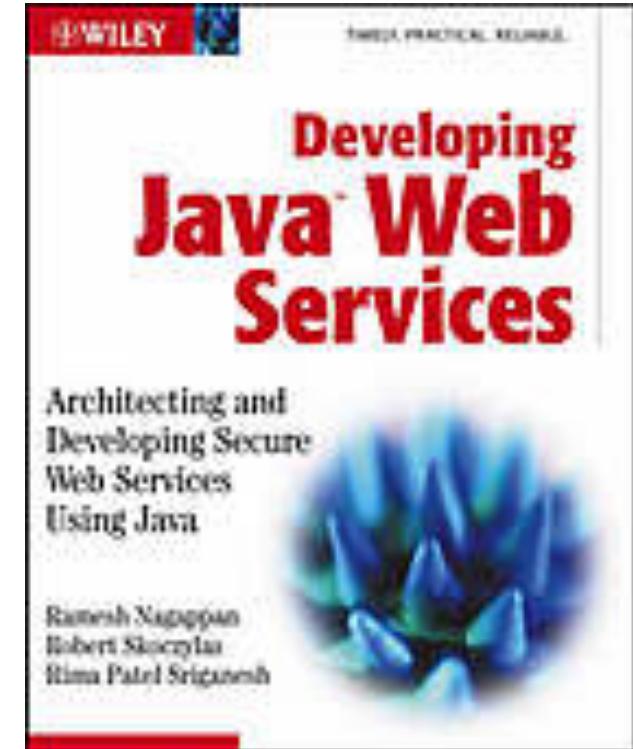
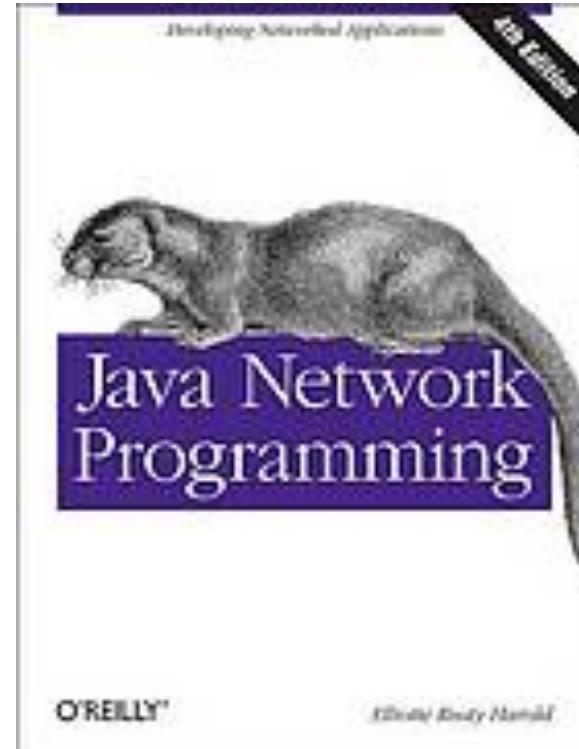
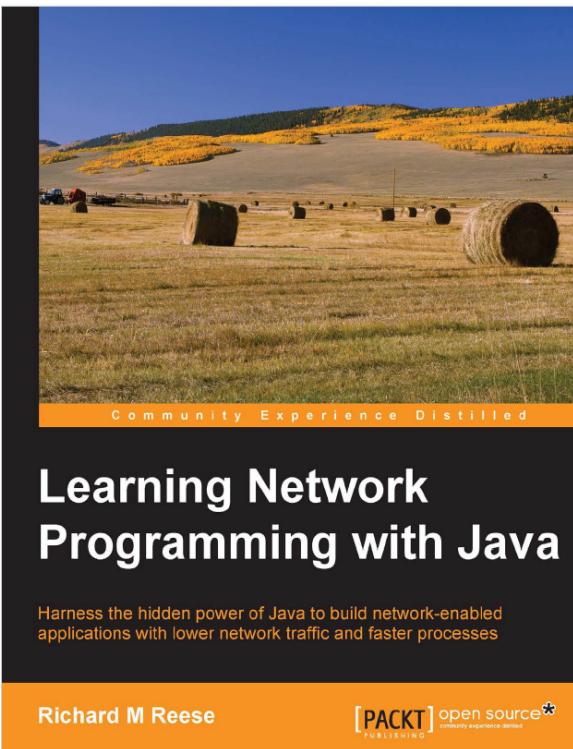
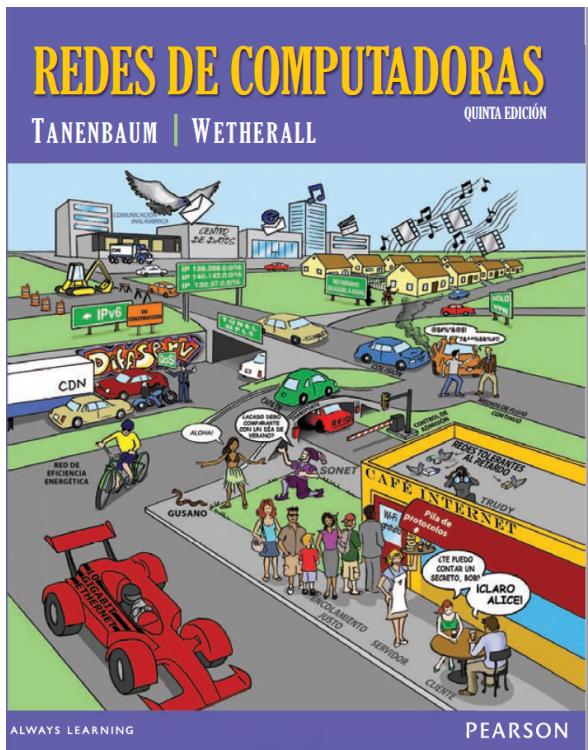


