

Introducción al Servidor Apache

Antonio Espín Herranz

Contenidos

- Historia y evolución de Apache HTTP Server.
- Arquitectura y principios de funcionamiento.
- Instalación en Windows.
- Estructura de archivos y configuración básica.

Historia y Evolución de Apache

- **Nacimiento 1995**, es la evolución de un servidor llamado: NCSA **HTTPd** (httpd es el nombre del servicio de Apache).
 - **NCSA** → National Center for Supercomputing Applications.
- El nombre proviene:
 - De la tribu nativa americana
 - Y viene también de las palabras (a patchy server → un servidor parcheado).

Evolución

- El equipo de desarrollo se llamó: **Apache Group** que luego se convirtió en **Apache Software Foundation**.
- **Código abierto y modularidad:**
 - Diseño de software libre.
 - *Arquitectura modular, se pueden añadir funcionalidades mediante módulos.*
- **Multiplataforma:** Windows, Linux y Mac

Impacto en la Web

- Líder desde **1996**: rápidamente se convirtió en el servidor Web más utilizado del mundo.
- En **2009**: superó los **100 millones** de sitios Web alojados.
- Competencia:
 - **Nginx**, LiteSpeed y servidores en la nube como Amazon o Google.

Arquitectura

- **Estructura modular:**

- Está compuesto por módulos que se pueden activar o desactivar según las necesidades del Servidor.

- **Módulos base:**

- El núcleo (**core**) y las directivas más comunes: (**mpm_common**)

- Módulos de procesamiento múltiple (**MPM**):

- Controlan como maneja Apache las conexiones concurrentes.
 - **prefork**: utiliza procesos separados sin hilos.
 - **worker**: combina procesos e hilos para mejor rendimiento
 - **event**: mejora la gestión de conexiones persistentes (**Keep-Alive**)
 - **mpm_winnt**: optimizado para Windows NT

Arquitectura

- Módulos adicionales:
 - Autenticación: **mod_auth**, **mod_auth_dbm**
 - Redirección de URLs (**mod_alias**)
 - Control de Acceso (**mod_access**)
 - Ejecución de scripts CGI (**mod_actions**)
 - Reescritura de URLs (**mod_rewrite**)
 - Seguridad SSL/TLS (**mod_ssl**)

Funcionamiento

- **Inicio:**
 - Apache se inicia y carga los módulos configurados
- **Escucha:**
 - El MPM se encarga de escuchar en los puertos definidos (por defecto el 80 para HTTP y 443 para HTTPS)
- **Gestión de peticiones:**
 - Cuando llega una petición, el MPM asigna un proceso / hilo para atender dicha solicitud.
- **Procesamiento:**
 - Los módulos interpretan la solicitud, aplican reglas, ejecutan scripts y generan respuestas.
- **Respuesta:**
 - Enviar la respuesta al cliente.

Instalación de Apache

- Enlace a la página de Apache: <https://www.apachelounge.com/download/>
- Descarga la versión de Apache para Windows:
 - Por ejemplo, httpd-2.4.57-win64-VS17.zip
- Extraer el contenido en C:\Apache24
- Abre C:\Apache24\conf\httpd.conf y verifica:
 - ServerRoot "C:/Apache24"
 - Listen 80
 - DocumentRoot "C:/Apache24/htdocs"

```
cd C:\Apache24\bin  
httpd.exe -k install  
httpd.exe -k start
```

<http://localhost>

It Works

Estructura de Archivos en Windows

- bin
- cgi-bin
- conf
- error
- htdocs
- icons
- include
- lib
- logs
- manual
- modules
- ABOUT_APACHE.txt
- CHANGES.txt
- INSTALL.txt
- LICENSE.txt
- NOTICE.txt
- README.txt

- En otros sistemas operativos puede no estar tan centralizado las subcarpetas de Apache.
- Las carpetas principales:

Carpeta	Descripción
bin/	Contiene los ejecutables principales (<code>httpd.exe</code> , <code>apachectl.exe</code>) y herramientas de línea de comandos.
conf/	Carpeta de configuración. Aquí está el archivo principal <code>httpd.conf</code> y otros como <code>extra/</code> para configuraciones adicionales.
htdocs/	Directorio raíz del sitio web. Es donde colocas tus archivos HTML, PHP, etc.
logs/	Archivos de registro: <code>access.log</code> y <code>error.log</code> .
modules/	Módulos dinámicos (<code>.so</code>) que extienden las funcionalidades de Apache.
cgi-bin/	Carpeta para scripts CGI (opcional, según configuración).

