

- 1) ¿Sobre que protocolo binario está montado el protocolo HTTP?**  
Está sostenido por encima de una conexión TCP entre un host cliente y un host servidor (mensaje de requerimiento/mensaje de respuesta).
- 2) ¿Cuáles son los clientes http y los servidores http más utilizados?**  
Los navegadores Chrome, Mozilla (clientes) y apache, nginx, Microsoft iis (servidores)
- 3) ¿Qué verbos admite un comando en el requerimiento HTTP?**  
GET (quiero), POST, CONNECT, DELETE, PUT, siendo los más comunes los métodos GET y POST.
- 4) ¿Qué contenido lleva el body de un requerimiento HTTP?**  
Puede contener datos en cualquier formato, pueden ir datos ingresados por el usuario en un formulario y archivos que el cliente quiera enviar al servidor.
- 5) ¿Qué diferencia existe entre un URL y un URI?**  
Una URL es un subconjunto de URI mientras que éste no una URL. La primera permite localizar recursos y se conforma principalmente por sí mismo, la segunda permite localizar e identificar recursos (puede cumplir ambas funciones) y se conforma por la URL junto al URN.  
**Ejemplo de sintaxis:**  
URL: http://miempresa.midominio  
URI: /mirecurso.html
- 6) ¿En qué variables quedan almacenados datos de las respuestas http?**  
En los headers (indican, entre otras cosas, el tipo y el tamaño de los archivos y datos devueltos).
- 7) ¿Qué significa Virtual hosting?**  
Significa que un mismo servidor web pueda manejar múltiples nombres de host distintos sobre esa misma IP que está funcionando.
- 8) ¿Por qué el HTTP se considera STATE LESS?**  
Porque no puede retener en memoria la identidad de cada cliente, el servidor no puede identificar si un requerimiento es de un mismo cliente o de uno diferente, es por eso que el servidor trata cada requerimiento como uno nuevo.

**9) ¿Cómo va a mejorar la velocidad de la WEB en poco tiempo?**

Con la estandarización de un protocolo donde se busca un protocolo mejor que el HTTP/1 que genere múltiples requerimientos multiplexados en el tiempo de forma simultánea. En HTTP/2 el navegador pediría al servidor todo junto y en el envío del servidor la entrega funcionaría de forma multiplexada.

Esto más la compresión de headers y la utilización de un formato binario mejoraría la velocidad.

**10) ¿Cómo va a mejorar el inicio de las aplicaciones WEB su comportamiento en ambientes ruidosos o de debilidad de conexión?**

Con la implementación de un nuevo protocolo llamado QUIC que es como el TCP en cuanto dirección y correcciones de errores y como el UDP en cuanto a la velocidad. Este nuevo protocolo si llega a encontrar un error solamente ese error va a afectar a la conexión de donde se estableció, con lo cual no se tiene que volver a repetir todo el envío frente a los errores, el error quedara acotado a la subconexión donde se produjo.