# **REDES DE COMPUTADORES Y LABORATORIO**

**Yor Castaño, MSc**



# BIBLIOGRAFÍA



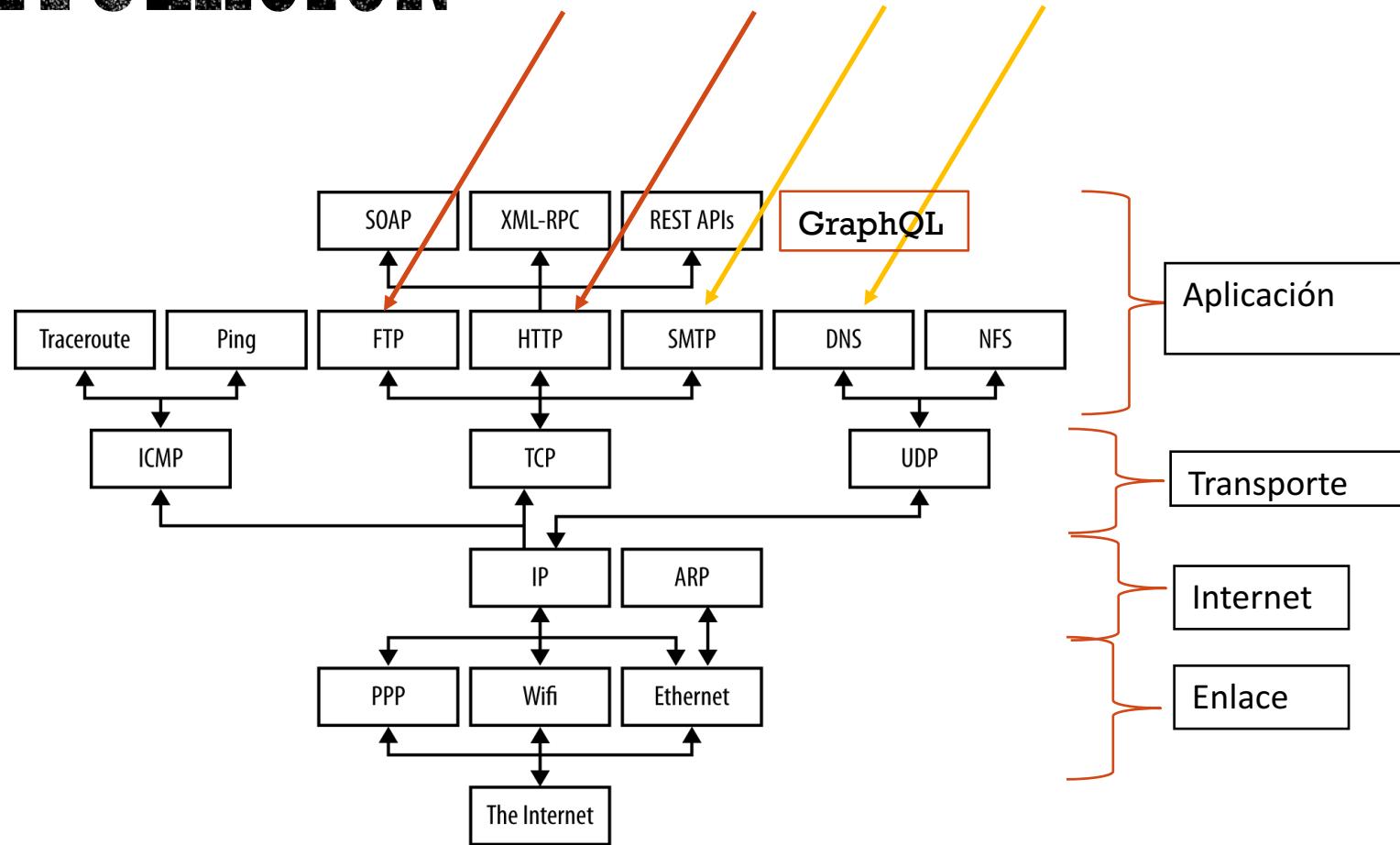
# COMPETENCIAS

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



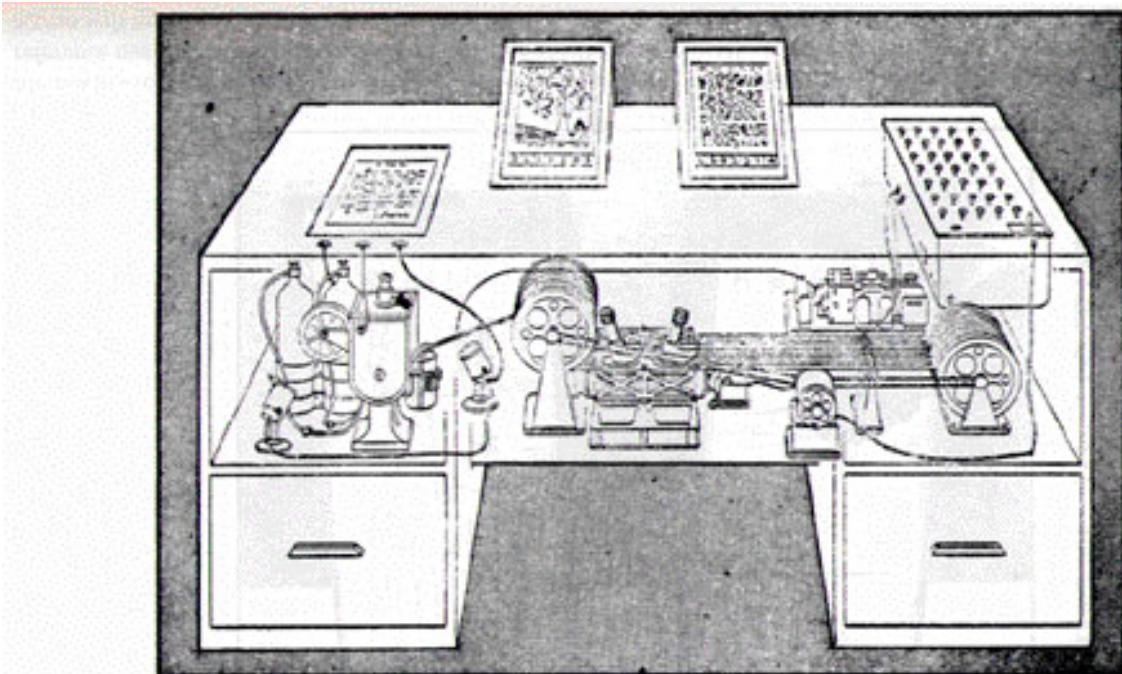
<https://www.youtube.com/watch?v=Bo4dY8X2QFg>

# RECAPITULACIÓN



# LA CAPA DE APLICACIÓN

# WORLD WIDE WEB



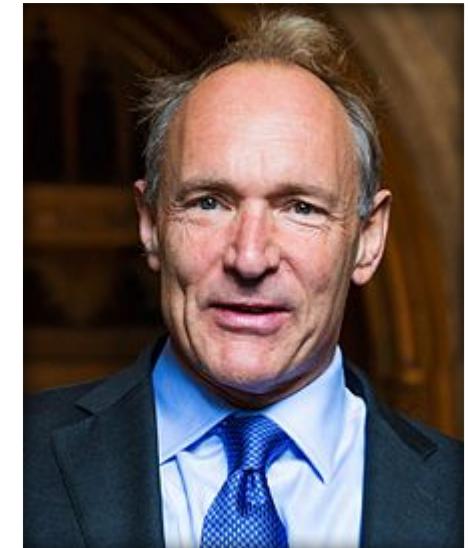
Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (LIFE 19(11), p. 123).

- MEMEX – Vannevar Bush
- 1945
- Dispositivo basado en fichas, proyectos e I/O mecanicos
- ¿A que se parece?