Estructura de Archivos en Windows II

- **htdocs:** es la carpeta de publicación de Apache.
 - Si publicamos varios sitios se organizarán en carpetas.
 - Una opción mejor es configurar **hosts virtuales** para poder almacenar el proyecto Web en otra ubicación.
- **manual:**
 - Contiene un fichero de texto con la URL de documentación:
 - <https://httpd.apache.org/docs/2.4/>

Estructura de Archivos en Windows III

- Las carpetas **include** y **lib** están relacionadas con el desarrollo personalizado de módulos en C / C++
- La carpeta **include** son ficheros de cabecera con declaraciones de funciones, clases, etc en C / C++
- La carpeta **lib** recopila las bibliotecas compiladas (.lib) que se vinculan en el proceso de compilación con los fuentes en C / C++
 - Las bibliotecas nos permiten interactuar con el núcleo de Apache

Estructura de Archivos en Windows IV

- Carpetas: **error y logs**

- **Logs:**

- Ubicación típica en Windows: C:\Apache24\logs\
 - Propósito: Almacena los archivos de registro (logs) generados por Apache.
 - **Archivos comunes:**
 - **access.log:** Registra cada solicitud HTTP recibida (IP, URL, método, código de respuesta).
 - **error.log:** Registra errores del servidor, fallos de configuración, problemas de permisos, etc.

- **Utilidad:**

- Diagnóstico de problemas.
 - Auditoría de tráfico.
 - Seguridad (detección de accesos sospechosos).

Estructura de Archivos en Windows V

- Carpeta **error**:
 - Esta carpeta no es estándar en todas las instalaciones de Apache.
 - Si existe, puede tener alguno de estos propósitos:
 - Archivos HTML personalizados para mostrar mensajes de error (como 404, 403, 500).
 - Páginas de error traducidas o adaptadas para distintos idiomas.
 - Recursos visuales (imágenes, estilos) usados en las páginas de error.

Personalizar páginas de error

- ***ErrorDocument 404 /error/404.html***
- ***ErrorDocument 500 /error/500.html***
- La configuración se ubica en:
 - En el archivo principal de configuración:
 - **C:\Apache24\conf\httpd.conf**
 - En un archivo adicional de configuración:
 - **C:\Apache24\conf\extra\httpd-vhosts.conf**
 - Si estamos gestionando varios sitios cada uno puede tener sus propias páginas de error.
 - En el archivo .htaccess
 - **C:\Apache24\htdocs\.htaccess**

Estructura de Archivos en Windows VI

- Carpeta: **cgi-bin**
 - Permite la ejecución de scripts ejecutables con extensiones cgi, pl, pero hoy en día se ha sustituido por otras tecnologías como PHP, Python etc.
- Carpeta: **conf**
 - La carpeta que contiene los archivos de configuración del Servidor.

Estructura de Archivos en Windows VII

Elemento	Descripción
<code>httpd.conf</code>	Archivo principal de configuración. Define cómo funciona Apache: puertos, módulos, rutas, permisos, etc.
<code>extra/</code>	Subcarpeta con configuraciones adicionales, como <i>Virtual Hosts</i> , SSL, autoindex, etc.
<code>mime.types</code>	Asocia extensiones de archivo con tipos MIME (por ejemplo, <code>.jpg</code> → <code>image/jpeg</code>).
<code>magic</code>	Ayuda a identificar tipos de archivo por contenido, no solo por extensión.
<code>charset.conv</code>	Tabla de conversión de caracteres (usado en configuraciones de codificación).

Configuración en httpd.conf

- Puerto de escucha (Listen 80)
- DocumentRoot (directorio raíz del sitio web)
- Carga de módulos (LoadModule)
- Permisos de acceso (<Directory>)
- Alias y redirecciones
- Páginas de error personalizadas
- Activación de CGI, SSL, Virtual hosts, etc.

Bloques de configuración dentro de httpd.conf

Bloque	¿Qué hace?
<code><Directory></code>	Aplica configuraciones a un directorio del sistema de archivos.
<code><VirtualHost></code>	Define configuraciones específicas para un dominio o subdominio.
<code><Location></code>	Aplica reglas a una URL específica (no al sistema de archivos).
<code><Files></code>	Aplica reglas a archivos específicos, sin importar su ubicación.
<code><IfModule></code>	Ejecuta directivas solo si el módulo está cargado.
<code><If></code>	Condicionales basadas en expresiones (Apache 2.4+).
<code><DirectoryMatch></code>	Igual que <code><Directory></code> , pero con expresiones regulares.
<code><LocationMatch></code>	Igual que <code><Location></code> , pero con expresiones regulares.
<code><FilesMatch></code>	Igual que <code><Files></code> , pero con expresiones regulares.

