

- 1) ¿Cuáles son las peticiones más comunes en el protocolo HTTP?, ¿Para que se usan? Ejemplifique

El protocolo HTTP utiliza varios tipos de peticiones o solicitudes que el cliente puede enviar al servidor para obtener o enviar información. Algunos de los tipos de peticiones más comunes incluyen:

GET: realiza una petición a un recurso específico. No permite el envío de datos a excepción si dichos datos se envían como parámetro en la Url que realiza la petición. Esta petición retorna tanto la cabecera como el contenido. Ahora, este método GET puede retornar una respuesta en formato HTML, JSON, XML, Imágenes o JavaScript. Semánticamente se utiliza para consultar información como una SELECT a la base de datos, se puede filtrar datos empleando los datos enviados por la URL.

Ejemplo de 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.

POST: Se utiliza cuando se tienen que enviar al servidor web paquetes grandes de datos, como imágenes o datos de formulario de carácter privado.

Este método no escribe el parámetro del URL en la dirección URL, sino que lo adjunta al encabezado HTTP.

Un ejemplo son los formularios digitales como en el caso a continuación:

```
<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>
```

PUT: es un método de solicitud utilizado para actualizar un recurso existente en un servidor web. Cuando se envía una solicitud PUT, el cuerpo de la solicitud incluye el recurso actualizado que se va a almacenar en el servidor, y el servidor reemplaza la versión anterior del recurso con la nueva versión proporcionada en la solicitud.

En el ejemplo siguiente se utiliza para actualizar un archivo de texto en un servidor web:

```
PUT /example.txt HTTP/1.1
```

```
Host: www.example.com
```

```
Content-Type: text/plain
```

```
Content-Length: 27
```

HEAD: se utiliza para solicitar que el servidor solo envíe el encabezado de la respuesta, sin el archivo. Esta alternativa es 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.

A continuación se muestra un 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":

| | |
|----------------|-------------------------------|
| age | 96529 |
| cache-control | max-age=604800 |
| content-length | 1495 |
| content-type | text/html; charset=UTF-8 |
| date | Fri, 06 Mar 2020 14:53:39 GMT |
| etag | "3147526947+gzip" |
| expires | Fri, 13 Mar 2020 14:53:39 GMT |
| last-modified | Thu, 17 Oct 2019 07:18:26 GMT |
| server | ECS (dcb/7F83) |
| vary | Accept-Encoding |
| x-cache | HIT |

Captura de pantalla de una respuesta HTTP a la petición HEAD

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

OPTIONS: es un método de solicitud utilizado para obtener información sobre las opciones de comunicación disponibles para un recurso en particular en un servidor web. Cuando se envía una solicitud OPTIONS, el servidor web responde con una lista de métodos HTTP y otros detalles de la comunicación permitidos para el recurso especificado. A continuación se muestra un ejemplo:

```
OPTIONS /example HTTP/1.1
```

```
Host: www.example.com
```

En este ejemplo, la solicitud OPTIONS se envía a la URL "/example" en el servidor web "www.example.com". La solicitud no incluye ningún cuerpo de mensaje, ya que no se envía ningún dato al servidor. El servidor responde con una lista de métodos HTTP permitidos para el recurso especificado, así como otra información relevante para la comunicación con el recurso.

DELETE: es un método de solicitud utilizado para solicitar que un recurso específico sea eliminado en un servidor web. Cuando se envía una solicitud DELETE, el servidor elimina el recurso especificado y responde con un código de estado HTTP para indicar si la operación de eliminación fue exitosa. A continuación se muestra un ejemplo:

```
DELETE /example.txt HTTP/1.1
```

```
Host: www.example.com
```

En este ejemplo, la solicitud DELETE se envía a la URL "/example.txt" en el servidor web "www.example.com". La solicitud no incluye ningún cuerpo de mensaje, ya que no se envía ningún dato al servidor. El servidor responde con un código de estado HTTP para indicar si la operación de eliminación fue exitosa.

TRACE: Esta petición se utiliza para realizar un seguimiento de la ruta que sigue una solicitud HTTP desde el cliente hasta el servidor y viceversa. El servidor devuelve la misma solicitud con una serie de encabezados adicionales que indican las diferentes etapas que ha pasado la solicitud. Un ejemplo, se puede ejecutar el siguiente comando desde la consola de Windows:

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

Métodos especiales:

Algunos métodos sólo 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.