

## **Tarea N° 1 – Redes de Computadores**

**Integrantes: Ricardo Diaz  
Felipe Navea  
Camilo Alviña**

**Profesor: Gabriel Astudillo**

## **HTTP (RFC2616)**

El Protocolo de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol o HTTP) es un protocolo de nivel de aplicación para sistemas de información distribuidos, colaborativos e hipermedia.

El protocolo HTTP es un protocolo de request/response. Un cliente envía una solicitud al servidor en forma de un método request, URI ("Uniform Resource Identifier") y versión de protocolo, seguido de un mensaje similar a MIME ("Multipurpose Internet Mail Extensions") que contiene modificadores request, información del cliente y posible contenido del cuerpo a través de una conexión con un servidor. El servidor responde con una línea de estado, incluida la versión del protocolo del mensaje y un código de éxito o error, seguido de un mensaje similar a MIME que contiene información del servidor, información de la entidad y posible contenido del cuerpo de la entidad.

Los 3 métodos más utilizados por el protocolo HTTP son:

POST: utilizado para enviar datos al servidor.

GET: pide un documento específico.

HEAD: se utiliza para solicitar información sobre un documento, envía poca información y es más rápido que el GET.

## **HTML**

El lenguaje de marcado de hipertexto (en inglés: HyperText Markup Language o HTML) es un lenguaje de marcas que se utiliza para el desarrollo de páginas web. Dicho lenguaje es una forma de codificar un documento, el cual contiene toda la información.

Se utiliza para estructurar un sitio web en forma de texto y determina desde, el propio texto, hasta imágenes, colores o cualquier elemento contenido en el mismo.

## **RELACION HTML Y HTTP**

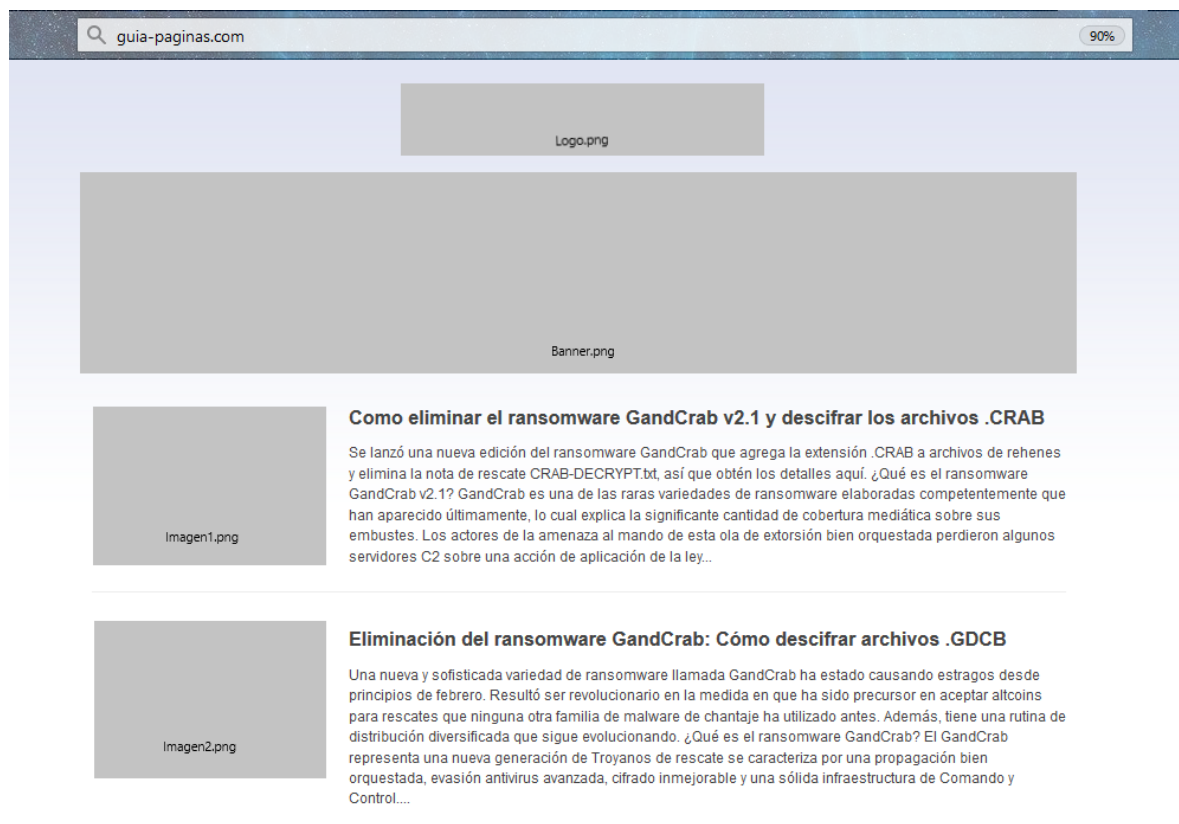
La forma en la que HTTP y HTML se relacionan se basa en que a través del protocolo HTTP el navegador se comunica con el servidor específico, solicita el archivo en cuestión en código HTML y lo interpreta, de esa forma poder visualizar cualquier sitio en internet. Es decir, HTTP es el transmisor del código HTML.