# 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.



# WWW

- **Hipertexto**, desarrollado en 1945 por el profesor Vannevar Bush. Es el encargado del direccionamiento entre páginas web
- **Navegador**, son los encargados de obtener la página web solicitada e interpretar su contenido con el fin de desplegarlo en el formato adecuado.
- **RFC URL:**
  - **1738, 3986**
- **RFC HTTP**
  - **1945, 2612**
- **RFC HTML**
  - **1866**

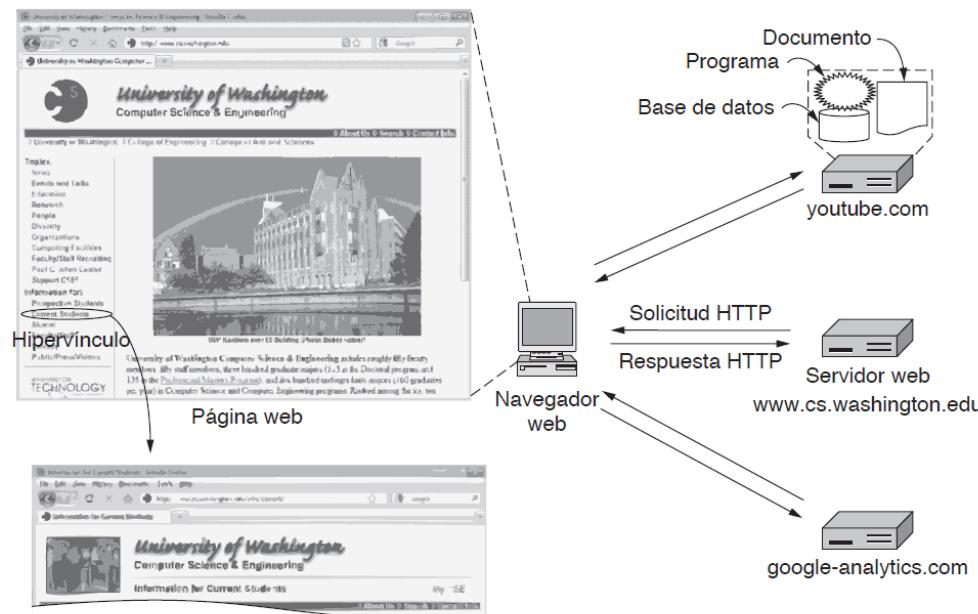


Figura 7-18. Arquitectura de la web.

# W3C

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



# 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.

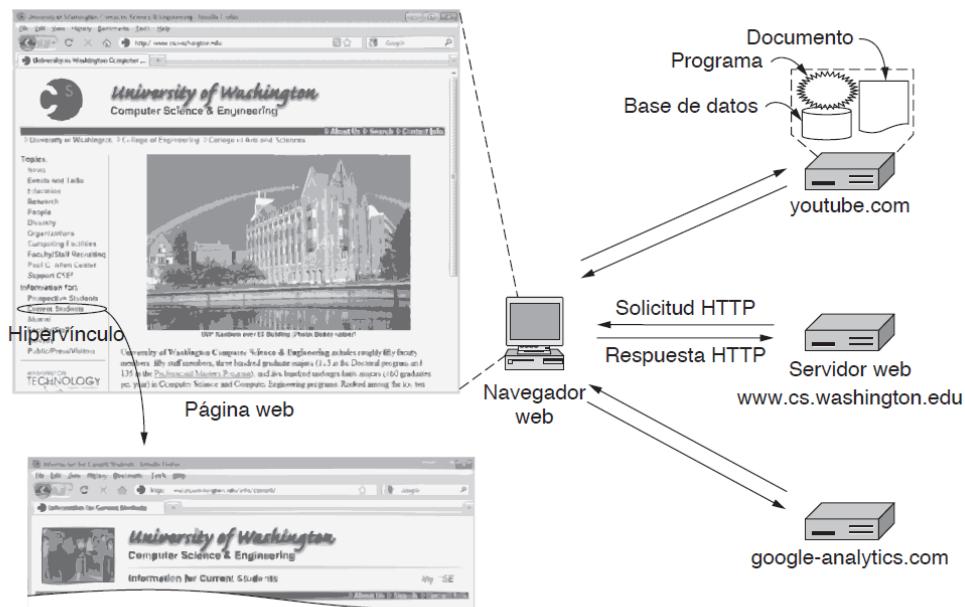


Figura 7-18. Arquitectura de la web.

# 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?
- ¿Alternativas de despliegue?
- Que hay en el education pack de git para lograr esto?

# EL LADO DEL CLIENTE

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.

<https://www.icesi.edu.co/>

<ftp://aeneas.mit.edu/>

<https://en.wikipedia.org/wiki/Internet#Terminology>

# EL LADO DEL CLIENTE

- 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 a 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.

# EL LADO DEL CLIENTE

[https://github.com/yorjaggy/networking\\_guides/tree/master/slides](https://github.com/yorjaggy/networking_guides/tree/master/slides)

- ¿Cuál es el protocolo?
- ¿El nombre de DNS?
- ¿La ruta que indica a la página específica?

# URL

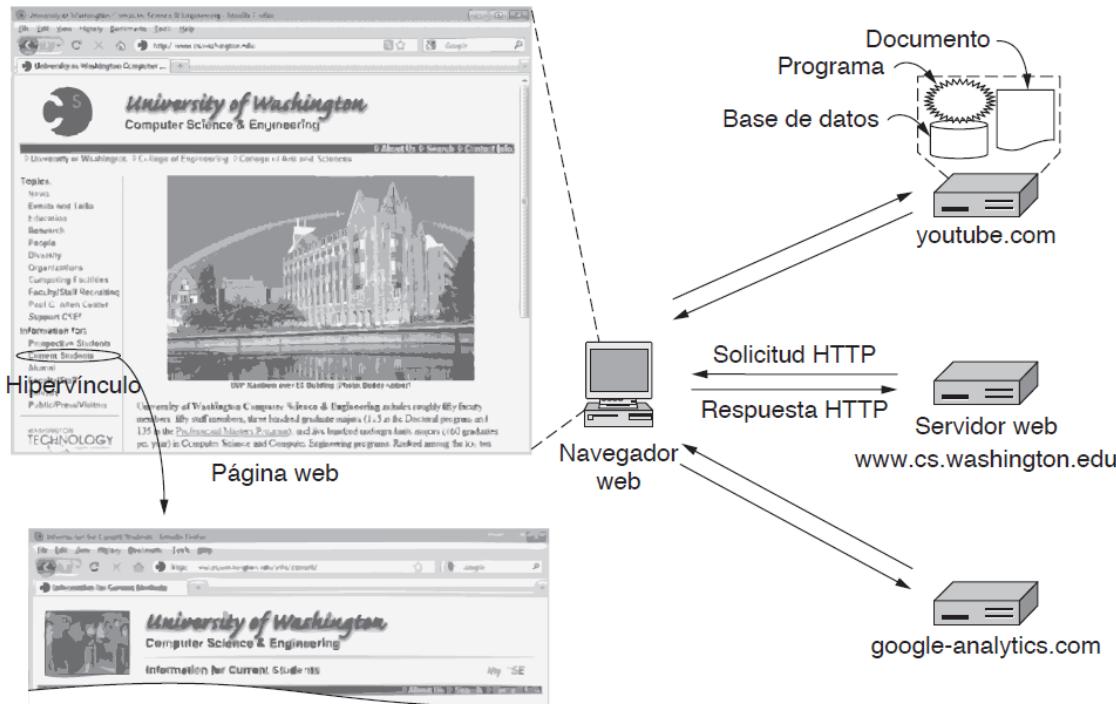


Figura 7-18. Arquitectura de la web.

