



Servizo Web

(versión 2.1)

Índice

1. Servidor web	2
2. URL e URI	2
3. Protocolo HTTP	3
3.1. Mensaxes HTTP	4
4. HTTPS	5
4.1. Certificados	5
5. Servidores Web virtuais	6
6. Apache	6
6.1. Directivas	7
6.2. Portos	7
6.3. Módulos	7
7. NGINX	8
8. Máis información	8

Introdución

En 1989, Tim Berners Lee e Robert Caillou, investigadores do CERN propuxeron a creación da World Wide Wide e colaboraron no seu desenvolvemento mediante a súa participación na elaboración de diversos estándares ou especificacións.

O funcionamento de Internet fundaméntase nunha rede de documentos enlazados a través de hipervínculos, hiperenlaces ou ligazóns. Cada documento equivale a un nodo da rede. Unha rede de servidores son os encargados de aloxar e distribuír eses documentos para que cheguen aos usuarios.

Programas clientes chamados navegadores reciben a información do servidor web e mostran os documentos coas ligazóns de forma que cando se pulsa co rato sobre eles se abren as páxinas web coas que estean enlazados.

Os documentos web deben estar escritos nunha linguaxe de marcas que os describe, sendo a máis utilizada HTML. Nos inicios os hiperenlaces enlazaban con outros documentos web polo que a estes documentos se lles chamou hipertextos e fóronselles engadindo novos elementos como imaxes, sons, animacións,... Actualmente os hiperenlaces ou ligazóns non só chaman a outros documentos web senón que poden enlazar con calquera outro tipo de arquivo ou con recursos doutros servizos como enderezos de correo electrónico.

1. Servidor web

Un servidor web almacena arquivos dun sitio web nun directorio e permite que os clientes accedan a el para consultar eses ficheiros. O servidor web e o cliente teñen asignadas cadansúa IP que os identifica na rede. A comunicación establécese seguindo o protocolo HTTPS que é un dos protocolos da capa de aplicación do modelo OSI e do TCP/IP.

Cada vez que un cliente desexa obter información dun servidor envía unha petición de datos. O navegador é o cliente web que realiza a solicitude, especificando na petición a IP (ou o nome DNS) do equipo que ten instalado o servidor web. Nalgúns casos engádese ruta e nome da páxina web solicitada. O servidor busca no directorio o arquivo correspondente e envíao de volta. O cliente recibe a resposta que se mostra no navegador e o usuario visualiza unha páxina web. Lembra que unha páxina web non é máis ca un ficheiro *.html* e un sitio web é unha carpeta que pode conter imaxes, vídeos,..

Na xestión do hardware dun servidor web cómpre ter en conta o número de clientes que poden acceder simultaneamente, o número de sitios que ofrece, o sistema operativo no que se traballa e as necesidades de memoria, unidades de disco e o ancho de banda da conexión. Dende o punto de vista do software os paquetes para instalar e configurar un servidor web máis utilizados son Apache e Nginx.

2. URL e URI

Unha URL ou mellor chamada, unha **URI** é unha secuencia de caracteres que permite localizar un recurso nunha rede calquera, tamén en Internet, e ten o seguinte formato:

PROTOCOLO://USUARIO:CONTRASINAL@MÁQUINA:PORTO/RUTA-RECURSO

Analicemos o que significa cada elemento fixado polo estándar empregado para nomear recursos en internet:

Protocolo: // indica o protocolo mediante o que se accede ao recurso. Para páxinas web o protocolo era http pero hoxe en día cómpre utilizar a versión segura que é HTTPS. Poden especificarse outros protocolos como FTP.

usuario:contrasinal poderíase usar cando no servidor se controla o acceso ao recurso mediante un nome de usuario e un contrasinal indicados na URL. Case nunca se utilizan e desaconséllase a súa utilización por razóns de seguridade. Hai outras formas de controlar o acceso máis seguras.

máquina vén sendo o nome DNS ou o enderezo IP do equipo onde se atopa o servidor que proporciona o recurso que se trata de localizar. Unha URL para unha páxina web consta como mínimo do protocolo e da máquina, como por exemplo *https://edu.xunta.gal*

:porto indica o porto do servidor co que hai que conectar. Se non se especifica asúmese un que estea estipulado para o protocolo usado. Por exemplo, para HTTP asígnase o porto 80 e para HTTPS será 443. Se o servidor ten aberto outro porto, digamos o 8888 diríamos que está escoitando por el. Neste caso no queda outra que especificalo.

ruta indica onde ten que localizar o recurso o servidor e o nome do recurso. Se non se cubre esta parte, o servidor asume un nome de recurso que teña na súa configuración (é habitual que sexa *index.html*).

outros datos van despois da ruta e serven para enviar datos ao servidor para que os procese. Cando solicitamos unha busca en Google na URL aparece a información que solicitamos:

<https://www.google.es/search?client=opera&q=i-rochiño&sourceid=opera&ie=UTF-8&oe=UTF-8>

3. Protocolo HTTP

O protocolo *HTTP HyperText Transfer Protocol* é un protocolo de transferencia de hipertexto que segue o modelo cliente/servidor na capa de aplicación do modelo OSI. Establece as normas para o intercambio de información contida nas páxinas web. Leva desenvolvéndose dende 1990 por W3C e IETF.