# PÁGINA 1

**Nombre:** guia-paginas.com

**IP:** 51.254.103.64

**Esquema:**



**Handshake inicial:**

| No. | Time     | Source        | Destination   | Protocol | Length | Info  |
|-----|----------|---------------|---------------|----------|--------|---|
| 45  | 0.733904 | 192.168.0.11  | 51.254.103.64 | TCP      | 66     | 52440 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1            |
| 57  | 0.995410 | 51.254.103.64 | 192.168.0.11  | TCP      | 66     | 80 → 52440 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 58  | 0.995507 | 192.168.0.11  | 51.254.103.64 | TCP      | 54     | 52440 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0                                  |

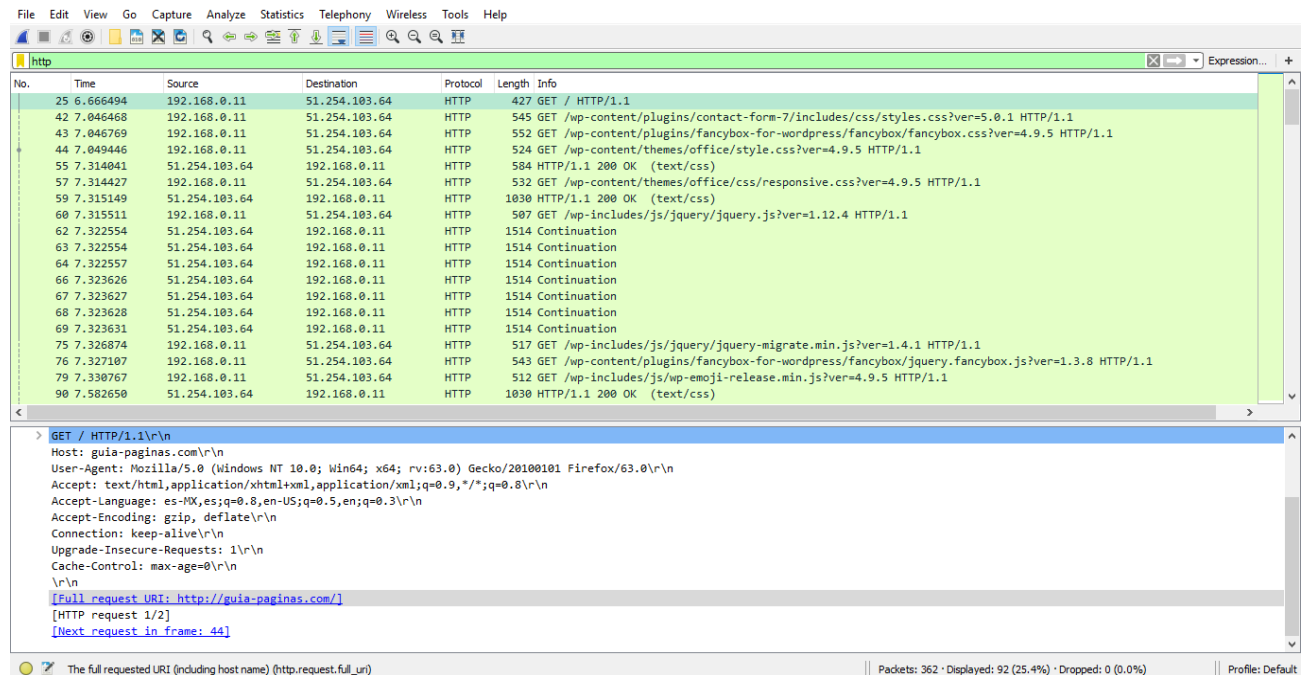
El handshake inicial (3-ways handshake) se encarga de establecer la conexión TCP, el cliente realiza una conexión enviando un paquete SYN al servidor, en el servidor se comprueba si el puerto está abierto, si el puerto no está abierto se le envía al cliente un paquete de respuesta RST, esto significa un rechazo de intento de conexión. Si el puerto está abierto, el servidor responde con un paquete SYN/ACK. Entonces el cliente respondería al servidor con un ACK, completando así la conexión.

