

Iván Turro Arroyo

Lucas Chacon Langa

David Barroso Martínez

Enunciado

Objetivo: entender el funcionamiento de un servidor web visto desde fuera: qué tipo de peticiones le llegan y qué información da en su respuesta.

Introducción: antes de meternos a ver la ejecución de código en un servidor web, necesitamos entender qué hace exactamente un servidor web, estudiar el ciclo de vida de las peticiones HTTP.

Desarrollo: en grupos de dos o tres alumnos, se trata de observar el flujo de información entre un cliente web (navegador) y un servidor para identificar la siguiente información:

- Protocolo (http) y puerto (80 conexión normal, 443 conexión segura https) y tipo de petición (de momento solo GET)
- Nombre de máquina (host) del servidor, aunque el cliente resolverá dicho nombre para poder ir a la IP concreta, también mandará esa información en las cabeceras (Headers) pues HTTP permite que una misma IP responda a varios sitios web distintos por nombre.
- Identificación del producto cliente.
- Identificación del producto de servidor.
- Tipos de respuestas que aceptará el cliente.
- Tipo de datos de la respuesta y codificación.
- Fecha y si hay alguna indicación para los proxy-cachés sobre validez temporal de la respuesta (cuándo caduca)
- Cualquier otro dato que te resulte interesante comentar al inspeccionar las peticiones.

Elige una web de alguna afición tuya para realizar la conexión y documentar cómo obtienes los datos anteriores y los resultados.

Herramientas:

- En el navegador Firefox con botón derecho del ratón en inspeccionar y pestaña Red, verás las conexiones que está realizando. No te sorprenda que en algunas páginas no pare de hacer peticiones, pues las respuestas llevan a su vez indicaciones para el navegador de volver a hacer peticiones refrescando contenidos, etc.
- Los clientes de línea de comandos curl o wget permiten bajar desde la terminal páginas web y mostrar las cabeceras recibidas. Pide al profesor una demostración, la opción de curl es -D - , la de wget -S
- Mediante clientes puros de red, que ni siquiera interpretan el protocolo, como nc o telnet, podemos ver el flujo completo en texto plano. Primero tendrás que obtener la dirección ip del servidor mediante ping o nslookup, ya que aquí no lo hace el navegador (no lo usamos) Atiende a la demostración del profesor para documentar cómo ver la parte de petición del cliente y de la respuesta del servidor mediante los siguientes comandos:

nc -C4t dirección-ip puerto, ej:

nc -C4t 185.199.109.153 80

se quedará esperando y tendrás que teclear la petición HTTP, por ejemplo:

GET / HTTP/1.1

Host: OscarMaestre.github.io

si la web es https (segura) podemos usar openssl, ej:

openssl s_client -connect server:443 (luego hacer la petición con el GET ..., CTRL-D para terminar)

- Para ver las cabeceras que manda el cliente (navegador) podemos simular un servidor fake y dirigir el navegador hacia el. Por ejemplo:

nc -l 4000 levantará un servidor en el puerto 4000, pero no http, solo escucha en un puerto

Poniendo en el navegador <http://localhost:4000/> hará la petición a ese servidor fake y podremos ver en la consola qué datos le ha mandado, el texto claro de la petición completa. Como no obtiene respuesta, el navegador no mostrará nada.

- La herramienta Postman (descargar y ejecutar como portable) la usaremos más adelante para depuración de API RESTs, ahora mismo no aportaría mucho sobre lo anterior, pero si quieres probarla vas avanzando materia.
- La herramienta Wireshark se utiliza sobre todo en seguridad para interceptar el tráfico de red, es demasiado para el punto en el que estamos, pero igualmente es una opción para ir

a por excelente. Te permitirá ver la diferencia entre las conexiones sin cifrar (http) y las cifradas (https o http sobre SSL o TLS) donde apenas verás información. Aquí verás claramente que las conexiones van a direcciones IP concretas tras la resolución DNS. Tendrás que pedir ayuda al profesor para configurar los filtros de Wireshark.

Entrega: un documento en formato abierto (PDF, ODF) con nombre Reto0-UD1 que incluya las pruebas, sus resultados (con capturas demostrativas comentadas) y conclusiones. En este último apartado os sugiero volver a leer el enunciado y reflexionar sobre si se entiende mejor al terminar el reto respecto a la primera lectura.

