



UAEM | Universidad Autónoma
del Estado de México

Centro Universitario UAEM Zumpango
Ingeniería en Computación

Servicios de Internet
M.T.I. Carlos Alberto Rojas Hernández



Julio 2016



Identificación de la Unidad de Aprendizaje (UA)

Nombre UA:

Servicios de Internet (L41081)

Total de horas a la semana: **3** Créditos: **4**

Carácter de la UA: Optativa

Modalidad: **Presencial**

UA Antecedente: **Ninguna**

UA Consecuente: **Ninguna**





Presentación del programa

Una de las principales actividades que tiene que desarrollar y enfrentarse el ingeniero en computación, es el conocer a la perfección los elementos con los cuales trabajara y desarrollara la mayoría de sus actividades en su vida profesional, que es el conocimiento, de las herramientas en equipo y sus respectivas instalaciones, así como los servicios que se puedan derivar de ello, como lo es en el área de redes el poder medir el alcance de dichas herramientas utilizando estándares y normas establecidas que se encuentran implantadas en ese ámbito, como lo son los servicios que se pueden brindar en Internet, ya que es la red de redes que se establece mundialmente y es una herramienta de gran uso en todos los ambientes.



Propósito de la unidad de aprendizaje

El alumno:

Aplicará el conocimiento adquirido en la teoría complementándolo con el del laboratorio e investigación, para el desarrollo de herramientas que se puedan desarrollar o bien implementar en los servicios de Internet utilizando técnicas, normas y estándares, las cuales le darán información sobre el desarrollo de herramientas que sean capaces de mantener tanto al alumno como al profesor inmersos en la red de redes más extensa a nivel mundial.



Unidades de competencia

1. Reconocer los diferentes S. O. de un servidor, enrutadores, switches, access points, PBX, acopladores de medios y todos los periféricos que integra una red.
2. Conocer los diferentes configuraciones de servidores, los cuales darán servicio de acceso a Internet en sus diferentes modalidades, basando en normas y estándares.
3. Establecer configuraciones de diferentes servidores como son de correo, de web, de asignación de direcciones IP, etc.
4. Establecer configuraciones de diferentes servidores, establecer redes virtuales en el entorno de red, monitorear dicha red y poder realizar pruebas de rendimiento a los diferentes servicios establecidos en el ambiente de Internet.



Unidad de competencia III

Propósito de la unidad

Establecer configuraciones de diferentes servidores como son de correo, **de web**, de asignación de IPs, etc.



¿ Internet o WWW ?

- La parte gráfica de Internet, lo que conocemos como World Wide Web o W3 comenzó al principio de los 90.
- Hasta entonces el sistema de comunicación era bastante rudimentario usando el correo electrónico, el Telnet para acceder a máquinas remotas, para acceder a los archivos situados en directorios compartidos.
- Todo esto se realizaba a través de HERMOSA pantalla negra de texto de un terminal UNIX, o una PC que emulase dicho terminal.



¿ Internet o WWW ?

```
Telnet www.google.com
Welcome to Microsoft Telnet Client
Escape Character is 'CTRL+I'
Microsoft Telnet> help
Commands may be abbreviated. Supported commands are:
c      - close           close current connection
d      - display        display operating parameters
o      - open hostname [port] connect to hostname (default port 23).
q      - quit           exit telnet
set    - set            set options (type 'set ?' for a list)
sen    - send           send strings to server
st     - status         print status information
u      - unset          unset options (type 'unset ?' for a list)
?/h   - help           print help information
Microsoft Telnet> open www.google.com
Connecting To www.google.com..._
```




¿ Internet o WWW ?

HTTP: Hyper Text Transfer Protocol

- Modelo petición/respuesta
- No tiene estado
- Desarrollado en el Laboratorio de física de partículas europeo (CERN) por Tim Berners Lee
 - Los desarrollos posteriores los realiza el W3C
- Distribuido: La información esta repartida en páginas no muy grandes enlazadas entre sí.



¿ Internet o WWW ?





World Wide Web

Tim Berners-Lee es el creador de lo que ahora se conoce como World Wide Web, en 1989 cuando trabajaba para la CERN propuso un proyecto global de hipertexto, el cual permitiría a las personas trabajar juntas y combinar sus conocimientos en una red de documentos de hipertexto.



<http://www.masterlink.com/blog/wp-content/uploads/2012/08/sir-tim-berners-lee.jpg>

