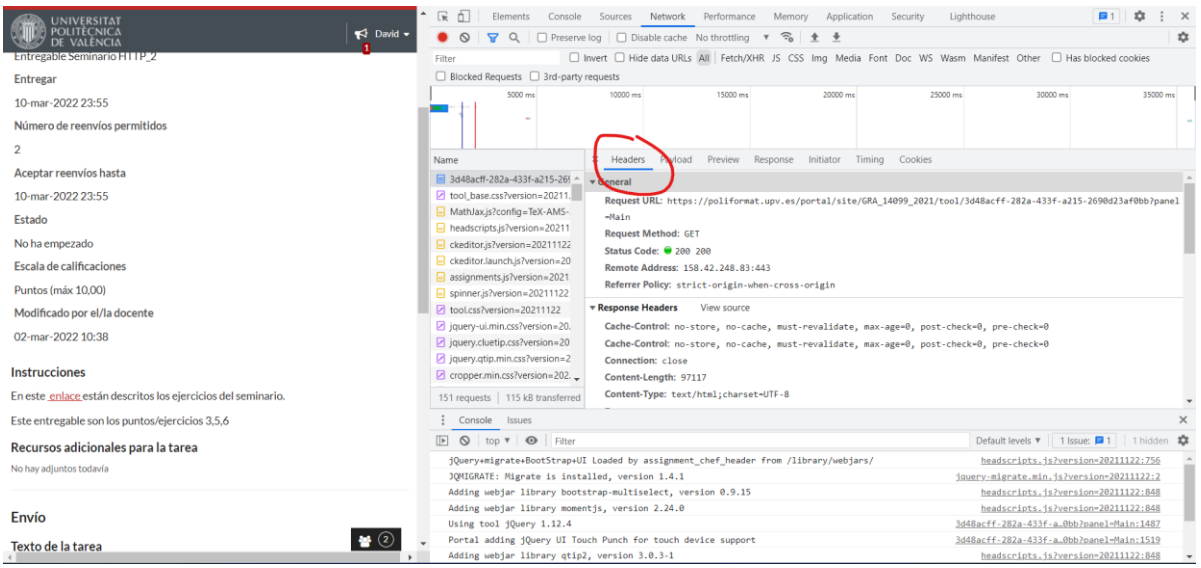
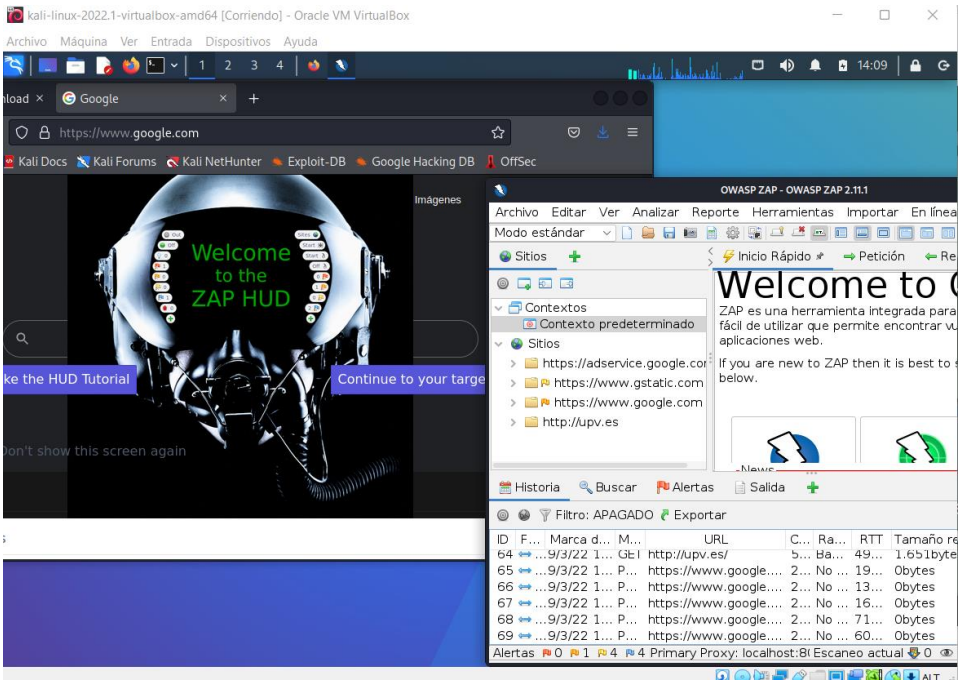
El documento describe varios ejercicios relacionados con tecnologías y herramientas TCP/HTTP:

1. Herramientas de desarrollador:

    1. Herramientas de desarrollador del navegador Firefox.https://es/docs/Tools
       Muestra en un pantallazo cómo se pueden ver las cabeceras HTTP en el navegador



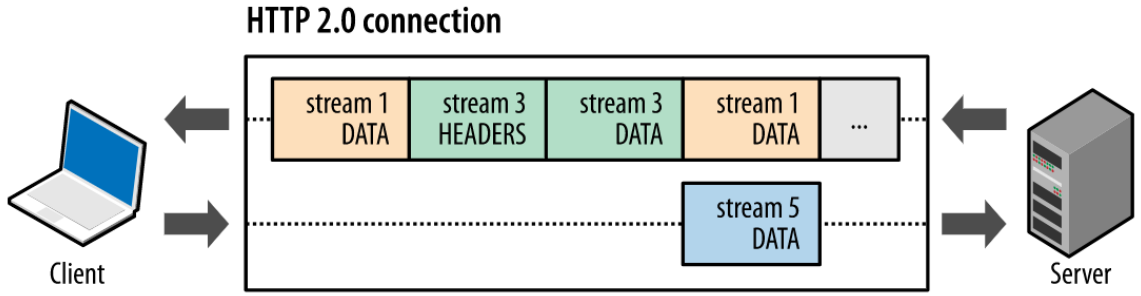
    2. Instalar badstore (prueba buscando badstore en docker hub)
    3. Instalación proxy ZAP
       ¿Es posible poner el proxy en medio de una comunicación HTTPS? Sí. En caso afirmativo ¿Cómo debería hacerse?



2. Explica quién, cómo y dónde se inicia una conexión http/2.0. https://rfc7540

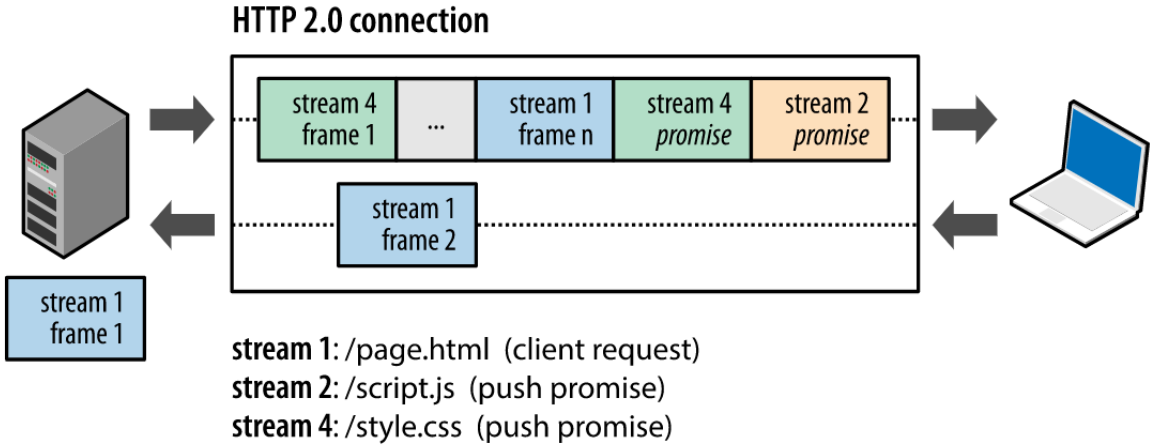
La principal diferencia con *HTTP/1.1* es que ahora se habilita una multiplexación total de solicitudes y respuestas, al permitir que el cliente y el servidor desglosen un mensaje de *HTTP* en tramas diferentes, intercalarlas y luego reensamblarlas en el otro extremo. Es decir, el cliente inicia todas las peticiones que desee en la tubería, que se distinguen gracias a su *stream ID* y, una vez que el servidor las procese, las devuelve etiquetadas con su *stream ID* correspondiente.