Evaluación:

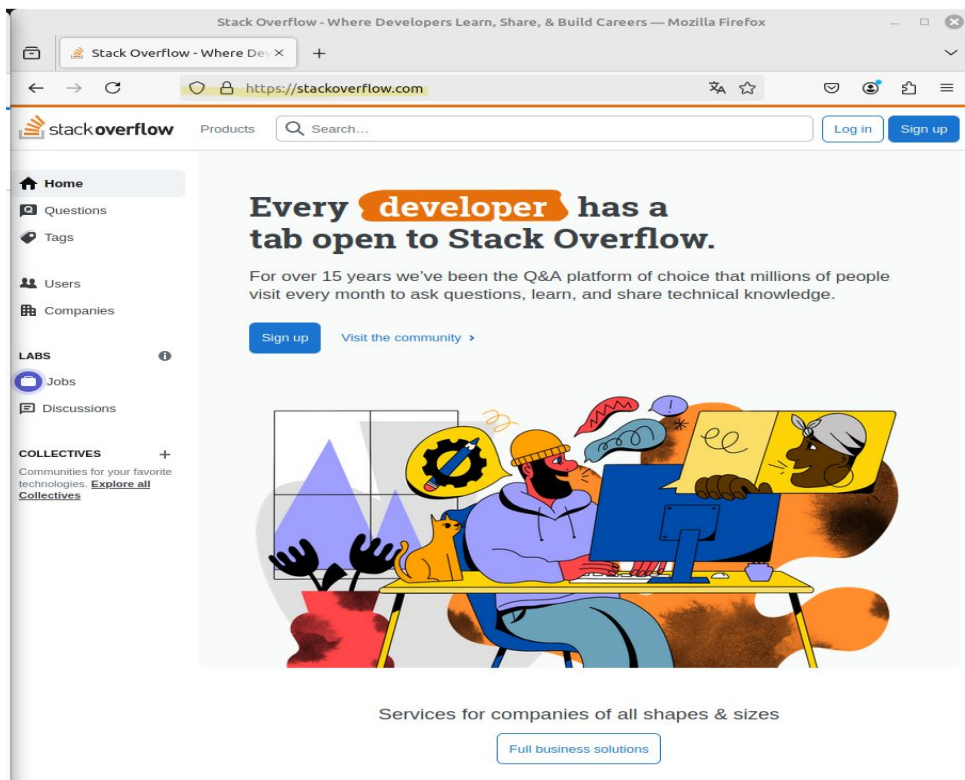
- Documentación de los pasos realizados, claridad y exhaustividad.

Calificación:

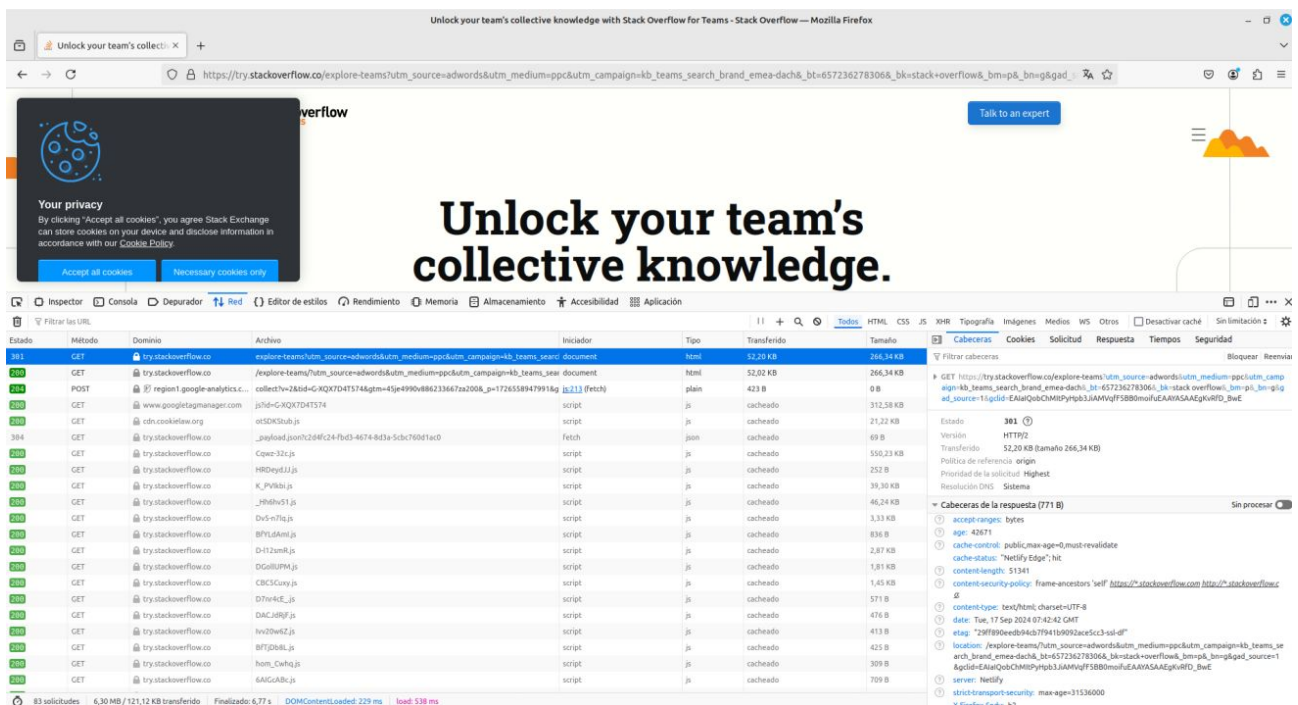
- Suficiente: documentación de los datos obtenidos usando dos herramientas distintas.
- Bien: lo anterior y descripción de las partes de una dirección web o URL (lo tienes en los materiales)
- Excelente: lo anterior y una tercera herramienta. Comparativa de tus resultados y los de algún otro compañero.

