

Actividad de presentación de la asignatura Daniel Damota Maldonado

Visualiza los siguientes videos y responde a las cuestiones planteadas a continuación

Actividad 0.1 - HTTP Introduction

<https://www.youtube.com/watch?v=eesqK59rhGA>

<https://www.youtube.com/watch?v=DuSURHrZG6I>

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

El primer servidor web fue creado por Tim Berners-Lee, un científico británico del CERN (Organización Europea para la Investigación Nuclear) en Suiza. Fue lanzado el 6 de agosto de 1991, y se considera el nacimiento oficial de la World Wide Web (WWW).

¿Qué es pila de protocolos usados por http?

La pila de protocolos usada por HTTP (Hypertext Transfer Protocol) se refiere a los distintos niveles o capas de protocolos de red que permiten la comunicación en la web.

¿Componentes de una URL?

Un **URL** se compone de los siguientes elementos:

1. Esquema/Protocolo: Indica el protocolo a usar (e.g., <http://>, <https://>).
2. Dominio/Host: El nombre del servidor o IP (e.g., www.ejemplo.com).
3. Puerto (opcional): Especifica el puerto (e.g., :8080).
4. Ruta/Path: La ubicación del recurso en el servidor (e.g., /articulos/).
5. Consulta/Query (opcional): Parámetros adicionales (e.g., ?id=123).
6. Fragmento (opcional): Un ancla dentro de la página (e.g., #seccion).

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

- **Resolución DNS:** El navegador convierte el dominio en una dirección IP.
- **Conexión TCP:** Se establece una conexión entre el navegador y el servidor.
- **Solicitud HTTP:** El navegador envía una solicitud HTTP al servidor.
- **Respuesta HTTP:** El servidor responde con el contenido de la página (HTML, CSS, etc.).
- **Renderizado:** El navegador interpreta y muestra la página web.

Diferencia entre páginas dinámicas y estáticas

Páginas estáticas:

- El contenido no cambia a menos que se edite manualmente el archivo.
- Están predefinidas en archivos HTML que el servidor entrega tal cual al navegador.
- Ejemplo: un sitio web de una empresa con información fija

Páginas dinámicas:

- El contenido se genera en tiempo real, basado en solicitudes del usuario o datos del servidor.
- Utilizan lenguajes de programación como PHP, Python o JavaScript para generar el HTML dinámicamente.
- Ejemplo: una red social donde el contenido cambia según el usuario que accede.

¿Cómo usar telnet para acceder a un servidor web? (solución en actividad 0.3)

Request. Métodos principales

Métodos: GET (envía recursos) y POST (envía datos para que sean procesados), PUT, HEAD y DELETE.

Response. Códigos:

- Códigos con formato 1xx: Respuestas informativas. Indica que la petición ha sido recibida y se está procesando.
- Códigos con formato 2xx: Respuestas correctas. Indica que la petición ha sido procesada correctamente.

- Códigos con formato 3xx: Respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.
- Códigos con formato 4xx: Errores causados por el cliente. Indica que ha habido un error en el procesado de la petición a causa de que el cliente ha hecho algo mal.
- Códigos con formato 5xx: Errores causados por el servidor. Indica que ha habido un error en el procesado de la petición a causa de un fallo en el servidor.

Content type. Tipos principales:

- text: Indica que el contenido es texto plano. Ejemplos de subtipos: html, xml
- multipart: Indica que tiene múltiples partes de datos independientes. Ejemplos de subtipos: form-data, digest
- message: Para encapsular un mensaje existente. Por ejemplo cuando queremos responder a un mensaje de correo incorporando el mensaje origen. Ejemplos de subtipos: partial, rfc822
- image: Indica que es una imagen. Ej de subtipos: png, gif
- audio: Indica que es un audio. Ejemplos de subtipos: mp3, 32kadpcm
- video: Indica que es un video. Ejemplos de subtipos: mpeg, avi
- application: Indica que se trata de datos de aplicación los cuales pueden ser binarios. Ejemplos de subtipos: json, pdf

Actividad 0.2 - UDP and TCP: Comparison of Transport Protocols

<https://www.youtube.com/watch?v=Vdc8TCESlg8>

Diferencias entre udp y tcp (min 2:46 y 4:15)

1. Conexión:

- a. **TCP:** Es un protocolo orientado a la conexión. Establece una conexión antes de enviar datos.
- b. **UDP:** Es un protocolo sin conexión. Envía datos sin establecer una conexión previa.