As características máis relevantes do funcionamento deste protocolo son as seguintes:

- Un servidor HTTP utiliza de oficio o porto 80, aínda que pode usar outros.
- Para establecer unha comunicación HTTP necesítase crear previamente unha conexión TCP.
- A comunicación realízase mediante mensaxes de petición e de resposta.
- Cada elemento dunha páxina web (documento, imaxes, vídeos, etc.), transfírese independentemente coa súa mensaxe de petición e a súa mensaxe de resposta. É a razón pola que o texto sempre aparece ao pedirmos unha páxina antes que as imaxes.

Cando nun navegador web escribimos a URL dunha páxina web prodúcese unha serie de transaccións de rede entre cliente e servidor:

1. O cliente solicita ao servidor DNS que resolva o nome do servidor web usado na URL.

2. Obtida a IP do servidor, establécese unha conexión TCP estable polo porto indicado do servidor, que permanece á escoita.
3. Establecida a conexión TCP, faise a petición do documento web ao servidor e este devolve a información solicitada cunha mensaxe de resposta.
4. Se o documento web inclúe elementos adicionais como imaxes, hai un proceso de envío petición/resposta por cada imaxe.
5. Nunha mesma conexión TCP pode producirse máis dunha transacción.

3.1. Mensaxes HTTP

As mensaxes que se intercambian cliente e servidor web son textos codificados en ASCII. As de petición son as que envía o cliente ao servidor e as de resposta son as que devolve o servidor contestando unha petición.

Na súa estrutura constan de catro campos como se ve na figura 1:

1. Liña de petición ou de resposta (segundo sexa o caso): contén a información principal. Na mensaxe de petición da figura 1 será: `GET /home.html HTTP/1.1`
2. Encabezados: conteñen información adicional sobre opcións relativas á mensaxe, (unha liña de encabezado por cada opción que se especifique). Na figura 1 na mensaxe de resposta un dos encabezados será: `Content-Type: text/html`
3. Unha liña co texto `--blank line ---` que sempre está presente.
4. Un corpo de mensaxe: este campo é opcional. Nas mensaxes de resposta úsase para enviar o contido do recurso solicitado.

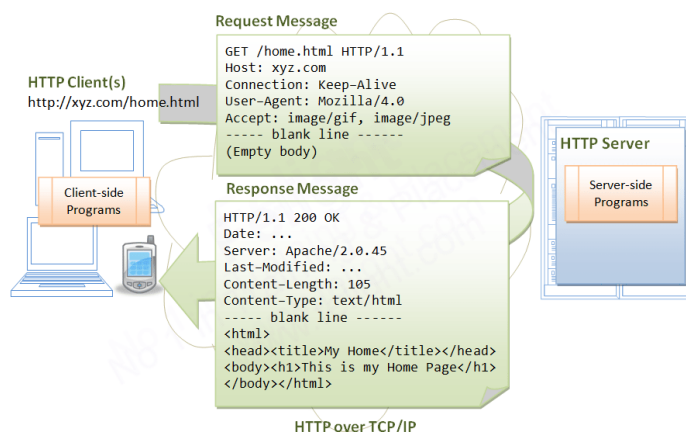


Figura 1: Mensaxes HTTP

Para obtermos información sobre o estado da conexión existen os **códigos de resposta**. Son números de tres cifras que indican se unha petición se recibiu e se atendeu correctamente ou se houbo algunha incidencia na súa tramitación.

- 1xx** Resposta informativa: Indícalle ao cliente que se recibiu a petición. Trátase dunha resposta provisional e hai que realizar unha nova transacción HTTP para que se poida obter a resposta definitiva. Non se aconsella que os servidores envíen estas respostas.

- 2xx** Petición correcta: Indica que a petición recibida anteriormente foi recibida, aceptada e procesada sen problemas no servidor.
- 3xx** Reenderezo: Indícaselle ao cliente HTTP que ten que realizar algunha acción adicional para que se poida resolver completamente a petición que se realizou.
- 4xx** Erro do cliente: Recibiuse unha petición cunha sintaxe errónea ou non se puido procesar a petición. O erro 404 é moi frecuente.
- 5xx** Erro de servidor: Produciuse un erro no servidor que lle impediu atender e procesar a petición.

