MEMORIA SC – PRÁCTICA II

Alonso Rodríguez Iglesias – SC 2019/20 – Práctica 2

ÍNDICE

ÍNDICE DE FIGURAS

INTRODUCCIÓN Y DETALLES

Esta práctica trata de realizar varias peticiones a diferentes sitios para familiarizarse con el intercambio de mensajes HTTP, sus códigos, dinámicas de carga y recarga, y la "seguridad" para realizar autenticación (inexistente en nuestro caso puesto que se realiza sobre texto plano).

La URL contra la que he realizado las prácticas es inefg.udc.es.

Para no insertar una captura de pantalla por respuesta, puesto que quedaría un trabajo enorme lleno de capturas, he realizado varias capturas en las que se puede ver toda la información necesaria. En ellas he marcado regiones con cuadrados de diferentes colores, y en las respuestas hago referencia a esas regiones de colores en las figuras.

Como leer el documento se hacía pesado en texto "plano", he decidido subrayar con un color, cada referencia al mismo, por ejemplo: verde, y a mi parecer agiliza mucho la lectura.

Las figuras se encuentran al final de cada sección, y la forma intencionada de leer este documento de forma ágil es abrir el PDF en dos ventanas separadas, una con el texto y otra con las imágenes. De esta forma la lectura es bastante cómoda. También se puede acceder a las imágenes originales en la carpeta image.

SOLICITUDES GET BÁSICAS

1. Solicita la versión HTTP/1.1 [Recuadro Naranja Figura 1]
2. Local: `192.168.0.16` (IP Local a la interfaz `enp2s0f0` de mi portátil)
   Servidor: `192.144.61.98` (IP Pública del servidor web)
   [Recuadros Rojos Figura 1]
3. La primera respuesta del servidor es la respuesta a la primera (y única solicitud) del cliente. Es el paquete número 20. [Recuadro Rojo Figura 2]
4. Acepta Español de España o cualquier tipo de español con un valor de calidad de 0.8, Inglés de Estados Unidos con q=0.5, y cualquier otro tipo de Inglés con q=0.3. [Recuadro Verde Figura 1]
   q indica el "quality value" o valor de calidad, es decir, del 0 siendo el mínimo hasta el 1 siendo el máximo, denota la preferencia del usuario por los idiomas que acompaña [`https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.4`].
5. Se especifica nuestro navegador y sistema operativo en cada paquete saliente de nuestro ordenador. En mi caso es:

   ```
   User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0
   ```

   Y para comprobarlo podemos hacer scroll por todos los paquetes salientes de mi pc filtrándolos con `http and ip.src==192.168.0.16/24`.
   [Recuadro Morado Figura 1]
6. Si, en la primera respuesta del servidor trae el campo Set-Cookie [Recuadro Verde Figura 2], y a partir de entonces todos los GET se realizan con el campo Cookie puesto a ese valor, como podemos ver por ejemplo en el paquete número 583. [Recuadro Rojo Figura 3]. La cookie es siempre la misma, y sirve para identificar al usuario que está navegando la web.
7. Para medir la diferencia de tiempos medimos Δt entre ambos paquetes, [Recuadro Negro Figura 1] que nos da $\Delta t = \Delta t_{fin} - \Delta t_{ini} \approx 0.3038 - 0.0674 \Rightarrow \Delta t = 0.2364s = 236.4\ ms$.
8. A nivel de HTTP, la respuesta viene toda en una misma respuesta, a nivel de TCP, se indica que se han reensamblado 5 paquetes TCP, y Wireshark nos deja ver el texto que se transmitió en ellos. En nuestro caso los segmentos TCP que se han reensamblado son el numero 12, 14, 16, 18, 20. El campo no es ninguno en particular. Es el texto que viene a partir de la línea en blanco.
   [Segmentos Involucrados: Recuadro Verde, Texto HTTP: Recuadro Rojo, Fig. 4]
9. Si el servidor dice la verdad, el ejecuta nginx. [Recuadro Morado Figura 2]

10. Dependiendo del contenido al que accedamos, tendrá una fecha de modificación anterior o posterior. Por ejemplo, el CSS de la página parece que lleva varios años sin cambiar, mientras que el HTML no.

```
HTML Principal: Last-Modified: Sat, 25 Apr 2020 17:36:39 GMT
CSS           : Last-Modified: Thu, 06 Apr 2017 07:16:57 GMT
```

