



Trabajo práctico de laboratorio #5 - 2016

World Wide Web - HTTP

(Fecha límite de entrega: 05-05-2016)

- 1) Describa someramente el protocolo HTTP. Indique modo de operación y primitivas básicas.
- 2) ¿Qué es HTML? ¿Qué especifica? Ejemplifique.
- 3) 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. Para realizar los últimos ejemplos, copie y pegue el texto a enviar en un archivo y luego utilice **nc** redirigiendo la entrada estandar desde dicho archivo.

a)
nc www.unlu.edu.ar 80 (enter)
GET / HTTP/1.0 (enter) (enter)

b)
nc 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 dos veces enter.

nc www.unlu.edu.ar 80

GET / HTTP/1.1
User-Agent: Yo_soy_mi_agente_de_usuario
Host: www.tyr.unlu.edu.ar
Accept: text/html
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 dos veces enter.

nc www.unlu.edu.ar 80

GET / HTTP/1.1
User-Agent: Yo_soy_mi_agente_de_usuario
Host: www.tyr.unlu.edu.ar
Accept: text/html
Connection: close

¿Cuales son los encabezados devueltos y qué contenido es transferido en cada caso?
¿Qué diferencia nota en la respuesta enviada por el servidor entre los dos últimos casos?

- 4) Instale e inicie el servidor HTTP Apache en su equipo (paquete **apache2**). Realice una captura de recuperación de una página HTML desde otro host utilizando al menos 2 navegadores gráficos diferentes (ej: Firefox/Iceweasel, Chrome/Chromium, Konqueror, Epiphany) y 1 navegador web en modo texto (ej: **w3m**, **lynx**, **links**) .

- a) ¿Qué encabezados envía cada cliente en la petición?
- b) Comente las características de la información en tránsito con respecto a la confidencialidad.

- 5) Describa cómo opera un cliente HTTP (por ejemplo un navegador web) para recuperar una página HTML que contiene varios objetos. Analice la captura de ejemplo y represente el intercambio de mensajes mediante un gráfico ideado por Ud. ¿Qué primitivas se utilizan en cada caso?.

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



7) Instale e inicie el servidor proxy HTTP Squid (paquete squid o squid3) en su equipo. Realice una captura de recuperación de una página HTML desde otro host utilizando un navegador web en el cual haya definido que utilice el proxy-http por ud. configurado.

- ¿Qué encabezados envía el cliente al proxy-http en la petición?
- ¿Qué encabezados envía el proxy-http al servidor web en la petición?
- Mencione las diferencias observadas con respecto a no utilizar un proxy-http (punto 4)

8) ¿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?

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

10) ¿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.

Para los ejercicios 11 a 14 deberá utilizar el intérprete de lenguaje Python versión 2.x disponible para múltiples plataformas y sistemas operativos. En los sistemas operativos Linux el intérprete usualmente está instalado por defecto. Si bien no es estrictamente necesario ninguna introducción “fuerte” en el lenguaje, se recomienda leer los primeros capítulos del tutorial en español indicado en las referencias de este trabajo.

11) Ejecute el siguiente comando en la consola de Linux, en el home del usuario (/home/alumno):

```
alumno@lab1:~$ python -m SimpleHTTPServer
```

Luego, abra un navegador web e ingrese a la URL <http://localhost:8000>

- ¿Qué es lo que se ve en el navegador?
- ¿Cuál es la salida por consola del programa Python? ¿Qué puede interpretar de ella?
- Abra un archivo de extensión .html en el directorio home del usuario. Si no existiera alguno, genere uno escrito por usted. ¿Qué sucede al abrirlo con el navegador web?

