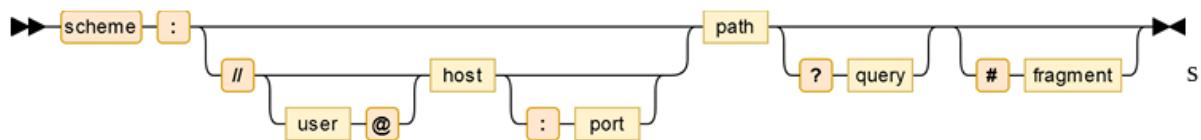


# PROTOCOL HTTP: SCHEME, PORT, PATH

Como hemos dicho, las URL son identificadores de recursos. Los recursos pueden ser muy diversos: páginas, imágenes, vídeos, archivos, etcétera. Por ejemplo:

- La url identifica la página principal del CERN
- La url identifica una imagen (de Tim, por supuesto)
- La url identifica un vídeo.

Una URL se compone de 7 componentes:



Indica brevemente para qué sirven cada uno de los siguientes componentes:

- scheme
- user
- host
- port
- path
- query
- fragment

Vamos a ver con mayor detalle estos componentes.

## Scheme

El protocolo de acceso al recurso indica la forma concreta en que deben comunicarse lo que hace la petición y lo que la responde.

Por ejemplo, el protocolo sirve para comunicarse con un servidor web. La especificación del protocolo es muy extensa, pero podemos ver un ejemplo sencillo.

En su forma más utilizada, HTTP sirve para obtener recursos.

Supongamos que queremos obtener la página principal del CERN. Entonces, debemos realizar una petición HTTP a su servidor (). Los datos que enviaremos al servidor deben seguir exactamente el protocolo.

Utilizaremos el programa para comunicarlo con el servidor del CERN.

Abre un terminal y ejecuta el siguiente pedido:

**nc info.cern.ch 80**

Con este pedido hemos conectado al puerto del servidor del CERN.

Ahora el servidor está esperando a que le enviemos datos.

Si enviamos cualquier dato al servidor, éste no será capaz de comprender la petición. Debemos enviarle los datos adecuados, siguiendo el protocolo HTTP.

Prueba a escribir algo, y pulsa Intro para enviar los datos al servidor.

El servidor responderá con un error. La petición no fue correcta.

```
user@pc:~$ nc info.cern.ch 80
sdfkljsdafklasfsfda
HTTP/1.1 400 Bad Request
Date: Tue, 15 Sep 2020 11:06:40 GMT
Server: Apache
Content-Length: 226
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>
</body></html>
```

Para realizar la petición correcta debemos hacer lo siguiente:

1. Conectamos con el servidor:

**nc info.cern.ch 80**

- Escribimos el pedido,
- después el recurso que queremos
- después la versión del protocolo
- después un "retorno de carro" Ctrl + V Ctrl + M
- y por último un "retorno de línea" Intro

**GET /index.html HTTP/1.0 Ctrl+V Ctrl+M Intro**

- Especificamos el Host al que estamos conectando introduciendo:
- "retorno de carro" Ctrl + V Ctrl + M
- y "retorno de línea" Intro

**Host: info.cern.ch Ctrl+V Ctrl+M Intro**

- Indicamos al servidor que ha finalizado la petición con "retorno de carro" Ctrl + V Ctrl + M
- y "retorno de línea" Intro

### **Ctrl+V Ctrl+M Intro**

Esta vez sí, el servidor nos responde con un 200 OK y nos devuelve el recurso solicitado.

**EJERCICIO:** Haz una conexión nc a <http://man.openbsd.org/> a través del terminal y luego repite el proceso en el navegador para comprobar los resultados de la conexión http. Envía Dos capturas de pantalla que demuestren que has hecho el ejercicio.

```
user@pc:~$ nc info.cern.ch 80
GET /index.html HTTP/1.0^M
Host: info.cern.ch^M
^M
HTTP/1.1 200 OK
Date: Tue, 15 Sep 2020 11:15:58 GMT
Server: Apache
Last-Modified: Wed, 05 Feb 2014 16:00:31 GMT
ETag: "40521bd2-286-4f1aadb3105c0"
Accept-Ranges: bytes
Content-Length: 646
Connection: close
Content-Type: text/html

<html><head></head><body><header>
<title>http://info.cern.ch</title>
</header>

<h1>http://info.cern.ch - home of the first website</h1>
<p>From here you can:</p>
<ul>
<li><a href="http://info.cern.ch/hypertext/WWW/TheProject.html">Browse the first website</a></li>
<li><a href="http://line-mode.cern.ch/www/hypertext/WWW/TheProject.html">Browse the first website using the line-mode browser simulator</a></li>
<li><a href="http://home.web.cern.ch/tonics/birth-web">Learn about the birth o
```