    Como podemos ver, esto se encuentra en el campo Last-Modified, se ubica en las respuestas del servidor, por ejemplo: [Recuadro Marrón Figura 2].
    El HTML Principal se modificó como se puede leer, el 25 de Abril de 2020, a 17:36:39 GMT, probablemente debido a la situación con el COVID-19, debido a la cual es importante mantener informado al profesorado, alumnos, y demás personas que visiten la página.
11. Tras analizar el HTML, éste contiene referencias a otros recursos, que en HTTP/1.1 no se transmiten todos juntos (server push de HTTP/2), sino que se tienen que ir solicitando uno por uno (al menos disponemos de keep-alive), por ejemplo CSS, JavaScript, las imágenes, etc.

SOLICITUDES GET BÁSICAS - FIGURAS

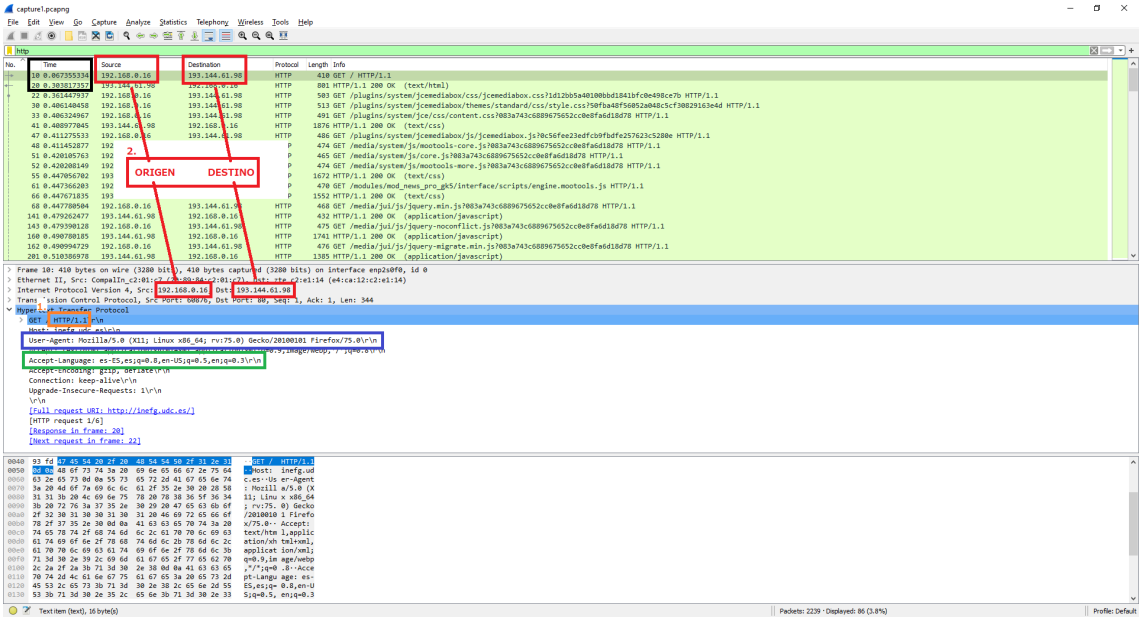

Figura 1: Análisis del primer paquete.

