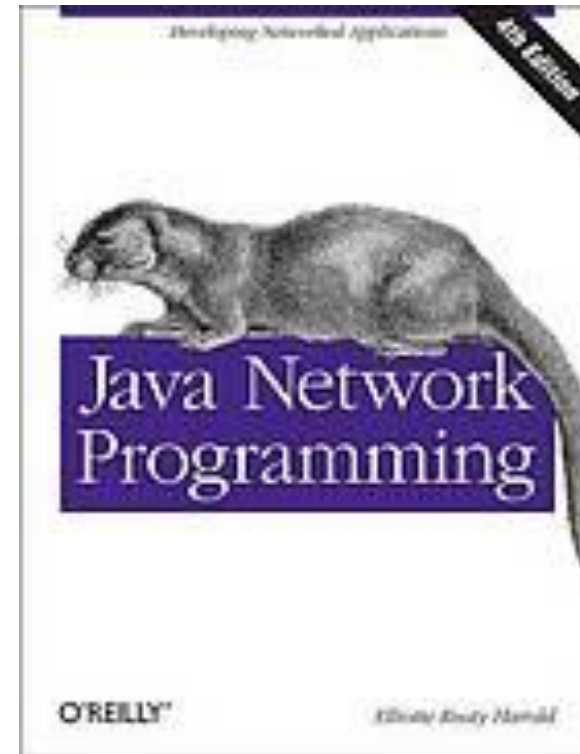
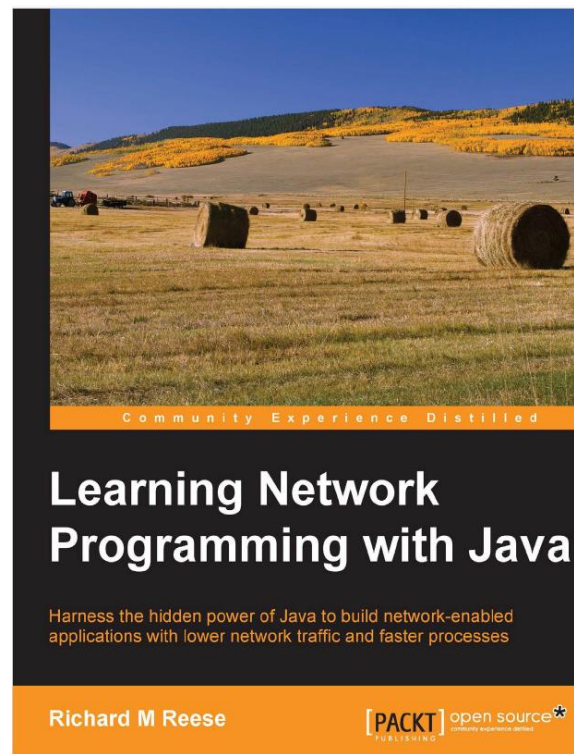


REDES DE COMPUTADORES Y LABORATORIO

Christian Camilo Urcuqui López, MSc



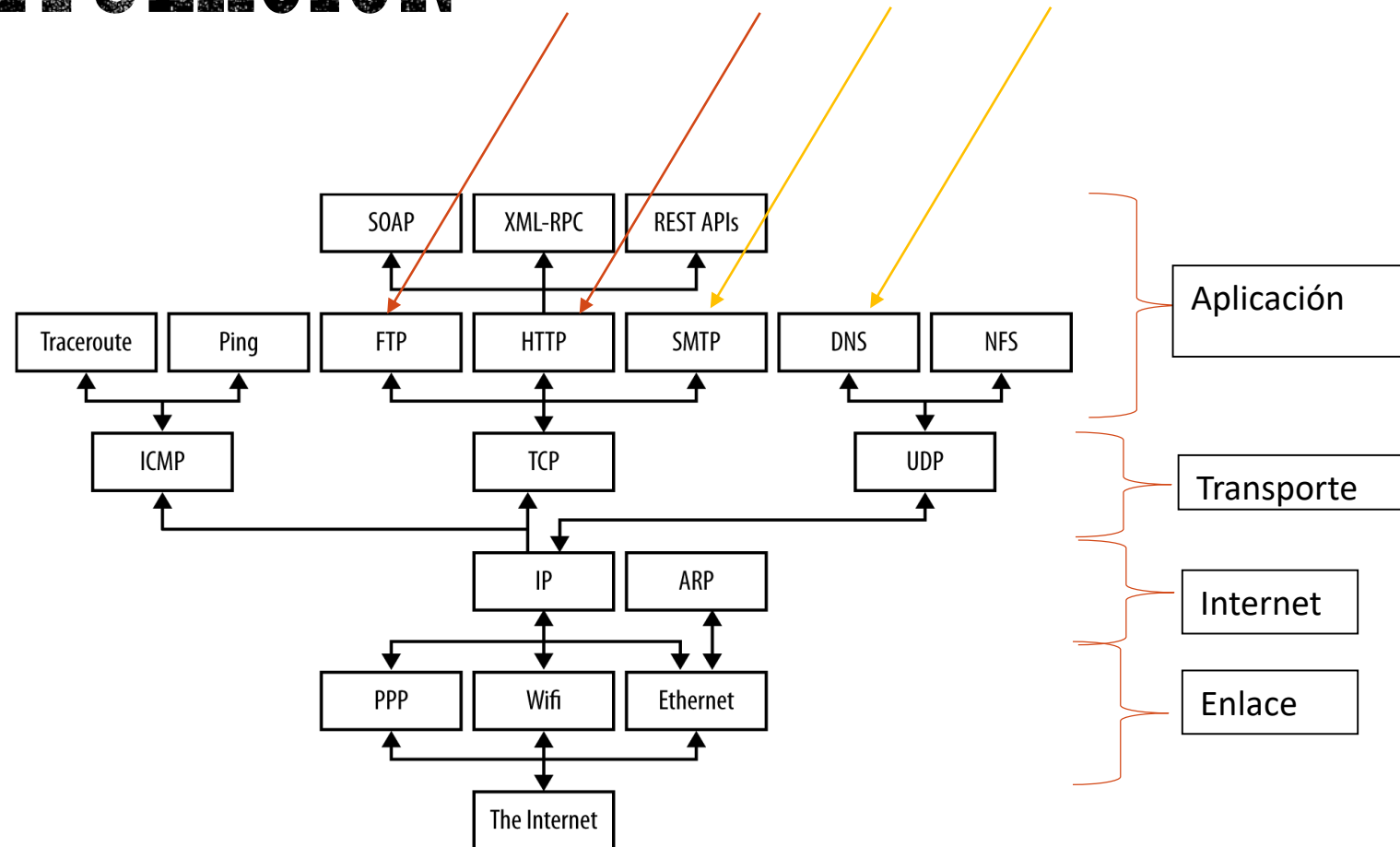
BIBLIOGRAFÍA



COMPETENCIAS

- Describir la capa de aplicación.
 - Describir World Wide Web
 - Páginas estáticas
 - Páginas dinámicas

RECAPITULACIÓN

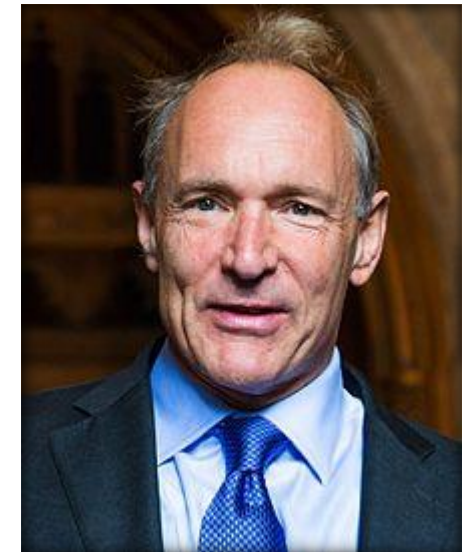


LA CAPA DE APLICACIÓN

WORLD WIDE WEB

- Es un marco arquitectónico para acceder a un contenido vinculado distribuido en millones de máquinas por toda Internet.
- Conocida como WWW comenzó en 1989 en el CERN (Centro Europeo de Investigación Nuclear), fue una propuesta de red de documentos vinculados por el físico Tim Berners-Lee, su equipo y él establecieron la primera comunicación entre cliente y servidor usando el protocolo HTTP.
- En 1993 se libero el primer navegador gráfico (Mosaic).

