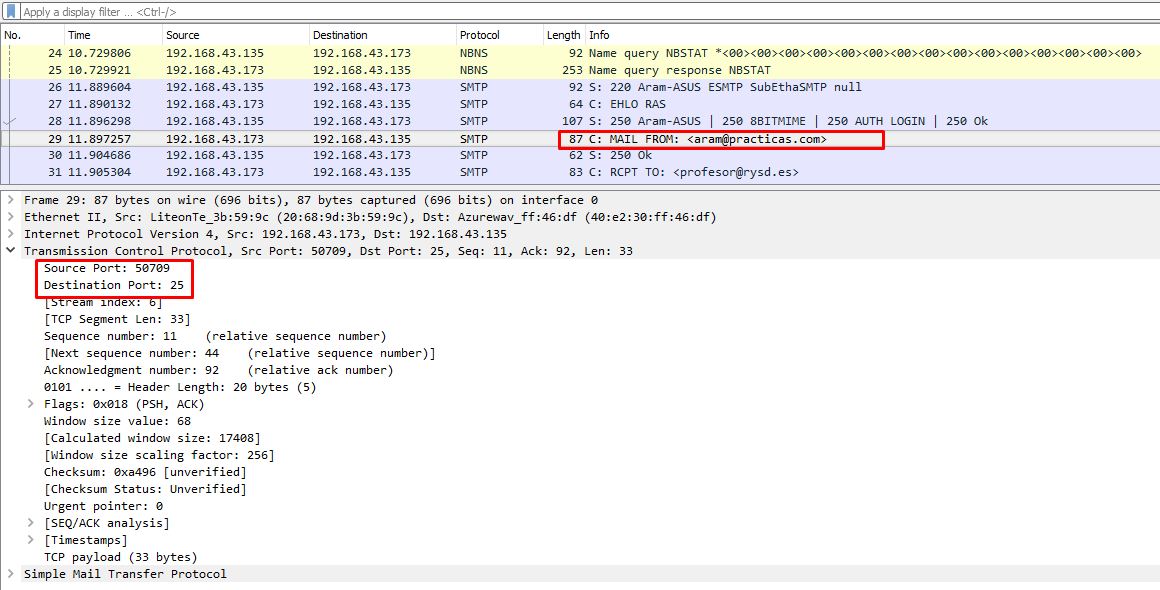
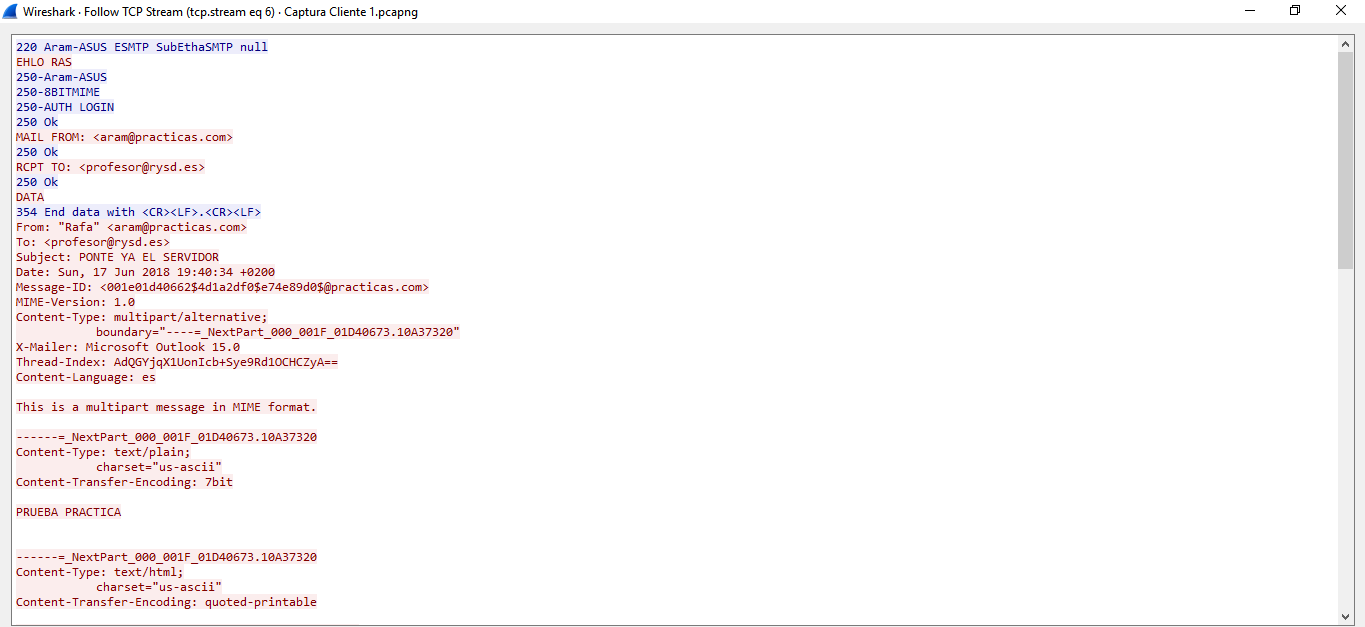