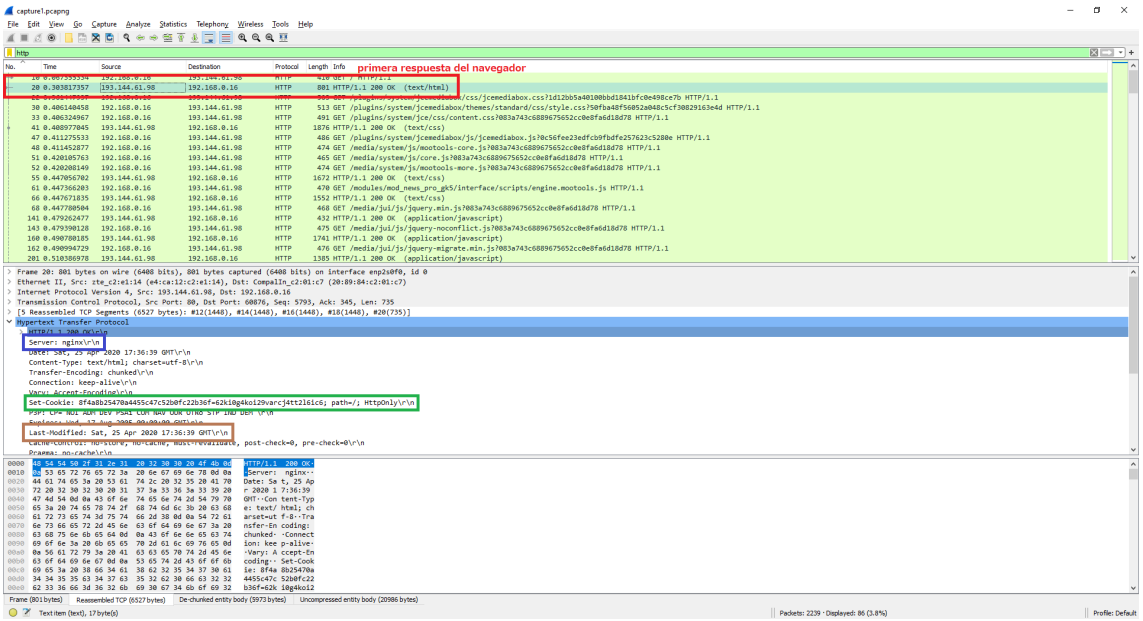

Figura 2: Análisis del segundo paquete.

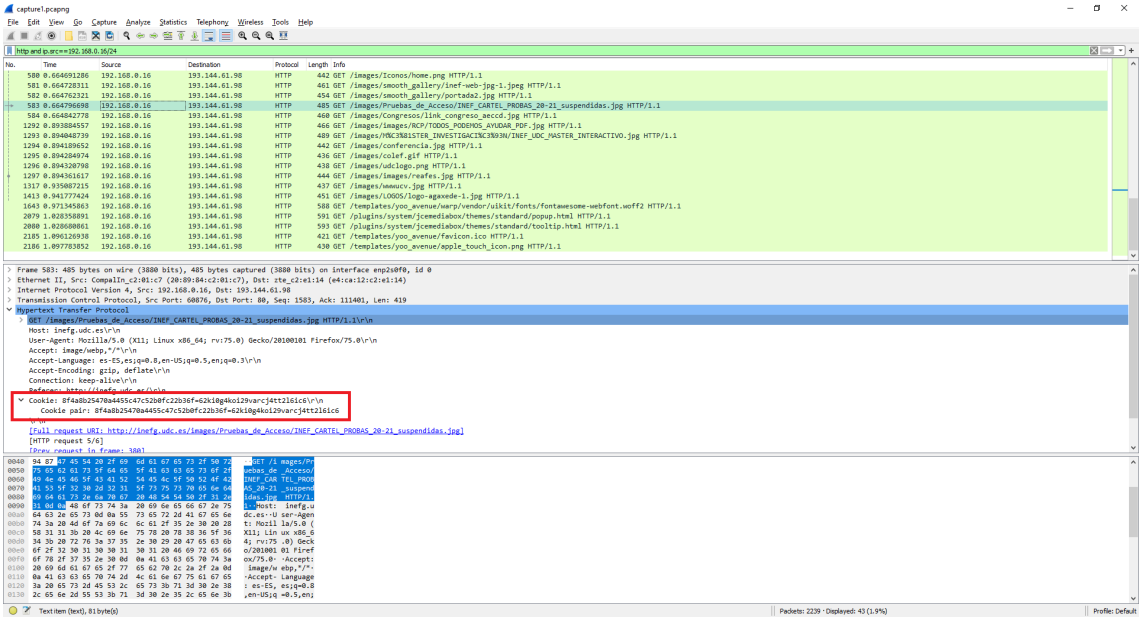

Figura 3: Cookie de un Paquete.