12) Escriba en un editor de texto el siguiente script y guárdelo en el archivo http1.py

```
# coding: utf-8
import BaseHTTPServer

HOST_NAME = 'localhost'
PORT = 8000

class Handler(BaseHTTPServer.BaseHTTPRequestHandler):

    def do_GET(server):
        """Respondo a una petición de tipo GET"""
        # Imprimo los encabezados
        print '-' * 80
        print server.command, server.path, server.request_version
        print server.headers
        # Devuelvo la respuesta
        server.send_response(200)
        server.end_headers()

    def log_message(*args):
        """Deshabilito la salida por defecto del servidor """
        pass

if __name__ == '__main__':
    http_server = BaseHTTPServer.HTTPServer((HOST_NAME, PORT), Handler)
    print "Ejecutando Server HTTP - %s:%s" % (HOST_NAME, PORT)
    print "Presione Ctrl+C para detener..."
    try:
        http_server.serve_forever()
    except KeyboardInterrupt:
        pass
    http_server.server_close()
    print "Deteniendo Server HTTP - %s:%s" % (HOST_NAME, PORT)
```



Es importante destacar al momento de escribir el código, que en el lenguaje Python los espacios son utilizados para definir el nivel de anidamiento de la sentencia (ya que como se ve, no se utilizan llaves); por lo tanto, debe respetarse la sintaxis y todos los espacios del ejemplo.

Ejecute el script servidor HTTP con el comando `python http1.py`
Luego, abra un navegador web e ingrese a la URL <http://localhost:8000>

- 12.a) ¿Qué es lo que se ve en el navegador?
12.b) ¿Cuál es la salida por consola del servidor http? ¿Qué puede interpretar de la misma?
12.c) Experimente y compare las diferencias en la salida del servidor web:
- Con otros navegadores o clientes http,
 - Recargando la página con Ctrl+F5,
 - Ingresando a la URL <http://localhost:8000/test/de/url/>

¿Qué diferencias aprecia y a causa de qué?

- 12.d) En el navegador (Firefox o Chromium/Chrome), presione Ctrl+Shift+I, lo que da lugar a la apertura del depurador del navegador; entre otras cosas, éste posee una pestaña o apartado llamado "Red", que permite examinar cómo la página web es obtenida por el navegador, recurso por recurso. Seleccione haciendo clic sobre la petición hecha al servidor web local y describa qué encabezados devolvió el servidor escrito en Python.

- 13) Escriba en un editor de texto el siguiente script¹ y guárdelo en el archivo `http2.py`

```
# coding: utf-8
import BaseHTTPServer

HOST_NAME = 'localhost'
PORT = 8000

class Handler(BaseHTTPServer.BaseHTTPRequestHandler):

    def do_GET(server):
        """Respondo a una petición de tipo GET"""
        # Imprimo los encabezados
        print '-' * 80
        print server.command, server.path, server.request_version
        print server.headers
        # Devuelvo la respuesta
        server.send_response(200)
        server.send_header('Content-Type', 'text/html')
        server.end_headers()
        server.wfile.write('<html><head><title>Pagina HTML de ejemplo</title>'
                           '</head><body><p>Esta es una prueba, con texto en <b>negrita</b>,'
                           '<i>cursiva</i> e incluso una imagen externa:</p>'
                           '</body></html>')

    def log_message(*args):
        """ Deshabilito la salida por defecto del servidor """
        pass

if __name__ == '__main__':
    http_server = BaseHTTPServer.HTTPServer((HOST_NAME, PORT), Handler)
    print 'Ejecutando Server HTTP - %s:%s' % (HOST_NAME, PORT)
    print 'Presione Ctrl+C para detener...'
    try:
        http_server.serve_forever()
    except KeyboardInterrupt:
        pass
    http_server.server_close()
    print 'Deteniendo Server HTTP - %s:%s' % (HOST_NAME, PORT)
```

Ejecute el script servidor HTTP con el comando `python http2.py`. Luego, abra un navegador web e ingrese a la URL <http://localhost:8000> con el depurador del navegador activo.

- 13.a) Analice cómo se obtiene la página principal y la imagen embebida en el documento.
13.b) Detenga el servidor, modifique el encabezado Content-Type a text/plain y vuelva a ejecutar la prueba. ¿Qué apariencia tiene la página web y por qué?

