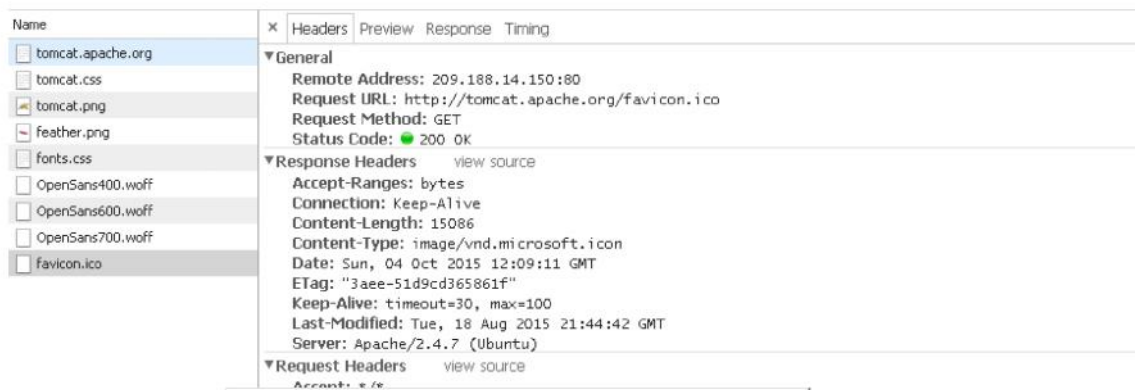


Hit Ctrl+R to capture frames.



## Práctica 2.1: Protocolo HTTP

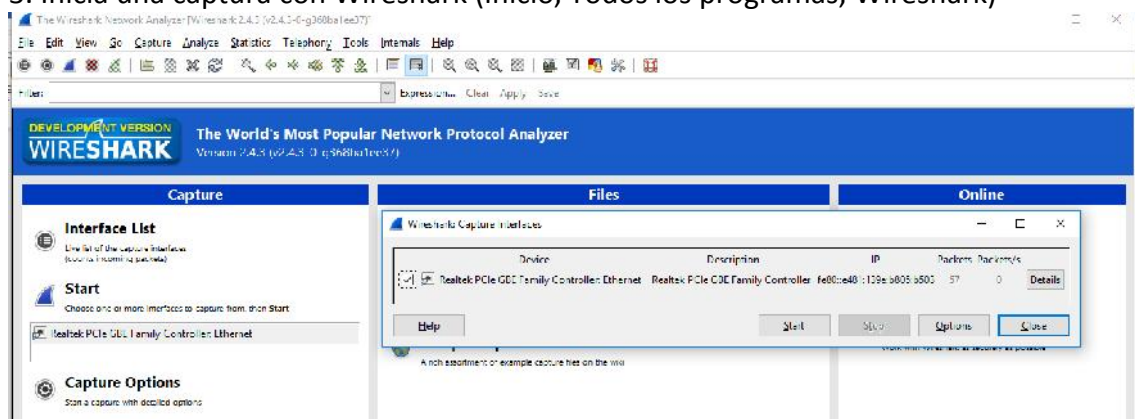
### Objetivo:

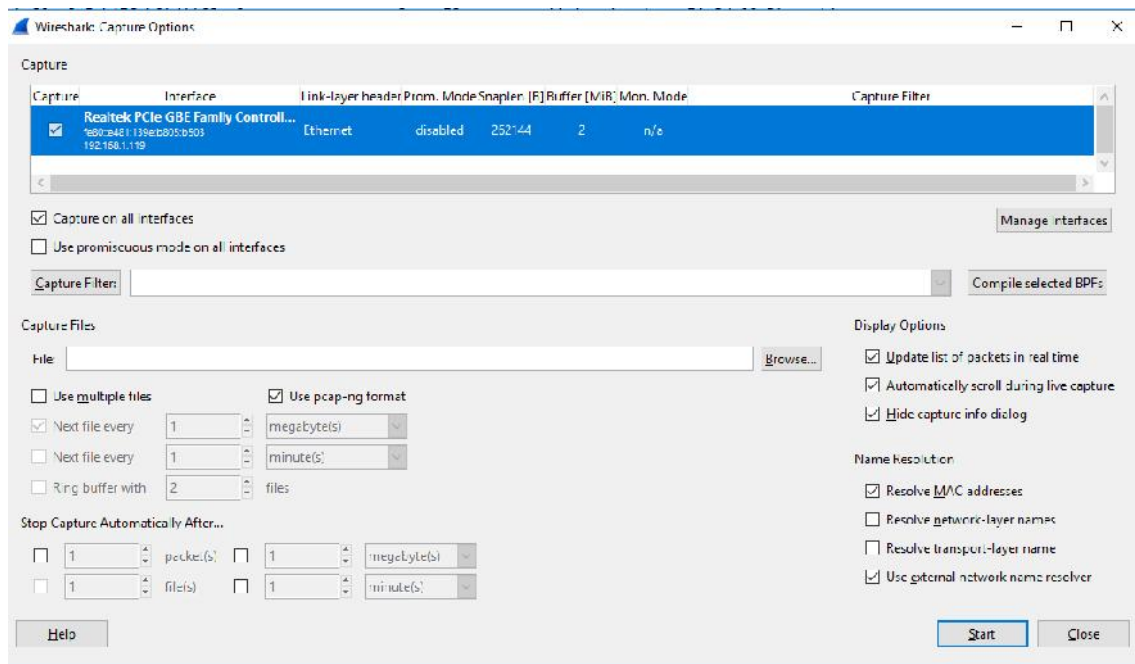
En esta práctica se analiza la información de los mensajes de petición y respuesta del protocolo HTTP.

### Pasos Previos:

Instalar el programa wireshark, cuyo ejecutable está en FTP.

1. Inicia sesión en Windows7.
2. Abre el navegador Google Chrome.
3. Inicia una captura con Wireshark (Inicio, Todos los programas, Wireshark)





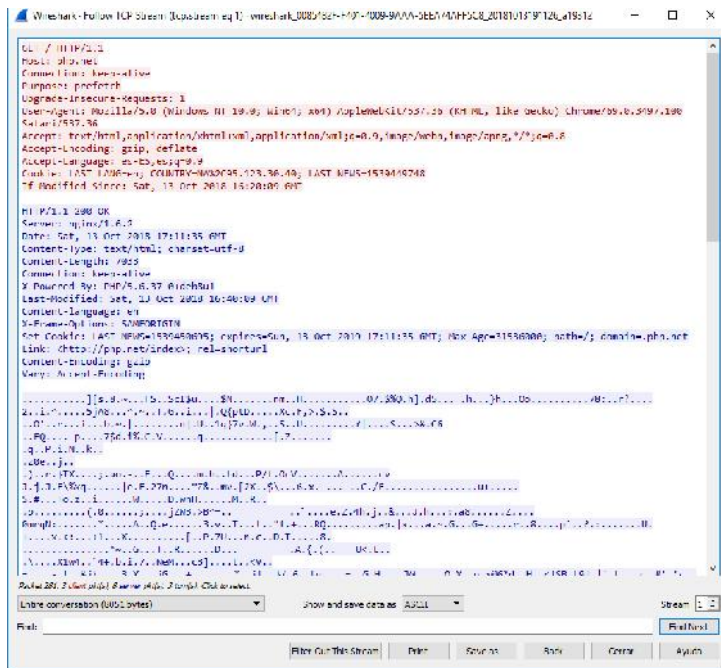
**Nota:** Desactivar el modo promiscuous

Capture=> Interfaces=>**Start**.

4. Desde el navegador Google Chrome, Firefox, Internet Explorer o Safari establece una conexión a un sitio web, por ejemplo a **http://www.php.net**.
5. Vuelve a Wireshark y para la captura (Capture, **Stop**).
6. Buscar una trama HTTP en donde la petición sea **GET / HTTP/1.1**, e incluye el pantallazo en la práctica.

277	5.939168	192.168.1.119	208.43.231.9	HTTP	570	GET / HTTP/1.1
-----	----------	---------------	--------------	------	-----	----------------

7. Haz clic con el botón derecho del ratón y selecciona **Follow TCP Stream**. Ver Figura



Incluye en la práctica tu pantallazo

8. Responde a las siguientes preguntas:

8.1. ¿Cuál es la IP de la máquina donde se ejecuta el servidor Web?

277 5.939168 192.168.1.119 208.43.231.9 HTTP 570 GET / HTTP/1.1  
Internet Protocol Version 4, Src: 192.168.1.119, Dst: 208.43.231.9

8.2. ¿Qué versión de HTTP se utiliza?