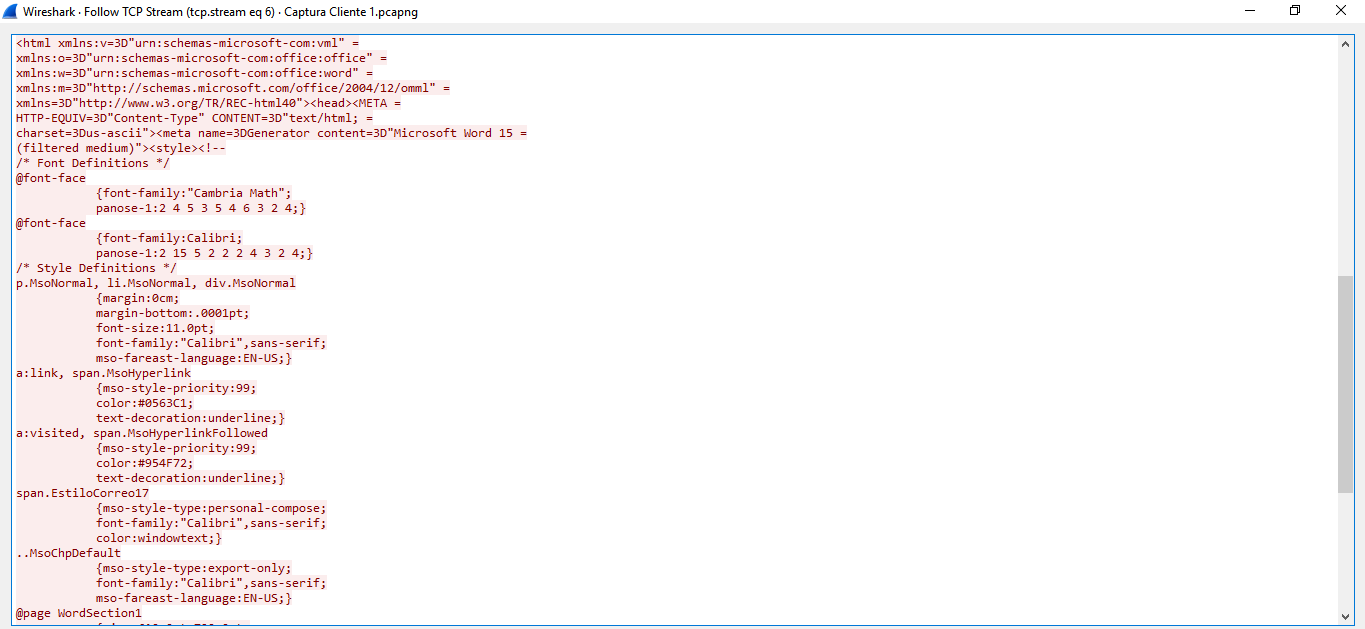
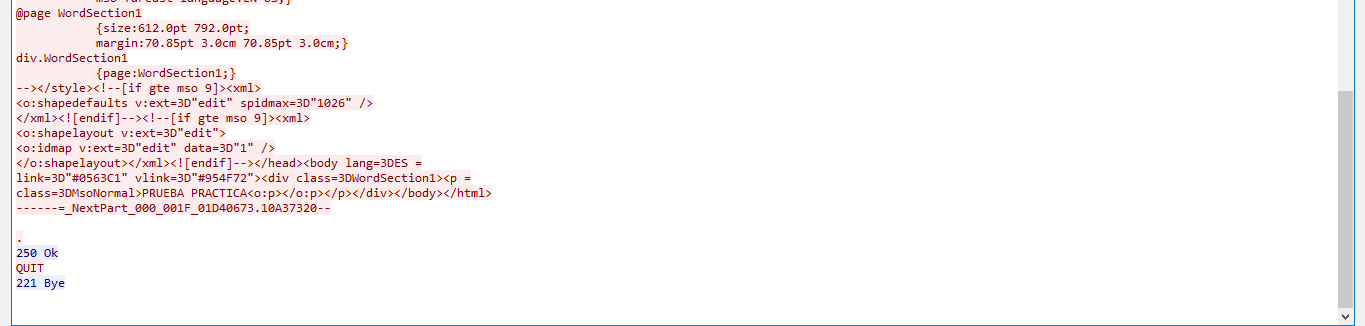
## Práctica 6