Desarrollado por Marc Andressen, fundador de Netscape Communications Corp.



W3C

- En 1994 el CERN y el MIT firmaron un acuerdo para establecer el W3C (Consortio World Wide Web, *World Wide Web Consortium*).
- www.w3.org
- Es una organización dedicada al desarrollo web, a la estandarización de protocolos y a fomentar la interoperabilidad entre los sitios.



WWW

- **Hipertexto**, direccionamiento entre páginas, desarrollo propuesto en 1945 por el profesor Vannevar Bush.
- **Navegador**, son los encargados de obtener la página solicitada, interpretar el contenido y desplegar la página en el formato adecuado.

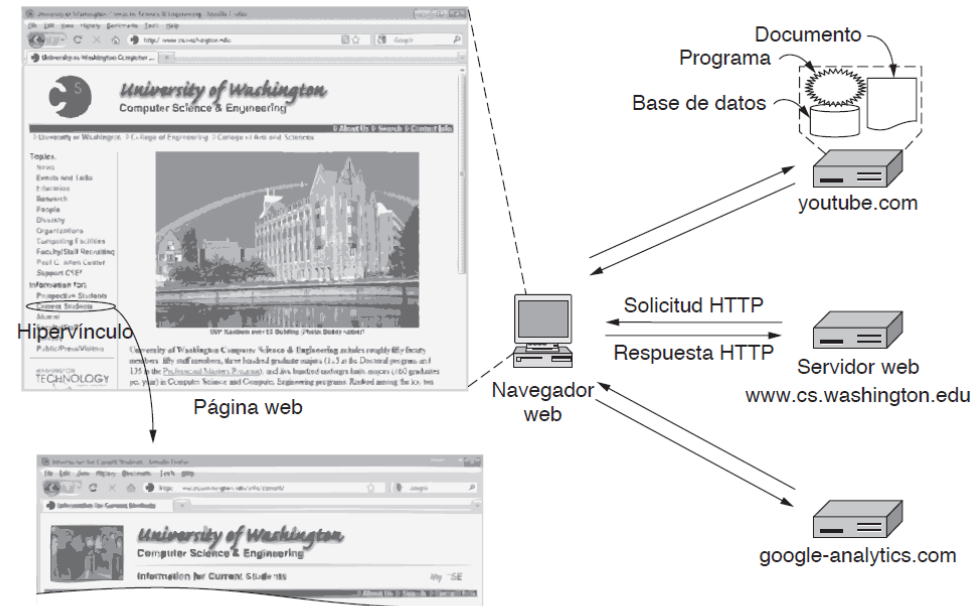


Figura 7-18. Arquitectura de la web.

HTTP (HYPERTEXT TRANSFER PROTOCOL)

- Es un protocolo de solicitud-respuesta para obtener páginas, sus datos son simples y basado en texto que se ejecuta sobre TCP.
- El contenido puede ser simplemente un documento que se lea de un disco, o el resultado de una consulta en una base de datos y la ejecución de un programa.

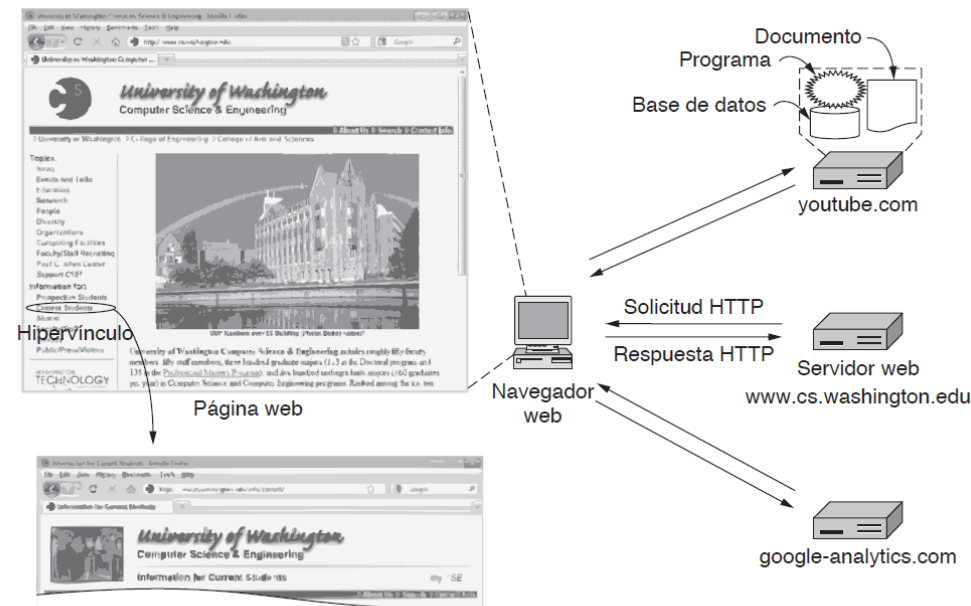


Figura 7-18. Arquitectura de la web.

EL LADO DEL CLIENTE

- ¿Que se debía tener en cuenta para desplegar una página?
 1. ¿Cómo se llama la página?
 2. ¿En dónde está ubicada?
 3. ¿Cómo se puede acceder a ella?
- 1. Para dar respuesta a las anteriores interrogantes, se propuso una solución que identifica a las páginas. A cada página se le asigna un **URL (Uniform Resource Locator)** que sirva de manera efectiva como el nombre mundial de la página.
- 2. Los URL tienen tres partes:
 1. El protocolo (conocido también como **esquema**).
 2. El nombre DNS de la máquina en la que se encuentra la página.
 3. La ruta que indica de manera única la página específica (un archivo a leer o un programa a ejecutar en la página). La ruta tiene un nombre jerárquico que modela la estructura de un directorio de archivos.

URL

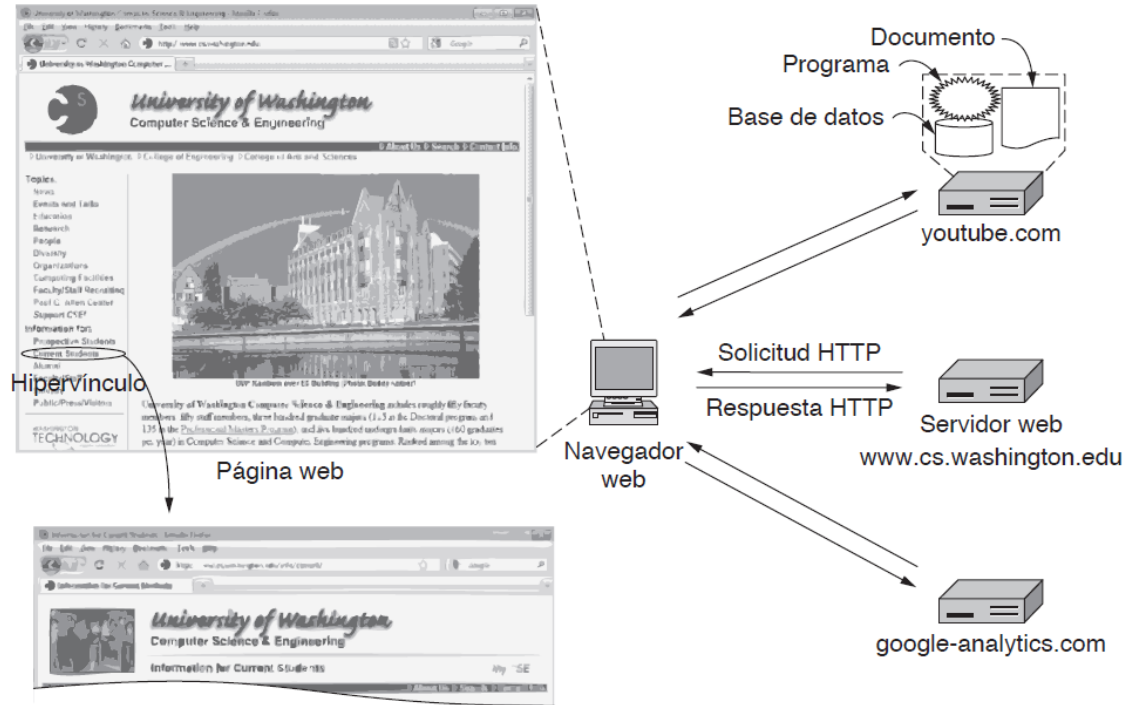


Figura 7-18. Arquitectura de la web.

