



## TPL 5 - World Wide Web - HTTP

**Fecha de Entrega:** 09/05/2022

**URL de Entrega:** <https://tinyurl.com/TyR-2022-TP5>

**Objetivo:** Permitir la comprensión del funcionamiento de la “World Wide Web”, a través del análisis de los principales protocolos y agentes que integran su arquitectura.

### Consignas

1. Utilizando la herramienta `nc`, conéctese a la dirección y al puerto del servidor web `www.unlu.edu.ar` y lleve a cabo las siguientes pruebas utilizando primitivas del protocolo HTTP. Guarde las respuestas obtenidas.

- a. Petición por protocolo HTTP versión 1.0

```
$ nc -v -C www.unlu.edu.ar 80 (enter)
GET / HTTP/1.0 (enter) (enter)
```

- b. Petición por protocolo HTTP versión 1.1

```
$ nc -v -C www.unlu.edu.ar 80 (enter)
GET / HTTP/1.1 (enter)
Host: www.unlu.edu.ar (enter) (enter)
```

- c. Petición HTTP. Copie el texto de la petición (indicada bajo la línea) y péguelo una vez establecida la conexión con `nc`. Finalice la petición pulsando **tres veces** la tecla Enter.

```
$ nc -v -C www.unlu.edu.ar 80
GET / HTTP/1.1
Host: www.labredes.unlu.edu.ar
Connection: keep-alive
```

- d. Petición HTTP. Copie el texto de la petición (indicada bajo la línea) y péguelo una vez establecida la conexión con `nc`. Finalice la petición pulsando **tres veces** la tecla Enter.

```
$ nc -v -C www.unlu.edu.ar 80
GET / HTTP/1.1
Host: www.labredes.unlu.edu.ar
Connection: close
```

Responda:

1. ¿Qué códigos numéricos de respuesta HTTP devuelve el servidor web para cada petición? ¿Qué significan según la RFC?
  2. ¿Cuales son los otros encabezados devueltos y qué contenido es transferido en cada caso?
  3. ¿Qué diferencia nota respecto a la duración de la conexión contra el servidor en los dos últimos casos?
  4. ¿Qué información acerca del software servidor web y configuración se obtiene?
2. Realice 3 capturas de peticiones HTTP al servidor web `www.unlu.edu.ar`. Para la primer y segunda captura utilice 2 navegadores gráficos distintos (ej: Firefox, Iceweasel, Chrome, Chromium, Konqueror, Epiphany, Explorer, Sarafi, etc.), y para la tercer captura use la herramienta de transferencias `curl` (<https://curl.haxx.se/>) o `wget` (<http://www.gnu.org/software/wget/>).



- a. ¿Qué encabezados envía cada cliente en la petición?
  - b. Comente las características de la información en tránsito con respecto a la confidencialidad.
3. Describa cómo opera un cliente HTTP (por ejemplo un navegador web) para recuperar una página HTML que contiene varios objetos. Analice la captura del archivo `captura_ejemplo_http.pcap` provisto por los docentes y represente el intercambio de mensajes mediante un gráfico ideado por Ud. ¿Qué primitivas se utilizan en cada caso?.
4. Instale e inicie en el entorno **kathará** el laboratorio de proxy HTTP provisto por los docentes, disponible en [https://github.com/redesunlu/netkit-labs/raw/master/tarballs/kathara-lab\\_proxy.tar.gz](https://github.com/redesunlu/netkit-labs/raw/master/tarballs/kathara-lab_proxy.tar.gz)

Este laboratorio comprende tres hosts: uno actúa como servidor web (con el servicio Apache2 en ejecución), uno actúa como cliente web (con el navegador Lynx instalado) y uno actúa como proxy HTTP (con el servicio Squid en ejecución).