La URL de la página es:

**http://www.cs.washington.edu/index.html**

- **Protocolo: http**
- **El nombre DNS: 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](http://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. Despues envía una solicitud HTTP para pedir la página */index.html*.

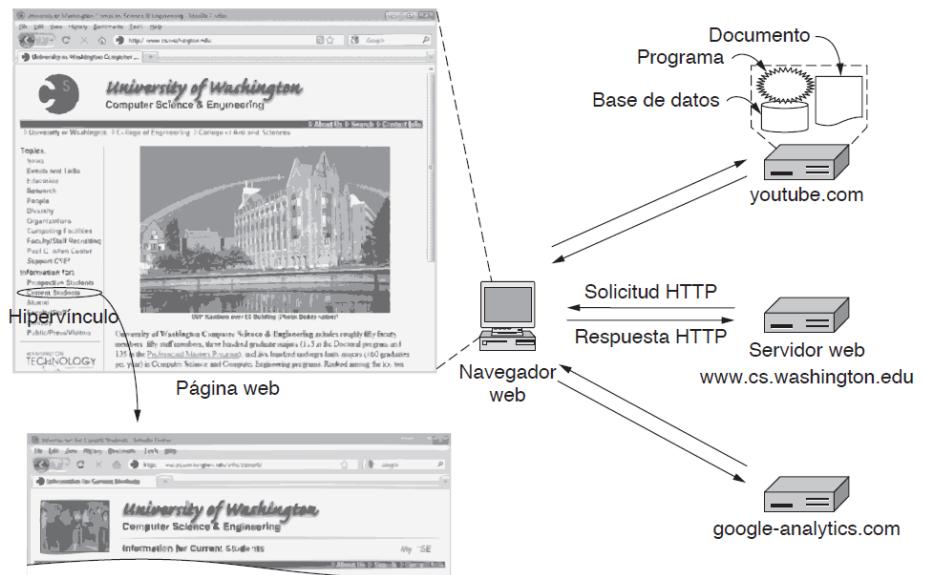


Figura 7-18. Arquitectura de la web.

# PASOS

6. El servidor *www.cs.washington.edu* envía la página como una respuesta HTTP, por ejemplo, enviando el archivo */index.html*.
7. 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*.

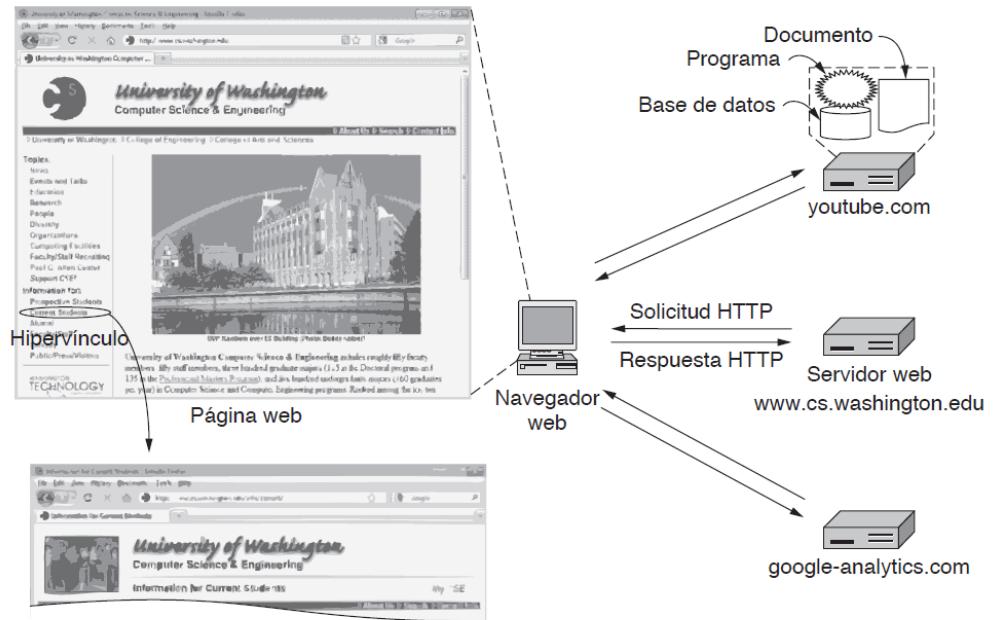


Figura 7-18. Arquitectura de la web.

# PASOS

8. El navegador despliega la página */index.html* como aparece
9. Se liberan las conexiones TCP si no hay más solicitudes para los mismos servidores durante un periodo corto.

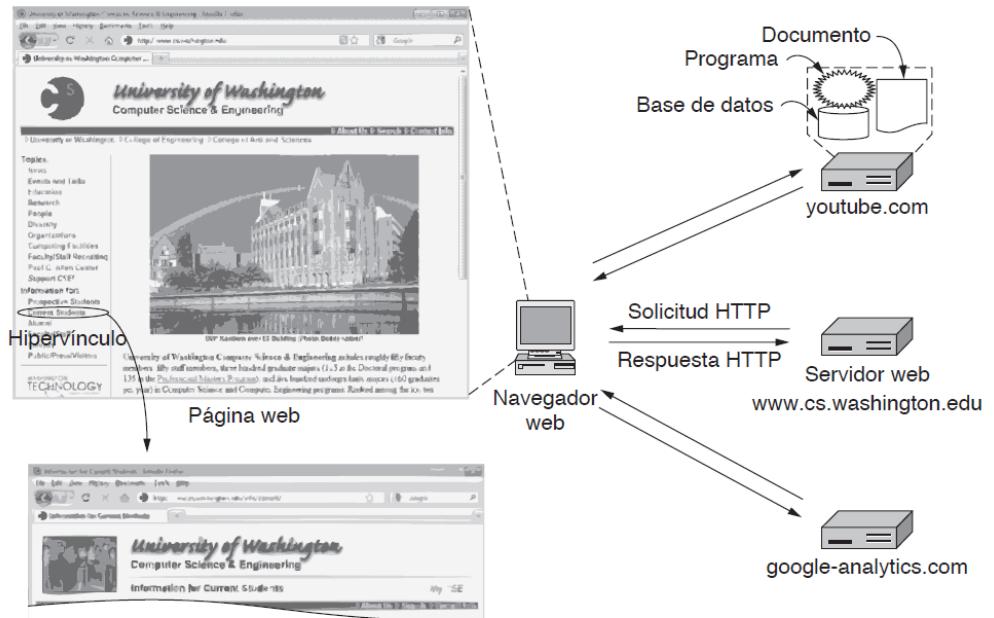


Figura 7-18. Arquitectura de la web.