en la barra de direcciones. Puedes comprobarlo de la siguiente forma:

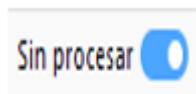
1. Abre una nueva pestaña de Firefox
2. Pulsa **F12** para abrir las **Herramientas de desarrollador**
3. Ir a la pestaña
4. Introduce en la barra de direcciones y navega a la página
5. Firefox te mostrará todos los recursos que está descargando del servidor
6. Selecciona el recurso
7. En la pestaña lateral,

[Cabeceras](#)

baja hasta

## ▼ Cabeceras de la petición (397 B)

y activa la opción



Aquí puedes ver toda la petición HTTP que se ha realizado:

The screenshot shows a browser window for <http://info.cern.ch>. The page title is "http://info.cern.ch - home of the first website". Below it, a list of links is provided: "Browse the first website", "Browse the first website using the line-mode browser simulator", "Learn about the birth of the web", and "Learn about CERN, the physics laboratory where the web was born". The browser's developer tools Network tab is open, showing two requests: "index.html" and "favicon.ico". The "Cabeceras" (Headers) tab is selected, displaying the following header information for the "index.html" request:

```
GET / HTTP/1.1
Host: info.cern.ch
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

## Path

Si no se especifica ningún archivo en el PATH, por defecto, el servidor apache servirá el archivo especificado en la directiva `DirectoryIndex`, “`index.html`”, y si este archivo no existe, servirá una página que listará los archivos del directorio.

Pruébalo:

Vamos a crear varios archivos en el directory :

Colocamos un archivo de imagen:imagen.jpg

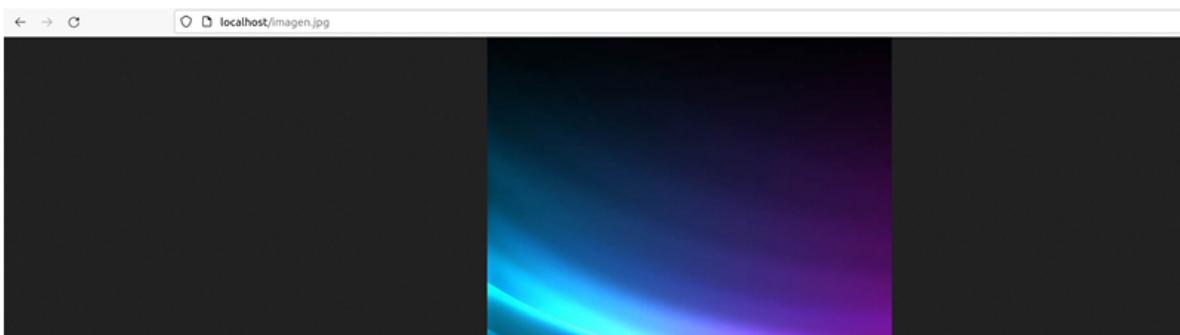
Creamos una carpeta “complementos” y ponemos en un archivo:imagen2.jpg