La URL de la página es:

<http://www.cs.washington.edu/index.html>

- Protocolo: http
- El nombre DNS: www.cs.washington.edu
- El nombre de la ruta: index.html

PASOS

1. El navegador determina el URL (al ver lo que se seleccionó).
2. El navegador pide al DNS la dirección IP del servidor *www.cs.washington.edu*.
3. El DNS responde con 128.208.3.88.
4. El navegador realiza una conexión TCP a 128.208.3.88 en el puerto 80, el puerto conocido para el protocolo HTTP.
5. Después envía una solicitud HTTP para pedir la página */index.html*.

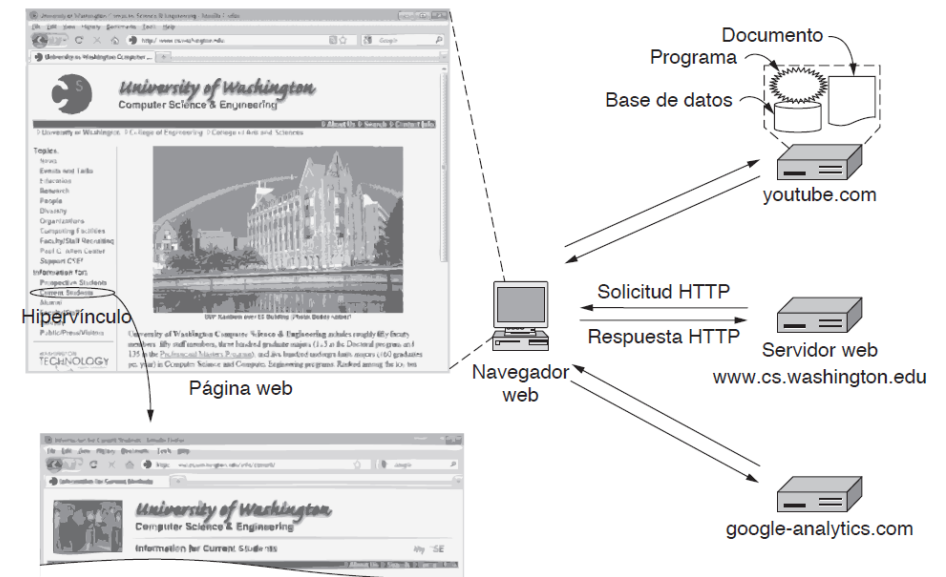


Figura 7-18. Arquitectura de la web.

PASOS

1. El servidor *www.cs.washington.edu* envía la página como una respuesta HTTP, por ejemplo, enviando el archivo */index.html*.
2. Si la página incluye los localizadores URL necesarios para desplegar en pantalla, el navegador obtiene los otros URL mediante el mismo proceso. En este caso, los URL incluyen varias imágenes incrustadas que también se obtienen de *www.cs.washington.edu*, así como un video de *youtube.com* y una secuencia de comandos (*script*) de *google-analytics.com*.
3. El navegador despliega la página */index.html* como aparece
4. Se liberan las conexiones TCP si no hay más solicitudes para los mismos servidores durante un periodo corto.

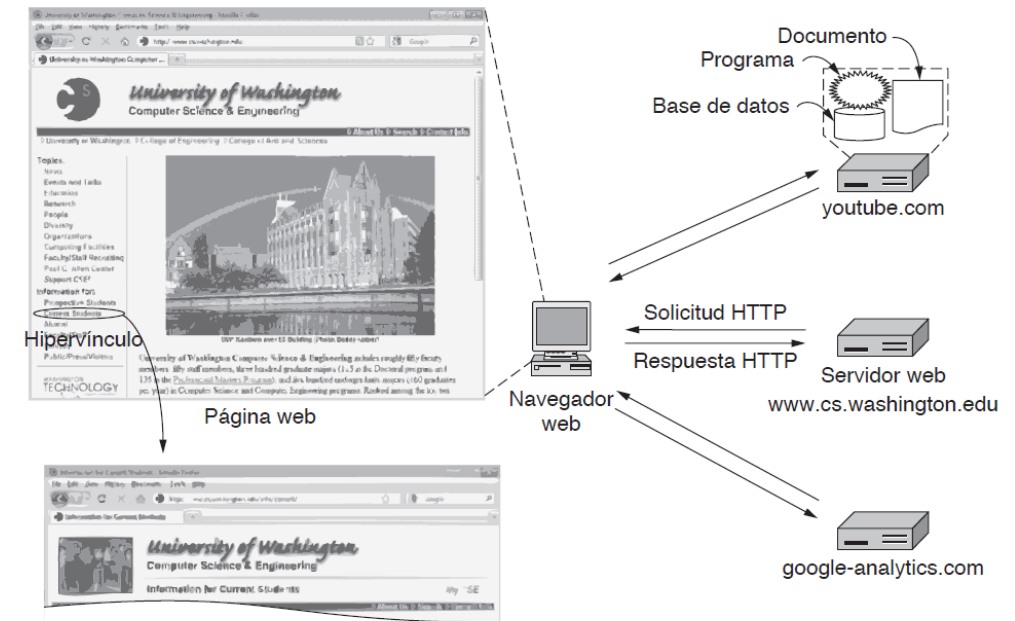


Figura 7-18. Arquitectura de la web.

Nombre	Se usa para	Ejemplo
http	Hipertexto (HTML).	http://www.ee.uwa-edu/~rob/
https	Hipertexto con seguridad.	https://www.bank.com/accounts/
ftp	FTP.	ftp://ftp.cs.vu.nl/pub/minix/README
file	Archivo local.	file:///usr/suzanne/prog.c
mailto	Enviar correo electrónico.	mailto:JohnUsuario@acm.org
rtsp	Medios de flujo continuo.	rtsp://youtube.com/montypython.mpg
sip	Llamadas multimedia.	sip:eva@adversario.com
about	Información del navegador.	about:plugins

Figura 7-19. Algunos esquemas comunes de URL.

FTP (FILE TRANSFER PROTOCOL)

- Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP.
- La web facilita el proceso para obtener archivos colocados en muchos servidores FTP en todo el mundo al proveer una interfaz simple en la que se puede hacer clic.
- Funciona en los puertos de red 20 y el 21.
- Las transferencias se realizan en texto plano, es decir, no se aplica un mecanismo de seguridad y se puede hacer un análisis del tráfico de red con la finalidad de obtener información sensible.

OTROS PROTOCOLOS . . .

- **File** es un protocolo que permite acceder a un archivo local como una página web.
- **Mailito** permite a los usuarios enviar correos electrónicos desde un navegador web. La mayoría de navegadores responderán cuando el usuario siga un vínculo *mailito* e iniciarán el agente de correo de ese usuario para escribir un mensaje.
- Los protocolos **rtsp** y **sip** son para establecer sesiones de medios de flujo continuo, y llamadas de audio y video.
- El protocolo **about** es una convención que provee información sobre el navegador. Por ejemplo, si el usuario sigue el vínculo *about:plugins*, la mayoría de navegadores mostrarán una página que lista los MIME que pueden manejar, con extensiones conocidas como complementos o *plu-gins*.

Los URL se diseñaron no sólo para permitir a los usuarios navegar en la web, sino también para ejecutar protocolos antiguos como FTP y el correo electrónico, así como los protocolos más recientes para audio y video, y para proveer un acceso conveniente a los archivos locales y la información del navegador.