Configuración dentro de httpd.conf

- Usa <**Directory**> para controlar acceso y comportamiento según la ruta física.
- Usa <**Location**> para controlar comportamiento según la URL solicitada.
- Usa <**VirtualHost**> para configurar sitios web diferentes en el mismo servidor.
- Usa <**Files**> para proteger o modificar el comportamiento de archivos concretos.
- Usa <**IfModule**> para evitar errores si un módulo no está cargado.

Ejemplos

- **<location>** Basado en URL
 - <Location "/admin">
 - Require ip 192.168.1.0/24 *# Ante una petición /admin solo admite IP en el rango*
 - </Location> *# de 192.168.1.0 a 192.168.1.24*
- **<files>** para un archivo específico
 - <Files "config.php">
 - Require all denied *# Bloquea a todos los usuarios, sin excepciones.*
 - </Files> *# Nadie puede acceder a config.php vía web.*
- **<IfModule>** solo si un módulo está active
 - <IfModule mod_rewrite.c> *# Verificar si el modulo mod_rewrite está cargado*
 - RewriteEngine On
 - RewriteRule ...
 - </IfModule>

Subcarpeta extra

- Esta carpeta permite modularizar la configuración. Algunos archivos útiles:
- **httpd-vhosts.conf**: Define múltiples sitios web (Virtual Hosts).
- **httpd-ssl.conf**: Configura HTTPS con certificados SSL/TLS.
- **httpd-autoindex.conf**: Personaliza la apariencia de listados de directorios.
- **httpd-multilang-errordoc.conf**: Páginas de error multilingües.

Configuración básica

- **Activar funcionalidades específicas cargando módulos:**
 - LoadModule rewrite_module modules/mod_rewrite.so
 - LoadModule ssl_module modules/mod_ssl.so
 - LoadModule cgi_module modules/mod_cgi.so
- **mod_rewrite:** para redirecciones y URLs amigables.
- **mod_ssl:** para habilitar HTTPS.
- **mod_cgi:** para ejecutar scripts CGI.

Configuración básica

- **Definir el puerto de escucha (útil si el puerto 80 ya está ocupado)**

- listen 8080

- Establecer el **DocumentRoot**:

- Define el directorio donde se encuentran tus archivos web.

```
DocumentRoot "C:/Apache24/htdocs/misitio"
```

```
<Directory "C:/Apache24/htdocs/misitio">
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

- **AllowOverride All** permite usar archivos **.htaccess**.
- **Options** controla el comportamiento del directorio (por ejemplo, si se listan los archivos).

Configuración básica

- **Configurar Virtual Hosts**

- En `conf/extra/httpd-vhosts.conf`, puedes definir múltiples sitios:
 - `<VirtualHost *:80>`
 - `ServerName www.ejemplo.local`
 - `DocumentRoot "C:/Apache24/htdocs/ejemplo"`
 - `</VirtualHost>`
- Agregar el dominio al archivo **hosts** de Windows:
 - `127.0.0.1 www.ejemplo.local`

Configuración básica

- **Páginas de error personalizadas:**
 - Puedes definir páginas para errores comunes:
 - ErrorDocument 404 /error/404.html
 - ErrorDocument 403 /error/403.html
- **Archivos .htaccess** (si están permitidos):
 - Permiten configurar reglas específicas por carpeta:
 - RewriteEngine On
 - RewriteRule ^inicio\$ index.html [L]

Configuración básica

- **Revisar el PATH de Windows.**
 - Tienen que funcionar los comandos de la carpeta **bin** de Apache.
- Verificar y reiniciar:
 - Verificar que no haya errores:
 - **httpd.exe -t**
 - Reinicia Apache desde el servicio o ejecutando:
 - **httpd.exe -k restart**
 - Ver que módulos tenemos activados en Apache
 - **httpd -M**

Configuración básica

- Para mantener la configuración organizada, puedes activar solo los archivos que necesites desde **httpd.conf** usando directivas como:
 - **Include conf/extra/httpd-vhosts.conf**
 - **Include conf/extra/httpd-ssl.conf**