- a. Servidor HTTP Apache: Comente cuáles son los parámetros de configuración básicos necesarios de un servidor HTTP. Sugerencia: Investigue el archivo de configuración del software Apache y detalle alguna de las posibilidades de configuración. ¿Qué estructura de directorios se utiliza, y cuál es su contenido? ¿Qué información se almacena en los archivos de logs estándares?
- b. En el cliente, configure que las peticiones HTTP se realicen a través de proxy. Para ello, establezca la variable de entorno `HTTP_PROXY` como se muestra a continuación. Esto es equivalente a configurar las variables "Servidor proxy" y "Puerto" en los navegadores gráficos y en los celulares.

```
export http_proxy=http://10.0.0.30:3128
```

- c. Inicie una captura desde el host redirigiendola a un archivo para su posterior análisis.
- d. En el cliente, navegue hacia la dirección `http://169.254.0.1/` utilizando un browser de consola:

```
lynx http://169.254.0.1/
```

Dicha dirección IP es la correspondiente al host servidor web.

- e. Detenga la captura y analice el mensaje que aparece en pantalla. ¿Qué código de respuesta HTTP se retornó? Cierre el navegador web pulsando la tecla `q`
- f. La configuración de fábrica del software proxy Squid impide que los clientes naveguen a través de él. Para resolverlo, busque el archivo `/etc/squid/squid.conf` dentro del host proxy, edítelo y reemplace la línea

```
# http_access allow localnet
```

por

```
http_access allow localnet
```

- g. Guarde los cambios y reinicie el proceso que actúa como proxy mediante el comando

```
service squid restart
```

De esta manera, el software Squid admitirá peticiones que realicen clientes que estén accediendo desde redes privadas (10.0.0.0/8, 192.168.0.0/16 y otras).



- h. Inicie una nueva captura y vuelva a realizar la petición del punto **d**.
- i. Indique qué mensaje aparece en pantalla. Cierre el navegador web pulsando la tecla **q**
- j. Detenga la captura, analícela y responda:
  1. ¿Qué encabezados envía el cliente al proxy-http en la petición?
  2. ¿Qué encabezados envía el proxy-http al servidor web en la petición?
  3. Mencione las diferencias que observa en los encabezados respecto a no utilizar un proxy-http (punto 4)
  4. ¿Es posible cambiar el número de puerto TCP en el que escucha el servidor proxy?  
¿Qué línea del archivo de configuración hay que cambiar para que Squid escuche por conexiones en el puerto 8080?
5. ¿Cómo un sistema que realiza caché local puede determinar si algún objeto en el servidor original fue modificado con respecto a la copia actual, sin realizar la transferencia completa del objeto?

### **Bibliografía**

- FOROUZAN, B.A. TCP IP Protocol Suite. McGraw-Hill Higher Education, 2009. Capítulo 22: "World Wide Web and HTTP"
- RFC 7230 - Hypertext Transfer Protocol (HTTP/1.1) Message Syntax and Routing
- RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1) Semantics and Content

### **Recursos en internet**

- <http://www.faqs.org> y <http://www.rfc-editor.org>
- Apache web server: <http://httpd.apache.org/>
- Squid proxy: <http://www.squid-cache.org/>
- Common Gateway Interface: <http://www.w3c.org/cgi/>
- HTML: <http://www.w3.org/TR/html/>
- Sitio web del lenguaje Python: <https://www.python.org/>
- Tutorial de Python: <http://docs.python.org.ar/tutorial/3/index.html>
- Guías de inicio en Python: <http://python.org.ar/AprendiendoPython>

### **Preguntas (guía de lectura)**

Describe someramente el protocolo HTTP, indicando modo de operación y primitivas básicas.

¿Cuál es el formato de mensajes del protocolo HTTP?

¿Qué es un servidor Proxy? ¿En qué situaciones se implementa? Brinde ejemplos.

¿Qué es HTML? ¿Qué especifica?

¿Qué es la interfaz CGI? ¿Para qué se utiliza?

¿De qué formas un programa puede recibir parámetros por medio de la interfaz CGI? Comente las diferencias en el modo de operación en cada caso.