Solución

1.- El protocolo utilizado en la consulta se puede observar en la barra de URL del cliente (Firefox) al tratarse del HTTPS se utiliza el puerto 443

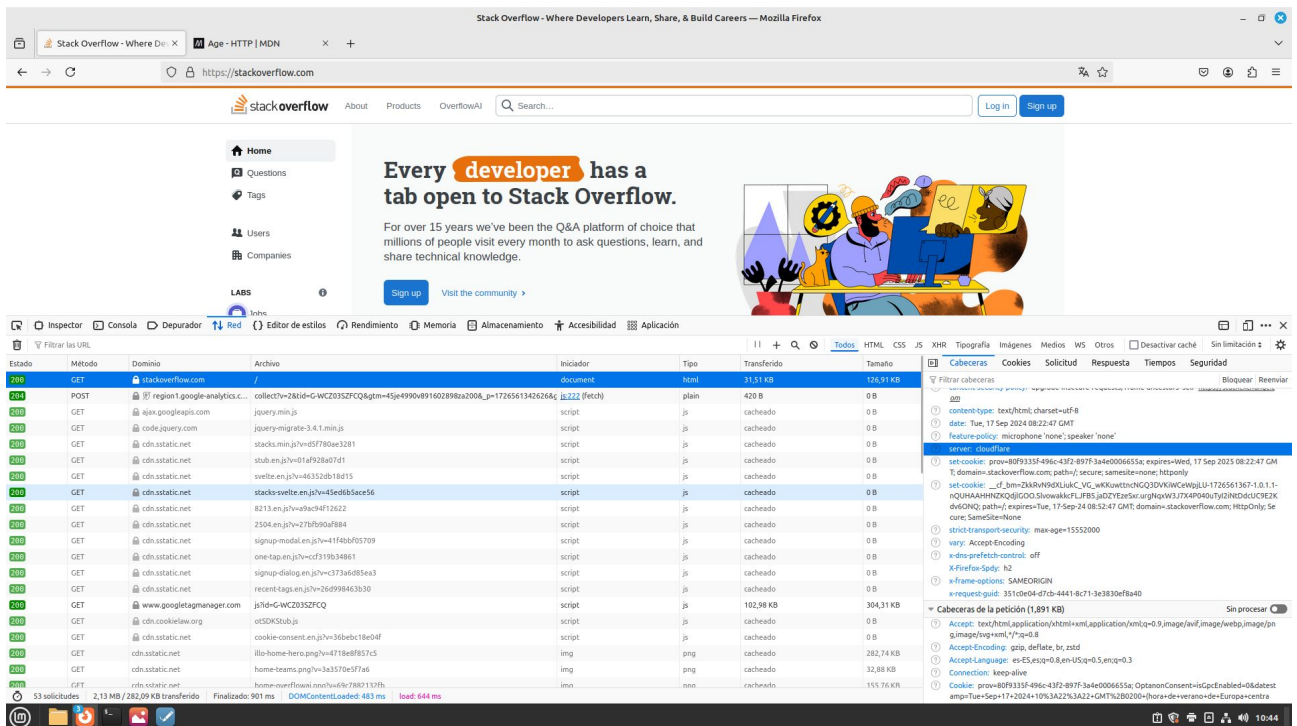


Para encontrar el tipo de petición inspeccionamos la pagina en cuestión haciendo click derecho en la pagina y en las opciones que aparecen seleccionamos Inspeccionar, seguidamente aparece el siguiente desplegable