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

**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.
- **Mailto** permite a los usuarios enviar correos electrónicos desde un navegador web. La mayoría de navegadores responderán cuando el usuario siga un vínculo *mailto* 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.**

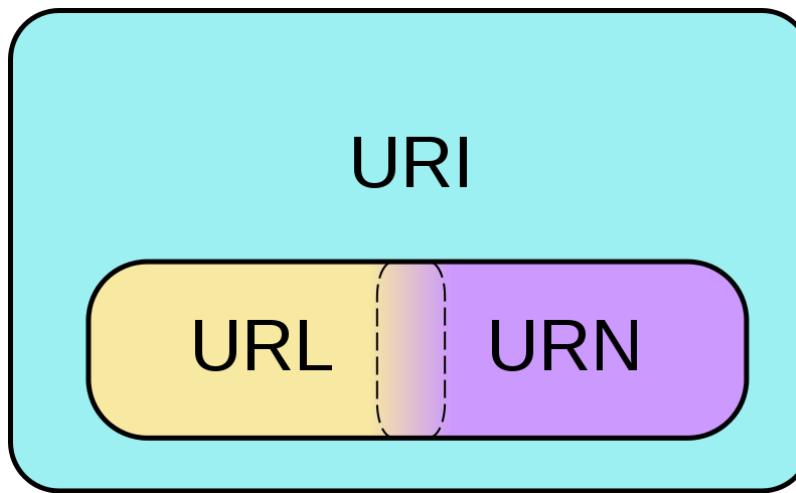
# DEBILIDAD

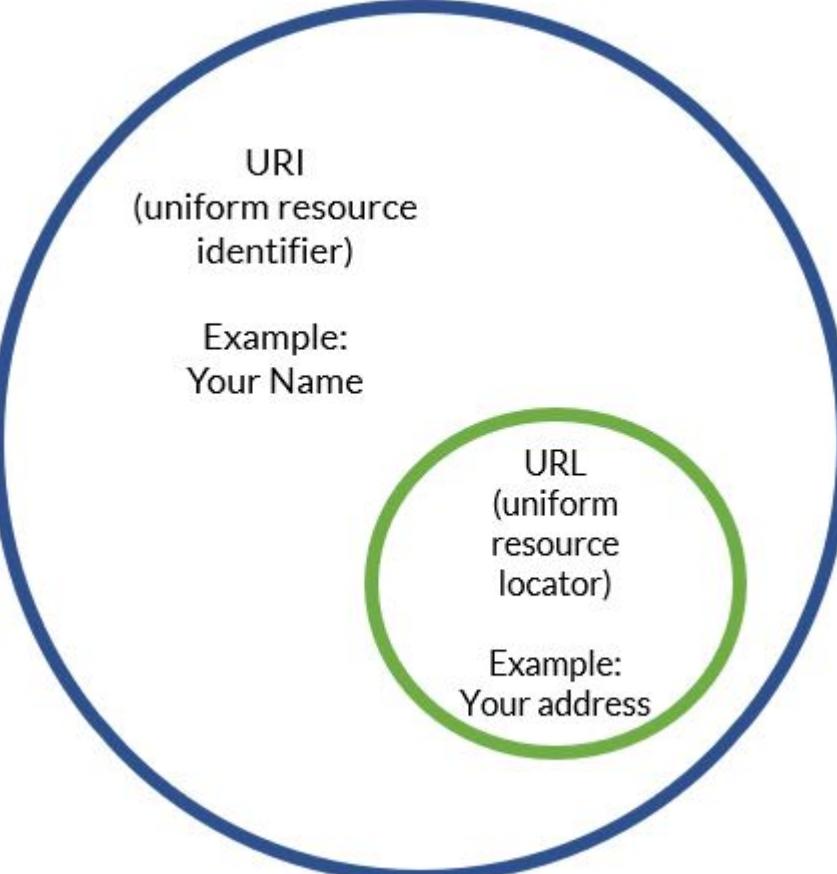
Un URL apunta a un host específico, pero algunas veces es conveniente hacer referencia a una página sin necesidad de decir al mismo tiempo en dónde está. Por ejemplo, en las páginas con muchas referencias es conveniente tener varias copias en ubicaciones distantes y separadas para reducir el tráfico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

# URI (UNIFORM RESOURCE IDENTIFIERS)

- Para resolver la debilidad mencionada, los URL se generalizaron en **URI**. [RFC 3986](#).
  - 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>





A diagram illustrating the relationship between URI and URL. It consists of two concentric circles. The outer circle is blue and contains the text "URI (uniform resource identifier)". The inner circle is green and contains the text "URL (uniform resource locator)". Below the blue circle, the text "Example: Your Name" is provided. Below the green circle, the text "Example: Your address" is provided.

URI  
(uniform resource  
identifier)

Example:  
Your Name

URL  
(uniform  
resource  
locator)

Example:  
Your address

# 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

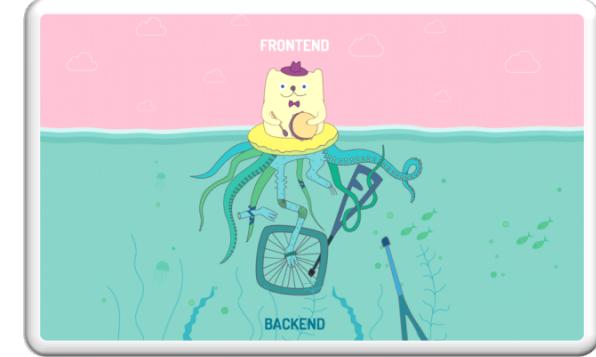
Tipo	Subtipos de ejemplo	Descripción
text	plain, html, xml, css	Texto en diversos formatos.
image	gif, jpeg, tiff	Imágenes.
audio	basic, mpeg, mp4	Sonidos.
video	mpeg, mp4, quicktime	Películas.
model	vrml	Modelo 3D.
application	octet-stream, pdf, javascript, zip	Datos producidos por aplicaciones.
message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