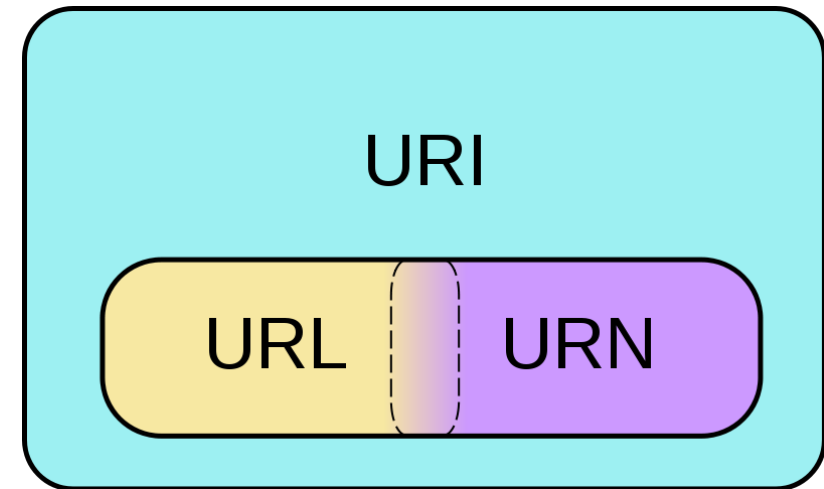
DEBILIDAD

Un URL apunta a un host específico, pero algunas veces es conveniente hacer referencia a una página sin necesidad de decir al mismo tiempo en dónde está. Por ejemplo, en las páginas con muchas referencias es conveniente tener varias copias en ubicaciones distantes y separadas, para reducir el tráfico de red.

URI (UNIFORM RESOURCE IDENTIFIERS)

- Para resolver la debilidad mencionada, los URL se generalizaron en **URI**. [RFC 3986](https://www.rfcs.org/rfcs/3986).
 - Algunos URI indican cómo localizar un recurso (estos son los URL).
 - Otros URI indican el nombre de un recurso, pero no en dónde encontrarlo (conocidos como **URN – Uniform Resource Names**).

<https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>



TIPOS DE MIME (MULTIPURPOSE INTERNET MAIL EXTENSIONS)

- Cuando un servidor devuelve una página, también devuelve cierta información adicional sobre ella. Esta información incluye el tipo MIME de la página

Tipo	Subtipos de ejemplo	Descripción
text	plain, html, xml, css	Texto en diversos formatos.
image	gif, jpeg, tiff	Imágenes.
audio	basic, mpeg, mp4	Sonidos.
video	mpeg, mp4, quicktime	Películas.
model	vrml	Modelo 3D.
application	octet-stream, pdf, javascript, zip	Datos producidos por aplicaciones.
message	http, rfc822	Mensaje encapsulado.
multipart	mixed, alternative, parallel, digest	Combinación de múltiples tipos.

Figura 7-13. Tipos de contenido MIME y subtipos de ejemplo.

- Un complemento es un módulo de código de terceros que se instala como una extensión para el navegador.
 - Algunos ejemplos comunes son los complementos para PDF, Flash y Quicktime, de modo que se puedan desplegar los documentos, además de reproducir audio y video
 - Antes de poder usar un complemento hay que instalarlo.

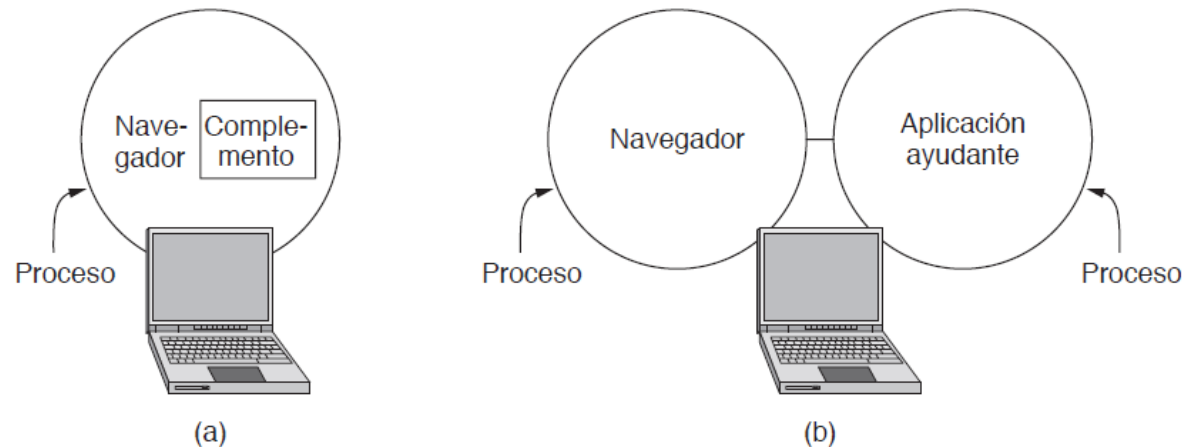


Figura 7-20. (a) Complemento del navegador. (b) Una aplicación ayudante.

- Hay un programa completo que se ejecuta en un proceso separado y tiene una interfaz que mantiene una distancia del navegador.
- Por lo general sólo acepta el nombre de un archivo de trabajo en el que se almacena el contenido, abre ese archivo y muestra el contenido en pantalla. Por ejemplo, Microsoft Word o PowerPoint. Muchas aplicaciones Utilizan el tipo MIME *application* (Por ejemplo, *application/vnd.ms-powerpoint* para archivos PowerPoint).

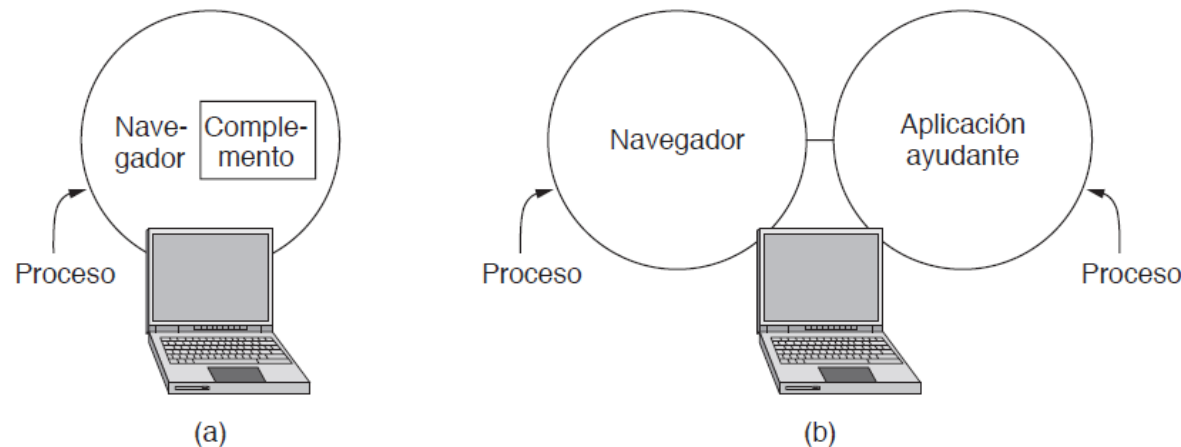


Figura 7-20. (a) Complemento del navegador. (b) Una aplicación ayudante.

- Los navegadores también puede abrir archivos locales, para ello el navegador necesita determinar el tipo de MIME del archivo. En una configuración típica, al abrir el archivo *foo.pdf* se abrirá en el navegador mediante el uso de un complemento *application/pdf*.