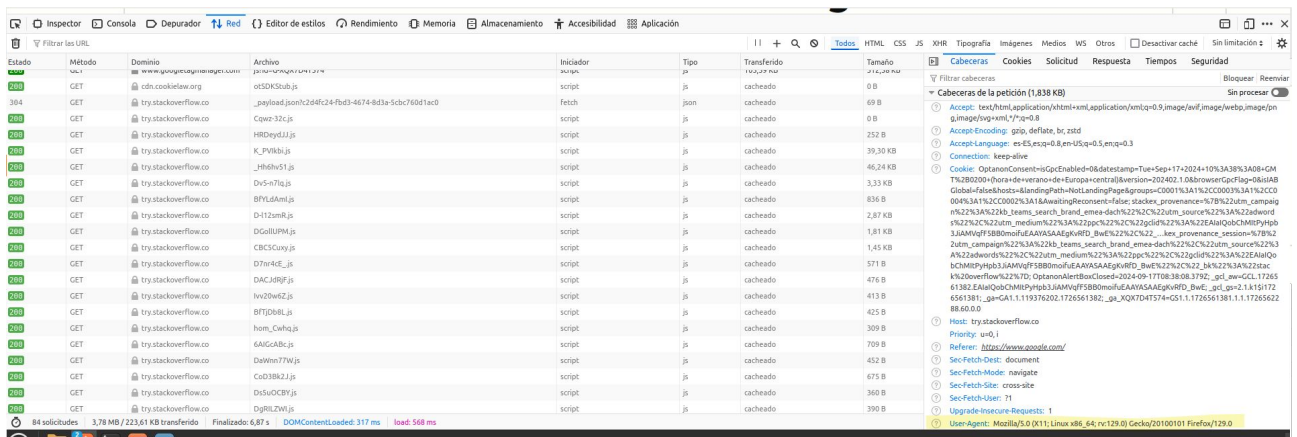


en la sección de RED aparece la tabla con todas las peticiones y la primera es la que ha realizado inicialmente el CLIENTE la cual esta resaltada en azul y es el método GET

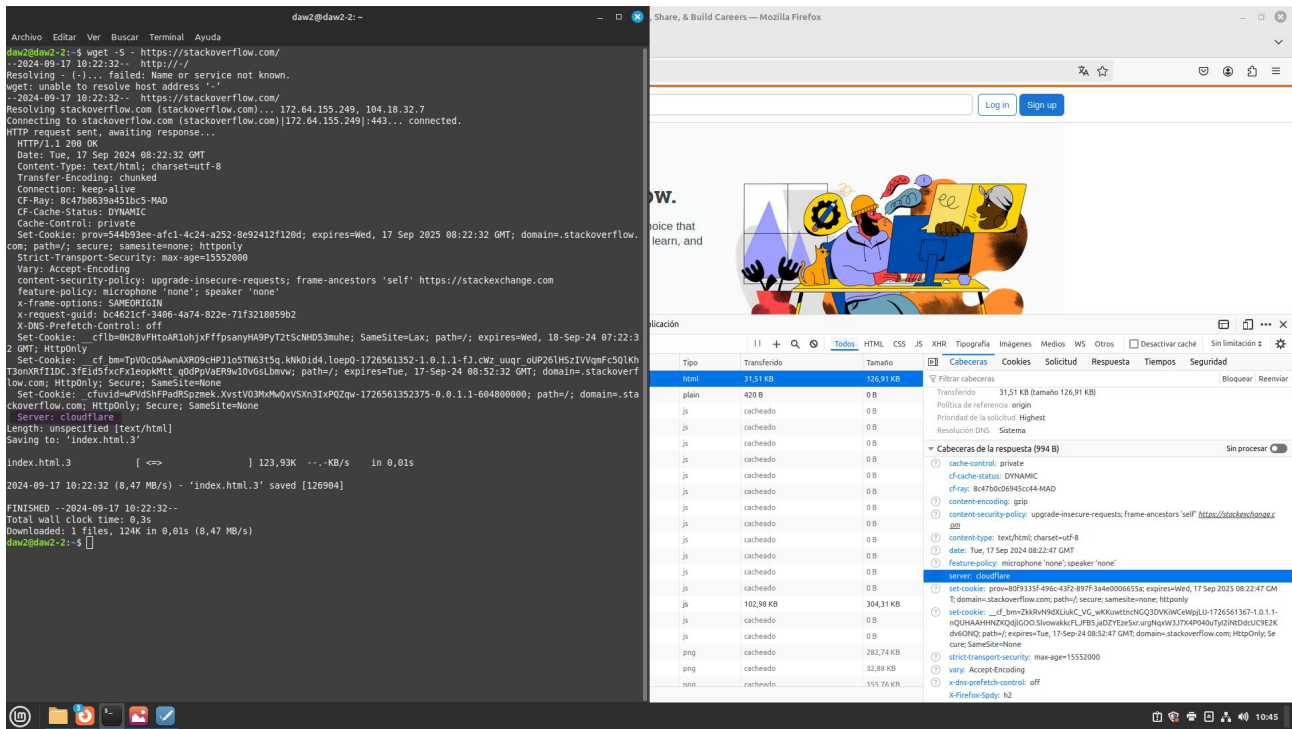
2.-Nombre Host es el nombre de la pagina que hemos usado pero sin el “<https://www.>”