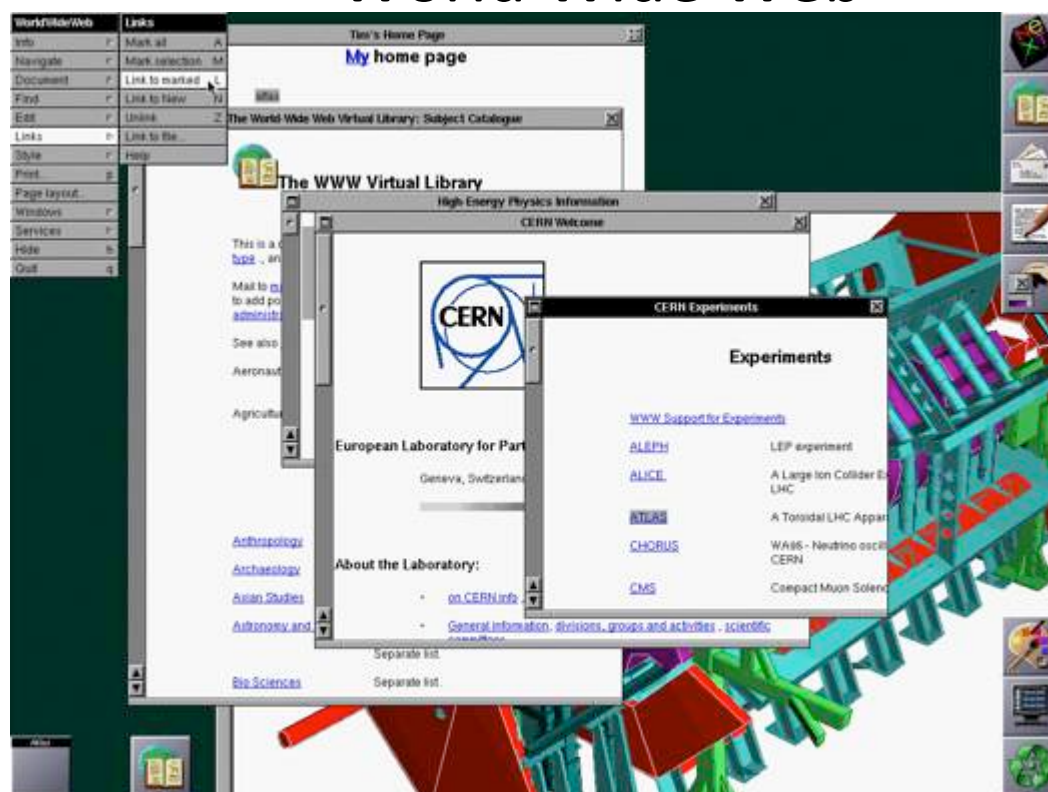


World Wide Web

Durante el mes de octubre del año 1990 inicio su trabajo usando una computadora NeXT, desarrollo el primer servidor web llamado httpd , y el primer cliente o navegador (browser) denominado simplemente WorldWideWeb, posteriormente renombrado Nexus para evitar confusiones con el concepto WWW.



World Wide Web



http://www.thegraphicmac.com/wp-content/uploads/web_history-nexus-browser.jpg





World Wide Web

La primera versión de utilidad práctica fue presentada el 25 de diciembre de 1990. En marzo 1991, el programa se distribuyó entre los compañeros de Berners-Lee en el CERN. En agosto, el público en general pudo acceder a la aplicación a través de los grupos de noticias.



World Wide Web

El RFC original que especifica el HTTP es RFC 2068 de enero de 1997, posteriormente se realiza una actualización en el RFC 2616 de junio de 1999.

Además existe diversas implementaciones en los RFC 2817, 5785, 6266 y 6585

En 2014, la W3C reconoce a las RFC del 7239 – 7237 como remplazo del original.



World Wide Web

El 30 de abril de 1993 el CERN liberó el código fuente de WorldWideWeb bajo el dominio público, convirtiéndolo en software libre

Inicialmente Berners-Lee consideró liberarlo bajo la GNU GPL pero finalmente escogió el dominio público para asegurar un mayor soporte corporativo



World Wide Web

Una página Web, más comúnmente llamada página de Internet (incorrecto) utiliza el puerto 80 de TCP y esta compuesta por objetos, estos pueden ser:

Un archivo HTML(Hypertext Mark Language) base, que incluye varios objetos referenciados

```
<html>
<body>
<!-- following code added by server. PLEASE REMOVE -->
<script language="JavaScript"
src="http://mx.geocities.com/js_source/ygIELib9.js?v3"></script><script
language="JavaScript">var
yviContents='http://geocities.yahoo.com/themis/as.php?
cc=mx&s=382036083&l=NE&b=1&t=1252640010',yviR='mx',yfiEA(0);</script>
<!-- preceding code added by server. PLEASE REMOVE -->

<!--<CENTER>

<IMG SRC="Libro.JPG" WIDTH=260 HEIGHT=149 BORDER=0 >

</CENTER> -->

<TABLE align=left>
<TBODY>
<TR>
<TD>
<p align=center>
```





World Wide Web

Una página Web, más comúnmente llamada página de Internet (incorrecto) utiliza el puerto 80 de TCP y esta compuesta por objetos, estos pueden ser:

Una imagen (JPEG , GIF, PNG, etc.).

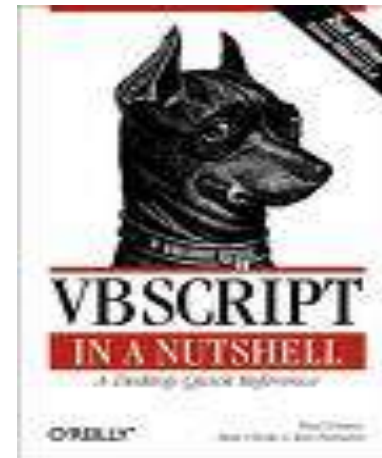




World Wide Web

Una página Web, más comúnmente llamada página de Internet (incorrecto) utiliza el puerto 80 de TCP y esta compuesta por objetos, estos pueden ser:

Un applet de Java, Visual Basic.





World Wide Web

Una página Web, más comúnmente llamada página de Internet (incorrecto) utiliza el puerto 80 de TCP y esta compuesta por objetos, estos pueden ser:

Un archivo de audio, video o animaciones.





World Wide Web

Para que cada objeto pueda ser conocido o referenciado es necesario, que exista una forma de conocer la ubicación en la que se encuentra, por lo que se utiliza un URL (Uniform Resource Locator)

Esto es una cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la página o en general en Internet, un ejemplo de URL es el siguiente:

http://sites.google.com/ico_cuz/ico_cuz.JPG



World Wide Web

http://sites.google.com/ico_cuz/ico_cuz.JPG

Consta de:

- Protocolo: necesario siempre
- Nombre de dominio: equipo en el que se encuentra el servidor, login y password (Opcional)
- Número de puerto: Opcional (se asume el valor por defecto)
- - Localización: directorio y archivo (Opcional)



World Wide Web

Una característica importante de HTTP es que un protocolo “stateless” (traducción literal desnaturalizado), es decir, el servidor no mantiene información sobre las solicitudes anteriores del cliente.