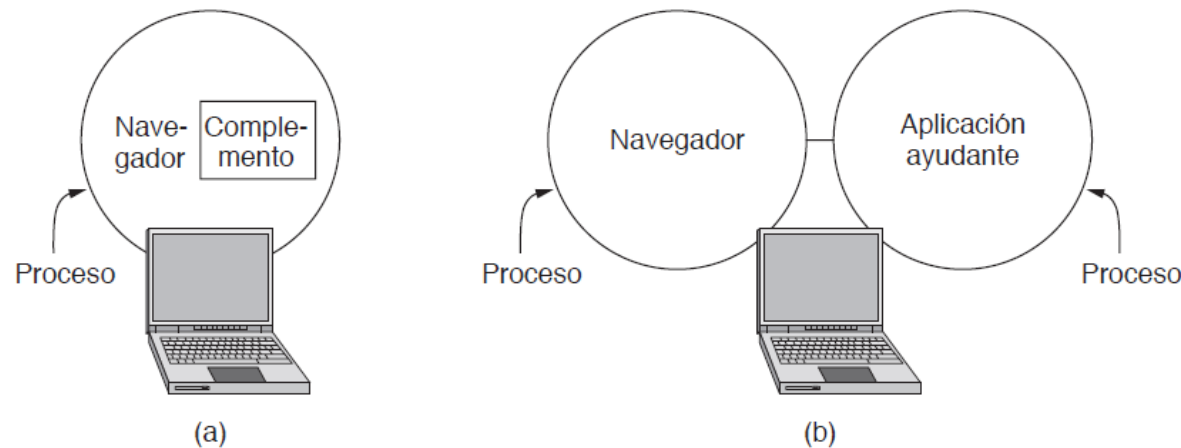
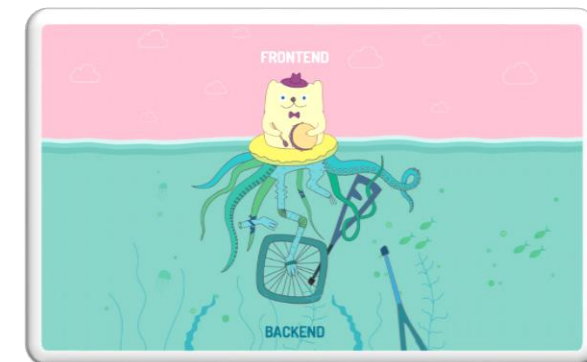


Figura 7-20. (a) Complemento del navegador. (b) Una aplicación ayudante.

EL LADO DEL SERVIDOR

Pasos que el servidor realiza en su ciclo principal

1. Aceptar una conexión TCP de un cliente (un navegador).
2. Obtener la ruta a la página, que viene siendo el nombre del archivo solicitado.
3. Obtener el archivo (del disco).
4. Enviar el contenido del archivo al cliente.
5. Liberar la conexión TCP.



- El “cuello de botella” del acceso concurrente a archivos. Una solución que existe es mantener los archivos en memoria cache por cierto periodo de tiempo.
- Para lidiar con el problema de atender una solicitud a la vez, una estrategia es hacer al servidor **multihilos**. En un diseño

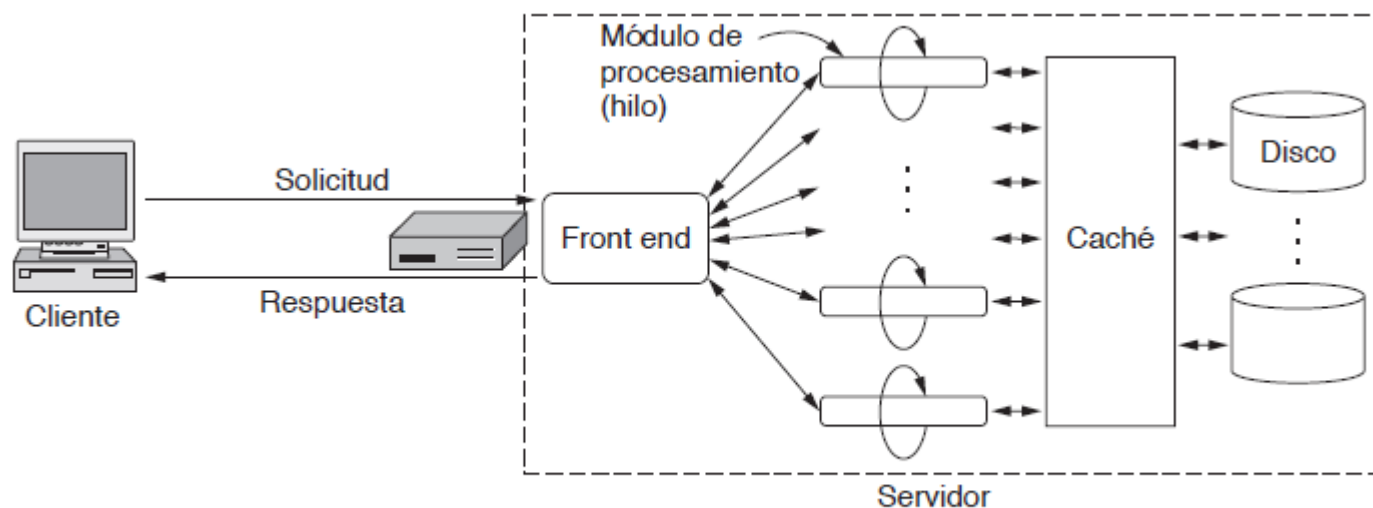
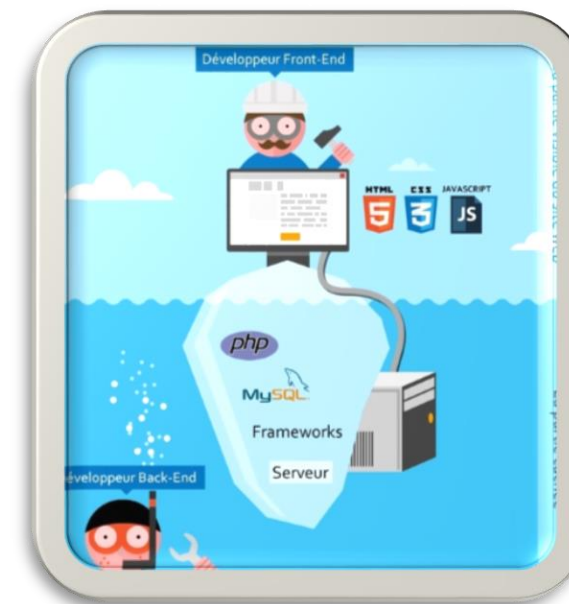


Figura 7-21. Un servidor web multihilos con un *front-end* y módulos de procesamiento.



1. Resuelve el nombre de la página solicitada.
2. Realiza control de acceso en la página web.
3. Verifica la caché.
4. Obtiene del disco la página solicitada o ejecuta un programa para construirla.
5. Determina el resto de la respuesta (por ejemplo, el tipo MIME a enviar)
6. Regresa la respuesta al cliente.
7. Realiza una entrada en el registro del servidor

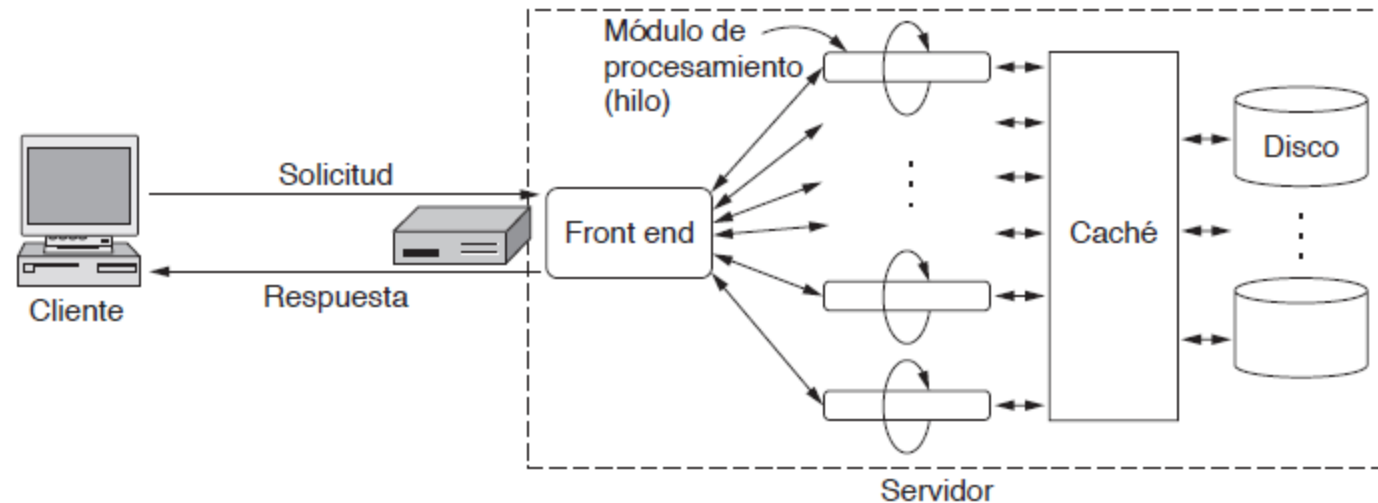


Figura 7-21. Un servidor web multihilos con un *front-end* y módulos de procesamiento.