**SYN:** Sincronización de números de secuencia.

**ACK:** Campo de reconocimiento.

**RST:** Reestablece la conexión.

## Tramas HTTP:



The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The main window is divided into three panes: the packet list, packet details, and packet bytes.

The packet list pane shows a list of captured packets. The first packet (No. 25) is selected, and its details are expanded in the packet details pane. The packet details pane shows the structure of the selected packet, which is an HTTP GET request. The request structure is as follows:

```
GET / HTTP/1.1\r\nHost: guia-paginas.com\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\nAccept-Encoding: gzip, deflate\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\nCache-Control: max-age=0\r\n\r\n[Full request URI: http://guia-paginas.com/]\n[HTTP request 1/2]\n[Next request in frame: 44]
```

The packet bytes pane shows the raw data of the selected packet, which is the full request URI: `http://guia-paginas.com/`.

El primer request (GET / HTTP/1.1) pide al servidor la página web en HTML y la interpreta.

Generalmente corresponde a la página principal de una página web. Los siguientes request corresponden a solicitudes de archivos específicos, como plugins y temas para la página.

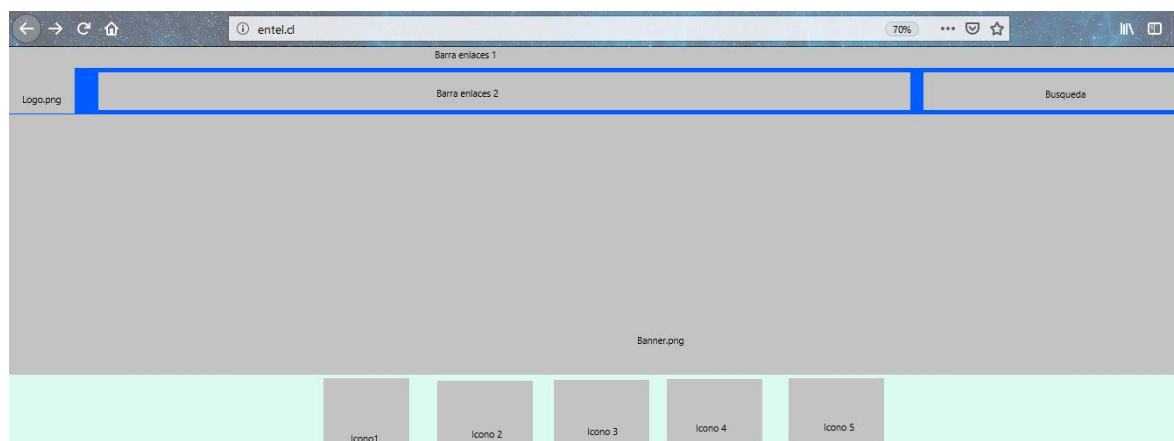
La respuesta del servidor viene a través de HTTP/1.1 200 OK, lo que quiere decir que el request ha tenido éxito. Dependiendo del método de solicitud HTTP que se utilice el significado de la respuesta varía. Para métodos GET quiere decir que el recurso ha sido recuperado y se transmite al body.