HTTP 1.1.

8.3. ¿Qué método de petición se utiliza?

GET.

8.4. ¿Qué recurso se solicita al servidor?

El directorio raíz.

8.5. ¿Qué valor tiene la cabecera Host?

www.php.net

8.6. ¿Se envían cookies en la petición HTTP?

Si.

8.7. ¿Qué lenguaje utiliza el navegador?

es-ES.

8.8. ¿Qué código de estado tiene la respuesta HTTP?

200.

8.9. ¿Qué servidor Web y versión se utiliza?

Server: nginx 1.6.2

8.10. ¿De qué tipo MIME es el recurso recibido?

text/html.

8.11. ¿Se han utilizado conexiones persistentes, es decir, en la misma conexión TCP hay varias peticiones y respuestas HTTP? ¿Qué significa Keep alive?

Si.

```
Keep-Alive: timeout=2, max=100\r\n
Connection: Keep-Alive\r\n
```

Es un mensaje enviado desde un dispositivo a otro para comprobar que el enlace entre estos dos está en funcionamiento o para prevenir que este enlace se rompa. Keep Alive es una forma de comunicación entre el servidor web y el navegador que posibilita que se mantenga una conexión TCP para distintos mensajes HTTP (peticiones/respuestas).

Cuando un navegador web quiere mostrar una página web, debe cargar primero el archivo HTML. Luego, leerá el HTML y solicitará otros recursos como CSS, imágenes u otro contenido que el archivo HTML tenga declarado. Si Keep Alive no está activado, este proceso se puede traducir en mucho tiempo para la descarga de una página web. Las páginas web son a menudo una colección de muchos archivos y, si una nueva conexión TCP tiene que ser ejecutada para todos y cada uno de esos archivos, podría ser más larga la visualización de la página web.

El header Connection: keep-alive que manda el servidor está indicando que acepta que dicha conexión sea persistente y que aceptará más de una petición por ella. En el caso que el servidor no quisiera mantener la conexión abierta (no soporte keepalive) en el header mandará Connection: close para notificar que cerrará la conexión una vez finalizada la transferencia del recurso solicitado.

Desde el punto de vista del cliente, en cualquier momento puede cerrar la conexión, por ejemplo si el usuario cierra el navegador o por su propio timeout.

Servidores como **Apache** se pueden configurar para que cierren la conexión según si:

- Supera un cierto **periodo de inactividad**, el valor de keepalive. Simplemente cerrará la conexión
- Supera un número **máximo de peticiones** por la misma conexión. En este caso lo notificará al cliente mediante un **Connection: close** y cerrará la conexión una vez haya servido la última petición.

8.12. ¿Existen peticiones y respuestas de imágenes? Obtener pantallazo

Si.

```
177.8.831821 192.168.1.119 208.43.231.9 HTTP 480 GET /images/logos/php-logo.svg HTTP/1.1
```

```

Frame 177: 480 bytes on wire (3840 bits), 480 bytes captured (3840 bits) on interface 0
Ethernet II, Src: WistronT_68:eh:61 (98:ee:rh:68:eh:61), Dst: AskeyCom_2c:94:52 (d8:fh:5e:2c:94:52)
Internet Protocol Version 4, Src: 192.168.1.119, Dst: 208.45.251.9
Transmission Control Protocol, Src Port: 65536, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
Hypertext Transfer Protocol
> GET /images/logos/php-logo.svg HTTP/1.1\r\n
Host: php.net\r\n
Connection: keep-alive\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36\r\n
Accept: image/webp,image/apng,image/*,*/*;q=0.8\r\n
Referer: http://www.php.net/\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: es-ES,es;q=0.9\r\n
> Cookie: LAST_LANG=en; COUNTRY=NA%2C95.123.30.40; LAST_NEW5=1539451805\r\n
\r\n

```

8.13.- Lanza de nuevo una captura de red con wireshark. ¿Qué observas al hacer una petición a [www.google.es](http://www.google.es)? ¿Qué protocolo se está utilizando a nivel de aplicación? ¿Qué diferencias observas con la petición realizada a [www.php.net](http://www.php.net)?

Se está utilizando el protocolo HTTPS, que utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de acceso, normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

El puerto estándar para este protocolo es el 443.