¹ Link al ejemplo 2: <http://pastebin.com/BThStjqY>



14) Escriba en un editor de texto el siguiente script¹ y guárdelo en el archivo http3.py

```
# coding: utf-8
import BaseHTTPServer

HOST_NAME = 'localhost'
PORT = 8000

def detectar_so(user_agent):
    # Ver listados en http://www.useragentstring.com/pages/useragentstring.php
    if 'Linux' in user_agent:
        return 'Veo que Ud. esta usando Linux como S.O.'
    elif 'Windows' in user_agent:
        return 'Veo que Ud. esta usando Windows como S.O.'
    else:
        return 'No conozco su S.O.'

def get_pagina_ok():
    """ Función que devuelve la página de exito de ejemplo """
    # Como alternativa podría abrirse un archivo del disco, leerlo y devolverlo
    # como cadena de texto.
    return ('<html><head><title>Pagina HTML de ejemplo</title>'
            '</head><body><p>Esta es una prueba, con texto en <b>negrita</b>,'
            '<i>cursiva</i> e incluso una imagen externa:</p>'
            '')

class Handler(BaseHTTPServer.BaseHTTPRequestHandler):
    def do_GET(server):
        """Respondo a una petición de tipo GET"""
        # Imprimo los encabezados por consola
        print '-' * 80
        print server.command, server.path, server.request_version
        print server.headers
        # Devuelvo la respuesta
        if server.path.startswith('/ir_a/'):
            ir_a = server.path.split('/')[1]
            server.send_response(302)
            server.send_header('Location', 'http://' + ir_a)
            server.end_headers()
        elif server.path.startswith('/no_existe'):
            server.send_response(404)
            server.send_header('Content-Type', 'text/plain')
            server.end_headers()
            server.wfile.write('Pagina no encontrada')
        else:
            server.send_response(200)
            server.send_header('Content-Type', 'text/html')
            server.end_headers()
            server.wfile.write(get_pagina_ok())
            server.wfile.write('<p>' + detectar_so(server.headers['User-Agent']))
            server.wfile.write('</p></body></html>')

    def log_message(*args):
        """ Deshabilito la salida por defecto del servidor """
        pass

if __name__ == '__main__':
    http_server = BaseHTTPServer.HTTPServer((HOST_NAME, PORT), Handler)
    print 'Ejecutando Server HTTP - %s:%s' % (HOST_NAME, PORT)
    print 'Presione Ctrl+C para detener...'
    try:
        http_server.serve_forever()
    except KeyboardInterrupt:
        pass
    http_server.server_close()
    print 'Deteniendo Server HTTP - %s:%s' % (HOST_NAME, PORT)
```

1 Link al ejemplo 3: <http://pastebin.com/9TW5kvA4>



Universidad Nacional de Luján
Departamento de Ciencias Básicas
Teleinformática y Redes

Ejecute el script servidor HTTP con el comando `python http3.py`. Luego, abra un navegador web e ingrese a la URL <http://localhost:8000> con el depurador del navegador activo.

14.a) ¿La página es siempre igual (estática) o dinámica? ¿De qué manera se intenta averiguar el Sistema Operativo del cliente?

14.b) ¿Qué sucede si se ingresa a la URL http://localhost:8000/ir_a/www.unlu.edu.ar? ¿Por qué? Analice el comportamiento con el depurador del navegador.

14.c) ¿Qué sucede si se ingresa a la URL http://localhost:8000/no_existe? ¿Por qué? Analice el comportamiento con el depurador del navegador.

Bibliografía

- "Comunicaciones y Redes de Computadoras", Sexta Edición, William Stallings, Prentice Hall. Capítulo 19.4: "Protocolo de Transferencia de Hipertexto (HTTP)".
- "Redes de Computadoras", Tercera Edición, Andrew Tannenbaum, Prentice Hall. Capítulo 7.6: "La World Wide Web".
- RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1

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/2/contenido.html>
- Guías de inicio en Python: <http://python.org.ar/AprendiendoPython>