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

# 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

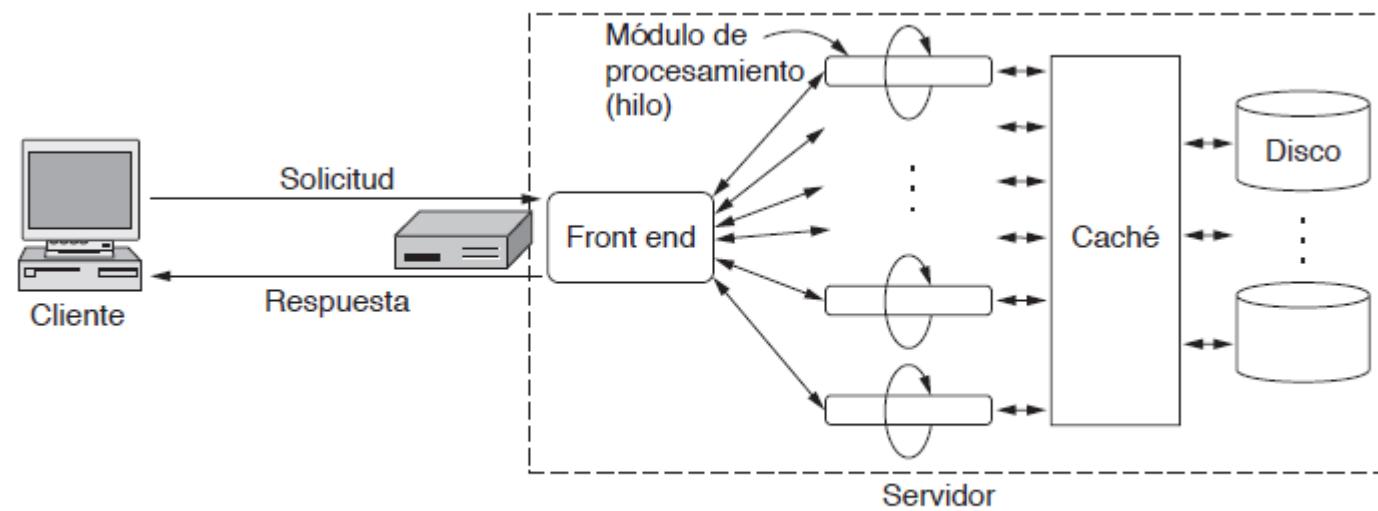
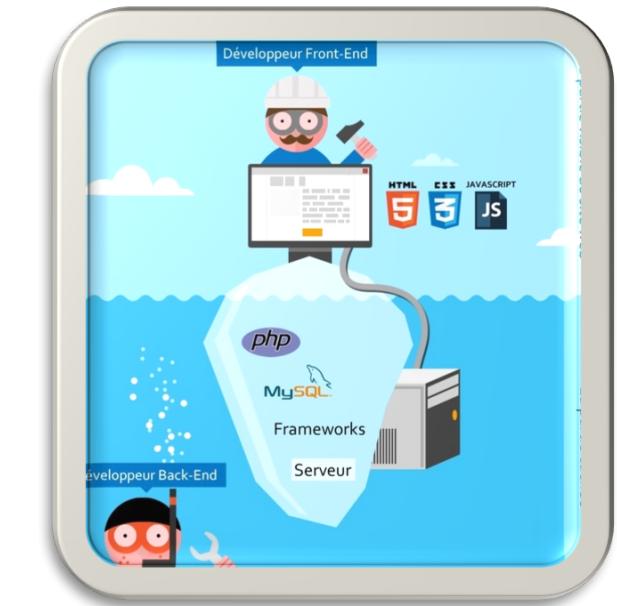


Figura 7-21. Un servidor web multihilos con un *front-end* y módulos de procesamiento.



# EL LADO DEL CLIENTE

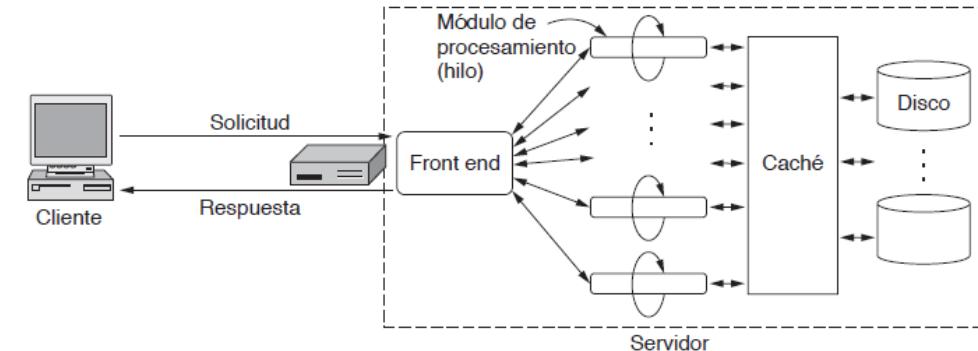


Figura 7-21. Un servidor web multihilos con un *front-end* y módulos de procesamiento.