2. Fiabilidad:

- a. **TCP:** Garantiza la entrega de paquetes y el orden correcto. Realiza corrección de errores.
- b. **UDP:** No garantiza la entrega ni el orden. Es más rápido, pero menos fiable.

¿Qué aplicaciones usan tcp?

HTTP, IMAC, POP, SMTP

¿Qué aplicaciones usan udp?

DNS, SNMP

¿Qué capa almacena el puerto?

Capa de Transporte: TCP y UDP utilizan números de puerto para identificar aplicaciones específicas.

¿Qué capa almacena la dirección IP?

Capa de Red: La dirección IP se utiliza para identificar dispositivos en una red.

¿Qué es three-way handshake?

El **three-way handshake** (apretón de manos en tres pasos) es el proceso que utiliza el protocolo **TCP** para establecer una conexión confiable entre un cliente y un servidor antes de comenzar a intercambiar datos. Este mecanismo garantiza que ambas partes estén listas para la comunicación.

Actividad 0.3 - Práctica telnet/http

<https://www.youtube.com/watch?v=xBpGC08f4Q&t=189s>

<http://www.profesordeinformatica.com/servicios/http/telnet>

Lee el artículo y prueba los ejemplos sugeridos en él.

Nota: Si usamos Windows 10, tenemos que activar "telnet"

<http://www.lawebdelprogramador.com/foros/Windows-10/1510815-Como-activar-Telnet-en-Windows-10.html>

Uso de comando telnet www.google.com 80 y luego GET / HTTP/1.1 para obtener el código de la página web:

```
Date: Mon, 23 Sep 2024 07:46:10 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-1nJCtuvnUKHdI4WS9SvUQ' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: AEC=AVYB7coMRHRNz3xZdtyMD0D5pm64HdTgFi8kL9YU-3H8Gj0hMnfWmbag; expires=Sat, 22-Mar-2025 07:46:10 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked

537f
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="es"><head><meta content="Google.es permite acceder a la información mundial en castellano, catalán, gallego, euskara e inglés." name="description"><meta content="noodp, " 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="1nJCtuvnUKHdI4WS9SvUQ">(function(){var _g={kEI:'whzxZva5FIif5NoPntLx0Qk',kEXPI:'0,793344,3356568,90133,2872,2891,562,42466,30022,16105,20362,2,6700,41948,64698,17667,2006,8155,23350,22436,9779,62657,76209,15816,1804,35269,11813,1632,13496,15783,21780,5217978,997,147,4,5991283,2841175,190,305,134,3,1,7,27981327,13290,2376,43886,3,1603,3,2124363,23029351,12799,14986,9525,2,40943,46567,98,101,10882,15164,8182,58664,12436,6752,156,2483,13505,7736,9138,4599,328,3217,5,1238,1766,2947,3813,826,5,13010,2,3763,3624,5633,687,2740,3,3849,1204,1,2,3015,2694,4476,11871,682,283,381,5842,2004,5802,54,2213,2,9,746,4,12661,2378,1484,316,662,529,4705,4446,3288,1824,5029,1539,4973,2392,3392,2893,2,2647,2728,2087,729,7112,3553,4,3993,153,285,1,2,166,5419,1002,1517,251,597,1771,31,299,201,433,366,14,3033,23,90,1408,3027,246,224,24,2959,470,219,4,1,2,628,823,41,1103,11,812,6,1,2,490,136,48,141,2268,523,80,514,206,1689,486,3365,415,10,441,492,384,86,134,1751,195,470,1453,91,47,4,140,576,38554,182,657,3,55,556,35,374,244,150,35,183,333,15,16,657,58,374,143,15,13,18,134,145,115,383,153,353,516,3,144,3,383,330
```

Obtención de la página de www.profesordeinformatica.com:

```
HTTP/1.1 301 Moved Permanently
Content-Type: text/html; charset=iso-8859-1
Content-Length: 242
Connection: keep-alive
Keep-Alive: timeout=15
Date: Mon, 23 Sep 2024 15:47:39 GMT
Server: Apache
Location: https://profesordeinformatica.com/

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
    <html><head>
        <title>301 Moved Permanently</title>
    </head><body>
        <h1>Moved Permanently</h1>
        <p>The document has moved <a href="https://profesordeinformatica.com/">here</a>.</p>
    </body></html>
```