**9.- CONCLUSIONES:** Se trata de realizar el seguimiento de las tramas que pertenecen a una determinada petición HTTP, introduciendo una URL determinada desde el navegador y haciendo el seguimiento de todos los protocolos implicados que posibilitan esa “conversación” (conjunto de peticiones/respuestas) entre los equipos origen y destino, en concreto el envío de mensaje de petición HTTP del navegador al servidor y la respuesta de éste al cliente que inició la comunicación.

**Se pide:**

Iniciar el analizador de red (wireshark) y lanzar una determinada petición http, esperar a que la página se haya cargado, parar la captura y a continuación, explicar el proceso seguido y los protocolos intervinientes. Si no obtenemos lo esperado, utilizad los comandos apropiados para borrar de la caché las direcciones MAC asociadas a IPs, así como las IPs asociadas a los nombres de dominio correspondientes (URL).

**9.1.- CONCLUSIONES:**

Para forzar a que funcione el protocolo ARP, vamos a empezar eliminando la caché arp de nuestro equipo con el comando (ejecutado en modo administrador):

>arp -d \*

Por otro lado, borraremos la caché de DNS de nuestro equipo con el comando:

>ipconfig /flushdns

De este modo obligaremos al servidor DNS configurado en nuestro cliente DNS, es decir, en nuestra interfaz de red (propiedades TCP/IP), a proporcionarnos la dirección



IP correspondiente al nombre de dominio solicitado desde nuestro navegador con la petición:



Como el servidor DNS no está en nuestra red local, es necesario conocer la MAC del router y para ello se realizará un ARP preguntando por ella.

### 1/ PROTOCOLO ARP:

```
Adaptador de Ethernet Conexión de área local:
    Sufijo DNS específico para la conexión. . . : 
    Descripción . . . . . : Adaptador de escritorio Intel(R)
    PRO/1000 MT
    Dirección física. . . . . : 08-00-27-26-58-55
    DHCP habilitado . . . . . : no
    Configuración automática habilitada . . . : sí
    Vínculo: dirección IPv6 local. . . : fe80::11f7:572:c22a:9e5b%11(Preferido)
    Dirección IPv4. . . . . : 192.168.1.120(Preferido)
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : 192.168.1.1
    IAID DHCPv6 . . . . . : 235405351
    DUID de cliente DHCPv6. . . . . : 00-01-00-01-1F-78-31-9F-08-00-27-
    26-58-55
    Servidores DNS. . . . . : 8.8.8.8
    8.8.4.4
    NetBIOS sobre TCP/IP. . . . . : habilitado
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CadmusCo_26:58:55	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.120
2	0.000603	Compa1Br_60:50:42	CadmusCo_26:58:55	ARP	60	192.168.1.1 is at dc:53:7c:60:50:42

#### 1.1/ PETICIÓN ARP:

Como se ve en la imagen una petición ARP consiste en el envío en broadcast de nivel 2 (a todos los equipos de la red) de una petición de dirección MAC correspondiente a la IP del router (o Gateway).

Con el filtro `arp and ip.addr==172.16.100.3` en el Wireshark encontramos más fácilmente la petición y respuesta arp.

#### 1.2/ RESPUESTA ARP

```

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: CadmusCo_26:58:55 (08:00:27:26:58:55), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: CadmusCo_26:58:55 (08:00:27:26:58:55)
  Type: ARP (0x0806)
Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: CadmusCo_26:58:55 (08:00:27:26:58:55)
  Sender IP address: 192.168.1.120
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.1.1

```

El router, que en este caso es el equipo con IP 192.168.1.1, contesta con su dirección MAC.