COOKIES

- Un programa llama a un procedimiento y recibe algo que tal vez necesite presentar después para realizar cierto trabajo. En este sentido, podemos considerar a un descriptor de archivos de UNIX o un manejador de objeto de Windows como una cookie. [RFC 2109](#).
- Cuando un cliente solicita una página web, el servidor puede proveer información adicional en forma de una cookie, junto con la página solicitada.
- **La cookie es una cadena con nombre muy pequeña (a lo más de 4KB) que el servidor puede asociar con un navegador. Los navegadores almacenan las cookies ofrecidas durante un intervalo de tiempo.**

COOKIES

Los campos de las cookies son:

- *Dominio* indica de dónde viene la cookie. Cada servidor puede almacenar hasta 20 cookies por cliente.
- *La ruta* es una trayectoria en la estructura del directorio del servidor que identifica qué partes del árbol de archivos del servidor pueden utilizar la cookie. Por lo general es (/), lo que significa el árbol completo.
- *Contenido* toma la forma *nombre = valor*.

Dominio	Ruta	Contenido	Expira	Seguro
toms-casino.com	/	CustomerID=297793521	15-10-10 17:00	Si
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-11 14:22	No
aportal.com	/	Prefs= Stk:CSCO+ORCL;Spt:Jets	31-12-20 23:59	No
sneaky.com	/	UserID=4627239101	31-12-19 23:59	No

Figura 7-22. Algunos ejemplos de cookies.

COOKIES

- Si el campo *Expira* está ausente, el navegador descarta la cookie al salir (este tipo se conoce como **no persistente**). Si se proporciona una hora y una fecha, se dice que la cookie es **persistente** y se mantiene hasta que expira. Para eliminar una cookie el servidor la envía de nuevo con una fecha pasada.
- *Seguro* le indica al navegador que solo puede regresar la cookie a un servidor mediante un transporte seguro, es decir, SSL/TLS
- Las cookies se han convertido en un punto central de debate en relación con la privacidad en línea.

Dominio	Ruta	Contenido	Expira	Seguro
toms-casino.com	/	CustomerID=297793521	15-10-10 17:00	Si
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-11 14:22	No
aportal.com	/	Prefs= Stk:CSCO+ORCL;Spt:Jets	31-12-20 23:59	No
sneaky.com	/	UserID=4627239101	31-12-19 23:59	No

Figura 7-22. Algunos ejemplos de cookies.

PÁGINA ESTÁTICA Y DINÁMICA

- Una **página es estática** cuando el contenido es el mismo cada vez que se despliega en pantalla.
- Una **página es dinámica** se puede presentar de manera distinta cada vez que se despliega en pantalla. Por ejemplo, la página principal de una tienda electrónica puede ser distinta para cada visitante. Si el cliente de una librería ha comprado novelas de misterio en sus anteriores visitas, es probable que la próxima vez le aparezcan las nuevas novelas de misterio en la página de inicio. Acá se involucran *cookies* para el almacenamiento temporal de los datos.

PÁGINAS WEB ESTÁTICAS

- HTML (*HyperText Markup Language*) permite a los usuarios introducir páginas web que incluyen texto, gráficos, video, apuntadores a otras páginas web y más. HTML es un lenguaje de marcado que sirve para describir cómo se va a dar formato a los documentos.
- LaTeX y TeX son otros ejemplos de lenguajes de marcado bien conocidos en la mayoría de los autores académicos.
- La ventaja clave de un lenguaje de marcado sobre uno con marcado no explícito es que separa el contenido e la forma en que se debe presentar.

```

<html>
<head> <title> WIDGETS AMALGAMADOS, S.A. </title> </head>
<body> <h1> Bienvenidos a la página de inicio de WASA </h1>
 <br>
Estamos muy contentos de que haya elegido visitar la página de inicio de <b>
Widgets Amalgamados </b>. Esperamos que <i> usted </i> encuentre aquí toda la información que necesita.
<p> A continuación presentamos vínculos con la información sobre nuestro surtido de productos finos. Puede
ordenar en forma electrónica (por WWW), por teléfono o por correo electrónico. </p>
<hr>
<h2> Información sobre nuestros productos </h2>
<ul>
  <li> <a href="http://widget.com/productos/grandes"> Widgets grandes </a> </li>
  <li> <a href="http://widget.com/productos/chicos"> Widgets chicos </a> </li>
</ul>
<h2> Información de contacto </h2>
<ul>
  <li> Teléfono: 1-800-WIDGETS </li>
  <li> Correo electrónico: info@widgets-amalgamados.com </li>
</ul>
</body>
</html>

```

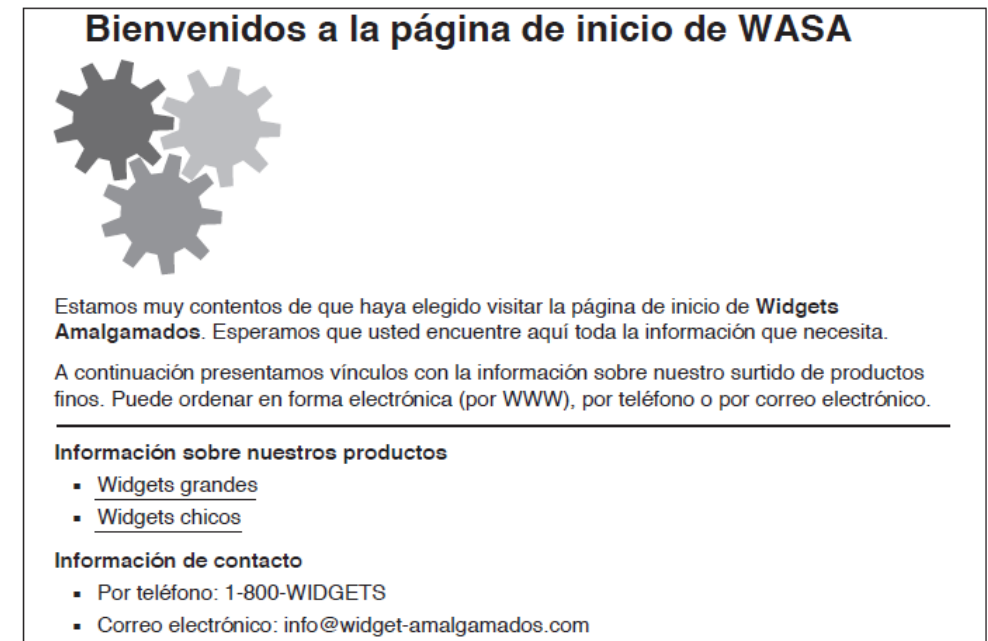


Figura 7.23 (a) El HTML para una página web de ejemplo. (b) La página con formato

PROGRAMACIÓN WEB

Etiquetas <html></html>

Las cadenas dentro de las etiquetas se llaman **directivas** <h1> hello directiva </h1>

Algunas etiquetas tienen parámetros (con nombre), llamados **atributos**.

```

```



Elemento	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
Hipervínculos	x	x	x	x	x
Imágenes	x	x	x	x	x
Listas	x	x	x	x	x
Mapas e imágenes activas		x	x	x	x
Formularios		x	x	x	x
Ecuaciones			x	x	x
Barras de herramientas			x	x	x
Tablas			x	x	x
Características de accesibilidad				x	x
Incrustación de objetos				x	x
Hojas de estilo				x	x
Secuencias de comandos				x	x
Video y audio					x
Gráficos vectoriales en línea					x
Representación de XML					x
Hilos en segundo plano					x
Almacenamiento del navegador					x
Lienzo de dibujo					x

Figura 7-24. Algunas diferencias entre versiones de HTML.