World Wide Web

Debido a lo anterior se hace uso de las “cookies”, de acuerdo a Netscape es:

"A general mechanism which server side connections (such as CGI scripts) can use to both store and retrieve information on the client side of the connection."

<http://www.cs.uccs.edu/~cs401/cookie.html>





World Wide Web

La cual no es más que un archivo de texto que algunos servidores piden a nuestro navegador (browser) que escriba en nuestro disco duro

```
0: HTTP/1.1 200 OK
1: Date: Tue, 15 Oct 2002 17:20:59 GMT
2: Server: Apache/1.3.26 (Unix) Debian GNU/Linux mod_gzip/1.3.19.1a PHP/4.1.2 v2h/1.5.1
3: X-Powered-By: PHP/4.1.2
4: Set-Cookie: lang=spanish; expires=Wed, 15-Oct-03 17:20:59 GMT
5: Set-Cookie: fXA[0]=1; expires=Tue, 15-Oct-02 18:20:59 GMT
6: Set-Cookie: fXA[3]=__News__; expires=Tue, 15-Oct-02 18:20:59 GMT
7: Connection: close
8: Content-Type: text/html
```





World Wide Web

Contienen información acerca de lo que hemos estado haciendo por sus páginas, y que proporciona una manera de conservar datos entre peticiones del cliente, extendiendo significativamente las capacidades de las aplicaciones cliente/servidor basadas en Web





World Wide Web

Las “cookies”, ¿son útiles?

Si son útiles ya que permiten a los sitios de Web, aprender algunas cosas acerca de los “gustos” o preferencias de aquel que accesa a la página.



World Wide Web

Las “cookies”, ¿son útiles?

Autorización: Verificar que ya se entro con anterioridad





World Wide Web

Las “cookies”, ¿son útiles?

Carros de compras: En la posible compra de algún producto es posible “guardarlo” y realizar la compra posteriormente





World Wide Web

Las “cookies”, ¿son útiles?

Estado de la sesión de usuario: En el caso del correo electrónico.

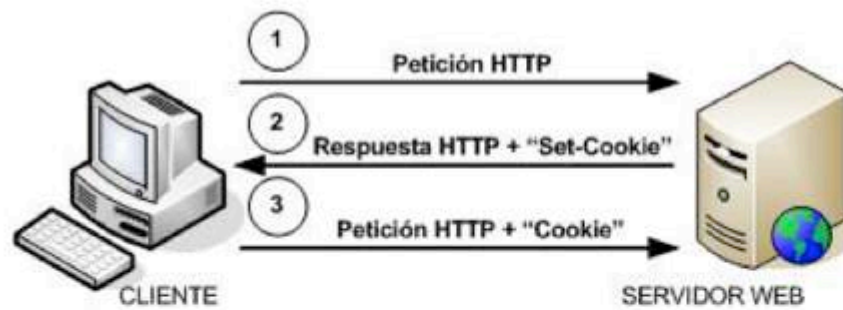
Acceda a Gmail con su
Cuenta de Google

Nombre de usuario:

Contraseña:

No cerrar sesión

[¿No puedes acceder a tu cuenta?](#)



1 Petición HTTP

```
GET /HTTP/1.1
Host: www.webserver.com
```

2 Respuesta HTTP + "Set-Cookie"

```
HTTP/1.1 200 OK
Date: Thu, 31 Jul 2006 18:05 GMT
Content-Length: 92
Connection: close
Content-Type: text/html
Set-Cookie: usuario=pepito; dominio=local; path=/home

<html>
<title>Ejemplo</title>
<h1>Ejemplo de Cookies</h1>
</html>
```

3 Petición HTTP + "Cookie"

```
GET /HTTP/1.1
Host: www.webserver.com
Cookie: usuario=pepito
```





World Wide Web

Las “cookies”, ¿son útiles?

Pero ...

Permiten a los sitios conocer cierto “gustos” o preferencias, algunas veces esto no es completamente deseable, por ejemplo, transacciones bancarias, compras en línea, etc.



World Wide Web

Las “cookies”, ¿son útiles?

Pero ...

Las motores de búsqueda utilizan redireccionamiento y las “cookies” aprenden aún más, es decir, que sitios que visitaste durante tu última sesión



World Wide Web

Las “cookies”, ¿son útiles?

Pero ...

Las compañías de publicidad obtienen información a través de los sitios web

(Spam ☹)



World Wide Web

El proceso que sigue una conexión HTTP es:

1. El cliente inicia la conexión TCP, se crea el socket al servidor (dirección IP + puerto 80).
2. El servidor acepta la conexión TCP solicitada por cliente.
3. Los mensajes HTTP que son mensajes entre las capas de aplicación se intercambian entre el navegador (browser o cliente HTTP) y el servidor Web (servidor HTTP).
4. Se cierra la conexión TCP.