## PÁGINA 2

**Nombre:** www.entel.cl

**IP:** 104.71.244.47

**Esquema:**



**Telefonia móvil, Internet y Hogar**

Nuestros Productos Destacados

**Handshake inicial:**

| No. | Time     | Source        | Destination   | Protocol | Length | Info  |
|-----|----------|---------------|---------------|----------|--------|---|
| 39  | 1.693565 | 192.168.0.11  | 104.71.244.47 | TCP      | 66     | 57545 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1            |
| 51  | 1.812256 | 104.71.244.47 | 192.168.0.11  | TCP      | 66     | 80 → 57545 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 52  | 1.812358 | 192.168.0.11  | 104.71.244.47 | TCP      | 54     | 57545 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0                                  |

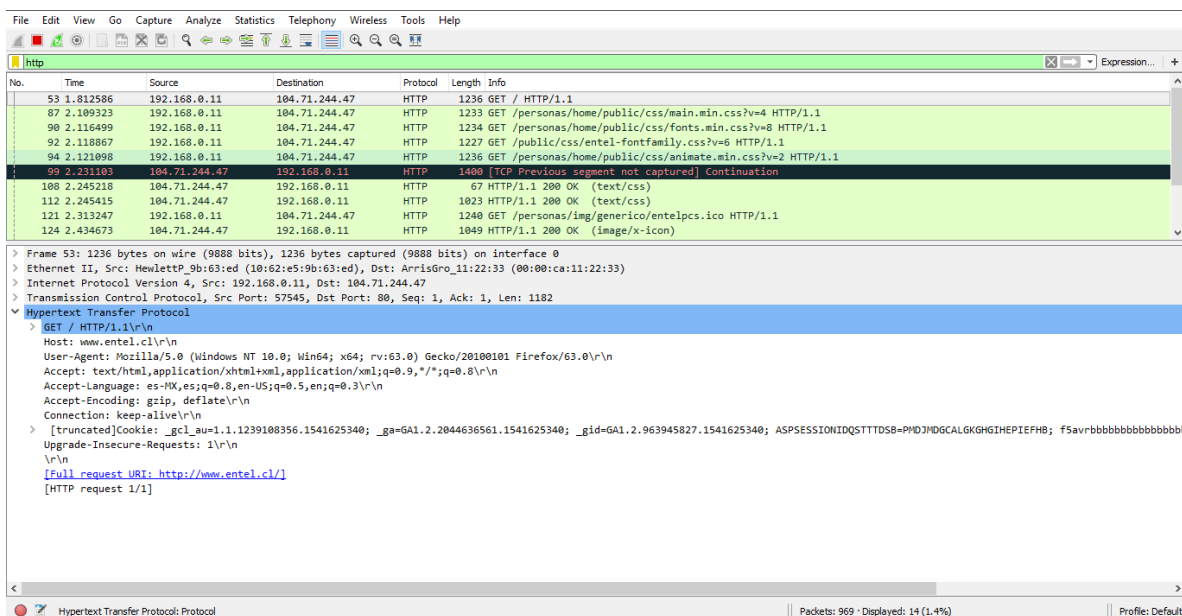
El handshake inicial (3-ways handshake) se encarga de establecer la conexión TCP, el cliente realiza una conexión enviando un paquete SYN al servidor, en el servidor se comprueba si el puerto está abierto, si el puerto no está abierto se le envía al cliente un paquete de respuesta RST, esto significa un rechazo de intento de conexión. Si el puerto está abierto, el servidor responde con un paquete SYN/ACK. Entonces el cliente respondería al servidor con un ACK, completando así la conexión.