3.- el identificador del producto cliente entendemos que lo podemos encontrar en “User-Agent” referenciando a el cliente (firefox) con el sistema operativo y sus versiones y la localización del navegador.



4.- El identificador del producto de servidor lo podemos encontrar en la etiqueta “server”



5.- Los tipos de respuesta que acepta el cliente en esta pagina son:

- 200: significa que indica que la solicitud ha tenido éxito.

- 204: La respuesta ha sido satisfactoria pero el cliente no necesita navegar fuera de la pagina actual, la respuesta es cacheable por defecto.

-301: Moved permanently indica que el recurso solicitado ha sido movido **de forma permanente** a la URL dada por las cabeceras "location". Un navegador redirecciona a esta página, pero los motores de búsqueda no actualizan sus enlaces al recurso.

- 302: indica que el recurso solicitado ha sido movido **temporalmente** a la URL dada por las cabeceras "location". Un navegador redirecciona a esta página, pero los motores de búsqueda no actualizan sus enlaces al recurso.

-304: Not Modified la respuesta de la petición indica que el recurso solicitado no es necesario que sea retransmitido

Unlock your team's collective knowledge with Stack Overflow - Stack Overflow - Mozilla Firefox

https://try.stackoverflow.com/explore-teams?utm_source=adwords&utm_medium=ppc&utm_campaign=kb_teams_search_brand_emea-dach&_bt=657236278306&_bk=stack+overflow&_bm=p&_bn=g&gad=...

Unlock your team's collective knowledge.

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies Necessary cookies only

Talk to an expert

Inspector Console Depurador Red Editor de estilos Rendimiento Memoria Almacenamiento Accesibilidad Aplicación