World Wide Web

Debido a que la comunicación entre el cliente y el servidor se debe de realizar de una forma fácil y rápida HTTP utiliza mensajes cortos para indicar las acciones que solicitara el “Cliente” y la respuesta que darán los “Servidores”, estos tipos de mensajes son: request (Solicitud o petición) y response (Respuesta)



World Wide Web

Formato general de un mensaje de petición o solicitud (Request) por el Cliente

Método			URL		Versión		/n
Nombre Campo1 Cabecera	:		Valor		/n		
Nombre Campo2 Cabecera	:		Valor		/n		
	:		:		:		
	:		:		:		
	/n						
Cuerpo de entidad							



World Wide Web

Formato general de un mensaje de petición o solicitud (Request) por el Cliente

Un ejemplo de un mensaje de petición de HTTP, es el siguiente:

```
GET /dir/pagina.html HTTP/1.1
```

```
Host: www.uaemex.mx
```

```
Connetion: close
```

```
User-agent: Mozilla/4.0
```

```
Accept-language: es/mx
```



World Wide Web

Los métodos más utilizados en la emisión de mensajes de petición son:

Para HTTP 1.0

- GET: es utilizado cuando el navegador demanda un objeto que identifica con el campo URL, para este caso no es utilizado el campo “cuerpo de entidad”



World Wide Web

Los métodos más utilizados en la emisión de mensajes de petición son:

Para HTTP 1.0

- POST: es utilizado principalmente cuando se envía alguna información hacia el servidor web, por ejemplo al rellenar un formulario o realiza le pide a un sitio web realizar una búsqueda.



World Wide Web

Los métodos más utilizados en la emisión de mensajes de petición son:

Para HTTP 1.0

- HEAD: es utilizado cuando el navegador demanda un objeto, similar al método GET, sin embargo el mensaje de respuesta no incluye el objeto, es principalmente utilizado en las aplicaciones de depuración de las página o aplicaciones asociadas (applets)



World Wide Web

Los métodos más utilizados en la emisión de mensajes de petición son:

Para HTTP 1.1

- PUT: es utilizado en aplicaciones que requieren guardar algún objeto en el servidor web.
- DELETE: es utilizado cuando un usuario o aplicación requiere borrar algún objeto del servidor web (si tiene los privilegios suficientes).



World Wide Web

Formato general de un mensaje de respuesta (Response) por el Servidor

Versión			Código de estado		Frase		/n
Nombre Campo1	Cabecera	:	Valor		/n		
Nombre Campo2	Cabecera	:	Valor		/n		
	:		:		:		
	:		:		:		
	/n						
Cuerpo de entidad (Datos)							



World Wide Web

Un ejemplo de un mensaje de respuesta de HTTP, es el siguiente:

HTTP/1.1 404 Not Found

Date: Fri, 21 Apr 2006 04:38:55 GMT

Server: Apache-1.3.29 <Unix> PHP-4.3.1 mod?ssl-2.8.16

OpenSSL-0.9.6i

Connetion: close

Content-Type: text-html; charset=iso-8859-1



World Wide Web

El código de estado es un número de 3 dígitos que indica si la petición ha sido atendida satisfactoriamente o no, y en caso de no haber sido atendida, indica la causa. Los códigos se dividen en cinco clases definidas por el primer dígito del código de estado



World Wide Web

- 1xx: Informativo. La petición se recibe y sigue el proceso. Esta familia de respuestas indican una respuesta provisional. Este tipo de respuesta está formada por la línea de estado y las cabeceras. Un servidor envía este tipo de respuesta en casos experimentales.



World Wide Web

- 2xx: Éxito. La acción requerida por la petición ha sido recibida, entendida y aceptada.
- 3xx: Redirección. Para completar la petición se han de tomar más acciones.



World Wide Web

- 4xx: Error del cliente. La petición no es sintácticamente correcta y no se puede llevar a cabo.
- 5xx: Error del servidor. El servidor falla al atender la petición que aparentemente es correcta.



World Wide Web

Algunos códigos de estado de HTTP, son los siguientes:

- 200 OK: Solicitud exitosa, el objeto solicitado va en este mensaje.
- 301 Moved Permanently: El objeto solicitado fue movido, la nueva ubicación se especifica posteriormente en este mensaje.
- 400 Bad Request: El mensaje de solicitud no fue entendido por el servidor.
- 404 Not Found: El documento solicitado no se encontró en este servidor.
- 505 HTTP Version Not Supported: Versión no soportada



World Wide Web

```
C:/> telnet www.uaemex.mx 80
Telnet> GET/index.html HTTP/1.1
```

```
C:\WINDOWS\system32\cmd.exe

HTTP/1.1 404 Not Found
Date: Fri, 21 Apr 2006 04:38:55 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.1 mod_ssl/2.8.16 OpenSSL/0.9.6i
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /index.html was not found on this server.<P>
<HR>
<ADDRESS>Apache/1.3.29 Server at www.uaemex.mx Port 80</ADDRESS>
</BODY></HTML>

C:\Documents and Settings\new user>
```



World Wide Web

Existen 2 tipos de conexiones HTTP:

HTTP no persistente

- HTTP/1.0 utiliza HTTP no persistente

Al menos un objeto es enviado sobre una conexión TCP



World Wide Web

HTTP no persistente

Supongamos que el usuario ingresa el URL en el navegador:

<http://sites.google.com/site/carojash/fotos.html>

El cual contiene un archivo HTML donde se hace referencia a las fotos tomadas en alguna clase



HTTP No persistente



El servidor HTTP en <http://sites.google.com/site/carojash/fotos.html> espera conexiones TCP en el puerto 80.
(Recuerda que el servidor siempre esta en estado de LISTEN o escucha)



HTTP No persistente



1. El cliente HTTP solicita e inicia la conexión TCP al servidor HTTP en <http://sites.google.com/site/carojash/fotos.html> en el puerto 80



HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.739130	192.168.0.100	192.168.0.1	DNS	Standard query A sites.google.com
3	0.907137	192.168.0.1	192.168.0.100	DNS	Standard query response CNAME www3.l.google.com A 74.125.
4	0.923145	192.168.0.100	www3.l.google.	TCP	rmlnk > http [SYN] Seq=0 Win=16384 Len=0 MSS=1460
5	1.003933	192.168.0.1	Broadcast	ARP	who has 192.168.0.119? Tell 192.168.0.1
6	1.117366	www3.l.google.	192.168.0.100	TCP	http > rmlnk [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=14
7	1.117435	192.168.0.100	www3.l.google.	TCP	rmlnk > http [ACK] Seq=1 Ack=1 win=17160 Len=0
8	1.117775	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/ HTTP/1.1
9	1.319190	www3.l.google.	192.168.0.100	TCP	http > rmlnk [ACK] Seq=1 Ack=824 win=6584 Len=0
10	1.768058	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
11	1.775479	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]