FORMULARIOS

- Los formularios contienen cuadros o botones que permiten a los usuarios proporcionar información o tomar decisiones, y después enviar dicha información al dueño de la página.
- Debemos tener en cuenta que los formularios siguen siendo contenido estático.

```
<html>
<head> <title> FORMULARIO DE PEDIDO DE CLIENTES WASA </title> </head>
<body>
<h1> Orden de compra de widgets </h1>
<form ACTION="http://widget.com/cgi-bin/pedido.cgi" method=POST>
<p> Nombre <input name="cliente" size=46> </p>
<p> Dirección <input name="direccion" size=40> </p>
<p> Ciudad <input name="city" size=20> Estado <input name="estado" size =4>
País <input name="pais" size=10> </p>
<p> Núm. tarjeta de crédito <input name="numtarj" size=10>
Expira <input name="expira" size=4>
M/C <input name="tc" type=radio value="mastercard">
VISA <input name="tc" type=radio value="visacard"> </p>
<p> Tamaño de widget Grande <input name="producto" type=radio value="costoso">
Pequeño <input name="producto" type=radio value="barato">
Enviar por mensajería rápida <input name="express" type=checkbox> </p>
<p><input type=submit value="Enviar pedido"> </p>
¡Gracias por ordenar un widget WASA, el mejor widget que el dinero puede comprar!
</form>
</body>
</html>
```

Orden de compra de widgets

Nombre

Dirección

Ciudad Estado País

Núm. tarjeta de crédito Expira M/C ☐ Visa ☐

Tamaño de widget Grande ☐ Chico ☐ Enviar por mensajería rápida ☐

¡Gracias por ordenar un widget WASA, el mejor widget que el dinero puede comprar!

(b)

Figura 7-25. (a) El HTML para un formulario de pedido. (b) La página con formato.

```

<html>
<head> <title> FORMULARIO DE PEDIDO DE CLIENTES WASA </title> </head>
<body>
<h1> Orden de compra de widgets </h1>
<form ACTION="http://widget.com/cgi-bin/pedido.cgi" method=POST>
<p> Nombre <input name="cliente" size=46> </p>
<p> Dirección <input name="direccion" size=40> </p>
<p> Ciudad <input name="city" size=20> Estado <input name="estado" size =4>
País <input name="pais" size=10> </p>
<p> Núm. tarjeta de crédito <input name="numtarj" size=10>
Expira <input name="expira" size=4>
M/C <input name="tc" type=radio value="mastercard">
VISA <input name="tc" type=radio value="visacard"> </p>
<p> Tamaño de widget Grande <input name="producto" type=radio value="costoso">
Pequeño <input name="producto" type=radio value="barato">
Enviar por mensajería rápida <input name="express" type=checkbox> </p>
<p><input type=submit value="Enviar pedido"> </p>
¡Gracias por ordenar un widget WASA, el mejor widget que el dinero puede comprar!
</form>
</body>
</html>

```

(b)

Figura 7-25. (a) El HTML para un formulario de pedido. (b) La página con formato.

```

cliente=John+Doe&direccion=100+Main+St.&ciudad=White+Plains&
estado=NY&pais=USA&numtarj=1234567890&expira=6/14&cc=mastercard&
producto=barato&express=on

```

Figura 7-26. Una posible respuesta del navegador para el servidor, con la información escrita por el usuario.

CSS: CASCADING STYLE SHEETS

- El objetivo original de HTML era especificar la **estructura** del documento, no su **apariciencia**.

```
<font face="helvética" size="24" color="red">Fotos de  
Débora</font>
```

- En los editores de texto, las hojas de estilo permiten a los autores asociar texto con un estilo lógico en vez de uno físico; por ejemplo, “párrafo inicial” en vez de “texto en cursiva”. La apariencia de cada estilo se define por separado. De esta manera, si el autor decide cambiar los párrafos iniciales de cursiva a 14 puntos en color azul, a negritas a 18 puntos en rosa impactante, todo lo que requiere es modificar una definición para convertir todo el documento.

CSS: CASCADING STYLE SHEETS

- CSS define un lenguaje simple para describir reglas que controlan la apariencia de contenido etiquetado

```
body {background-color:linen; color:navy; font-family:Arial;}  
h1 {font-size:200%}  
h2 {font-size:150%}
```

Figura 7-27. Ejemplo de CSS.

- Podemos colocar las hojas de estilo en un archivo HTML (mediante la etiqueta <style>), pero es más común colocarlas en un archivo separado y hacer referencia a ellas.

```
<head>  
<title>WIDGETS AMALGAMADOS, S.A. </title>  
<link rel="stylesheet" type="text/css" href="awistyle.css" />  
</head>
```

→ MIME

Figura 7-28. Inclusión de una hoja de estilo CSS.

PÁGINAS WEB DINÁMICAS Y APLICACIONES WEB

- El contenido dinámico se puede generar mediante programas que se ejecuten en el servidor o en el navegador (o en ambos lugares).

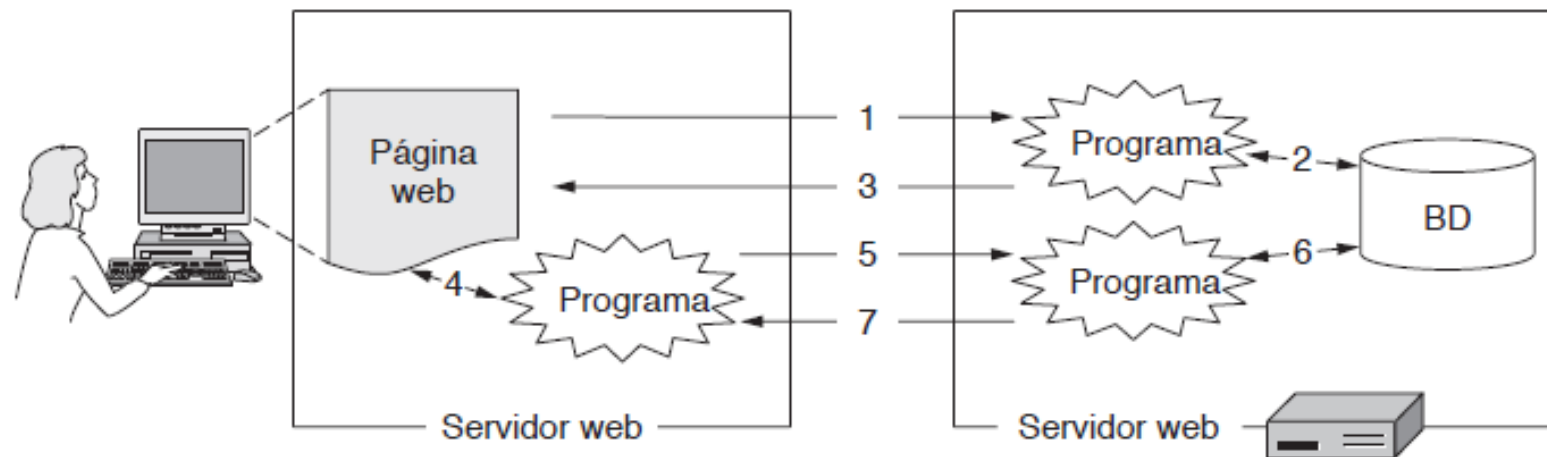


Figura 7-29. Páginas dinámicas.

PÁGINAS WEB DINÁMICAS Y APLICACIONES WEB

Se han desarrollado varias API estándar para que los servidores web invoquen programas. La existencia de estas interfaces facilita a los desarrolladores el proceso de extender distintos servidores con aplicaciones web. A continuación analizaremos dos API para que el lector se dé una idea de lo que implican.

1. **CGI (Interfaz de Puerta de Enlace Común**, del inglés *Common Gateway Interface*) y se define en el [RFC 3875](#). **CGI provee una interfaz para permitir que los servidores web se comuniquen con programas de soporte y secuencias de comandos que pueden aceptar entradas (por ejemplo, de los formularios) y generar páginas HTML en respuesta.**

Estos programas pueden estar escritos en cualquier lenguaje que sea conveniente para el desarrollador, por lo general un lenguaje de secuencias de comandos para facilitar el avance. Puede elegir Python, Ruby, Perl o su lenguaje favorito.

Por convención, los programas que se invocan a través de CGI viven en un directorio llamado cgibin, el cual es visible en el URL.