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

# 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	Sí
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	Sí
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>
```

## Bienvenidos a la página de inicio de WASA



Estamos muy contentos de que haya elegido visitar la página de inicio de **Widgets Amalgamados**. Esperamos que usted encuentre aquí toda la información que necesita.

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.

### Información sobre nuestros productos

- Widgets grandes
- Widgets chicos

### Información de contacto

- Por teléfono: 1-800-WIDGETS
- Correo electrónico: info@widget-amalgamados.com

Figura 7.12 (a) El HTMT 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.

# 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).

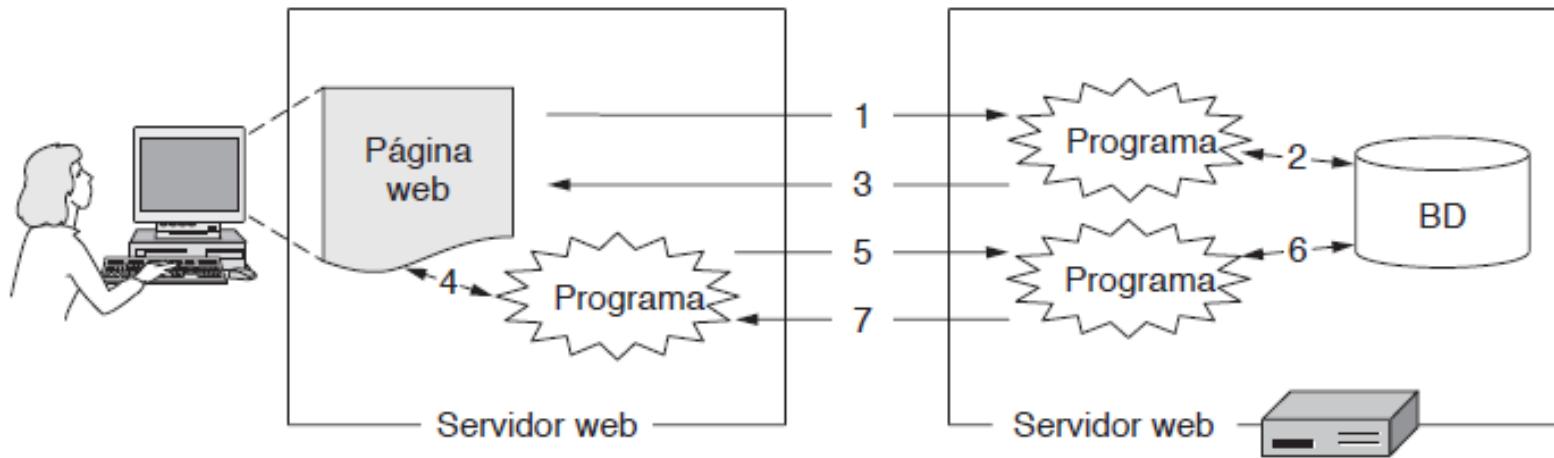
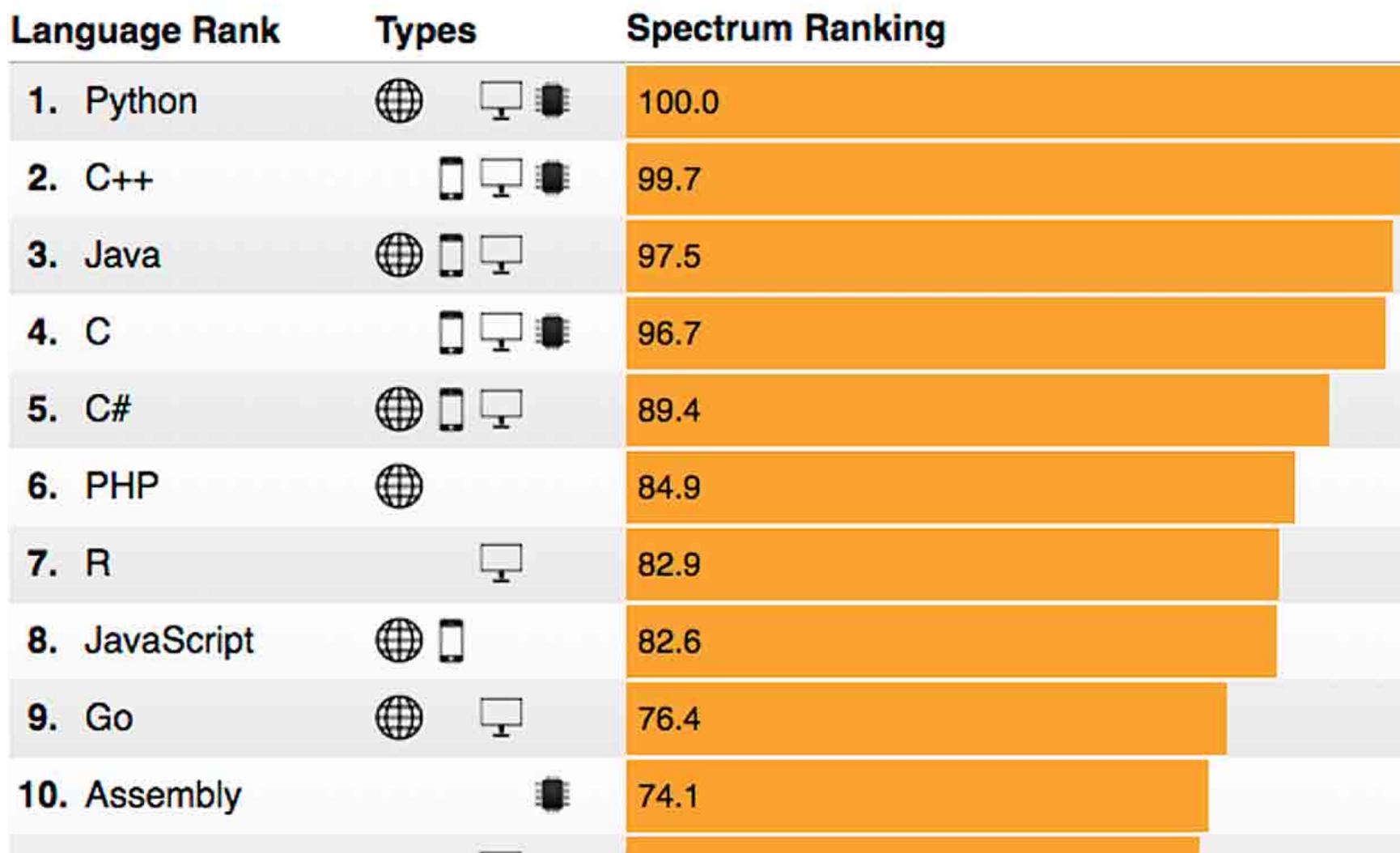


Figura 7-29. Páginas dinámicas.



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



## QUÍMICA FARMACÉUTICA RECIBE ACREDITACIÓN DE ALTA CALIDAD

MÁS INFORMACIÓN



Tu profesión  
A Otro Nivel

¿Qué programa estás buscando?

## Noticias y Eventos



EVENTO  
Nov/09  
Club De Lectura

Screenshot of the Network tab in the Chrome DevTools developer console, showing network requests for the website https://www.icesi.edu.co.

**Request URL:** https://www.icesi.edu.co/  
**Request Method:** GET  
**Status Code:** 200 OK  
**Remote Address:** 200.3.192.46:443  
**Referrer Policy:** no-referrer-when-downgrade

**Response Headers** (view parsed)

Header	Value
HTTP/1.1	200 OK
Date	Fri, 09 Nov 2018 22:09:29 GMT
Server	Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips PHP/7.1.11
X-Powered-By	PHP/7.1.11
Expires	Wed, 17 Aug 2005 00:00:00 GMT
Last-Modified	Fri, 09 Nov 2018 22:09:29 GMT
Cache-Control	no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma	no-cache
Vary	Accept-Encoding
Content-Encoding	gzip
Content-Length	21493
Connection	close
Content-Type	text/html; charset=utf-8

**Request Headers** (view parsed)

Header	Value
GET	/ HTTP/1.1
Host	www.icesi.edu.co
Connection	keep-alive
Cache-Control	max-age=0
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding	gzip, deflate, br
Accept-Language	en-US,en;q=0.9,es;q=0.8
Cookie	_ga=GA1.3.2006869504.1524161431; __atuvc=1%7C16; 923cee6bec1c49638b8e40cffc911e13=es-ES; b4691f0f8df12a21e4071f350e754ab9=en-GB; _gcl_au=1.1.28557620.1537548397; amplitude_id_06d3064c4b7aa1187920f7d63c355898icesi.edu.co=eyJkZXZpY2VJZCI6IjV1Y2I0MGQ3LWY2YzEtNDQzNy05NzQwLWQzNmVkJTY3YTdjZViLCJ1c2VysWQ;

82 requests | 134 KB transferred | Finish: 1...

▼ General

Request URL: <https://www.icesi.edu.co/>  
Request Method: GET  
Status Code: 200 OK  
Remote Address: 200.3.192.46:443  
Referrer Policy: no-referrer-when-downgrade

---

▼ Response Headers

[view parsed](#)

HTTP/1.1 200 OK  
Date: Fri, 09 Nov 2018 22:09:29 GMT  
Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips PHP/7.1.11  
X-Powered-By: PHP/7.1.11  
Expires: Wed, 17 Aug 2005 00:00:00 GMT  
Last-Modified: Fri, 09 Nov 2018 22:09:29 GMT  
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0  
Pragma: no-cache  
Vary: Accept-Encoding  
Content-Encoding: gzip  
Content-Length: 21493  
Connection: close  
Content-Type: text/html; charset=utf-8

---

▼ Request Headers

[view parsed](#)

GET / HTTP/1.1  
Host: www.icesi.edu.co  
Connection: keep-alive  
Cache-Control: max-age=0  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, br  
Accept-Language: en-US,en;q=0.9,es;q=0.8  
Cookie: \_ga=GA1.3.2006869504.1524161431; \_\_atuvc=1%7C16; 923cee6bec1c49638b8e40cffc911e13=es-ES; b4691f0f8df12a21e4071f350e754ab9=en-620.1537548397; amplitude\_id\_06d3064c4b7aa1187920f7d63c355898icesi.edu.co=eyJkZXZpY2VJZCI6IjV1Y2I0MGQ3LWY2YzEtNDQzNy05NzQwLWQzMmVkJTY

# MÉTODOS

Método	Descripción
GET	Leer una página web.
HEAD	Leer el encabezado de una página web.
POST	Adjuntar a una página web.
PUT	Almacenar una página web.
DELETE	Eliminar la página web.
TRACE	Repetir la solicitud entrante
CONNECT	Conectarse a través de un proxy
OPTIONS	Consultar las opciones para una página

## ▼ General

Request URL: <https://www.icesi.edu.co/>

Request Method: GET

Status Code: ● 200 OK

Remote Address: 200.3.192.46:443

Referrer Policy: no-referrer-when-downgrade

Figura 7-37. Los métodos de solicitud HTTP integrados.

# CÓDIGOS DE ESTADO

Código	Significado	Ejemplos
1xx	Información	100 = el servidor acepta manejar la solicitud del cliente.
2xx	Éxito	200 = la solicitud es exitosa; 204 = no hay contenido.
3xx	Redirección	301 = se movió la página; 304 = la página en caché aún es válida.
4xx	Error del cliente	403 = página prohibida; 404 = no se encontró la página.
5xx	Error del servidor	500 = error interno del servidor; 503 = intentar más tarde.

Figura 7-38. Los grupos de respuesta del código de estado.

## ▼ General

Request URL: <https://www.icesi.edu.co/>

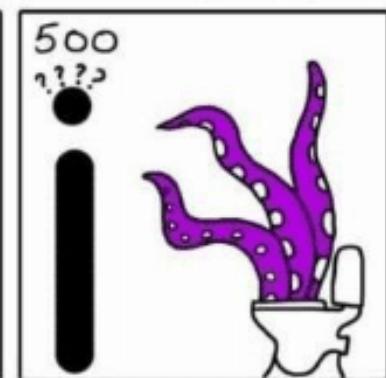
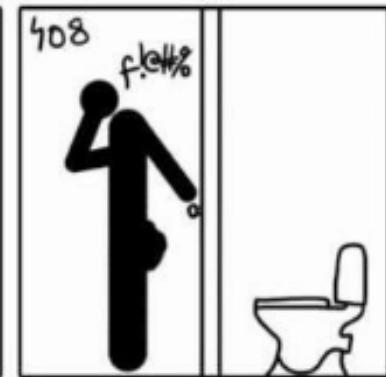
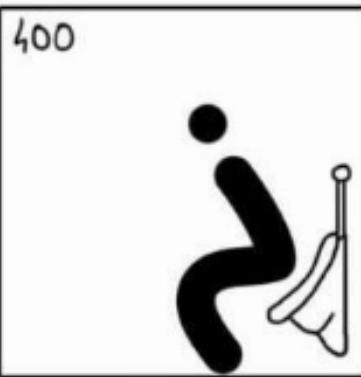
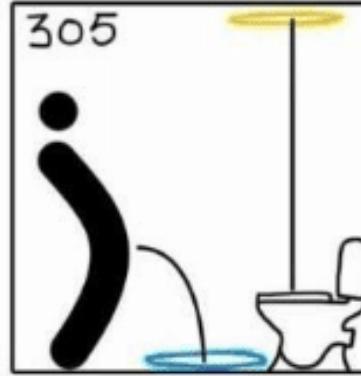
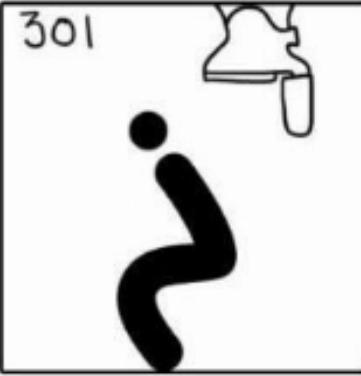
Request Method: GET

Status Code: ● 200 OK

Remote Address: 200.3.192.46:443

Referrer Policy: no-referrer-when-downgrade

## HTTP STATUS CODES