Ethernet II, Src: IntelCor_4b:09:ca (00:15:00:4b:09:ca), Dst: 192.168.0.1 (00:17:3f:40:36:38)
 Internet Protocol, Src: 192.168.0.100 (192.168.0.100), Dst: www3.l.google.com (74.125.65.100)
 Transmission Control Protocol, Src Port: rmlnk (2818), Dst Port: http (80), Seq: 0, Len: 0
 Source port: rmlnk (2818)
 Destination port: http (80)
 [Stream index: 1]
 Sequence number: 0 (relative sequence number)
 Header length: 28 bytes
 Flags: 0x02 (SYN)
 Window size: 16384
 Checksum: 0xce96 [validation disabled]
 Options: (8 bytes)

```

0000  00 17 3f 40 36 38 00 15 00 4b 09 ca 08 00 45 00  ..?@68.. .K....E.
0010  00 30 41 72 40 00 80 06 6c 68 c0 a8 00 64 4a 7d  .0Ar@... lh...dJ}
0020  41 64 0b 02 00 50 27 cb f4 7d 00 00 00 00 70 02  Ad...P'. .}....p.
0030  40 00 ce 96 00 00 02 04 05 b4 01 01 04 02      @
  
```

File: "C:\Documents and Settings\Carlos\Desktop\... Packets: 271 Displayed: 271 Marked: 0 Profile: Default

Windows taskbar showing Start button, Windows Media Player, Diapositivas, OpenOffice.org, Untitled - Notepad, HTTP.pcap - Wire..., and untitled - Paint. System tray shows ES, network, volume, and time 09:39 p.m.



HTTP No persistente



2. Si el servidor acepta la conexión notifica al cliente.

HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.739130	192.168.0.100	192.168.0.1	DNS	Standard query A sites.google.com
3	0.907137	192.168.0.1	192.168.0.100	DNS	Standard query response CNAME www3.l.google.com A 74.125.
4	0.923145	192.168.0.100	www3.l.google.	TCP	rmlnk > http [SYN] Seq=0 win=16384 Len=0 MSS=1460
5	1.003933	192.168.0.1	Broadcast	ARP	Who has 192.168.0.119? Tell 192.168.0.1
6	1.117366	www3.l.google.	192.168.0.100	TCP	http > rmlnk [SYN, ACK] Seq=0 Ack=1 win=5720 Len=0 MSS=14
7	1.117435	192.168.0.100	www3.l.google.	TCP	rmlnk > http [ACK] Seq=1 Ack=1 win=17160 Len=0
8	1.117775	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/ HTTP/1.1
9	1.319190	www3.l.google.	192.168.0.100	TCP	http > rmlnk [ACK] Seq=1 Ack=824 win=6584 Len=0
10	1.768058	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
11	1.775479	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]

Ethernet II, Src: 192.168.0.1 (00:17:3f:40:36:38), Dst: IntelCor_4b:09:ca (00:15:00:4b:09:ca)

Internet Protocol, Src: www3.l.google.com (74.125.65.100), Dst: 192.168.0.100 (192.168.0.100)

Transmission Control Protocol, Src Port: http (80), Dst Port: rmlnk (2818), Seq: 0, Ack: 1, Len: 0

Source port: http (80)
 Destination port: rmlnk (2818)
 [Stream index: 1]
 Sequence number: 0 (relative sequence number)
 Acknowledgement number: 1 (relative ack number)
 Header length: 28 bytes

Flags: 0x12 (SYN, ACK)
 Window size: 5720
 Checksum: 0x94dc [validation disabled]

```

0000  00 15 00 4b 09 ca 00 17 3f 40 36 38 08 00 45 00  ...K.... ?@68..E.
0010  00 30 9c c0 00 00 35 06 9c 1a 4a 7d 41 64 c0 a8  .0....5. ..J}Ad..
0020  00 64 00 50 0b 02 79 a5 e9 c9 27 cb f4 7e 70 12  .d.P..y. ..'...~p.
0030  16 58 94 dc 00 00 02 04 05 96 01 01 04 02      y
  
```

File: C:\Documents and Settings\Carlos\Desktop\... Packets: 271 Displayed: 271 Marked: 0 Profile: Default

start Windows Media Pl... Diapositivas OpenOffice.or... Untitled - Notepad HTTP.pcap - Wire... untitled - Paint ES 09:40 p.m.



HTTP No persistente



CLIENTE



SERVIDOR WEB



3. El cliente HTTP envía un mensaje de solicitud (request message) que contiene el URL del objeto hacia su socket de conexión TCP. El mensaje indica que el cliente desea el objeto
`http://sites.google.com/site/carojash/fotos.html`

HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.739130	192.168.0.100	192.168.0.1	DNS	Standard query A sites.google.com
3	0.907137	192.168.0.1	192.168.0.100	DNS	Standard query response CNAME www3.l.google.com A 74.125.
4	0.923145	192.168.0.100	www3.l.google.	TCP	rmlnk > http [SYN] Seq=0 win=16384 Len=0 MSS=1460
5	1.003933	192.168.0.1	Broadcast	ARP	who has 192.168.0.119? Tell 192.168.0.1
6	1.117366	www3.l.google.	192.168.0.100	TCP	http > rmlnk [SYN, ACK] Seq=0 Ack=1 win=5720 Len=0 MSS=14
7	1.117435	192.168.0.100	www3.l.google.	TCP	rmlnk > http [ACK] Seq=1 Ack=1 Win=17160 Len=0
8	1.117775	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/ HTTP/1.1
9	1.319190	www3.l.google.	192.168.0.100	TCP	http > rmlnk [ACK] Seq=1 Ack=824 win=6584 Len=0
10	1.768058	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
11	1.775479	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]