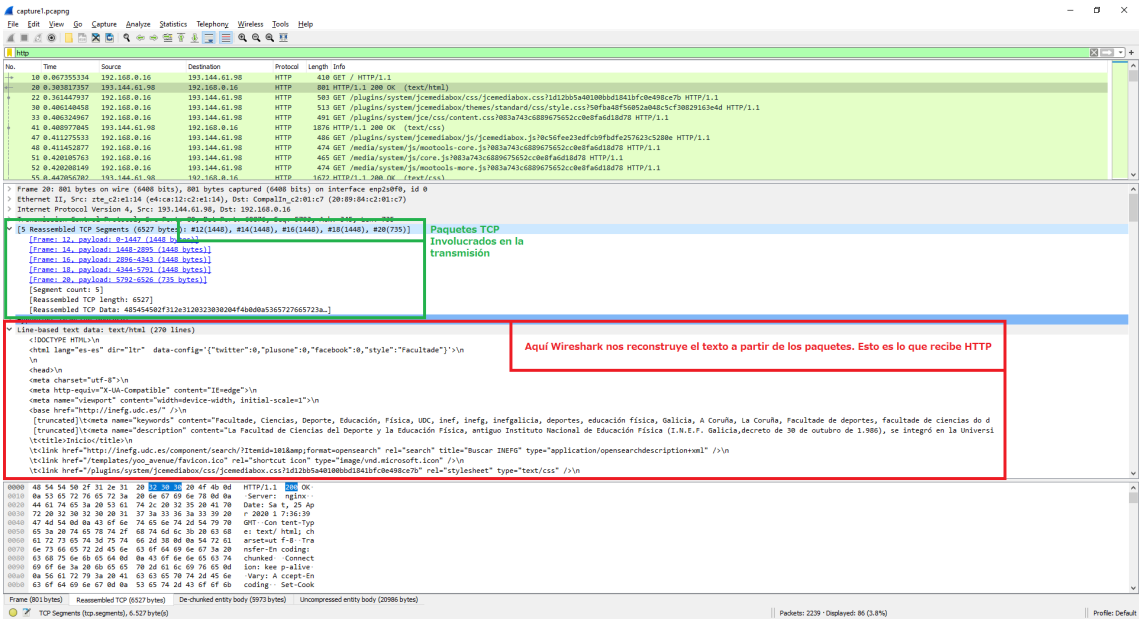

Figura 4: Paquetes TCP formantes de la transmisión HTTP y reconstrucción de Wireshark.

SOLICITUDES GET CONDICIONALES

1. Parece que Firefox al presionar F5, hace siempre una petición nueva al servidor, por lo que no incluye If-Modified-Since, ya que indicamos explícitamente que queremos recargar la página, pero para los recursos que enlaza si que incluye el campo en la cabecera. Por eso, si comparamos por ejemplo la primera request a `/images/colef.gif` veremos que no tiene la línea en la cabecera, [Figura 5] pero en la siguiente vez tras recargar, si. [Figura 6]
2. Como ya he comentado en la respuesta anterior, tras la recarga si que aparecen, porque el ordenador ya tiene cacheado el valor del campo Last-Modified para cada recurso, así que pone ese valor en el campo If-Modified-Since del GET, para ahorrar ancho de banda si no fuese necesaria una retransmisión.
   El campo es del formato:

   ```
   If-Modified-Since: Mon, 27 Jan 2020 17:47:10 GMT
   ```

3. Para los ítems que se piden sin el campo If-Modified-Since, y si hubiese, para los ítems que se piden y que no han sido actualizados desde el valor del mismo, se emite una respuesta con status code y response phrase:

   ```
   HTTP/1.1 200 OK
   ```

   Sin embargo, para los que envía un GET con una fecha en el campo If-Modified-Since mayor o igual a la que el servidor enviaría en Last-Modified (es decir a la última fecha de modificación del recurso), la respuesta comienza con:

   ```
   HTTP/1.1 304 Not Modified
   ```

SOLICITUDES GET CONDICIONALES – FIGURAS

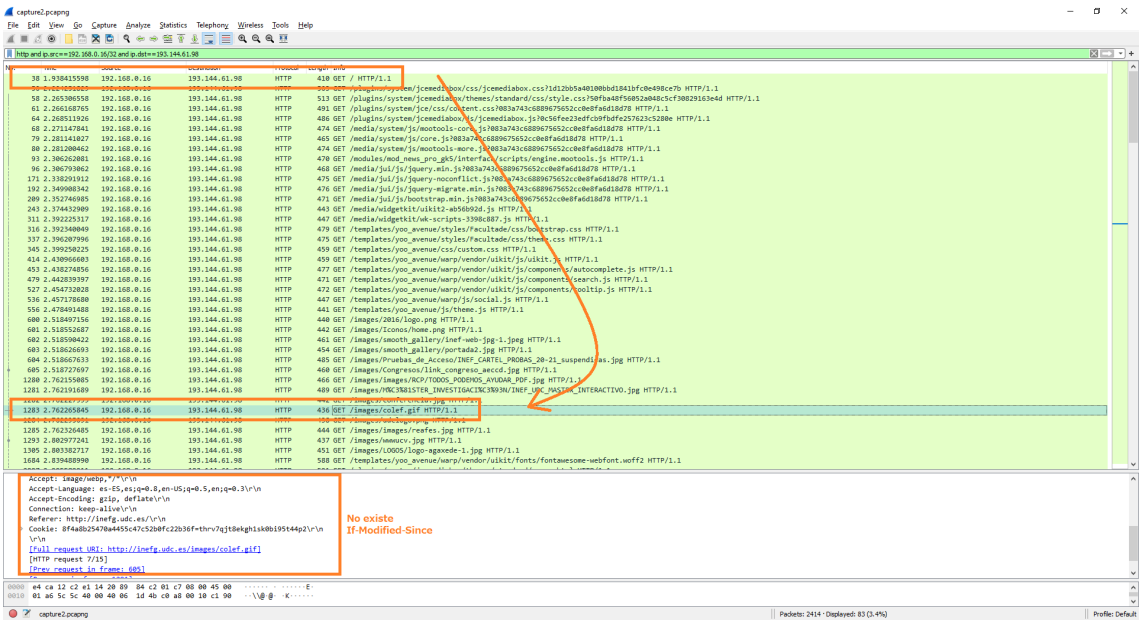

Figura 5: Secuencia de petición de carga de / y siguientes peticiones. Se muestra la petición a /images/colef.gif. Nótese la ausencia de If-Modified-Since.

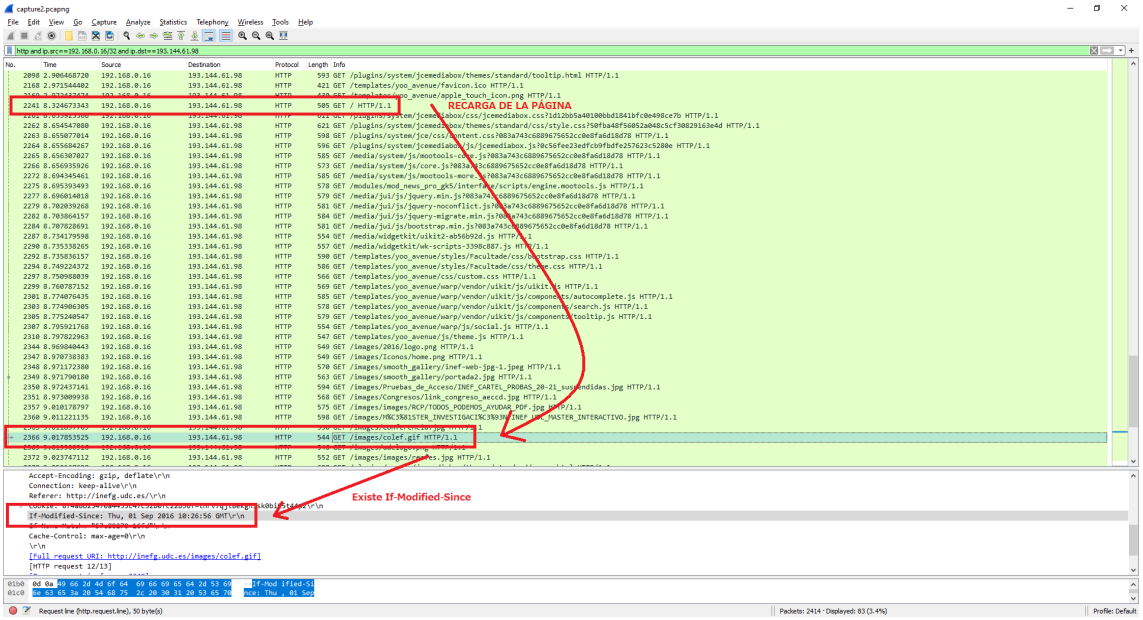

Figura 6: Secuencia de petición de recarga de / y siguientes peticiones. Se muestra la petición a /images/colef.gif. Nótese la presencia de If-Modified-Since.

