

Tema 2. Protocolo HTTP

José Antonio Fajardo Naranjo

fajardonaranjoja@gmail.com

Resumen

Estos apuntes pertenecen al tema 2 de la asignatura Desarrollo Web en entorno servidor impartida en el 2º curso de FPGS de Desarrollo de aplicaciones web en el IES Martín Rivero durante el curso 2023/2024.

Índice

1. Introducción	3
1.1. Funcionamiento	3
1.2. Las URL	4

1. Introducción

El protocolo HTTP busca la simpleza y la eficacia para el intercambio de documentos. Aunque su uso principal es el intercambio de los documentos de hipertexto se puede usar para transferir otro tipo de elementos.

1.1. Funcionamiento

El principio en el que se basa este protocolo es el tándem pregunta/respuesta. El cliente genera una pregunta con la forma de una petición HTTP que contiene, al menos, los datos que permiten identificar el recurso que le ha sido solicitado por el cliente. Esta petición no es más que un bloque de texto que viaja del cliente al servidor. Se compone de dos partes separadas por una línea en blanco (retorno de carro o salto de línea). La primera parte es la cabecera de la petición HTTP, que es obligatoria, mientras que la segunda es el cuerpo, que es opcional, dependiendo su aparición del tipo de petición HTTP. La línea en blanco ha de aparecer siempre, siendo completamente **obligatoria**. La respuesta mantiene el mismo formato de las peticiones.

Una respuesta HTTP sólo existe si se ha enviado previamente una petición HTTP al servidor, no tomando este último la iniciativa de enviar datos que no han sido solicitados por el cliente (con la tecnología de los **WebSockets** esto no es necesario, los servidores actualizan en tiempo real sin necesidad de petición).

El protocolo TCP se utiliza para el transporte de los bloques de texto de petición y respuesta HTTP. Por defecto, esta conexión mediante TCP se establece para cada par petición/respuesta. Sin embargo, la versión 1.1 propone una solución para transportar varios pares petición/respuesta con la misma conexión TCP. A continuación, encontramos el ejemplo utilizado en clase de paquete capturado.

```
POST /patch HTTP/1.1
Content-Type: text/html
Content-Length: 139
User-Agent: Blizzard Web Client
Host: deDE.patch.battle.net:1119
Cache-Control: no-cache
```

Figura 1: Captura de una petición HTTP

```

HTTP/1.1 200
Accept-Ranges: bytes
ETag: W/"1011-1544263226000"
Last-Modified: Sat, 08 Dec 2018 10:00:26 GMT
Content-Type: text/html
Content-Length: 1011
Date: Thu, 21 Sep 2023 11:36:41 GMT
Keep-Alive: timeout=20
Connection: keep-alive

...
<html>
  <head>
    <meta http-equiv="cache-control" content="no-cache">
  </head>
  <body>
    <form action="FormularioDavid" >
      Nombre:<input type="text" name="nombre" /><br>
      Apellidos:<input type="text" name="apellidos" /><br>
      Seleccione sistema operativo:<select name="sistema">
        <option value="Windows">Windows</option>
        <option value="Linux">Linux</option>
        <option value="Mac">Mac</option>
      </select><br>
      Lengua favorita:<select name="lenguaje" multiple>
        <option value="ingles">ingles</option>
        <option value="italiano">italiano</option>
        <option value="espa..ol">espa..ol</option>
        <option value="aleman">aleman</option>
      </select><br>
      deportes favoritos <br>
      tenis <input type="checkbox" value="tenis" name="check1">
      futbol <input type="checkbox" value="futbol" name="check1">
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>

```

Figura 2: Captura de una respuesta HTTP

1.2. Las URL

Las URL (Uniform Resource Locator) permiten localizar los recursos que se desean recuperar mediante peticiones HTTP. Se componen de una cadena de caracteres de cinco campos. Se ha de seguir el siguiente formato:

`protocolo://identificador del servidor:número de puerto/recurso?parámetros`

- Protocolo: indica el protocolo utilizado para el acceso al recurso. Nosotros usaremos HTTP aunque hay otros como FTP o mailto.
- Identificador del servidor: esta parte permite localizar el servidor en la red. Normalmente es el FQDN (Fully Qualified Domain Name), el nombre de dominio del servidor, que no da lugar a ambigüedades. Se tiene que transformar en la dirección IP para que el protocolo TCP pueda establecer una conexión con el servidor, de lo cual se encarga un servidor DNS (Domain Name System). Se puede usar una IP en la URL pero es más sencillo el FQDN.
- Número de puerto: este dato es el número de puerto TCP con el que se debe establecer la conexión. Este número funciona para que el protocolo TCP identifique una aplicación particular en una máquina. Este sistema permite el alojamiento de más de una aplicación por servidor. El puerto 80 es el que se asigna por defecto al protocolo HTTP.
- Recurso: esta parte de la URL permite determinar el recurso que se desea obtener. Puede estar compuesto por varios elementos separados por el carácter .
- Parámetros: son los datos que pasamos cuando la petición HTTP se hace, por ejemplo, sobre un recurso dinámico. Aparecen de la forma nombreDeParam=valorDeParam. Con el carácter & se pueden separar estos parámetros.

A continuación se muestran equivalencias entre los caracteres y su valor en ASCII, ya que algunos caracteres tienen una función en las URL:

Carácter	Codificación
Espacio	%20 o a veces +
”	%22
%	%25
&	%26
+	%2B
.	%2E
/	%2F
?	%3F
,	%60