Transmission Control Protocol, Src Port: rmlnk (2818), Dst Port: http (80), Seq: 1, Ack: 1, Len: 823

Hypertext Transfer Protocol

GET /site/carojash/ HTTP/1.1\r\n

Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, application/vr\n

Accept-Language: es-mx\r\n

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; msn Opt\r\n

Accept-Encoding: gzip, deflate\r\n

If-Modified-Since: Sun, 15 Nov 2009 04:05:44 GMT\r\n

Host: sites.google.com\r\n

Connection: Keep-Alive\r\n

[truncated] Cookie: SitesNav2bd=gx:b9ae51490cbc71e; rememberme=false; PREF=ID=78455a4d0ad6d31c:U=e988c

0030 43 08 fc ec 00 00 47 45 54 20 2f 73 69 74 65 2f C....GE T /site/
 0040 63 61 72 6f 6a 61 73 68 2f 20 48 54 54 50 2f 31 carojash / HTTP/1
 0050 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d 61 67 .1..Acce pt: imag
 0060 65 2f 67 69 66 20 20 69 6d 61 67 65 2f 63 70 65 e/gif image/ine

Text item 0, 30 bytes Packets: 271 Displayed: 271 Marked: 0 Profile: Default

start Windows Media Pl... Correoe 2 OpenOffice.or... Untitled - Notepad HTTP.pcap - Wire... untitled - Paint ES 09:45 p.m.



HTTP No persistente



4. El servidor HTTP recibe el mensaje de solicitud, construye un mensaje de respuestas (response message) que contiene el objeto solicitado, y envía el mensaje hacia su socket.

HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
19	1.955096	192.168.0.100	www3.l.google.	TCP	fc-faultnotify > http [SYN] Seq=0 Win=16384 Len=0 MSS=146
20	1.968918	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
21	1.968962	192.168.0.100	www3.l.google.	TCP	rmInk > http [ACK] Seq=824 Ack=8873 Win=17160 Len=0
22	1.969402	www3.l.google.	192.168.0.100	HTTP	HTTP/1.1 200 OK (text/html)
23	2.007905	192.168.0.1	Broadcast	ARP	Who has 192.168.0.120? Tell 192.168.0.1
24	2.101179	192.168.0.100	www3.l.google.	TCP	rmInk > http [ACK] Seq=824 Ack=9031 Win=17002 Len=0
25	2.101704	www3.l.google.	192.168.0.100	TCP	http > fc-faultnotify [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0
26	2.101738	192.168.0.100	www3.l.google.	TCP	fc-faultnotify > http [ACK] Seq=1 Ack=1 Win=17160 Len=0
27	2.101945	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/_/rsrc/1258057443297/system/app/css/ov
28	2.288884	www3.l.google.	192.168.0.100	TCP	http > fc-faultnotify [ACK] Seq=1 Ack=756 Win=6795 Len=0

Reassembled TCP segments (9030 bytes): #19(9030), #21(17160), #23(17160), #25(17160), #27(17160), #28(17160)

Hypertext Transfer Protocol

Line-based text data: text/html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">\n
<head>\n
<script type="text/javascript">/* Copyright 2008 Google. */ (function() { // Input 0\n
var a=window;\n

```

0000	48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d	HTTP/1.1 200 OK.
0010	0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74	.Content -Type: t
0020	65 78 74 2f 68 74 6d 6c 3b 20 63 68 61 72 73 65	ext/html ; charse
0030	74 3d 75 74 66 2d 38 0d 0a 58 2d 46 72 61 6d 65	t=utf-8. .X-Frame
0040	2d 4f 70 74 69 6f 6e 73 3a 20 41 4c 4c 4f 57 41	-Options : ALLOWA
0050	4c 4c 0d 0a 4c 61 73 74 2d 4d 6f 64 69 66 69 65	LL..Last -Modifie
0060	64 3a 20 53 75 6e 2c 20 31 35 20 4e 6f 76 20 32	d: Sun, 15 Nov 2

Frame (212 bytes) Reassembled TCP (9030 bytes) De-chunked entity body (8637 bytes) Uncompressed entity body (28550 bytes)

Hypertext Transfer Protocol (http), 375 bytes Packets: 271 Displayed: 271 Marked: 0 Profile: Default



HTTP No persistente



5. El servidor HTTP cierra la conexión TCP.



HTTP No persistente



6.El cliente HTTP recibe el mensaje de respuesta que contiene el archivo HTML

Si se desea revisar o almacenar los “n” objetos se repiten los pasos anteriores “n” veces.



HTTP

Existen 2 tipos de conexiones HTTP:

HTTP persistente

- HTTP/1.1, utiliza conexiones persistentes
- Múltiples objetos pueden ser enviados sobre una misma conexión TCP.





HTTP Persistente

Al igual que en el ejemplo anterior, supongamos que el usuario ingresa el URL en el navegador:

<http://sites.google.com/site/carojash/fotos.html>

El cual contiene un archivo HTML donde se hace referencia a las fotos tomadas en alguna clase



HTTP Persistente



El servidor HTTP en <http://sites.google.com/site/carojash/fotos.html> espera conexiones TCP en el puerto 80.
(Recuerda que el servidor siempre esta en estado de LISTEN o escucha)



HTTP Persistente



1. El cliente HTTP solicita e inicia la conexión TCP al servidor HTTP en <http://sites.google.com/site/carojash/fotos.html> en el puerto 80



HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.739130	192.168.0.100	192.168.0.1	DNS	Standard query A sites.google.com
3	0.907137	192.168.0.1	192.168.0.100	DNS	Standard query response CNAME www3.l.google.com A 74.125.
4	0.923145	192.168.0.100	www3.l.google.	TCP	rmlnk > http [SYN] Seq=0 Win=16384 Len=0 MSS=1460
5	1.003933	192.168.0.1	Broadcast	ARP	who has 192.168.0.119? Tell 192.168.0.1
6	1.117366	www3.l.google.	192.168.0.100	TCP	http > rmlnk [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=14
7	1.117435	192.168.0.100	www3.l.google.	TCP	rmlnk > http [ACK] Seq=1 Ack=1 win=17160 Len=0
8	1.117775	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/ HTTP/1.1
9	1.319190	www3.l.google.	192.168.0.100	TCP	http > rmlnk [ACK] Seq=1 Ack=824 win=6584 Len=0
10	1.768058	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
11	1.775479	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]

⊕ Ethernet II, Src: IntelCor_4b:09:ca (00:15:00:4b:09:ca), Dst: 192.168.0.1 (00:17:3f:40:36:38)

⊕ Internet Protocol, Src: 192.168.0.100 (192.168.0.100), Dst: www3.l.google.com (74.125.65.100)

⊖ Transmission Control Protocol, Src Port: rmlnk (2818), Dst Port: http (80), Seq: 0, Len: 0

- Source port: rmlnk (2818)
- Destination port: http (80)
- [Stream index: 1]
- Sequence number: 0 (relative sequence number)
- Header length: 28 bytes
- ⊕ Flags: 0x02 (SYN)
- Window size: 16384
- ⊕ Checksum: 0xce96 [validation disabled]
- ⊕ Options: (8 bytes)