Si ahora hacemos una petición al servidor sin indicar ningún archivo el servidor nos enviará el archivo index.html



Si especificamos algún archivo en la petición (ej:imagen.jpg) Apache nos servirá este archivo.



Si el archivo que ponemos en la url (ex:Imagenxxx.jpg ) no existe, apache nos servirá una página " **Not Found** "



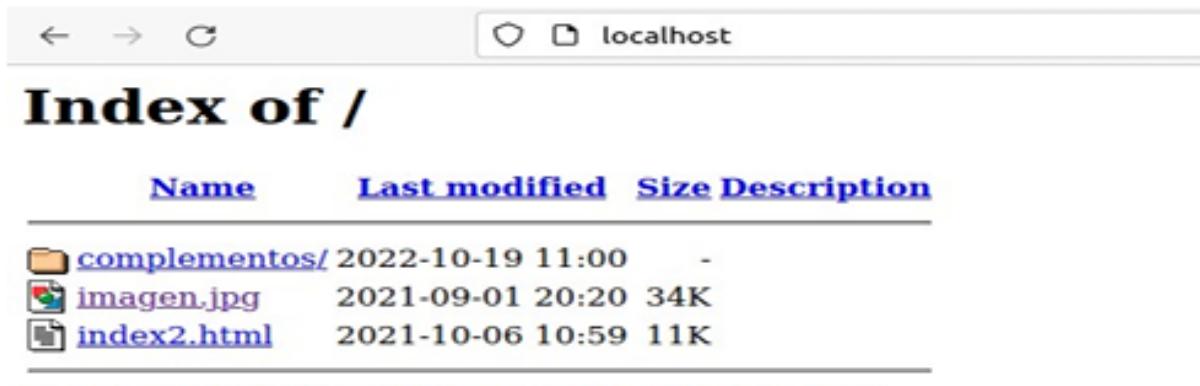
## Not Found

The requested URL was not found on this server.

Apache/2.4.41 (Ubuntu) Server at localhost Port 80

Por último, cambiamos el nombre al archivo index.html a index2.html para que no exista el index.html :

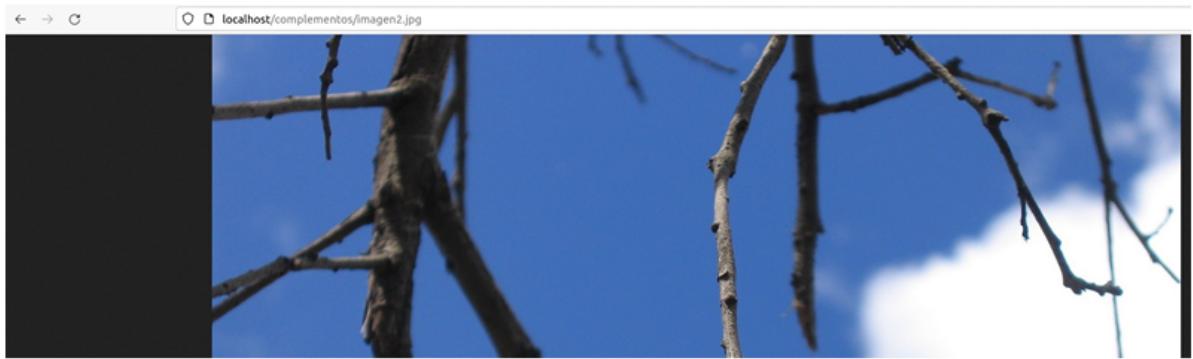
... y comprobamos que Apache nos muestra una página con la lista de archivos del directorio, cuando no especificamos ningún archivo en la url y no existe un index.html (no existe el archivo especificado en la directiva DirectoryIndex):



**IMPORTANTE** : Vuelve a poner el nombre de index.html en el index2.html

En **Path** de la url también se pueden especificar los directorios donde se encuentra el archivo que se solicita. Apache buscará estos archivos a partir del **DocumentRoot** .

Por ejemplo, si **DocumentRoot** es **/var/www/html** , y hacemos esta petición:  
**http://localhost/complementos/imagen2.jpg** , Apache buscará este archivo  
**/var/www/html/complementos/imagen2.jpg** . Compruébalo:



**Ejercicio:** Lanza los siguientes comandos, indica para qué sirven y añade una captura del resultado tras lanzarlos:

- a. ifconfig
- b. ip route show
- c. ping [google.com](http://google.com)
- d. ping -c 3 [google.com](http://google.com)
- e. ping -o [google.com](http://google.com)
- f. netstat -i
- g. netstat -r
- h. netstat -a
- i. who
- j. man wget
- k. ps ax
- l. htop -u nombre\_usuario
- m. fuser -v -n tcp 8080