



**Universidad  
de Valparaíso**  
CHILE

# **Implementación de cliente http simple en C++14**

Integrantes: Eduardo Cabrera  
Javier Figueroa  
Mauricio Moraga

Profesor: Gabriel AStudillo

## 1.1 Captura de tráfico HTTP.

The screenshot shows the Wireshark interface with a packet capture of HTTP traffic. The packet list on the left shows several GET requests to various CSS and JavaScript files. The packet details pane on the right shows the structure of an HTTP GET request, including the request line, headers (Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, Connection, Cookie), and the body.

Time: 73.2163873065 Source: 192.168.1.212 Destination: 192.168.1.212 Protocol: HTTP Length: 940 Info: GET / HTTP/1.1

98 3.149796248 192.168.1.212 192.168.1.212 HTTP 940 HTTP/1.1 200 OK (text/html)

100 3.337690321 192.168.1.212 192.168.1.212 HTTP 730 GET /templates/home-muni\_antofagasta/css/bootstrap.css HTTP/1.1

107 3.346926502 192.168.1.212 192.168.1.212 HTTP 278 HTTP/1.1 304 Not Modified

109 3.347302032 192.168.1.212 192.168.1.212 HTTP 722 GET /media/system/js/modal.js?3b50923530e55893e000385592a5902c HTTP/1.1

112 3.350254833 192.168.1.212 192.168.1.212 HTTP 644 GET /index.php?option=com\_ajax&plugin=arktypography&format=json HTTP/1.1

115 3.351459918 192.168.1.212 192.168.1.212 HTTP 725 GET /plugins/content/pdf\_embed/assets/css/style.css HTTP/1.1

116 3.353244275 192.168.1.212 192.168.1.212 HTTP 571 GET /modules/mod\_news\_pro\_gk5/interface/css/style.css HTTP/1.1

121 3.353698997 192.168.1.212 192.168.1.212 HTTP 738 GET /media/system/css/modal.css?3b50923530e55893e000385592a5902c HTTP/1.1

122 3.353784745 192.168.1.212 192.168.1.212 HTTP 724 GET /media/com\_attachments/css/attachments\_hide.css HTTP/1.1

126 3.355094380 192.168.1.212 192.168.1.212 HTTP 726 GET /media/com\_attachments/css/attachments\_list.css HTTP/1.1

127 3.358541782 192.168.1.212 192.168.1.212 HTTP 277 HTTP/1.1 304 Not Modified

Frame 73: 643 bytes on wire (5144 bits), 643 bytes captured (5144 bits) on interface 0

Ethernet II, Src: SamsungE\_29:e2:d4 (50:b7:c3:29:e2:d4), Dst: Mitrasta\_d7:b5:3c (34:57:60:d7:b5:3c)

Internet Protocol Version 4, Src: 192.168.1.212, Dst: 192.168.1.212

Transmission Control Protocol, Src Port: 44950, Dst Port: 80, Seq: 1, Ack: 1, Len: 577

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

Host: www.municipalidaddeantofagasta.cl\r\n

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; 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-CL,es;q=0.8,en-US;q=0.5,en;q=0.3\r\n

Accept-Encoding: gzip, deflate\r\n

Referer: https://www.google.com/\r\n

Connection: keep-alive\r\n

Cookie: r1\_modals=6; \_ga=GA1.2.3961610158.1541086120; 8d8b09f92a9584f058d8920be2989e03=gg87n4cs5f087v293ss93quqs7; \_gid=GA1.2.580081111.1541641473; \_gat=1\r\n

0040 92 c2 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 --GET / HTTP/1.1\r\n

0050 00 00 48 6f 73 74 3a 20 77 77 77 2e 6d 75 6e 69 --Host: www.muni

0060 63 69 70 61 6c 69 64 61 64 64 65 65 6e 74 6f 66 --cipalida deantof

0070 61 67 61 73 74 61 2e 63 6c 0d 6a 55 79 65 72 2d --agasta.c l- User-

0080 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 --Agent: Mozilla/5

0090 2e 30 20 28 58 31 31 3b 20 55 62 75 6e 74 75 3b --.0 (X11; Ubuntu;

00a0 20 4c 69 6e 75 78 20 78 38 36 5f 36 34 3b 20 72 --Linux x 86\_64; r

00b0 76 3a 36 33 2e 39 29 29 47 65 63 6b 6f 2f 32 39 --v:63.0) Gecko/20

00c0 31 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f 36 --100101 Firefox/6

00d0 33 2e 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 --3.0 Acc ept: tex

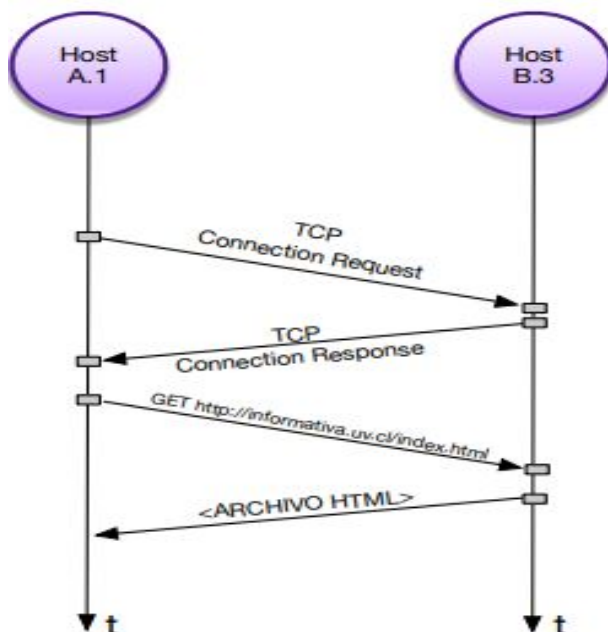
00e0 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 --t/html,a pplicati

00f0 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 --on/xhtml +xml,app

0100 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 --lication /xml;q=0

0110 2e 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 --.9,\*/\*;q =0.8 Ac

Primero dentro del wireshark lo que hago es iniciarlo y refrescar la pagina que voy a utilizar que en este caso es <http://www.municipalidaddeantofagasta.cl/> una vez hecho esto filtró por protocolo que en este caso es http, al tener todo esto reviso cual es mi primera acción desde mi ip al destinatario que es la página y lo primero que hace es el handshake.



Después de pedir todo por el puerto origen 44950 al destino del puerto 80 que es la página estaría lograda la comunicación entre los puertos.

# Qué es HTML?

Html o en sus siglas HyperText Markup Language es un lenguaje que sirve para el desarrollo de páginas web según la definición, pero más que eso se separa en etiquetas y siempre comienza con la etiqueta html.

# Que es HTTP?

Http o en sus siglas Hypertext Transfer Protocol es una aplicación a nivel protocolo que es usada por la world wide web desde 1990, la primera versión http fue http/0.9 y después fue subiendo hasta http/1.1 qué es la que usamos actualmente dentro de las páginas capturadas y el http es el idioma para que el computador origen y el destino puedan conversar entre sí y mandarse la información.

# Cómo están relacionados ambos?

Ambos están relacionados de la manera de que sin uno no podría funcionar el otro, como el http es el idioma que ocupan las computadoras para mandarse información y esa información es la que muestra el html que son todas las imágenes, objetos y texto que contenga la página a pedir del computador origen.

# Esquema