Un ejemplo visual sería:

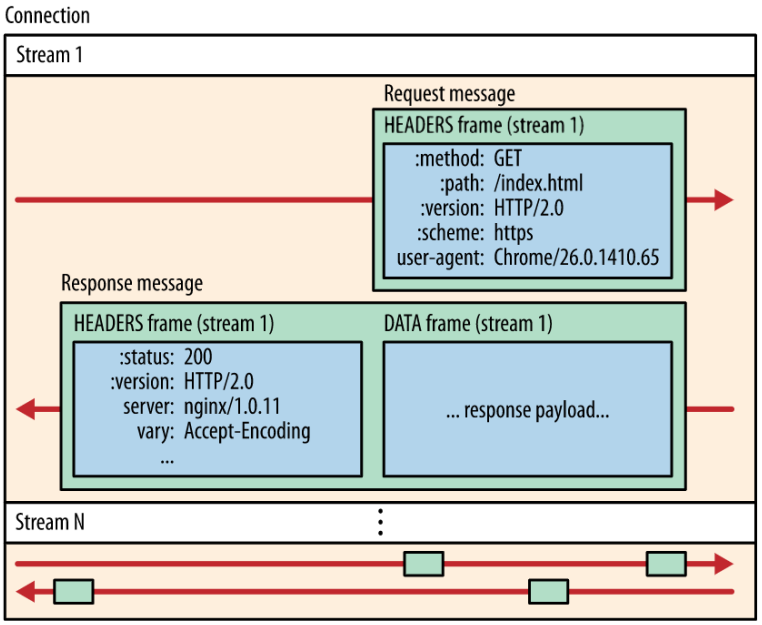


La imagen captura múltiples transmisiones en tránsito dentro de la misma conexión. El cliente transmite un marco *DATA* (*stream* 5) al servidor, mientras este transmite una secuencia intercalada de marcos al cliente para las transmisiones 1 y 3. En consecuencia, hay tres transmisiones paralelas en tránsito.

Otra nueva función interesante de *HTTP/2* es la capacidad del servidor de enviar respuestas múltiples para una única solicitud del cliente. Es decir, además de la respuesta a la solicitud original, el servidor puede insertar recursos adicionales en el cliente (ver la siguiente figura), sin necesidad de que este los solicite de manera explícita.



Podríamos resumir la conexión de la siguiente manera:



En resumen, *HTTP/2* desglosa la comunicación del protocolo *HTTP* en un intercambio de tramas con codificación binaria, que luego se asignan a los mensajes que pertenecen a una transmisión específica, todo está multiplexado dentro de una única conexión de *TCP*.

Un ejemplo de conexión sería:

```
GET / HTTP/1.1
Host: server.example.com
Connection: Upgrade, HTTP2-Settings
Upgrade: h2c
HTTP2-Settings: <base64url encoding of HTTP/2 SETTINGS payload>
```

Que lo enviaría el cliente y el servidor podría no aceptarla por no soportar HTTP/2 respondiendo ignorando la cabecera Upgrade, de la siguiente manera:

```
HTTP/1.1 200 OK
Content-Length: 243
Content-Type: text/html
```

En el caso de aceptar la conexión, aceptaría la actualización con una respuesta 101, por ejemplo:

```
HTTP/1.1 101 Switching Protocols
Connection: Upgrade
Upgrade: h2c
```

3. Explica quién, cómo y dónde se inicia una conexión http/3.0. https://draft-ietf-quic-http-34

HTTP/3 es la tercera adaptación del protocolo *HTTP*. A diferencia de la primera y segunda versión de *HTTP*, que se basan en *TCP*, *HTTP/3* usa *QUIC* (*Quick UDP Internet Connections*), un nuevo estándar de código abierto desarrollado inicialmente por Google. *QUIC* es muy similar a *HTTP/2*, pero implementado sobre *UDP*.

Igual que en *HTTP/2*, el cliente es quien inicia la conexión de la siguiente forma, como indican desde *https://draft-ietf-quic-http-34*:

A client MAY attempt access to a resource with an "https" URI by resolving the host identifier to an IP address, establishing a QUIC connection to that address on the indicated port (including validation of the server certificate as described above), and sending an HTTP/3 request message targeting the URI to the server over that secured connection. Unless some other mechanism is used to select HTTP/3, the token "h3" is used in the Application Layer Protocol Negotiation (ALPN; see [RFC7301]) extension during the TLS handshake.

Además, algunas de las mejoras clave de *QUIC* respecto a *HTTP/2* incluyen:

- Latencia de establecimiento de la conexión
- Mejora del control de la congestión
- Multiplexación sin bloqueo de cabecera
- Corrección de errores hacia adelante
- Migración de la conexión