

TEMA 2.1: WWW Y PROTOCOLO HTTP

WWW es un servicio de distribución de información que da acceso a millones de recursos electrónicos y aplicaciones distribuidos en servidores por todo internet. Se identifican y localizan por direcciones URIs o URLs, y se conectan entre sí a través de hipervínculos.

Fue desarrollado por el W3C (World Wide Web Consortium), quien controla el desarrollo de WWW además de desarrollar los estándares web XHTML, CSS y XML.

1. PÁGINAS, SITIOS Y APLICACIONES WEB

Una **página web** es un documento hipermedia o conjunto de información electrónica relacionada (texto, audio, imágenes, vídeo, etc.) que normalmente contiene hiperenlaces a otras páginas web o recursos. Está escrita en lenguaje que son interpretados y/o ejecutados por los navegadores (XHTML, CSS, JavaScript, Flash, etc.). Su contenido puede ser estático o dinámico.

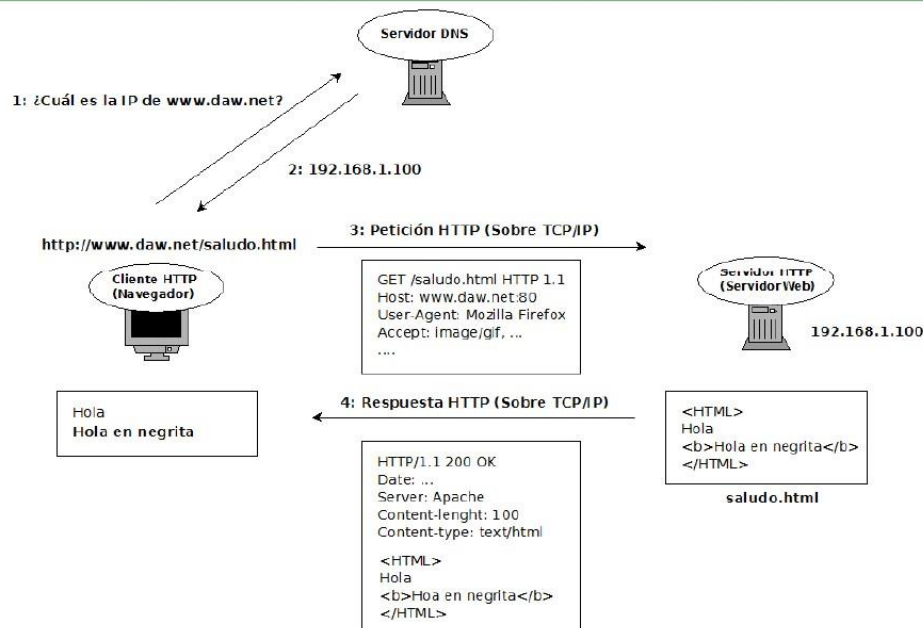
Un **sitio web** es un conjunto de páginas web relacionadas y accesibles a partir de un mismo nombre de dominio DNS. El conjunto de sitios web de Internet constituyen la WWW.

Una **aplicación web** es una aplicación distribuida cuya interfaz de usuario es accesible desde un navegador web. El usuario interactúa con un navegador que accede a los servicios y recursos que ofrece un servidor web (E.G. un buscador, una tienda electrónica, un cliente de correo web, etc.). Ejemplos: Gmail, Ebay, Facebook, etc.

2. COMPONENTES Y FUNCIONAMIENTO

- Recursos (documentos, vídeos, imágenes, etc.) conectados por hipervínculos.
- Clientes web (clientes HTTP o navegadores).
- Servidores web (o servidores HTTP).
- Nombres y direcciones (URIs y URLs).
- Protocolo HTTP.
- Tecnologías web (XHTML, CSS, XML, Ajax, XPath, etc.).

Componentes y funcionamiento



3. CLIENTES WEB (NAVEGADORES)

Son programas con los que interactúa el usuario. Se usan URIs o URLs para acceder a recursos disponibles en la red. Pueden existir clientes de diferentes protocolos, pero su función principal es ejercer como clientes HTTP.

Mantienen una memoria caché de las direcciones a las que han accedido en un historial, de los recursos procesados, de las contraseñas introducidas por el usuario en las aplicaciones, etc. Ejemplos: Internet Explorer, Firefox, Chrome, Opera, etc.

Estos clientes permiten ampliar su funcionalidad con la instalación de plantillas, idiomas, extensiones y complementos. Los navegadores, en determinados aspectos, no cumplen completamente los estándares.

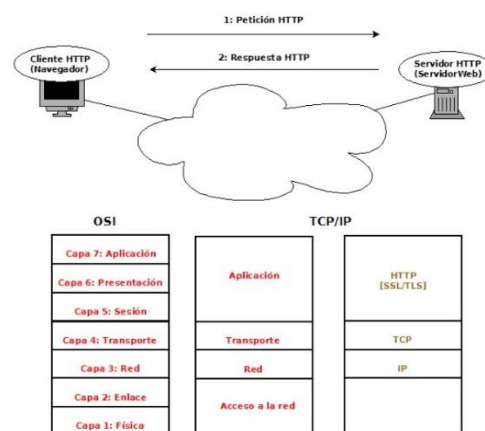
4. SERVIDORES WEB O HTTP

Son los encargados de atender las peticiones HTTP, procesando e interpretando el código escrito en diferentes lenguajes para enviar a los clientes los recursos solicitados. Ofrecen múltiples opciones de configuración y su arquitectura modular permite ampliar o quitar funcionalidades fácilmente. Las peticiones HTTP se escuchan en el puerto 80/TCP. Ejemplos: Apache HTTP Server, Microsoft IIS, NGINX, Lighttpd, etc.

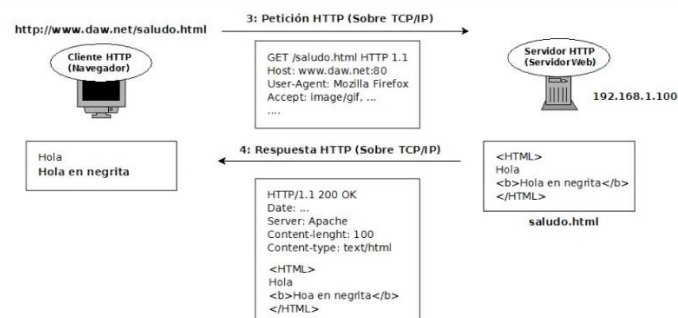
5. PROTOCOLO HTTP

HTTP (Hyper Text Transfer Protocol) es el protocolo de comunicación en la Web. Este protocolo es sin estado, es decir, trata cada petición como una transacción independiente que no tiene relación con cualquier solicitud anterior, de modo que la comunicación se compone de pares independientes de solicitud y respuesta. Utiliza TCP como protocolo de transporte.

Protocolo HTTP



Protocolo HTTP Funcionamiento



- 1- El usuario introduce una URI o URL en la barra de direcciones del navegador o hace clic sobre un hipervínculo.
- 2- El navegador analiza la URL y establece una conexión TCP con el servidor web.
- 3- El navegador envía un mensaje HTTP de petición que depende de la URI o URL.
- 4- El servidor envía un mensaje de respuesta que depende de la petición enviada y de estado del servidor.
- 5- Se cierra la conexión TCP.

PROTOCOLO HTTP: Mensajes HTTP

Los mensajes HTTP son líneas en texto plano que siguen el formato ASCII y son de dos tipos: mensajes de petición y mensajes de respuesta.

- Mensajes de petición.

Están formados por tres partes:

- Línea inicial de petición.
- Líneas de cabecera.
- Cuerpo del mensaje (opcional). Parámetros o ficheros a enviar al servidor.

```
GET / HTTP/1.1
Host: www.daw.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20100101 Firefox/12.0
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
Connection: keep-alive
```



```
GET /saludo.html HTTP/1.1
Host: www.daw.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20100101 Firefox/12.0
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
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Date: Fri, 27 Apr 2012 07:40:28 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Fri, 27 Apr 2012 07:40:10 GMT
ETag: "20016-20-4bea436cd2425"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 48
Keep-Alive: timeout=5, max=100
Connection: keep-alive
Content-Type: text/html

.....(.....IT(MU(I.QPTT....H....J. ...]
```

- Mensajes de respuesta.

