



Actividad de presentación de la asignatura

Índice:

1. Introducción	3
2. Actividad 0.1 - HTTP Introduction	3
3. Actividad 0.2 - UDP and TCP: Comparison of Transport Protocols	4
4. Actividad 0.3 - Práctica telnet/http.....	5
5. Actividad 0.4 - Usando cUrl	6
6. Actividad 0.5 - Práctica servidor web	7
7. Actividad 0.6 - Repositorio Github	8
8. Conclusión	8
9. Bibliografía	9

1. Introducción

Vamos a realizar unos ejercicios de introducción a las redes y a los servidores web.

2. Actividad 0.1 - HTTP Introduction

¿Quién, dónde y cuándo se crea el primer servidor web?

El primer servidor web lo crea Tim Berners Lee, físico del CERN, en 1990.

¿Qué es pila de protocolos usados por http?

TCP/IP.

¿Componentes de una URL?

Protocolo, dominio o anfitrión, puerto, ruta de archivos o recursos y cadena de búsquedas.

¿Pasos en la recuperación de una página web mediante HTTP?

El cliente manda una petición hacia el servidor web, y este envía una respuesta junto con un código de respuesta. La petición suele ser del tipo GET para obtener páginas web.

Diferencia entre páginas dinámicas y estáticas

Una página dinámica es aquella que se construye sobre la marcha, mientras que una estática suele ser un fichero, una página que no cambia y siempre está ahí.

Request. Métodos principales

GET, que solicita un recurso; POST, que envía datos para que sean procesados por el recurso; HEAD, igual que GET, pero solo recibe la cabecera; PUT, igual que POST, pero hace referencia a los datos y no al recurso; y DELETE, que borra el recurso

Response. Códigos

Códigos con formato 1xx son respuestas informativas. Indica que la petición ha sido recibida y se está procesando.

Códigos con formato 2xx son respuestas correctas. Indica que la petición ha sido procesada correctamente.

Códigos con formato 3xx son respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.

Códigos con formato 4xx son errores causados por el cliente. Indica que ha habido un error en el procesamiento de la petición a causa de que el cliente ha hecho algo mal.

Códigos con formato 5xx son errores causados por el servidor. Indica que ha habido un error en el procesamiento de la petición a causa de un fallo en el servidor.

Content type. Tipos principales

El Content-Type indica el tipo de contenido de la petición en POST o PUT; consiste en un tipo: type y un subtipo: subtype. Los tipos principales son: text/html, text/css, audio/aac e image/jpeg.

3. Actividad 0.2 - UDP and TCP: Comparison of Transport Protocols

Diferencias entre udp y tcp?

La cabecera de UDP tiene 8 bytes mientras que la de TCP tiene 20 bytes, UDP no tiene un control de la congestión y TCP sí, los paquetes que se envían por UDP pueden llegar desordenados mientras que los de TCP no, y TCP no envía los paquetes inmediatamente y UDP sí.

¿Qué aplicaciones usan tcp?

HTTP, FTP, SMTP y Telnet.

¿Qué aplicaciones usan udp?

DHCP, DNS, SNMP, TFTP, VoIP e IPTV.

¿Qué capa almacena el puerto?

La capa de transporte.

¿Qué capa almacena la dirección IP?

La capa de red.

¿Qué es three-way handshake?

Primero el emisor pregunta al receptor si quiere iniciar una conexión, este responde y una vez que el emisor recibe la respuesta envía un paquete al emisor de que la conexión ha sido establecida.

4. Actividad 0.3 - Práctica telnet/http

Probamos los siguientes ejemplos.

- telnet www.profesordeinformatica.com 80:

```
C:\Users\usuario>telnet www.profesordeinformatica.com 80_
```

Y escribimos GET / HTTP/1.1.

```
HTTP/1.1 400 Bad Request
Server: nginx
Date: Thu, 26 Sep 2024 06:41:01 GMT
Content-Type: text/html
Content-Length: 150
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

- telnet localhost 80:

```
C:\Users\usuario>telnet localhost 80
```

Y escribimos GET / HTTP/1.1.

```
HTTP/1.1 400 Bad Request
Date: Thu, 26 Sep 2024 06:42:08 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Por último, probamos OPTIONS / HTTP/1.1.

```
HTTP/1.1 400 Bad Request
Date: Thu, 26 Sep 2024 06:42:46 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
Content-Length: 325
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
    <html><head>
        <title>400 Bad Request</title>
    </head><body>
        <h1>Bad Request</h1>
        <p>Your browser sent a request that this server could not understand.<br />
        </p>
        <hr>
        <address>Apache/2.4.58 (Win64) O
        </address>
    </body></html>

Se ha perdido la conexión con el host.
```

5. Actividad 0.4 - Usando cUrl

cUrl es una abreviatura de “Client URL”. Los comandos de cURL están diseñados para funcionar como una forma de verificar la conectividad a las URL y como una gran herramienta para transferir datos.