Se nos fixamos na figura 1 o código de resposta está na primeira liña da mensaxe de resposta e resulta ser `HTTP/1.1 200 OK` ou sexa a transacción resolveuse correctamente.

4. HTTPS

O protocolo HTTPS *HTTP Secure* transmite as mensaxes HTTP cifradas para garantir a identidade do servidor ante os clientes. Utiliza en realidade dous protocolos: o protocolo HTTP combinado co protocolo SSL *Secure Sockets Layer* ou mellor co protocolo TLS *Transport Layer Security*.

Un cliente web para conectarse cun servidor HTTPS ten que indicar na URI explicitamente o protocolo `https://` en troques de `http://` e enviará unha mensaxe polo porto de escoita que teña asignado o servidor. Un servidor HTTPS adoita ter aberto o porto 443 como porto de escoita.

Nunha conexión HTTPS garántese:

- A confidencialidade xa que as mensaxes transmitidas están cifradas.
- A integridade posto que se unha mensaxe se modifica accidentalmente ou intencionadamente, o receptor detecta que se produciu esa modificación.
- A autenticación, podendo asegurar a identidade do servidor e/ou do cliente mediante o uso de certificados dixitais.

4.1. Certificados

Un servidor web HTTPS debe dispoñer dun certificado para poder establecer conexións seguras cos clientes. Un certificado dixital ou electrónico é un ficheiro emitido por unha entidade **Autoridade de Certificación (AC)** que serve para demostrar a identidade nunha conexión.

Para establecer conexións seguras HTTPS, o propio servidor web ten que ter instalado un certificado para enviar as mensaxes asinadas coa chave privada que se xenerou cando o certificado. A primeira vez que un cliente se conecta cun servidor HTTPS para pedir unha páxina web recibe a chave pública que lle manda o servidor. Pide confirmación sobre se consideramos válido o certificado e se desexamos usar esa chave pública para conectarnos. Esta chave servirá para descifrar as mensaxes que recibe e verificar a súa identidade. Como é habitual nos casos de uso de criptografía de chave pública-privada, o cliente cifrará as mensaxes coa chave pública do servidor para que o servidor as descifre coa chave privada que só ten el.

Unha vez que aceptamos, as seguintes conexións do cliente co servidor verificarán de forma automática a identidade do servidor e non pedirán autorización.

Se traballamos na fase de produción, é dicir se o noso servidor web está operativo en internet, os certificados deben ser emitidos por unha autoridade certificadora recoñecida. Os navegadores web normalmente dispoñen dunha lista de autoridades certificadoras que lles permite establecer conexións automaticamente (sen ter que confirmar) no caso de que os certificados estean emitidos por algunha autoridade coñecida. Empresas como VeriSign ou CyberTrust poden certificar legalmente.

A Fábrica Nacional de Moneda y Timbre é unha entidade que depende do goberno e actúa coma autoridade certificadora en España. Pódese solicitar un certificado dixital e a meirande parte dos navegadores a recoñecen como entidade certificadora válida. Co uso do DNI electrónico esta tecnoloxía está desenvolvéndose e incorporándose aos nosos fogares.

Os certificados podemos fabricalos nós mesmos para facer probas: son os chamados **certificados autoasinaados**. O procedemento para ter un certificado para un sitio web en fase de desenvolvemento, ou sexa que non sae a internet, e só se usa para facer probas pode ser usando OpenSSL. Outras opcións como Let's Encrypt son tamén válidas.

5. Servidores Web virtuais

Os chamados *virtual host* permiten que unha mesma máquina poida traballar como servidor para varios sitios web. Usando servidores virtuais poderemos facer que se administre tanto `www1.sitioXX.com` coma `www2.sitioXX.com` sen ter que instalar un servidor para cada sitio web.

Pódense crear tres tipos de servidores virtuais:

baseados en nomes: o servidor web pode aloxar varios sitios web pertencentes a distintos dominios usando para todos eles o mesmo enderezo e o mesmo porto de escoita.

baseados en enderezos IP: por cada enderezo IP do equipo servidor web terase un servidor web virtual. Para ter esta configuración necesitaríase ter no equipo varios adaptadores de rede ou crear varias interfaces virtuais (p.e. `eth0`, `eth1` e `eth2`) e dar unha IP a cada unha.

baseados en portos: o servidor usará un porto de escoita por cada servidor virtual. Isto realízase por exemplo cando un servidor ten dous virtuais e un escoita no porto 80 e o outro escoita no porto 8080 ou no 443 se desexamos usar o protocolo *HTTPS*.