```

GET / HTTP/1.1
Host: www.icesi.edu.co
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,es;q=0.8
Cookie: __ga=GA1.3.2006869504.1524161431; __atuvc=1%7C16; 923cee6bec1c49638b8e40cffc911e13=es-ES; b4691f0f8df12a21e4071f350e754ab9=en-620.1537548397; amplitude_id_06d3064c4b7aa1187920f7d63c355898icesi.edu.co=eyJkZXZpY2VJZCI6IjV1Y2I0MGQ3LWY2YzEtNDQzNy05NzQwLwQzNmVkJT

```

---

Encabezado	Tipo	Contenidos
User-Agent	Solicitud	Información sobre el navegador y su plataforma.
Accept	Solicitud	El tipo de páginas que puede manejar el cliente.
Accept-Charset	Solicitud	Los conjuntos de caracteres que son aceptables para el cliente.
Accept-Encoding	Solicitud	Las codificaciones de página que puede manejar el cliente.
Accept-Language	Solicitud	Los idiomas naturales que puede manejar el cliente.
If-Modified-Since	Solicitud	Hora y fecha para verificar la actualidad de un mensaje.
If-None-Match	Solicitud	Etiquetas enviadas previamente para verificar la actualidad de un mensaje.
Host	Solicitud	El nombre DNS del servidor.
Authorization	Solicitud	Una lista de las credenciales del cliente.
Referer	Solicitud	El URL anterior desde el cual provino la solicitud.
Cookie	Solicitud	La cookie establecida previamente que se regresa al servidor.

▼ Response Headers [view parsed](#)

HTTP/1.1 200 OK  
Date: Fri, 09 Nov 2018 22:09:29 GMT  
Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips PHP/7.1.11  
X-Powered-By: PHP/7.1.11  
Expires: Wed, 17 Aug 2005 00:00:00 GMT  
Last-Modified: Fri, 09 Nov 2018 22:09:29 GMT  
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0  
Pragma: no-cache  
Vary: Accept-Encoding  
Content-Encoding: gzip  
Content-Length: 21493  
Connection: close  
Content-Type: text/html; charset=utf-8

Content-Range	Respuesta	Identifica una parte del contenido de la página.
Last-Modified	Respuesta	Hora y fecha de la última modificación de la página.
Expires	Respuesta	Hora y fecha en que la página dejará de ser válida.
Location	Respuesta	Indica al cliente a dónde enviar su solicitud.
Accept-Ranges	Respuesta	Indica que el servidor aceptará solicitudes de rango de bytes.
Date	Ambas	Fecha y hora en que se envió el mensaje.
Range	Ambas	Identifica una parte de una página.
Cache-Control	Ambas	Directivas para manejar las cachés.
ETag	Ambas	Etiqueta para el contenido de la página.
Upgrade	Ambas	El protocolo al que el emisor desea comutar.

# DEBERES:

- Use postman para hacer peticiones http hacia algun REST API publico
- Defina REST API
- Investigue sobre GraphQL
- Instalación de docker y ejecución de docker run hello-world \*