**SYN:** Sincronización de números de secuencia.

**ACK:** Campo de reconocimiento.

**RST:** Reestablece la conexión.

## Tramas HTTP:



| No. | Time     | Source        | Destination   | Protocol | Length | Info   |
|-----|----------|---------------|---------------|----------|--------|--|
| 53  | 1.812586 | 192.168.0.11  | 104.71.244.47 | HTTP     | 1236   | GET / HTTP/1.1   |
| 87  | 2.109323 | 192.168.0.11  | 104.71.244.47 | HTTP     | 1233   | GET /personas/home/public/css/main.min.css?v=4 HTTP/1.1    |
| 90  | 2.116499 | 192.168.0.11  | 104.71.244.47 | HTTP     | 1234   | GET /personas/home/public/css/fonts.min.css?v=8 HTTP/1.1   |
| 92  | 2.118867 | 192.168.0.11  | 104.71.244.47 | HTTP     | 1227   | GET /public/css/entel-fontfamily.css?v=6 HTTP/1.1          |
| 94  | 2.121098 | 192.168.0.11  | 104.71.244.47 | HTTP     | 1236   | GET /personas/home/public/css/animate.min.css?v=2 HTTP/1.1 |
| 99  | 2.231103 | 104.71.244.47 | 192.168.0.11  | HTTP     | 1408   | [TCP Previous segment not captured] Continuation           |
| 108 | 2.245218 | 104.71.244.47 | 192.168.0.11  | HTTP     | 67     | HTTP/1.1 200 OK (text/css)                                 |
| 112 | 2.245415 | 104.71.244.47 | 192.168.0.11  | HTTP     | 1023   | HTTP/1.1 200 OK (text/css)                                 |
| 121 | 2.313247 | 192.168.0.11  | 104.71.244.47 | HTTP     | 1240   | GET /personas/img/generico/entelpcs.ico HTTP/1.1           |
| 124 | 2.434673 | 104.71.244.47 | 192.168.0.11  | HTTP     | 1049   | HTTP/1.1 200 OK (image/x-icon)                             |

> Frame 53: 1236 bytes on wire (9888 bits), 1236 bytes captured (9888 bits) on interface 0  
> Ethernet II, Src: HewlettP\_9b:63:ed (10:62:e5:9b:63:ed), Dst: ArrisGro\_11:22:33 (00:00:ca:11:22:33)  
> Internet Protocol Version 4, Src: 192.168.0.11, Dst: 104.71.244.47  
> Transmission Control Protocol, Src Port: 57545, Dst Port: 80, Seq: 1, Ack: 1, Len: 1182  
▼ Hypertext Transfer Protocol  
    > GET / HTTP/1.1\r\n  
        Host: www.entel.cl\r\n  
        User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0\r\n  
        Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8\r\n  
        Accept-Language: es-MX,es;q=0.8,en-US;q=0.5,en;q=0.3\r\n  
        Accept-Encoding: gzip, deflate\r\n  
        Connection: keep-alive\r\n  
        Cookie: [truncated]Cookie: \_gcl\_au=1.1.1239108356.1541625340; \_ga=GA1.2.2044636561.1541625340; \_gid=GA1.2.963945827.1541625340; ASPSESSIONIDQ\$TTTDSB-PMDJNDGALGKHGHIIEFHB; f5avrbbbbbbbbbbbbbb\r\n  
        Upgrade-Insecure-Requests: 1\r\n  
        \r\n  
        [Full] request URI: http://www.entel.cl/  
        [HTTP request 1/1]

⏏ Hypertext Transfer Protocol: Protocol      Packets: 969 • Displayed: 14 (1.4%)      Profile: Default

El primer request (GET / HTTP/1.1) pide al servidor la página web en HTML y la interpreta. Generalmente corresponde a la página principal de una página web. Los siguientes request corresponden a solicitudes de archivos específicos, como plugins y temas para la página.

La respuesta del servidor viene a través de HTTP/1.1 200 OK, lo que quiere decir que el request ha tenido éxito. Dependiendo del método de solicitud HTTP que se utilice el significado de la respuesta varía. Para métodos GET quiere decir que el recurso ha sido recuperado y se transmite al body.

Nos encontramos con TCP Previous segment not captured, lo cual se puede tratar de una pérdida de paquete o el inicio tardío en la captura.