- 1- Obtén la página principal de un servidor web: curl https://www.example.com/

```
C:\Users\usuario>curl https://www.google.com
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="es"><head><meta content="Google.es permite
acceder a la informaci n mundial en castellano, catal n, gallego, euskara e ingl s." name="description"><meta content="n
oodp, " name="robots"><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding
/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="95dqs4SSGV3hL01U_gkro
A">(function(){var _g={fkt:'NVcx7r EGZGh PEP1r Hm-Op', kEXPT:' .793108.2907188.653.432.3.45.538616.2872.2891.89155.203622
```

- 2- Obtén una página web de un servidor usando el puerto 443: curl http://www.example.com:443/

```
C:\Users\usuario>curl https://www.google.com:443
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="es"><head><meta content="Google.es permite
acceder a la informaci n mundial en castellano, catal n, gallego, euskara e ingl s." name="description"><meta content="n
oodp, " name="robots"><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding
/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="jNVq2YYLXkh3z6EZqN28G
```

- 3- Obtén la página principal de un servidor FTP: curl ftp://ftp.rediris.com

```
C:\Users\usuario>curl ftp://ftp.rediris.com
drwxr-xr-x  4 14      50      4096 Sep 20  2017 .
drwxr-xr-x  4 14      50      4096 Sep 20  2017 ..
lrwxrwxrwx  1 14      50           23 Jun  8  2017 debian -> sites/debian.org/debian
lrwxrwxrwx  1 14      50           26 Jul 18  2017 debian-cd -> sites/debian.org/debian-cd
drwxr-xr-x  2 14      50           8192 Jun 26  2021 mirror
drwxrwxr-x 61 14      50           8192 Apr 25  2023 sites
-rw-r--r--  1 14      50           93 Jun  8  2017 welcome.msg
```

- 4- Obtén una página web y guárdala en un archivo local con un nombre específico: curl -o thatpage.html http://www.example.com/

```
C:\Users\usuario>curl -o googlearchivo.html https://www.google.com/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left   Speed
100 22354    0 22354    0     0  48359      0 --:--:-- --:--:-- --:--:-- 48490
```

- 5- Obtén un archivo de mensaje de un servidor FTP: `curl ftp://ftp.rediris.com/welcome.msg`

```
C:\Users\usuario>curl ftp://ftp.rediris.com/welcome.msg
Bienvenido al servicio de replicas de RedIRIS.
Welcome to the RedIRIS mirror service.
```

6. Actividad 0.5 - Práctica servidor web

- Simple Web Server: escribimos `python -m http.server 8000`

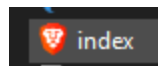
```
C:\Users\usuario>python -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

Y al ir a la dirección `localhost:8000` nos aparecerá la página que tenemos por index.



Esto es un encabezado

Y esto es un párrafo.



- HTTP Server: hay algún tipo de error en el código.

```
C:\Users\usuario>python server.py 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
Exception occurred during processing of request from ('::1', 50183, 0, 0)
Traceback (most recent call last):
  File "C:\Users\usuario\AppData\Local\Programs\Python\Python312\Lib\socketserver.py", line 692, in process_request_thread
    self.finish_request(request, client_address)
  File "C:\Users\usuario\server.py", line 1324, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "C:\Users\usuario\server.py", line 678, in __init__
    super().__init__(*args, **kwargs)
  File "C:\Users\usuario\AppData\Local\Programs\Python\Python312\Lib\socketserver.py", line 761, in __init__
    self.handle()
  File "C:\Users\usuario\server.py", line 442, in handle
    self.handle_one_request()
  File "C:\Users\usuario\server.py", line 430, in handle_one_request
    method()
  File "C:\Users\usuario\server.py", line 682, in do_GET
    f = self.send_head()
        ^^^^^^^^^^^^^^^^^
  File "C:\Users\usuario\server.py", line 727, in send_head
    ctype = self.guess_type(path)
        ^^^^^^^^^^^^^^^^^
  File "C:\Users\usuario\server.py", line 907, in guess_type
    guess, _ = mimetypes.guess_file_type(path)
        ^^^^^^^^^^^^^^^^^
```

- Dummy Web Server: guardamos el código de la página en un archivo llamado server.py y escribimos python server.py 8000; y entramos en localhost:8000



Esto es un encabezado

Y esto es un párrafo.

7. Actividad 0.6 - Repositorio Github

Crea una cuenta en Github, si no la tienes ya. Crea un repositorio en Github con el nombre del módulo. El repositorio incluirá una carpeta para cada tema: “Tema0”, “Tema1”, ... “TemaN” en el que incluirás las actividades que se te indiquen expresamente. El repositorio incluirá un archivo “README” en el que se enlazarán la solución a los ejercicios incluidos en las carpetas anteriores.

<https://github.com/ericher836/ASIR-SREI>

8. Conclusión

Hemos aprendido cómo funciona la web en su nivel más bajo y cómo se realiza la conexión de los servidores y clientes, además de realizar algunos ejemplos de funcionamiento.

9. Bibliografía

<https://curl.se/docs/manual.html>
<https://www.hostinger.es/tutoriales/comando-curl>
<https://www.sapalomera.cat/moodlecf/RS/1/course/module7/7.2.4.1/7.2.4.1.html>
<https://www.sapalomera.cat/moodlecf/RS/1/course/module7/7.2.4.2/7.2.4.2.html>
<https://www.youtube.com/watch?v=eesqK59rhGA>
<https://www.youtube.com/watch?v=DuSURHrZG6I>
<https://www.youtube.com/watch?v=Vdc8TCESIg8>
<https://www.youtube.com/watch?v=xpBpGC08f4Q>
<http://www.profesordeinformatica.com/servicios/http/telnet>
<http://www.lawebdelprogramador.com/foros/Windows-10/1510815-Como-activar-Telnet-en-Windows-10.html>
<https://docs.python.org/3/library/http.server.html>
<https://github.com/python/cpython/blob/main/Lib/http/server.py>
<https://gist.github.com/kabinpokhrel/6fd1275603e9d5f1e284be717cbd1bff>
<https://github.com/Github-Classroom-Cybro/Learn-Github>
<https://guides.github.com/features/mastering-markdown/>