Actividad 0.4 - Usando cUrl

<https://curl.se/docs/manual.html>

Busca información sobre el comando curl y muestra al menos cinco ejemplos de uso

a) Conseguir la página principal de un servidor web:

```
C:\Users\usuario>curl -o google2.html www.google.com
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 22264 0 22264 0 0 107k 0 --:--:-- --:--:-- --:--:-- 108k

C:\Users\usuario>
```

b) Obtener archivo README de un servidor FTP:

```
C:\Users\Usuario>curl -u dlpuser:rNrKYTX9g7z3RgJRmxWuGHbeu ftp://ftp.dlptest.com/README -o README
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0
curl: (78) The file does not exist
```

c) Listar directorios de servidor FTP:

```
C:\Users\Usuario>curl ftp://ftp.slackware.com/
lrwxrwxrwx 1 0 0 17 Mar 19 2024 debian -> pub/debian/debian
lrwxrwxrwx 1 0 0 18 Mar 19 2024 debian-cd -> pub/debian-cdimage
lrwxrwxrwx 1 0 0 18 Mar 19 2024 debian-cdimage -> pub/debian-cdimage
drwxr-xr-x 4 0 0 4096 Mar 19 2024 pub
-rw-r--r-- 1 0 0 754 Mar 19 2024 welcome.msg
```

d) Obtener dos documentos a la vez:

```
C:\Users\Usuario>curl -O ftp://ftp.dlptest.com/hello.txt & curl -O http://www.example.com:8000/index.html &
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- 0:00:03 --:--:-- 0
```

e) Descargar archivo de servidor FTPS:

```
C:\Users\Usuario>curl -O ftps://test.rebex.net/readme.txt
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 379 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0
curl: (18) transfer closed with 379 bytes remaining to read
```

Actividad 0.5 - Práctica servidor web

1. Visita los siguientes enlaces:

Simple web server (ejemplo 1)

<https://docs.python.org/3/library/http.server.html>

```
python -m http.server 8000
```

http server (ejemplo 2)

<https://github.com/python/cpython/blob/main/Lib/http/server.py>

dummy web server (ejemplo 3)

<https://gist.github.com/kabinpokhrel/6fd1275603e9d5f1e284be717cbd1bff>

2. Instala Python.
3. Ejecuta los ejemplos mostrados con anterioridad.
4. Publica en GitHub los ejemplos llevados a cabo. Los ejemplos se acompañarán con capturas de pantalla en las que se muestre su funcionamiento.

Ejemplo 1:

Inicializamos el servidor web en el puerto 8000 y luego accedemos a localhost:8000 desde el navegador:

```
C:\Users\usuario>python -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
::1 - - [25/Sep/2024 08:37:24] "GET / HTTP/1.1" 200 -
::1 - - [25/Sep/2024 08:37:24] code 404, message File not found
::1 - - [25/Sep/2024 08:37:24] "GET /favicon.ico HTTP/1.1" 404 -
::1 - - [25/Sep/2024 08:37:32] "GET /.VirtualBox/ HTTP/1.1" 200 -
::ffff:10.6.0.5 - - [25/Sep/2024 08:39:44] "GET / HTTP/1.1" 200 -
::ffff:10.6.0.5 - - [25/Sep/2024 08:39:44] code 404, message File not found
::ffff:10.6.0.5 - - [25/Sep/2024 08:39:44] "GET /favicon.ico HTTP/1.1" 404 -
::ffff:10.6.0.5 - - [25/Sep/2024 08:39:47] "GET /Documents/ HTTP/1.1" 200 -
::ffff:10.6.0.5 - - [25/Sep/2024 08:40:06] "GET /ntuser.ini HTTP/1.1" 200 -
::ffff:10.6.0.5 - - [25/Sep/2024 08:40:11] "GET /Downloads/ HTTP/1.1" 200 -
```

Directory listing for /

- [.swt/](#)
- [.VirtualBox/](#)
- [.vscode/](#)
- [3D Objects/](#)
- [AppData/](#)
- [Configuración local/](#)
- [Contacts/](#)
- [Cookies/](#)
- [Datos de programa/](#)
- [Desktop/](#)
- [Documents/](#)
- [Downloads/](#)
- [Entorno de red/](#)
- [Favoritos/](#)
- [Impresoras/](#)
- [Links/](#)
- [Menú Inicio/](#)
- [Mis documentos/](#)
- [Music/](#)
- [NTUSER.DAT](#)
- [ntuser.dat.LOG1](#)
- [ntuser.dat.LOG2](#)
- [NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR_0.regtrans-ms](#)
- [NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR_1.regtrans-ms](#)
- [NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR_2.regtrans-ms](#)
- [NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.blf](#)
- [NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf](#)
- [NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000001.regtrans-ms](#)
- [NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000002.regtrans-ms](#)
- [ntuser.ini](#)
- [OneDrive/](#)
- [Pictures/](#)
- [Plantillas/](#)
- [Reciente/](#)
- [Saved Games/](#)
- [Searches/](#)
- [SendTo/](#)
- [Videos/](#)