6. Apache

O proxecto de servidor HTTP Apache é un software de desenvolvemento colaborativo que ten como obxectivo a creación dunha implementación dun servidor HTTP con acceso libre ao seu código fonte. Manteñen este proxecto un grupo de voluntarios que se comunican a través de Internet e planifican e desenvolven o servidor e a súa documentación de uso.

Para traballar cun servidor web con `apache2` precisamos dous elementos principais: un ficheiro de configuración e un directorio que sexa o noso sitio web coas súas páxinas web, imaxes, vídeos,...

Normalmente en Debian o ficheiro de configuración principal de `apache2` está en `/etc/apache2/apache2.conf` e o ficheiro de configuración dos virtual host están almacenados no directorio `/etc/apache2/sites-available`. Noutras plataformas pode estar en `.htaccess`

Apache2 vén de fábrica configurado cun único host virtual que se pode modificar e usar como modelo para outros que se desexen configurar. Para modificalo cópiase ou edítase o ficheiro `/etc/apache2/sites-available/000-default.conf`. Este host virtual funcionará coma sitio de probas ou cando a URI que introduce o usuario non encaixa con ningún outro host virtual.

6.1. Directivas

A sintaxe de apache2 baséase no uso de **directivas** que veñen sendo unha especie de comandos nun ficheiro que permiten personalizar o servidor web. A súa sintaxe é simple, abonda con indicar o nome da directiva e o valor que se lle quere dar entre aqueles que son válidos. Existen centos de directivas pero afortunadamente para unha configuración sinxela dun servidor chegan con a penas: **VirtualHost**, **ServerName**, **ServerAdmin**, **DocumentRoot**, **Listen**.

Para monitorizar o funcionamento apache2 crea ficheiros `.log` que se gardarán onde indiquen as directivas **ErrorLog**, **CustomLog** que teñen que estar na mesma definición do virtual host.

Cómpre ter moito tino coa sintaxe dos ficheiros de configuración se os manipulamos dende un terminal. Como normas xerais lembrar:

- Os ficheiros de configuración conteñen unha directiva por liña. Os argumentos das directivas sepáranse por espazos en branco. Se un argumento, por exemplo unha ristra de caracteres contén espazos, deberase encerrar entre comiñas dobres.
- As directivas nos ficheiros de configuración non son sensibles a maiúsculas, pero os argumentos con frecuencia si que o son. Cómpre ter tino con como escribimos.
- As liñas que comezan co carácter almohadilla `#` considéranse comentarios e son ignorados.
- Os espazos en branco que preceden a calquera directiva son ignorados, e por iso, pódense indentar as liñas para maior claridade. Tampouso se teñen en conta as liñas completas en branco.

Pódese comprobar a sintaxe dos ficheiros de configuración de apache2 antes de iniciar o servidor mediante a execución do comando `apachectl configtest`

6.2. Portos

Cando se usa o protocolo HTTP ábrese algún porto do equipo local e espera por peticións.

No ficheiro `/etc/apache2/ports.conf` está a directiva **Listen** que acepta peticións so en algún porto ou combinacións de enderezos e portos. Se unicamente especificamos unha directiva **Listen** o servidor escoita peticións nese porto en todos os seus enderezos IP. Se se especifica unha IP xunto co porto, atenderá peticións nese porto e interface. Pódense poñer múltiples directivas **Listen** xunto con portos e enderezos IP. O servidor atenderá peticións en calquera deles.

6.3. Módulos

Apache ten unha grande cantidade de módulos que lle permiten aumentar a súa funcionalidade. O caso máis coñecido sería habilitar e deshabilitar un módulo para facer o servidor máis seguro ou para controlar o inicio de sesión.

7. NGINX

Unha das eivas que lle custa xestionar a Apache é como traballar cando o tráfico é masivo. Paseniño está facendo un oco un servidor web chamado Nginx que está moi ben considerado polo administradores web pola alta concurrencia, escalabilidade e a eficiencia dos recursos que utiliza. Fronte a Apache e a súa comunidade de desenvolvedores, Nginx pertence a unha empresa.

Comparado con Apache que ten un fío asociado a cada conexión, o que se chama arquitectura de procesos, o rendemento de Nginx fundaméntase na xestión de miles de conexións que se tartan de xeito simultáneo. A arquitectura de Nginx utiliza eventos asíncronos que permiten administrar varias solicitudes de clientes en paralelo o que lle permite ser bo no balanceo de carga.

8. Máis información

- [Documentación oficial de Apache](#)
- [Directivas de Apache](#)
- [Servizo Web - Manuais Informática - IES San Clemente](#)
- [Instalar un servizo web con Apache \(digitalocean.com\)](#)
- [Traballar con virtual hosts](#)
- [Comparativa Apache vs Nginx \(ionos\)](#)

Material elaborado para traballar na asignatura *Servizos en Rede* de SMR conxuntamente con:

