Introducción al WWW. Índice

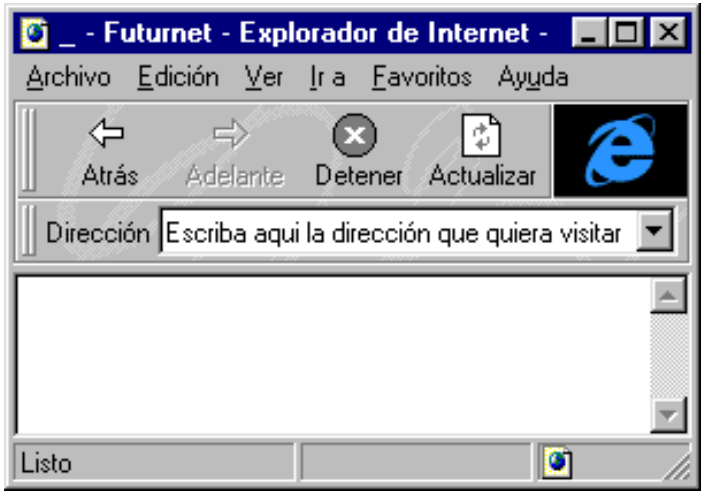
1. Introducción.
    a. Historia.
    b. Arquitectura.
    c. Funcionamiento
2. Protocolo HTTP.
    d. Formato.
    e. Peticiones.
    f. Respuestas.
    g. Ejemplos.
    h. HTTPS.
    i. SSL.
    j. Cookies.
3. URL.
4. HTML.

Introducción

Distribución de la información: publicación, consulta y servicios.

HISTORIA:

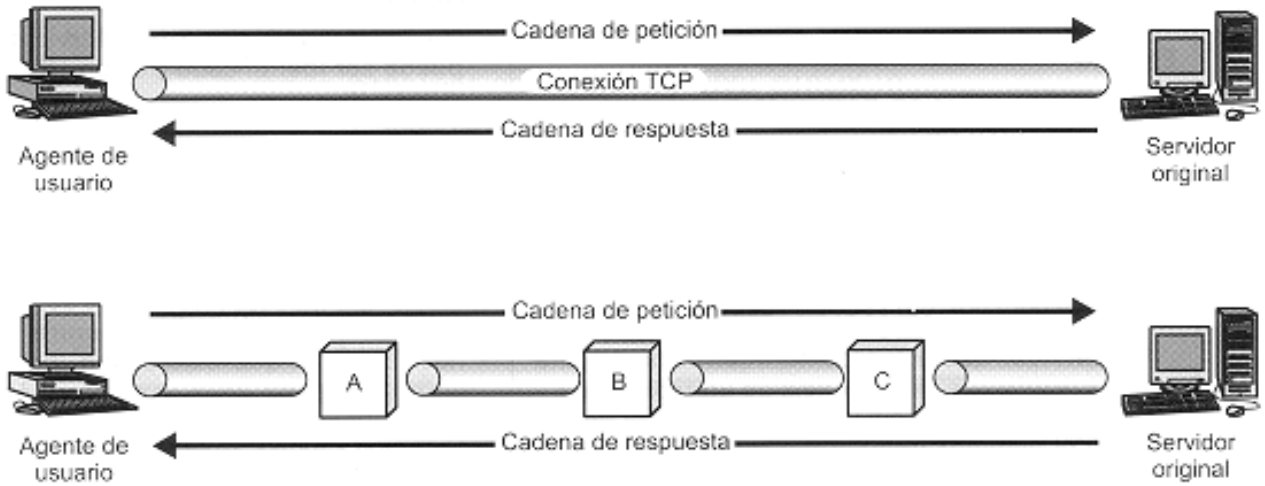
- 1945. Enlaces entre documentos en microfichas.
- 1960. Edición de hipertexto (texto interactivo y fotos) con dispositivo señalador.
- 1980. En el CERN se escribe un programa con enlaces entre notas.
- 1989. Propuesta de sistema de información hipertexto.
- 1990. Prototipo de browser hipertexto.
- 1993. Mosaic: distribución del cliente de hipertexto: Web browser.
- 1994. WWW Consortium. Desarrollo de protocolos y estándares.



Arquitectura WWW

- Arquitectura distribuida cliente-servidor.
- Dos programas:
  - Cliente: envia peticiones de documentos a mostrar al servidor.
  - Servidor: almacena y responde a las peticiones transfiriendo documentos hipertexto.

- Se utiliza el puerto 80 TCP. El servicio se denomina http y el daemon httpd.
- Ambos programas se pueden ejecutar en cualquier sistema, incluso en el mismo.
- El funcionamiento es independiente del sistema en el que se ejecute.



Hipertexto en Cliente WEB (I)

Existen programas clientes que inician las peticiones a los servidores, interpreta el texto y comandos de formato y muestra la página de forma correcta: Internet Explorer, Netscape, Mozilla, Opera, etc..

Para que funcione ha sido necesario:

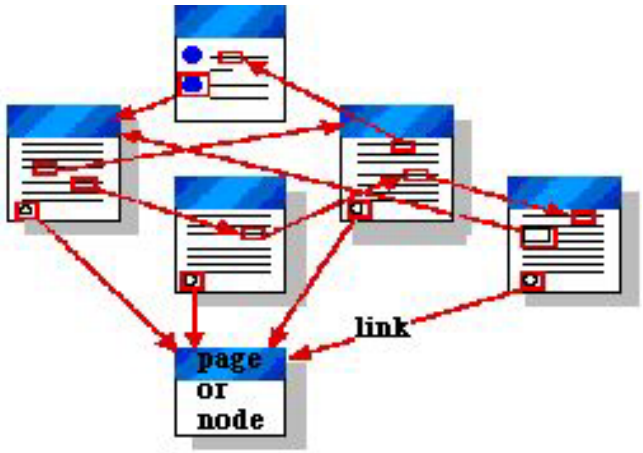
- Protocolo que permita saltos hipertexto. **HTTP** (Hipertext Transfer Protocol).
- Lenguaje para representar contenido hipertexto y formato de página. **HTLM** (Hipertext Markup Language).
- Codificar las instrucciones para los saltos hipertexto. **URL** (Uniform Resource Locator).
- Aplicaciones cliente para todo tipo de plataformas para acceder de modo uniforme a través de diversos protocolos y formatos (texto, animaciones, gráficos, etc).

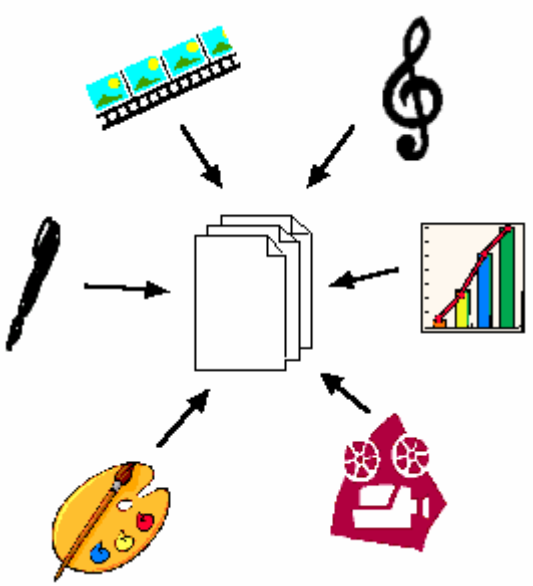
Las paginas contienen texto, hipertexto, iconos, mapas, fotografias, cada uno de los cuales puede vincularse a otra página.

Hipertexto en Cliente WEB (II)

- Mezclar las páginas con otros medios como audio, video y programas se conoce como sistema **hipermedia**.

- Los navegadores suelen usar el disco duro local como caché de las páginas visitadas para de esa forma acelerar el funcionamiento.





Servidor WWW

Los servidores usan el protocolo HTTP para comunicarse con los clientes. Se realiza a través del puerto 80 TCP. HTTP es un protocolo "**sin estados**".

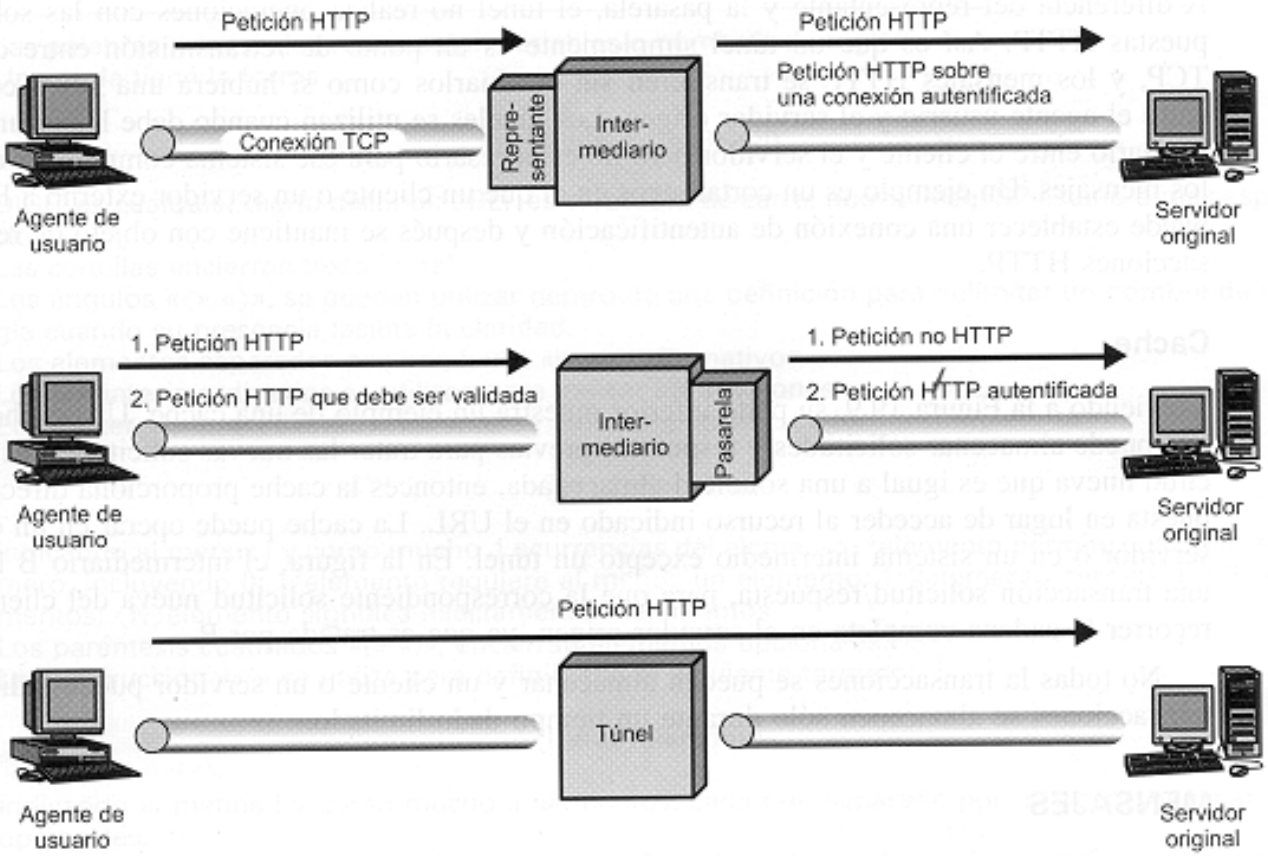
- Permite servir ficheros reales o **virtuales** (generados por programas) con **formularios** y zonas activas.
- Es flexible en cuanto a los **formatos** que puede tratar a través de listas de prioridades que envia el cliente.
- La **respuesta** incluye información de estado, código de éxito/error y mensaje tipo MIME que contiene información del servidor, y la respuesta misma.

Cuando el usuario hace clic en algún enlace: http://www.uhu.es/index.html

- El visualizador determina la URL y solicita al DNS la dirección IP.
- Establece conexión TCP con el puerto 80 en la IP que el DNS ha indicado.
- Emite un comando GET /index.html
- El servidor envia el archivo index.html y se libera la conexión.
- El visualizador formatea y muestra el texto del archivo.
- El visualizador trae y presenta todas las imágenes incluidas.

Tipos de Conexión (I)

- **Directamente** con el servidor origen. El agente de usuario (browser) es el cliente que inicia la solicitud, actuando de parte de un usuario final. El **servidor origen** es el servidor donde reside el recurso de interés.



Tipos de Conexión (II)

- Empleando uno o más **sistemas intermedios** entre agente usuario y servidor origen:

  - **Representante** (***proxy***). Un representante actúa en nombre del cliente y presenta solicitudes de otros clientes a un servidor:
    - De seguridad. Cliente y servidor están separados por un cortafuegos.
    - Diferentes versiones http. Representante traduce formatos.

  - **Pasarela** (***gateway***). Actúa como el servidor original en nombre de otros servidores que no son capaces de comunicarse directamente:
    - Intermediario de seguridad. Clientes externos se deben autentificar con la pasarela para usar el servicio.
    - Servidor no http. La pasarela proporciona capacidad de conexión con otros protocolos.

  - **Túnel** (***tunnel***). Simplemente retransmite las peticiones y respuestas. Cortafuegos.

Protocolo HTTP

- Intercambio de mensajes entre clientes y servidores WWW.

- Caracteristicas:

  - **Ligereza**: intercambios discretos que no sobrecargan la red y permiten saltos rápidos.
  - **Generalidad**: puede usarse para transferir cualquier tipo de datos (emplea estándar MIME)
  - **Extensibilidad**: contempla tipos de transacciones entre clientes y servidores.

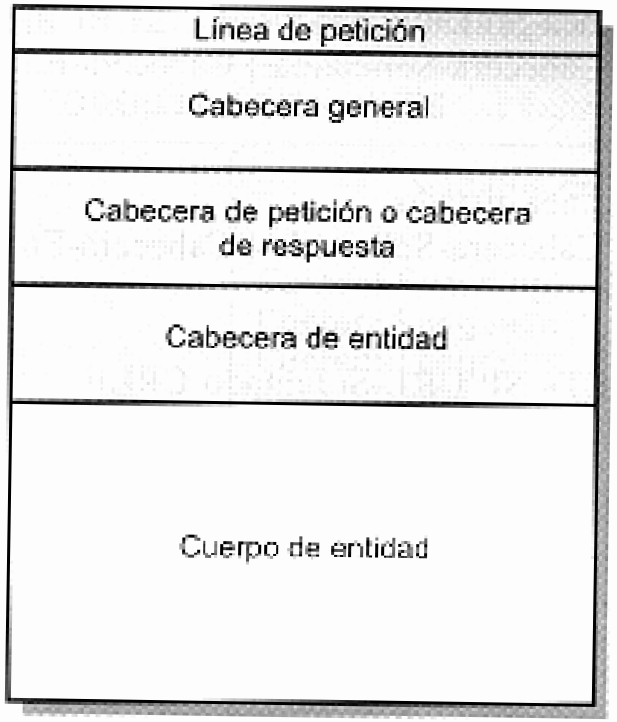
- El protocolo http **no mantiene el estado**. Reconoce dos tipos de solicitudes:

  - **Solicitud Sencilla**. Línea GET que nombra la página deseada, sin versión de protocolo, la respuesta es la página sin información adicional.
  - **Solicitud completa**, contiene GET, el nombre de la página, la versión del protocolo y varias líneas de cabecera.

Formato de los Mensajes (I)

- Los mensajes están compuestos de líneas de texto:
  - Línea de petición o respuesta.
  - Líneas de cabecera.
  - Línea de cabecera de la entidad.
  - Cuerpo del mensaje o entidad.

- La **línea inicial** en las peticiones especifica que recurso que solicita:
  - Nombre de método:
    - GET. Petición simple.
    - HEAD. Petición sin transferir cuerpo.
    - PUT. Escribir recurso en servidor.
    - POST. Añadir información a recurso.
    - DELETE. Borrar recurso.
    - LINK. Estable enclaces entre recursos.
    - UNLINK. Elimina enlaces entre recursos.
  - Camino.
  - Versión de http.



Formato de los Mensajes (II)

- Las respuestas incluyen:
  - Versión de http.
  - Código númerico de estado, del tipo:
    - 1xx: Mensaje informativo.
    - 2xx: Resultado exitoso.
    - 3xx: Redireccion del cliente a otra URL.
    - 4xx: Error en lado cliente.
    - 5xx: Error en lado servidor.
  - Código de estado textual.
- Las líneas de cabecera tienen el mismo formato que las del correo electrónico (MIME).

Ejemplos HTTP

- Get de petición:

```
GET /~jgb/test.html HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.07 [en] (X11; I; Linux 2.2.15 i586; Nav) ...
Host: gsyc.escet.urjc.es
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, i ...
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

- Get de respuesta:

```
HTTP/1.1 200 OK
Date: Tue, 23 Jan 2001 12:44:27 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU
Last-Modified: Tue, 23 Jan 2001 12:39:45 GMT
ETag: "19e89f-22-3a6d7b91"
Accept-Ranges: bytes
Content-Length: 34
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>\nEsto es una prueba\n</html>
```

- POST de petición:

```
POST /comments.pl HTTP/1.0
From: jgb@gsyc.escet.urjc.es
User-Agent: MegaNavigator/0.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 18

section=all&rank=10
```

HTTPS

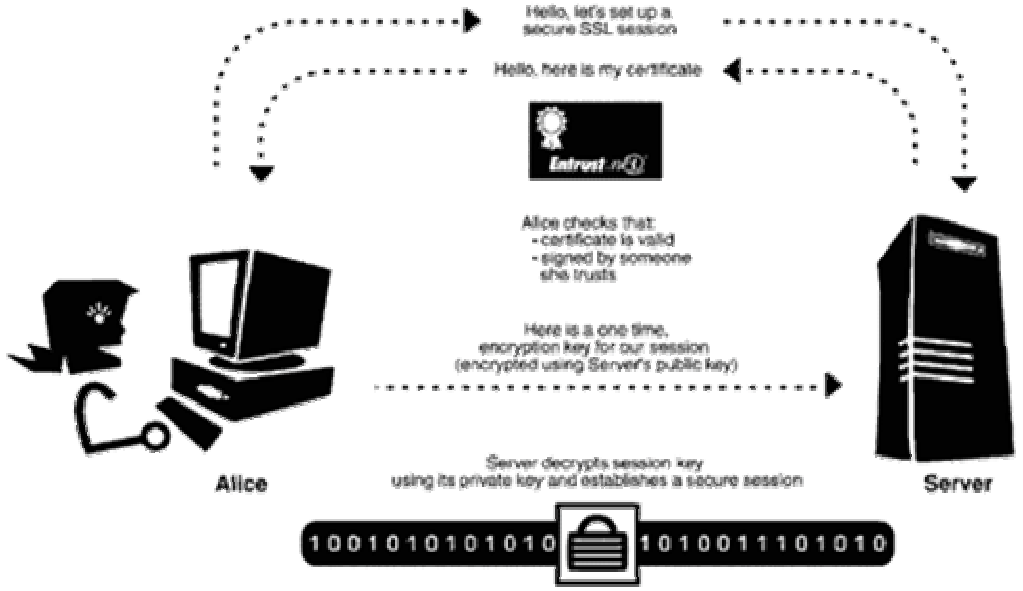
- HTTPS (**Secure HTTP**) es un procolo similar al http que permite enviar además de forma segura mensajes individuales.
- HTTPS emplea un servidor http a través del puerto 443 TCP. Necesita otros mecanismos (certificados, etc) para ofrecer un nivel de seguridad razonable. Las URLs comienzan con https://.
- El navegador pide y comprueba los certificados antes de pasar a encriptar la información. Y lo indica al usuario.

- El creador de contenidos o programador no tiene que realizar nada especial para que las páginas funcionen. Solo es necesario un certificado de seguridad válido y aceptado por el cliente.

SSL

- SSL (**Secure Socket Layer**) es un protocolo desarrollado por Netscape para transmitir datos de forma privada. La conexión TCP está cifrada, de forma que no puede ser interceptada.

- SSL se emplea para obtener información confidencial del usuario, como números de tarjetas de creditos o transacciones autenticadas.



Cookies

- Fichero que **envia el servidor** para registras las actividades del usuario en un sitio web.
- Cuando el navegador pide **ficheros adicionales**, se devuelve al servidor la información del cookie.
- Se emplea para **personalizar** los sitios web. Asocian un estado a un conjunto de transacciones.
- Original de Netscape, propuesta como estándar a través de RFC 2109.
- Cuando el servidor **manda un cookie** se usa la cabecera “Set-Cookie” :
  - Nombre del cookie.
  - Fecha de caducidad del cookie.
  - Dominio, camino, para distinguir cookies.
  - Marca de seguridad, se envia solo por https.
- Cuando el cliente **pide URL**, buscara si tiene que enviar alguna. Las envia todas en una única cabecera.

URL

- A cada página se le asigna una URL (Uniform Resource Locator).
- Relacionan un servicio, recurso y puerto con un servidor.
- URL tiene varias partes: **protocolo://servidor:puerto/recurso#seccion**
  - **Protocolo**: indica el protocolo empleado para acceder al recurso, generalmente será http, puede ser: telnet, ftp, https, etc.
  - **Servidor**: nombre o dirección IP del servidor.
  - **Puerto**: puerto remoto en el servidor al que se efectua la conexión, si no se indica se emplea el conocido para el servicio.
  - **Recurso**: camino y nombre de la página web o elemento para el que realizamos la petición si hay alguno.
  - **Marca**: una posición determinada en una página.
- Ejemplo: http://www.uhu.es, http://10.0.0.1:631/, telnet://platero.uhu.es .
- **URI**: Uniform Resource Identifier. URL generalizada.

HTML (I)

- **Lenguaje de Marcaje de Hipertexto**. Formateo de documentos mediante comandos.
- **Separa la presentación del contenido**. Estándar para formato de documentos.
- Es un lenguaje muy sencillo. Emplea **"tags" o marcas** para formateo o enlaces a otros recursos. Se colocan entre simbolos: <xxx>
- Los tags se emplean para:
  - Estilos del texto: <H1> … </H1> <B> …</B>
  - Titulos de documentos: <TITLE> … </TITLE>
  - Parrafos. <p>
  - Listas. … …
  - Enlaces. …
  - Formularios. <INPUT TYPE="text" NAME="…">

HTML (II)

- Una página web consiste en la cabecera y el cuerpo encerrado entre etiquetas: <HTML> </HTML>

```
<HTML>
<HEAD>
    <TITLE>Editeur web site</TITLE>
    <LINK REL=stylesheet TYPE="text/css" HREF="Styles.css">
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1"
    <META NAME="GENERATOR" CONTENT="Mozilla/4.03 [en] (Win95; I) [Netscape]
    <META NAME="DESCRIPTION" CONTENT="Editeur web site">
    <META NAME="KEYWORDS" CONTENT="java, perl, html, php, python">
    <SCRIPT language="JavaScript">
    <!--
    function animateAnchor() {
        var el=event.srcElement;
            if ("A"==el.tagName) {  // Initialize effect if none specified
                if (null==el.effect) el.effect = "highlight"
                // Swap effect with the class name.
                temp = el.effect;
                el.effect = el.className;
                el.className = temp;
                }
          }
    // Initialize event handlers
    document.onmouseover = animateAnchor;
    document.onmouseout = animateAnchor;
```