http://www.extreme-dm.com/tracking/>
- IntelliBanner Site Analysis: <http://www.intellibanner.com/>

Nosotros mismos podemos programar aplicaciones que analicen nuestros archivos "log", utilizando el lenguaje de nuestra preferencia (Perl es especialmente adecuado para ello)

Referencias

- [1] Stephen Spainhour. *WebMaster in a Nutshell*. O'Reilly, 1996
<http://www.oreilly.com/catalog/webmaster/>
- [2] Shishir Gundavaram. *CGI Programming on the World Wide Web*. O'Reilly, 1996
<http://www.ora.com/catalog/cgi/noframes.html>
- [3] Servidor Web Apache:
<http://apache.org/>
- [4] Servidor Web NCSA:
<http://hoo.hoo.ncsa.uiuc.edu/>
- [5] Apache, the Definitive Guide:
<http://www.oreilly.com/catalog/apache/>
- [6] WebMonkey:
<http://www.webmonkey.com>

CeCalCULA



www.cecalc.ula.ve

Gracias por su asistencia y atención !

• **e-mail:** *torrens@cecalc.ula.ve*