```

adaptador de Ethernet Conexión de área local:
Sufijo DNS específico para la conexión. . . :
Descripción . . . . . : Adaptador de escritorio Intel(R)
PRO/1000 MT
Dirección física. . . . . : 08-00-27-26-58-55
DHCP habilitado . . . . . : no
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . : fe80::11f7:572:c22a:9e5b%11(Preferido)
Dirección IPv4. . . . . : 192.168.1.120(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.1.1
IAID DHCPv6 . . . . . : 235405351
DUID de cliente DHCPv6. . . . . : 00-01-00-01-1F-78-31-9F-00-00-27-
26-58-55
Servidores DNS. . . . . : 8.8.8.8
8.8.4.4
NetBIOS sobre TCP/IP. . . . . : habilitado

```

## 2/ PROTOCOLO DNS

14 7.634150000	192.168.1.119	62.81.16.213	DNS	67 Standard query 0xef95 A php.net
15 7.693100000	62.81.16.213	192.168.1.119	DNS	83 Standard query response 0xe95 A 72.52.91.14

### 2.1/ PETICIÓN DNS

Nuestro equipo, en este caso con IP 192.168.1.119, solicita al servidor DNS configurado en nuestra tarjeta de red (8.8.8.8) como se comprueba con el comando:

La dirección IP del nombre `www.php.net`

```

* Frame 30: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0
* Ethernet II, Src: CadmusCo_26:58:55 (08:00:27:26:58:55), Dst: CompalBr_60:50:42 (dc:53:7c:60:50:42)
* Internet Protocol Version 4, Src: 192.168.1.120, Dst: 8.8.8.8
* User Datagram Protocol, Src Port: 57732, Dst Port: 53
  Source Port: 57732
  Destination Port: 53
  Length: 37
  Checksum: 0xd266 [unverified]
  [Checksum Status: unverified]
  [Stream index: 1]
* Domain Name System (query)
  [Response In: 36]
  Transaction ID: 0x4f10
  Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    * www.php.net: type A, class IN
      Name: www.php.net
      [Name Length: 11]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)

```

## 2.2/ RESPUESTA DNS

```

* Domain Name System (response)
  [Request In: 30]
  [Time: 0.030619000 seconds]
  Transaction ID: 0x4f10
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  Queries
    * www.php.net: type A, class IN
      Name: www.php.net
      [Name Length: 11]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  Answers
    * www.php.net: type A, class IN, addr 194.50.97.16
      Name: www.php.net
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 178
      Data length: 4
      Address: 194.50.97.16

```

El servidor DNS envía la IP correspondiente a ese nombre de dominio (194.50.97.16):

## 3/ ESTABLECIMIENTO DE LA CONEXIÓN A NIVEL DE LA CAPA DE TRANSPORTE

Mediante el Saludo de tres vías, se establece la conexión entre el Cliente Web y el Servidor:



37	6.432306	192.168.1.120	194.50.97.16	TCP	66 49186-80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
----	----------	---------------	--------------	-----	--

59	6.474559	194.50.97.16	192.168.1.120	TCP	66 80-49186 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128
60	6.474592	192.168.1.120	194.50.97.16	TCP	54 49186-80 [ACK] Seq=1 Ack=1 Win=65536 Len=0

Observando los segmentos TCP de establecimiento de la conexión que envía el cliente al servidor hay que fijarse en que el puerto destino es el 80 (puerto por defecto usado por los servidores web) y el origen uno de los puertos dinámicos (>49150). En este caso, 49186. También observad IPs origen/destino

```

Frame 59: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: CompalBr_60:50:42 (dc:53:7c:60:50:42), Dst: CadmusCo_26:58:55 (08:00:27:26:58:55)
Internet Protocol Version 4, Src: 194.50.97.16, Dst: 192.168.1.120
Transmission Control Protocol, Src Port: 80, Dst Port: 49186, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 49186
  [Stream index: 12]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 32 bytes
  Flags: 0x012 [SYN, ACK]
  Window size value: 14600
  [Calculated window size: 14600]
  Checksum: 0xfdd5 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  [SEQ/ACK analysis]

```

**Respuesta del Servidor:**

```

Frame 60: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
Ethernet II, Src: CadmusCo_26:58:55 (08:00:27:26:58:55), Dst: CompalBr_60:50:42 (dc:53:7c:60:50:42)
Internet Protocol Version 4, Src: 192.168.1.120, Dst: 194.50.97.16
Transmission Control Protocol, Src Port: 49186, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
  Source Port: 49186
  Destination Port: 80
  [Stream index: 12]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 20 bytes
  Flags: 0x010 [ACK]
  Window size value: 256
  [Calculated window size: 65536]
  [Window size scaling factor: 256]
  Checksum: 0xe37d [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  [SEQ/ACK analysis]

```

**Confirmación del Cliente:**

En el caso de esta URL el navegador abre varias conexiones TCP y puede enviar peticiones GET por cada una de ellas.

#### 4/ PETICIÓN HTTP

61	6.482440	192.168.1.120	194.50.97.16	HTTP	487 GET / HTTP/1.1
----	----------	---------------	--------------	------	--------------------

Se observa que hay varias solicitudes para esta página:

```

Frame 61: 487 bytes on wire (3896 bits), 487 bytes captured (3896 bits) on interface 0
Ethernet II, Src: cadmusco_26:58:55 (08:00:27:26:58:55), Dst: compaler_60:50:42 (dc:53:7c:60:50:42)
Internet Protocol Version 4, Src: 192.168.1.120, Dst: 194.50.97.16
Transmission Control Protocol, Src Port: 49186, Dst Port: 80, Seq: 1, Ack: 1, Len: 433
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: www.php.net\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2783.143 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
Accept-Encoding: gzip, deflate, sdch\r\n
Accept-Language: es-ES;q=0.8\r\n
Cookie: COUNTRY=NA&C89.141.112.117; LAST_NEWS=1476293682\r\n
\r\n
[Full request URI: http://www.php.net/]
[HTTP request 1/1]
[Response in frame: 84]

```

84	6.560794	194.50.97.16	192.168.1.120	HTTP	1092	HTTP/1.1 200 OK (text/html)
85	6.560821	192.168.1.120	194.50.97.16	TCP	54	40186-80 [ACK] Seq=434 Ack=5262 Win=

Y Respuestas, ésta contiene un código de Operación realizada satisfactoriamente: Podría haber sido parcial (206), y después de varias Peticiones / Respuestas, se conseguiría cargar toda la página.

```

Frame 84: 1092 bytes on wire (8736 bits), 1092 bytes captured (8736 bits) on interface 0
Ethernet II, Src: compaler_60:50:42 (dc:53:7c:60:50:42), Dst: cadmusco_26:58:55 (08:00:27:26:58:55)
Internet Protocol Version 4, Src: 194.50.97.16, Dst: 192.168.1.120
Transmission Control Protocol, Src Port: 80, Dst Port: 49186, Seq: 4224, Ack: 434, Len: 1038
[4 Reassembled TCP segments (5261 bytes): #81(1460), #82(1460), #83(1303), #84(1038)]
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Server: nginx/1.11.5\r\n
Date: Sun, 16 Oct 2016 16:36:37 GMT\r\n
Content-Type: text/html; charset=utf-8\r\n
Transfer-Encoding: chunked\r\n
Connection: keep-alive\r\n
X-Powered-By: PHP/7.1.0RC3\r\n
Last-Modified: Sun, 16 Oct 2016 16:10:12 GMT\r\n
Content-Language: en\r\n
X-Frame-Options: SAMEORIGIN\r\n
Set-Cookie: LAST_NEWS=1476635797; expires=Mon, 16-Oct-2017 16:36:37 GMT; Max-Age=31536000; path=/; domain=.php.net\r\n
Link: <http://php.net/index>; rel=shorturl\r\n
Content-Encoding: gzip\r\n
Vary: Accept-Encoding\r\n

```

85	6.560821	192.168.1.120	194.50.97.16	TCP	54	49186-80 [ACK] Seq=434 Ack=5262 win=65536 Len=0
----	----------	---------------	--------------	-----	----	---

## 5/ CIERRE DE LA CONEXIÓN:

Finalmente se cerrarán todas las conexiones TCP abiertas

39	8.049778000	192.168.1.119	72.32.91.14	TCP	54	49760 > http [FIN, ACK] Seq=907 Ack=308 Win=63392 Len=0
----	-------------	---------------	-------------	-----	----	---

## 9.2.- Comandos:

> **arp -a**

Observamos las asociaciones IP – MAC que se han ido almacenando en la caché de la máquina.