AUTENTICACIÓN BÁSICA

1. Devuelve un 401: `HTTP/1.1 401 Unauthorized`. [Paquete 33 en Figura 7]
2. Como podemos ver en la figura 7, el flujo es:

   `GET -> 401 Unauthorized -> Get (+Authorization) -> 200 OK`

   En la figura se aprecia el flujo tal cual [Recuadro Naranja Figura 7], en ese orden. Las últimas dos líneas son la petición GET y la recepción del favicon.

   También podemos ver como la autenticación se envía codificada en Base64 en el campo Authorization, pero Wireshark ya nos la decodifica automáticamente si desplegamos el menú, [Recuadro Rojo Figura 7] con lo que podemos comprobar que efectivamente, muy seguro quizás no es, y deberíamos plantearnos utilizar un canal de comunicación seguro; cosa que nos puede proveer SSL o TLS.

AUTENTICACIÓN BÁSICA – FIGURAS

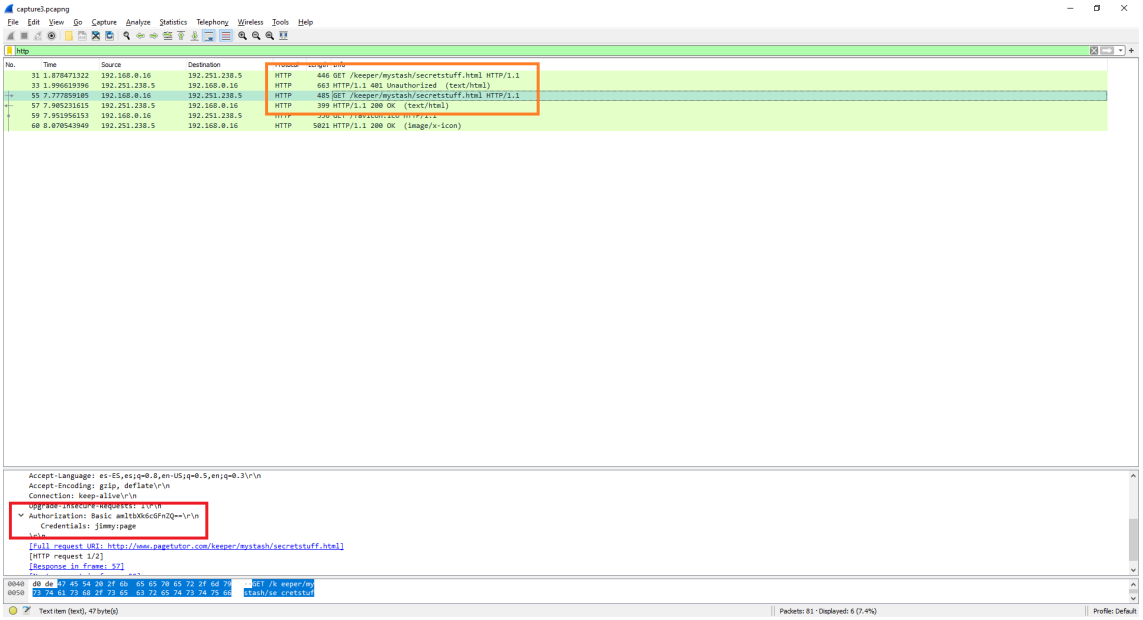

Figura 7: Secuencia de mensajes entre Cliente y 192.251.238.5. Primero no autorizado, se envía página correctamente tras enviar credenciales correctas.

CONCLUSIONES

Los resultados son los esperados, esta práctica la consideraría como un refresco de la teoría que vimos en Redes, con la parte práctica con Wireshark que realizamos en LSI, lo cual es a la vez una ventaja y un inconveniente: ventaja porque por mi parte la he realizado en una tarde, lo cual se agradece por la alta carga de trabajo que estamos teniendo debido a esta situación causada por el virus COVID-19. Desventaja porque no he aprendido nada nuevo haciéndola, pero siempre está muy bien refrescar conocimientos y adquirir un poco de fluidez con programas que supongo utilizaremos en subsecuentes prácticas.

Alonso Rodriguez Iglesias. 25-Abril-2020