



TECNICATURA SUPERIOR EN
Telecomunicaciones

ARQUITECTURA Y CONECTIVIDAD

Módulo II: Familia de Protocolos IoT

Protocolo HTTP

¿Cuáles son las Peticiones más comunes en el Protocolo HTTP? Para que se usan? Ejemplifique

En la web, los clientes, como un navegador, por ejemplo, se comunican con los distintos servidores web con ayuda del protocolo HTTP, el cual regula cómo ha de formular sus peticiones el cliente y cómo ha de responder el servidor.

El protocolo HTTP emplea varios métodos de petición diferentes, los más comunes son:

- GET
- POST
- HEAD
- OPTIONS
- TRACE
- MÉTODOS ESPECIALES

GET: Se utiliza para **solicitar un recurso**, como un archivo HTML, **del servidor web**. Cuando escribes la dirección URL www.ejemplo.com en tu navegador, este se conecta con el servidor web y le envía una petición GET:

GET /index.php

El archivo *index.php* de esta muestra de código de petición es la página de inicio de un sitio web, que el servidor enviará como respuesta al navegador.

La petición de la dirección www.ejemplo.com/test.html se formularía de forma análoga:

GET /test.html

El servidor enviaría el archivo *test.html* como respuesta.

Parámetros de URL: A la petición GET puede añadirse **más información**, con la intención de que el servidor web también la procese. Estos llamados parámetros de URL se adjuntan a la dirección URL. La sintaxis es bastante simple:

- La secuencia de petición (*query string*) se inicia con un signo de interrogación “?”.
- Todos los parámetros se componen de un nombre y un valor: “Nombre=Valor”.
- Si se han de adjuntar varios parámetros, se unen con un signo “&”.

Ejemplo: para buscar ciertas ofertas en la página web de una empresa de software, en la petición GET se indicará “Windows” como plataforma y “Office” como categoría:

GET /search?platform=Windows&category=office

Codificación del URL de la secuencia de petición: Las secuencias de petición necesitan una codificación especial, porque muchos **caracteres especiales** tienen un significado diferente. Por ejemplo, el texto “Lista HTTP” se ha de codificar como se indica a continuación para que sea aceptado como secuencia de petición:

GET /search?thema=Lista%20HTTP

POST : Cuando se tienen que enviar al servidor web paquetes grandes de datos, como imágenes o datos de formulario de carácter privado, acá, el método GET se queda corto, porque todos los datos que se transmiten se escriben en abierto en la barra de direcciones del navegador.

Usamos el método POST, que no escribe el parámetro del URL en la dirección URL, sino que lo adjunta al encabezado HTTP. Las peticiones POST suelen emplearse con **formularios digitales**.

formulario que recoge un nombre y una dirección de correo electrónico y lo envía al servidor por medio de POST:

```
<html>
<body>
<form          action="newsletter.php"          method="post">
Name:          <input          type="text"          name="name"><br>
E-mail:        <input          type="text"          name="email"><br>
<input                                          type="submit">
</form>
</body>
</html>
```

HEAD: El método de petición de HTTP HEAD se utiliza para solicitar que el servidor solo envíe el encabezado de la respuesta, sin el archivo. Alternativa conveniente cuando se han de transferir archivos muy voluminosos, ya que, con esta petición, el cliente conoce primero el **tamaño del archivo** para luego poder decidir si acepta recibirlo o no.

Por ejemplo:

```
HEAD /downloads/video1.mpeg HTTP/1.0
```

En el encabezado que el servidor le envía de respuesta, el cliente encuentra los datos sobre el tamaño del archivo en el campo “content-length”:

El servidor responde a la petición HEAD con los datos clave sobre el archivo en cuestión.

OPTIONS

Con el método OPTIONS, el cliente puede preguntar al servidor **qué métodos soporta** para el archivo de que se trata.

```
OPTIONS /download.php
```

La respuesta podría consistir en algo parecido a esto:

Respuesta del servidor a la petición OPTIONS

En el campo “allow” el servidor informa al cliente de que soporta los métodos OPTIONS, GET, HEAD y POST. En el campo “content-length”, la cifra 0 indica que no se ha enviado ningún archivo, sino solamente el encabezado.

TRACE

Con el método TRACE, puede seguirse la **ruta** que sigue una HTTP Request hacia el servidor y, desde allí, de regreso al cliente. Este seguimiento puede ejecutarse

en Windows con el comando tracert. Solo tienes que introducir en la consola (*cmd.exe*) el siguiente comando:

```
tracert www.example.com
```

Métodos especiales

Algunos métodos solo pueden aplicarse en ciertos contextos. Este es el caso del método CONNECT, que crea una conexión directa y protegida por medio de un proxy (*tunneling*), o varios métodos que guardan relación con WebDAV: PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK.

PUT, PATCH y DELETE se utilizan para **guardar archivos en el servidor**, modificarlos o eliminarlos. En la programación web habitual, estos métodos apenas desempeñan un papel, ya que el servidor los bloquea por seguridad, pero sí se emplean en el contexto de WebDAV y API REST.