Filtrar las URL

Estado	Método	Domino	Archivo	Iniciador	Tipo	Transferido	Tamaño
301	GET	try.stackoverflow.co	explore-teams?utm_source=adwords&utm_medium=ppc&utm_campaign=kb_teams_search_brand_emea-dach&_bt=657236278306&_bk=stack+overflow&_bm=p&_bn=g&gad=...	document	html	52,20 KB	266,34 KB
304	GET	try.stackoverflow.co	explore-teams?utm_source=adwords&utm_medium=ppc&utm_campaign=kb_teams_search_brand_emea-dach&_bt=657236278306&_bk=stack+overflow&_bm=p&_bn=g&gad=...	document	html	52,20 KB	266,34 KB
200	POST	region1.google-analytics.c...	collect?v=2&tid=G-XQKTD4754&utm_source=adwords&utm_medium=ppc&utm_campaign=kb_teams_search_brand_emea-dach&_bt=657236278306&_bk=stack+overflow&_bm=p&_bn=g&gad=...	fetch()	plain	423 B	0 B
200	GET	www.googletagmanager.com	js?id=G-XQKTD4754	script	js	cached	0 B
200	GET	cdn.cookiecutter.org	o5D0K3n.js	script	js	cached	0 B
304	GET	try.stackoverflow.co	_payload.json?c2d4f2c4-fbd3-4674-8d3a-5cb760d1ac00	fetch	json	cached	69 B
200	GET	try.stackoverflow.co	Cqew-32c.js	script	js	cached	0 B
200	GET	try.stackoverflow.co	HRDeYdJJ.js	script	js	cached	252 B
200	GET	try.stackoverflow.co	K_Pviki.js	script	js	cached	39,30 KB
200	GET	try.stackoverflow.co	_Hh6hv51.js	script	js	cached	46,24 KB
200	GET	try.stackoverflow.co	Dv5-n7q.js	script	js	cached	3,33 KB
200	GET	try.stackoverflow.co	BYUdAm1.js	script	js	cached	836 B
200	GET	try.stackoverflow.co	D-112mR.js	script	js	cached	2,87 KB
200	GET	try.stackoverflow.co	DGolUPM.js	script	js	cached	1,81 KB
200	GET	try.stackoverflow.co	CBC5Cuy.js	script	js	cached	1,45 KB
200	GET	try.stackoverflow.co	D7w4E...js	script	js	cached	571 B
200	GET	try.stackoverflow.co	DAC4JRf.js	script	js	cached	476 B
200	GET	try.stackoverflow.co	lvv2w6Z.js	script	js	cached	413 B
200	GET	try.stackoverflow.co	BTfjDBL.js	script	js	cached	425 B
200	GET	try.stackoverflow.co	hom_Cwhq.js	script	js	cached	309 B
200	GET	try.stackoverflow.co	NAICABc.js	script	js	cached	709 B

83 solicitudes 4,85 MB / 128,51 KB transferido Finalizado: 7,09 s DOMContentLoaded: 434 ms load: 649 ms

Capcaberas Cookies Solicitud Respuesta Caché Tiempos Traza de la pila

Filtrar cabeceras

- age: 55863
- cache-control: max-age=86400
- cf-cache-status: HIT
- cf-ray: 804773b168b92798-MAD
- content-encoding: gzip
- content-length: 12808
- content-md5: JaQNgVc3a736cMdoYk3BQ==
- content-type: application/json
- date: Tue, 17 Sep 2024 07:42:28 GMT
- etag: 0w6DCA5DEB83DC8A4
- last-modified: Tue, 16 Jul 2024 21:31:57 GMT
- server: Cloudflare
- strict-transport-security: max-age=31536000; includeSubDomains; preload
- vary: Accept-Encoding
- x-content-type-options: nosniff
- X-Firefox-Spdy: h2
- x-ms-blob-type: BlobBlob
- x-ms-dep-status: unlocked
- x-ms-request-id: 46f04ee-201e-0076-80cb-e91736000000
- x-ms-version: 2009-09-19

▼ Cabeceras de la petición (461 B)

- Accept: */*
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: es-ES,en;q=0.8,en-US;q=0.5,en;q=0.3
- Connection: keep-alive
- Host: cdn.cookiecutter.org

Sin procesar

6.- Los tipos de datos de la respuesta y codificación lo podemos encontrar en dos lugares diferentes, primero podemos filtrar en el buscador de la sección de “RED” o la segunda opción es clickando en cada petición y podemos observarlo en la cabecera de respuesta “Content-type”.

7.

The screenshot displays a terminal window on the left and a web browser on the right. The terminal shows the command `dsw2@dsw2-2:~$ wget -S - https://stackoverflow.com/` and its output, including the HTTP status code 200 and various headers like `Date: Tue, 17 Sep 2024 08:22:32 GMT` and `Content-Type: text/html; charset=utf-8`. The browser shows the Stack Overflow homepage with the headline "Every developer has a tab open to Stack Overflow." and a list of cookies in the developer tools, including `accept-ranges`, `cache-control`, `cf-cache-status`, `cf-ray`, `content-encoding`, `content-length`, `content-type`, `date`, `expires`, `last-modified`, `server`, `set-cookie`, `vary`, and `x-dns-prefetch-control`.

Curiosidades:

- Cada vez que se actualiza la pagina, el numero de solicitudes cambia y/o aparecen nuevas solicitudes con identificadores de petición diferentes.