Ejemplo 3:

Utilizamos el código de github y lo guardamos en la carpeta de usuario del equipo:

Nombre	Fecha de modificación	Tipo	Tamaño
.swt	19/09/2024 9:15	Carpeta de archivos	
.VirtualBox	19/09/2024 8:23	Carpeta de archivos	
.vscode	18/09/2024 13:53	Carpeta de archivos	
Búsquedas	18/09/2024 13:47	Carpeta de archivos	
Contactos	18/09/2024 13:46	Carpeta de archivos	
Descargas	25/09/2024 8:53	Carpeta de archivos	
Documentos	24/09/2024 12:04	Carpeta de archivos	
Escritorio	24/09/2024 11:46	Carpeta de archivos	
Favoritos	18/09/2024 13:46	Carpeta de archivos	
Imágenes	18/09/2024 13:47	Carpeta de archivos	
Juegos guardados	18/09/2024 13:46	Carpeta de archivos	
Música	18/09/2024 13:46	Carpeta de archivos	
Objetos 3D	18/09/2024 13:46	Carpeta de archivos	
OneDrive	18/09/2024 13:48	Carpeta de archivos	
Videos	23/09/2024 9:01	Carpeta de archivos	
Vínculos	18/09/2024 13:46	Carpeta de archivos	
Sin confirmar 381286.crdownload	25/09/2024 8:53	Archivo CRDOWN...	48 KB
server	25/09/2024 8:55	Archivo de origen ...	1 KB

Ahora ejecutamos el comando `python server.py:8000`

```
C:\Users\usuario>python server.py 8000
serving at port 8000
127.0.0.1 - - [25/Sep/2024 08:56:03] "GET / HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:03] "GET / HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:05] code 404, message No permission to list directory
10.6.0.5 - - [25/Sep/2024 08:58:05] "GET /Plantillas/ HTTP/1.1" 404 -
10.6.0.5 - - [25/Sep/2024 08:58:07] "GET /server.py HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:10] "GET /Videos/ HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:11] "GET /Videos/Captures/ HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:12] "GET /Videos/Captures/desktop.ini HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:16] "GET /AppData/ HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:17] "GET /AppData/Local/ HTTP/1.1" 200 -
10.6.0.5 - - [25/Sep/2024 08:58:23] code 404, message No permission to list directory
10.6.0.5 - - [25/Sep/2024 08:58:23] "GET /Datos%20de%20programa/ HTTP/1.1" 404 -
```

Actividad 0.5. 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.

La página README del repositorio debe tener un aspecto parecido al mostrado a continuación:

Nombre del módulo Este repositorio incluye actividades llevadas a cabo en el módulo <i>nombredelmódulo</i>	
Tema 0 - Nombre tema 0	
Ejercicio 1	Breve descripción 0.1
Ejercicio 2	Breve descripción 0.2
...	...
Tema 1 - Nombre tema 1	
Ejercicio 1	Breve descripción 0.1
Ejercicio 2	Breve descripción 0.2
...	...

Nota: Si no has utilizado antes Github, es recomendable que cree un repositorio nuevo llamado “prueba” que incluya una página “README.md”. Utiliza markdown para que incluya varias cabeceras, texto, una lista, un gráfico y una tabla. Previamente se recomienda leer:

- <https://github.com/Github-Classroom-Cybro/learn-github>
- <https://guides.github.com/features/mastering-markdown/>

Servicios de Redes e Internet ASIR2

Este repositorio incluye actividades llevadas a cabo en el módulo de servicios de redes e internet por

Tema 0 - Introducción

Nombre	Descripción
Actividad de introducción	Ejercicios básicos sobre servidores web y sus protocolos

Tema 1 - Servidores Web



Servicios de Redes e Internet ASIR2

Este repositorio incluye actividades llevadas a cabo en el módulo de servicios de redes e internet por

Tema 0 - Introducción

Nombre	Descripción
Actividad de introducción	Ejercicios básicos sobre servidores web y sus protocolos

Tema 1 - Servidores Web

Enlace al repositorio: <https://github.com/DanielDamota04/Servicios-de-Redes-e-Internet-ASIR2>