```
C:\Windows\system32>arp -a

Interfaz: 192.168.1.120 --- 0xb
Dirección de Internet      Dirección física      Tipo
192.168.1.1                dc-53-7c-60-50-42    dinámico
192.168.1.255              ff-ff-ff-ff-ff-ff    estático
224.0.0.251                01-00-5e-00-00-fb    estático
224.0.0.252                01-00-5e-00-00-fc    estático
239.255.255.250           01-00-5e-7f-ff-fa    estático
```

**El protocolo de resolución de direcciones (ARP, Address Resolution Protocol)** es un protocolo de comunicaciones de la capa de enlace de datos, responsable de encontrar la dirección de hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (*ARP request*) a la dirección de difusión de la red (*broadcast*, MAC = FF FF FF FF FF FF) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (*ARP reply*) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet (dirección lógica) ser independiente de la dirección Ethernet (dirección física).

>arp -d \* Elimina la cache de IP – MAC

```
C:\Windows\system32>arp -d *
C:\Windows\system32>arp -a
No se encontraron entradas ARP.
C:\Windows\system32>
```

>Ipconfig /displaydns

En la imagen se aprecian resoluciones inversas

Si lanzamos el comando nslookup [www.google.es](http://www.google.es) se irá llenando la caché con nuevas

```
C:\Windows\system32>ipconfig /displaydns

Configuración IP de Windows

be2.php.net
-----
Nombre de registro . . : be2.php.net
Tipo de registro . . . : 5
Período de vida . . . : 1270
Longitud de datos . . : 8
Sección . . . . . : respuesta
Registro CNAME. . . . : php.cu.be

teredo.ipv6.microsoft.com
-----
No existe el nombre.

www.php.net
-----
Nombre de registro . . : www.php.net
Tipo de registro . . . : 1
Período de vida . . . : 205
```

asociaciones IP – nombre dominio

```
safebrowsing-cache.google.com
-----
Nombre de registro . : safebrowsing-cache.google.com
Tipo de registro . . : 5
Período de vida . . : 153
Longitud de datos . . : 8
Sección . . . . : respuesta
Registro CNAME. . . : safebrowsing.cache.l.google.com
```

O también:

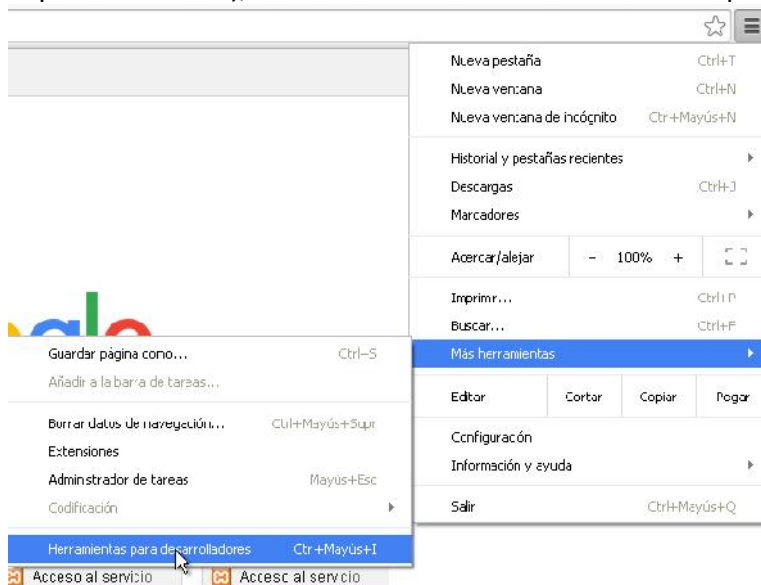
```
C:\Windows\system32>ping www.palomatica.info

Haciendo ping a www.palomatica.info [134.0.14.235] con 32 bytes de datos:
Respuesta desde 134.0.14.235: bytes=32 tiempo=197ms TTL=55
Respuesta desde 134.0.14.235: bytes=32 tiempo=21ms TTL=55
Respuesta desde 134.0.14.235: bytes=32 tiempo=21ms TTL=55
Respuesta desde 134.0.14.235: bytes=32 tiempo=20ms TTL=55
```

>ipconfig /displaydns

```
www.palomatica.info
-----
Nombre de registro . : www.palomatica.info
Tipo de registro . . : 1
Período de vida . . : 854
Longitud de datos . . : 4
Sección . . . . : respuesta
Un registro (host). : 134.0.14.235
```

10. Accede a las opciones de configuración de Google Chrome (cuadrado en la parte superior derecha), Más Herramientas => Herramientas para desarrolladores.



10. Accede a <http://tomcat.apache.org/> y analiza las peticiones y respuestas HTTP, qué métodos usan, los códigos de respuesta, los recursos que envía el servidor. Obtén el pantallazo.