PÁGINAS WEB DINÁMICAS Y APLICACIONES WEB

2. La metodología en este caso es incrustar pequeñas secuencias de comandos dentro de las páginas de HTML y hacer que las ejecute el mismo servidor para generar la página. Un lenguaje popular para escribir estas secuencias de comandos es **PHP** (**P**reprocesador de **H**ipertexto, del inglés *Hypertext Preprocessor*).
- El servidor tienen que entender PHP. Por lo general, los servidores identifican las páginas web que contienen PHP con base en la extensión de archivo ***php***.

```
<html>
<body>
<form action="accion.php" method="post">
<p> Por favor introduzca su nombre: <input type="text" name="nombre"> </p>
<p> Por favor introduzca su edad: <input type="text" name="edad"> </p>
<input type="submit">
</form>
</body>
</html>
```

(a)

```
<html>
<body>
<h1> Respuesta: </h1>
Hola <?php echo $nombre; ?>.
Predicción: el siguiente año tendrá <?php echo $edad + 1; ?>
</body>
</html>
```

(b)

```
<html>
<body>
<h1> Respuesta: </h1>
Hola Bárbara.
Predicción: el siguiente año tendrá 33
</body>
</html>
```























(c)

Figura 7-30. (a) Una página web que contiene un formulario. (b) Una secuencia de comandos de PHP para manejar la salida del formulario. (c) Salida de la secuencia de comandos de PHP cuando las entradas son “Bárbara” y “32”, respectivamente.

PÁGINAS WEB DINÁMICAS Y APLICACIONES WEB

- Existen otras opciones disponibles:
- **JSP** (*JavaServer Pages*) la parte dinámica se escribe en el lenguaje de programación Java. Las páginas tienen la extensión de archivo *.jsp*.
- **ASP.NET** (*Active Server Pages .NET*). Es la versión de Microsoft que utiliza programas escritos en el framework en red .NET para generar el contenido dinámico. Las páginas escritas en esta tecnología tienen extensión *.aspx*.

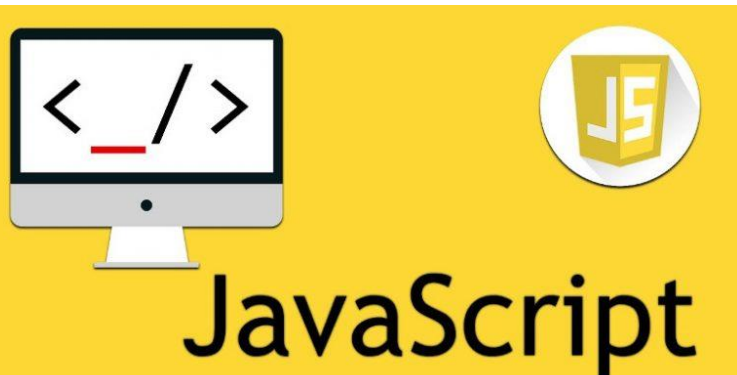


Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

IEEE ESPECTRUM - The 2018 Top Programming Languages.
<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

GENERACIÓN DE PÁGINAS WEB DINÁMICAS EN EL CLIENTE

- Lo que ninguno de los lenguajes previamente mencionados puede realizar es responder a interacciones directas con el usuario (por ejemplo, los movimientos del ratón).
- Para ello es necesario tener secuencias de comandos incrustados en páginas HTML que se ejecuten en la máquina del cliente y no en el servidor.
- Los comandos se pueden ingresar mediante la etiqueta `<script>` y las páginas web interactivas se conocen como **HTML dinámico**.
- JavaScript es un lenguaje de alto nivel para el desarrollo de páginas interactivas.



```

<html>
<head>
<script language="javascript" type="text/javascript">
function respuesta(form_prueba) {
    var persona = form_prueba.nombre.value;
    var anios = eval(form_prueba.edad.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hola " + persona + "<br>");
    document.writeln("Predicción: el siguiente año tendrá " + anios + "!");
    document.writeln("</body> </html>");
    document.close();
}
</script>
</head>

<body>
<form>
Por favor introduzca su nombre: <input type="text" name="nombre">
<p>
Por favor introduzca su edad: <input type="text" name="edad"> </p>
<p>
<input type="button" value="submit" onclick="respuesta(this.form)">
</form>
</body>
</html>

```

Figura 7-31. Uso de JavaScript para procesar un formulario.

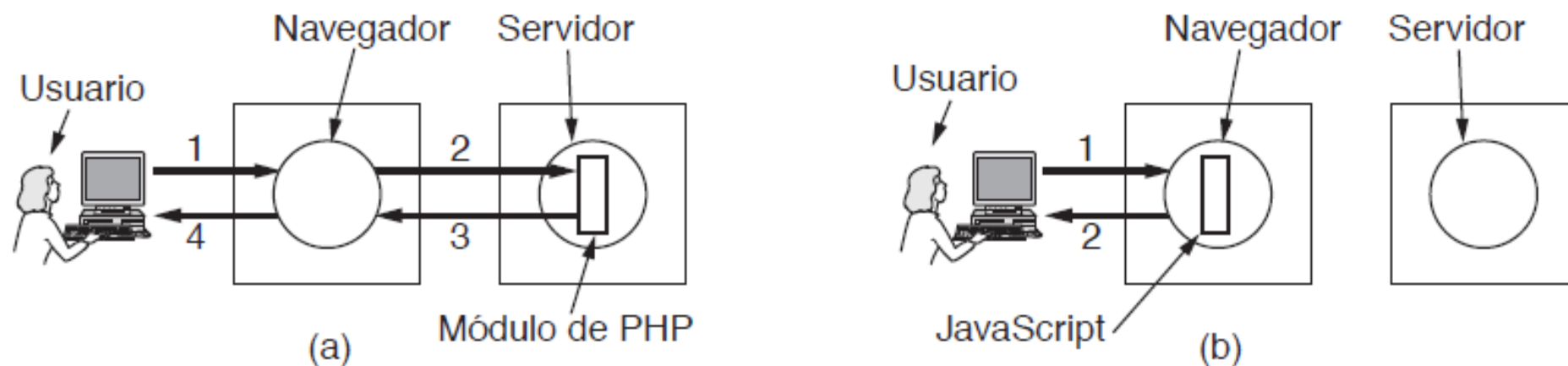


Figura 7-32. (a) Secuencias de comandos del lado servidor con PHP. (b) Secuencias de comandos del lado cliente con JavaScript.

Otras alternativas a JavaScript:

- En las plataformas Windows es **VBScript**, que se basa en Visual Basic.
- **Applets**, son pequeños programas en Java compilados en instrucciones de máquina para una computadora virtual, conocida como **JVM (Java Virtual Machine)**



<https://barcamp.se/>

24 de noviembre

PLANEACIÓN

Material utilizado	<p>1. Arboleda, L. (2012). Programación en Red con Java.</p> <p>2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.".</p> <p>3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación.</p> <p>4. Reese, R. M. (2015). Learning Network Programming with Java. Packt Publishing Ltd.</p>
VISTO	A1. Secciones 7.1 y 7.2 (revisado)
HOY	A2. Leer del libro 3 la sección 7.3
Actividades DESPUÉS clase – viernes (9 noviembre)	A3. Leer la sección 12 del libro 1

PLANEACIÓN

Material utilizado	<p>1. Arboleda, L. (2012). Programación en Red con Java.</p> <p>2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.".</p> <p>3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación.</p> <p>4. Reese, R. M. (2015). Learning Network Programming with Java. Packt Publishing Ltd.</p>
Actividades DESPUÉS clase – jueves (15 noviembre)	A1. Leer del libro 1 la sección 13 y del libro 4 las páginas 37-45
Actividades DESPUÉS clase – viernes (16 noviembre)	A2. Leer del libro 4 la sección 4

REFERENCIAS

- [https://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/Sir Tim Berners-Lee %28cropped%29.jpg/220px-Sir Tim Berners-Lee %28cropped%29.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/Sir_Tim_Berners-Lee_%28cropped%29.jpg/220px-Sir_Tim_Berners-Lee_%28cropped%29.jpg)
- [https://upload.wikimedia.org/wikipedia/commons/thumb/7/75/Marc Andreessen .jpg/200px-Marc Andreessen.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/7/75/Marc_Andreessen.jpg/200px-Marc_Andreessen.jpg)
- https://commons.wikimedia.org/wiki/File:URI_Venn_Diagram.svg
- <https://qph.fs.quoracdn.net/main-qimg-1ce27ba8e1e52a0aa78897222951c538.webp>
- <https://www.optimisthub.com/up/frontend-nedir.png>