```

0000  00 17 3f 40 36 38 00 15 00 4b 09 ca 08 00 45 00  ..?@68.. .K....E.
0010  00 30 41 72 40 00 80 06 6c 68 c0 a8 00 64 4a 7d  .0Ar@... lh...dJ}
0020  41 64 0b 02 00 50 27 cb f4 7d 00 00 00 00 70 02  Ad...P'. .}....p.
0030  40 00 ce 96 00 00 02 04 05 b4 01 01 04 02      @

```

File: "C:\Documents and Settings\Carlos\Desktop\... Packets: 271 Displayed: 271 Marked: 0 Profile: Default

Windows taskbar showing Start button, Windows Media Player, Diapositivas, OpenOffice.org, Untitled - Notepad, HTTP.pcap - Wire..., untitled - Paint, and system tray with time 09:39 p.m.



HTTP persistente



2. Si el servidor acepta la conexión notifica al cliente.

HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.739130	192.168.0.100	192.168.0.1	DNS	Standard query A sites.google.com
3	0.907137	192.168.0.1	192.168.0.100	DNS	Standard query response CNAME www3.l.google.com A 74.125.
4	0.923145	192.168.0.100	www3.l.google.	TCP	rmlnk > http [SYN] Seq=0 win=16384 Len=0 MSS=1460
5	1.003933	192.168.0.1	Broadcast	ARP	Who has 192.168.0.119? Tell 192.168.0.1
6	1.117366	www3.l.google.	192.168.0.100	TCP	http > rmlnk [SYN, ACK] Seq=0 Ack=1 win=5720 Len=0 MSS=14
7	1.117435	192.168.0.100	www3.l.google.	TCP	rmlnk > http [ACK] Seq=1 Ack=1 Win=17160 Len=0
8	1.117775	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/ HTTP/1.1
9	1.319190	www3.l.google.	192.168.0.100	TCP	http > rmlnk [ACK] Seq=1 Ack=824 win=6584 Len=0
10	1.768058	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
11	1.775479	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]

Ethernet II, Src: 192.168.0.1 (00:17:3f:40:36:38), Dst: IntelCor_4b:09:ca (00:15:00:4b:09:ca)

Internet Protocol, Src: www3.l.google.com (74.125.65.100), Dst: 192.168.0.100 (192.168.0.100)

Transmission Control Protocol, Src Port: http (80), Dst Port: rmlnk (2818), Seq: 0, Ack: 1, Len: 0

Source port: http (80)

Destination port: rmlnk (2818)

[Stream index: 1]

Sequence number: 0 (relative sequence number)

Acknowledgement number: 1 (relative ack number)

Header length: 28 bytes

Flags: 0x12 (SYN, ACK)

Window size: 5720

Checksum: 0x94dc [validation disabled]

