

Ejercicio 1: HTTP

Este es un ejemplo de una solicitud (request) HTTP.

```
GET /collaborators HTTP/1.1
Host: wareware.bunsan.io
User Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en)
AppleWebKit/124
(KHTML, like Gecko) Safari/125
Accept: ext/xml, application/xml, application/xhtml+xml,
text/html;q=0.9,
text/plain;q=0.8, image/png,*;q=0.5
Accept-Language: it
Keep-Alive: 300
Connection: keep-alive
```

Responde a las siguientes preguntas, indicando dónde (por ejemplo, en qué campo) en la solicitud HTTP puedes encontrar la respuesta:

1) ¿Cuál es la URL solicitada?

La URL solicitada es la que está marcada en el siguiente campo

```
GET /collaborators HTTP/1.1
```

2) ¿Qué versión de HTTP se utiliza?

La versión HTTP es la siguiente:

```
GET /collaborators HTTP/1.1
```

3) ¿El navegador solicita una conexión persistente o no persistente?

A través del siguiente comando se indica que el cliente desea usar una conexión persistente

```
Connection: keep-alive
```

4) ¿Cuál es, en tu opinión, la utilidad de indicar el tipo (y versión) de navegador utilizado por el cliente en la solicitud HTTP?

- Creo que le ayuda al servidor HTTP a identificar desde qué dispositivo se realiza la solicitud, lo que permite que por ejemplo si se quiere acceder a una página web, que esta pueda ajustar su código según las condiciones que se requieran. Es decir esto tiene importancia en cuanto al posicionamiento ya que esto depende que los usuarios puedan navegar bien por la página.

Ejercicio 2: HTTP

Un cliente HTTP envía el siguiente mensaje:

```
GET /assignments HTTP/1.1
Host: wareware.bunsan.io
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
If-modified-since: 27 Feb 2017 08:10:00
```

a) Escribe dos respuestas posibles del servidor HTTP (solo la línea de estado)

- HTTP/1.1 200 OK
- HTTP/1.1 404 Not Found

b) Suponiendo que el mensaje se envía a través de un Proxy, especifica el comportamiento del propio Proxy.

- Entiendo que el proxy reenviara la solicitud al servidor HTTP y también devolverá la respuesta al cliente, es decir el proxy actuará como intermediario entre el cliente y el servidor.

Investigando un poco más sobre el trabajo del proxy, encontré que el proxy ayuda a examinar el tráfico web, esto para identificar contenido sospechoso como un virus o intrusiones, también protege de ataques al servidor HTTP y niega cookies de sitios web. Además de realizar otras funciones como: caching, filtrado, balanceo de carga de peticiones, autenticación y registro de eventos