Están formados por tres partes:

- Una línea inicial de respuesta (línea de estado).
 - Versión HTTP, código de estado y texto explicativo.
- Líneas de cabecera.
- Cuerpo del mensaje (opcional). Determinado por el tipo de recurso solicitado.

```
HTTP/1.1 200 OK
Date: Fri, 27 Apr 2012 07:47:19 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Fri, 27 Apr 2012 07:47:06 GMT
ETag: "2096e-34-4bea44fa4264c"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 71
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

.....(....I.O....O.sI-N,*....WHIUH,...LNL...K-V(QM...*.4...|
```



```
GET /saludo.html HTTP/1.1
Host: www.daw.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20100101 Firefox/12.0
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
Connection: keep-alive

HTTP/1.1 200 OK
Date: Fri, 27 Apr 2012 07:40:28 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Fri, 27 Apr 2012 07:40:10 GMT
ETag: "20016-20-4bea436cd2425"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 48
Keep-Alive: timeout=5, max=100
Connection: keep-alive
Content-Type: text/html

.....(.....IT(MU(I,QPTT....H....J). ...|
```

Métodos de petición: tipos.

Especifican la operación que quiere realizar el cliente en el servidor: GET, POST, OPTIONS, HEAD, PUT, DELETE, TRACE, CONNECT, PATH.

- **Get.**

Métodos de petición. GET

```
GET /datosGET.php?nombre=Aida&edad=33&Enviar=Enviar HTTP/1.1
Host: www.daw.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20100101 Firefox/12.0
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
Connection: keep-alive
Referer: http://www.daw.net/formularioGET.html
```

- **Post.**

Métodos de petición. POST

```
POST /datosPOST.php HTTP/1.1
Host: www.daw.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20100101 Firefox/12.0
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
Connection: keep-alive
Referer: http://www.daw.net/formularioPOST.html
Content-Type: application/x-www-form-urlencoded
Content-Length: 36

nombre=Antonio&edad=40&Enviar=Enviar
```

PROTOCOLO HTTP: Cabeceras.

Son pares de nombre/valor que se pueden incluir en los mensajes de petición y respuesta HTTP. Definen información (metadatos) sobre:

- Los datos que se intercambian los clientes y servidores.
- Los propios clientes y servidores.
- Sobre la propia transferencia de información.

Existen múltiples tipos de cabeceras:

- Generales: (Date, Transfer-Encoding...).
- De petición o de cliente (User-Agent, Accept...).
- De respuesta o de servidor (Server, Age...).
- De entidad (Content-Encoding, Content-Language, Content-Type...).

PROTOCOLO HTTP: Códigos de estado y error.

Son códigos que envían los servidores en las respuestas HTTP e informan a los clientes de cómo ha sido procesada la petición. Se acompañan de un texto descriptivo. Estos códigos se componen de tres dígitos que se clasifican en función del primero:

- 100 -199: Informativo, informacional.
- 200 – 299: Éxito, Successful.
- 300 – 399: Redirección, Redirection.
- 400 – 499: Errores del cliente, Client error.
- 500 – 599: Errores en el servidor, Server error.



PROTOCOLO HTTP: Cookies.

Son fragmentos de información que envía un servidor web en una respuesta HTTP y es almacenada por el navegador. El navegador puede enviar la cookie en solicitudes posteriores al mismo navegador.

► Cabeceras: *Cookies* y *Set-Cookie*

```
GET /cookies.html HTTP/1.1
Host: www.daw.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20100101 Firefox/12.0
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
Connection: keep-alive
Cookie: veces=3; visitante=Alicia
If-Modified-Since: Fri, 27 Apr 2012 08:16:39 GMT
If-None-Match: "20019-975-4bea4b951dbeb"
Cache-Control: max-age=0
```