```

0000  00 15 00 4b 09 ca 00 17 3f 40 36 38 08 00 45 00  ...K.... ?@68..E.
0010  00 30 9c c0 00 00 35 06 9c 1a 4a 7d 41 64 c0 a8  .0....5. ..J}Ad..
0020  00 64 00 50 0b 02 79 a5 e9 c9 27 cb f4 7e 70 12  .d.P..y. ..'...~p.
0030  16 58 94 dc 00 00 02 04 05 96 01 01 04 02      y

```

File: C:\Documents and Settings\Carlos\Desktop\... Packets: 271 Displayed: 271 Marked: 0 Profile: Default

start Windows Media Pl... Diapositivas OpenOffice.or... Untitled - Notepad HTTP.pcap - Wire... untitled - Paint ES 09:40 p.m.



HTTP persistente



CLIENTE



SERVIDOR WEB



3. El cliente HTTP envía un mensaje de solicitud (request message) que contiene el URL del objeto hacia su socket de conexión TCP. El mensaje indica que el cliente desea el objeto
`http://sites.google.com/site/carojash/fotos.html`

The image shows a Wireshark window titled "HTTP.pcap - Wireshark". The packet list pane shows 11 packets. Packet 2 is a DNS standard query for sites.google.com. Packet 3 is the response. Packet 4 is a TCP SYN from rmlnk to www3.l.google.com. Packet 5 is an ARP broadcast. Packet 6 is a TCP SYN-ACK from www3.l.google.com to rmlnk. Packet 7 is a TCP ACK from rmlnk to www3.l.google.com. Packet 8 is an HTTP GET request for /site/carojash/. Packets 9 and 10 are TCP ACKs from www3.l.google.com to rmlnk. Packet 11 is a TCP segment of a reassembled PDU.

The packet details pane for packet 8 shows:
Transmission Control Protocol, Src Port: rmlnk (2818), Dst Port: http (80), Seq: 1, Ack: 1, Len: 823
Hypertext Transfer Protocol
GET /site/carojash/ HTTP/1.1\r\n
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, application/vr
Accept-Language: es-mx\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; msn Opt
Accept-Encoding: gzip, deflate\r\n
If-Modified-Since: Sun, 15 Nov 2009 04:05:44 GMT\r\n
Host: sites.google.com\r\n
Connection: Keep-Alive\r\n
[truncated] Cookie: SitesNav2bd=gx:b9ae51490cbc71e; rememberme=false; PREF=ID=78455a4d0ad6d31c:U=e988c

The packet bytes pane shows hex and ASCII data for the GET request.



HTTP persistente



4. El servidor HTTP recibe el mensaje de solicitud, construye un mensaje de respuestas (response message) que contiene el objeto solicitado, y envía el mensaje hacia su socket.

HTTP.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
19	1.955096	192.168.0.100	www3.l.google.	TCP	fc-faultnotify > http [SYN] Seq=0 Win=16384 Len=0 MSS=146
20	1.968918	www3.l.google.	192.168.0.100	TCP	[TCP segment of a reassembled PDU]
21	1.968962	192.168.0.100	www3.l.google.	TCP	rmInk > http [ACK] Seq=824 Ack=8873 Win=17160 Len=0
22	1.969402	www3.l.google.	192.168.0.100	HTTP	HTTP/1.1 200 OK (text/html)
23	2.007905	192.168.0.1	Broadcast	ARP	Who has 192.168.0.120? Tell 192.168.0.1
24	2.101179	192.168.0.100	www3.l.google.	TCP	rmInk > http [ACK] Seq=824 Ack=9031 Win=17002 Len=0
25	2.101704	www3.l.google.	192.168.0.100	TCP	http > fc-faultnotify [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0
26	2.101738	192.168.0.100	www3.l.google.	TCP	fc-faultnotify > http [ACK] Seq=1 Ack=1 Win=17160 Len=0
27	2.101945	192.168.0.100	www3.l.google.	HTTP	GET /site/carojash/_/rsrc/1258057443297/system/app/css/ov
28	2.288884	www3.l.google.	192.168.0.100	TCP	http > fc-faultnotify [ACK] Seq=1 Ack=756 Win=6795 Len=0

Reassembled TCP segments (9030 bytes): #19(9030), #21(1717), #23(1717), #25(1717), #26(1717), #28(1717)

Hypertext Transfer Protocol

Line-based text data: text/html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">\n
<head>\n
<script type="text/javascript">/* Copyright 2008 Google. */ (function() { // Input 0\n
var a=window;\n

```

0000	48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d	HTTP/1.1 200 OK.
0010	0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74	.Content -Type: t
0020	65 78 74 2f 68 74 6d 6c 3b 20 63 68 61 72 73 65	ext/html ; charse
0030	74 3d 75 74 66 2d 38 0d 0a 58 2d 46 72 61 6d 65	t=utf-8. .X-Frame
0040	2d 4f 70 74 69 6f 6e 73 3a 20 41 4c 4c 4f 57 41	-Options : ALLOWA
0050	4c 4c 0d 0a 4c 61 73 74 2d 4d 6f 64 69 66 69 65	LL..Last -Modifie
0060	64 3a 20 53 75 6e 2c 20 31 35 20 4e 6f 76 20 32	d: Sun, 15 Nov 2

Frame (212 bytes) Reassembled TCP (9030 bytes) De-chunked entity body (8637 bytes) Uncompressed entity body (28550 bytes)

Hypertext Transfer Protocol (http), 375 bytes Packets: 271 Displayed: 271 Marked: 0 Profile: Default



HTTP persistente



5. El cliente HTTP recibe el mensaje de respuesta que contiene el archivo HTML y los “n” objetos sobre la misma conexión TCP



HTTP persistente



5. El servidor HTTP cierra la conexión TCP.



Referencias bibliográficas

- [1] Andrew S. Tanenbaum, David J. Wetherall, “Redes de computadoras”, 5a edición, Pearson Educación, 2012.
- [2] James F. Kurose, Keith W. Ross, “Redes de computadores : un enfoque descendente basado en internet”, 2a edición, Pearson Educación, 2004.
- [3] Molina Francisco J., “Instalación y Mantenimiento de servicios de internet”, Editorial Alfa Omega, 2007.



Referencias bibliográficas

[4] Página Web Tim Berners-Lee ,”Email Home”, <https://www.w3.org/People/Berners-Lee/#Bio>

[5] RFC 2068, “Hypertext Transfer Protocol - HTTP/1.1”, <https://www.ietf.org/rfc/rfc2068.txt>

[6] RFC 2616, ” Hypertext Transfer Protocol - HTTP/1.1”, <https://www.ietf.org/rfc/rfc2616.txt>



HUMANISMO QUE TRANSFORMA

www.uaemex.mx