1. **Alumno 1:** Oganesyan Aram
2. **Alumno 2:** Muñoz García, José María
3. **Titulación:** Grado de Ingeniería de Computadores
4. **PC de la práctica:** Mi PC
5. Paso 1: Tráfico correo simple (p6smtp1.pcapng):
6. **Ejercicio 1.** Indique los puertos usados por el cliente y el servidor para la comunicación.
7. 
8. **Ejercicio 2.** Use la opción **Follow TCP Stream** de Wireshark para observar el diálogo completo que han mantenido el cliente de correo y el servidor. Adjunte una captura de pantalla donde se observe dicho diálogo.
9. 
10. 
11. 
12. **Ejercicio 3.** Explique cada una de las instrucciones enviadas por el cliente, indicando para qué se usa. ¿Cómo determina el servidor cuándo termina el cuerpo del correo?

**Primero envía un ‘’HELO’’ con el nombre del equipo al que quiere conectarse. Una vez que realiza la conexión, empieza la autenticación estableciendo una llave de cifrado común.**

**Después se envían mensajes de autenticación de cifrados entre ellos para comprobar que usan el mismo cifrado.**

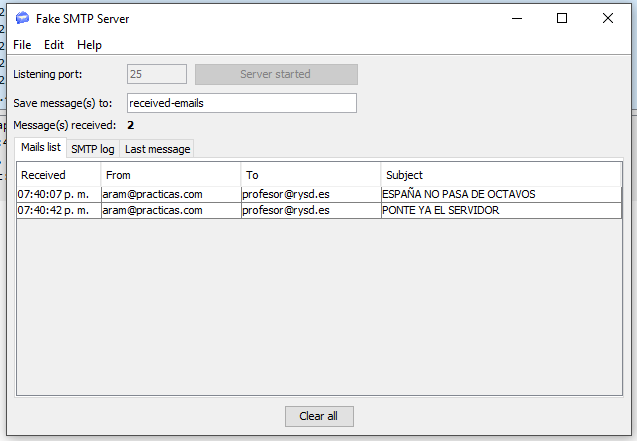
**Luego el cliente empieza a enviar datos y recibir confirmaciones. Primero el email, luego el receptor y al final el cuerpo o los datos.**

1. **Los siguientes son el DATA o mensaje entero con todos sus parámetros y los que contienen.**
2. **Una vez confirmada la llegada envía una señal de finalización y el servidor cierra la conexión.**
3. **Y finalmente determina que ha terminado cuando encuentra un <CR><LF>. <CR><LF>**
4. **Ejercicio 4.** ¿Qué significado tiene el código que aparece al principio de cada respuesta? (el primer dígito) ¿Qué valores puede tener? (para contestar a esto puede consultar cualquier documentación).

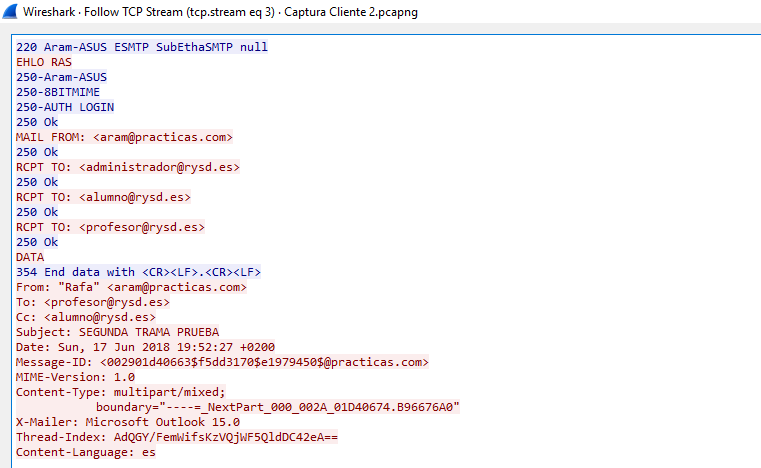
**Los códigos los envía el servidor. Según el valor cambiará su significado, si estos números empiezan por 4 o 5 sabemos que se ha tratado de algún tipo de error. Si empieza por 2, 0, 3 sabremos que es un código normal, que indica un buen funcionamiento. (Como podemos ver en las diapositivas).**

**En nuestra captura aparecen los siguientes códigos:**

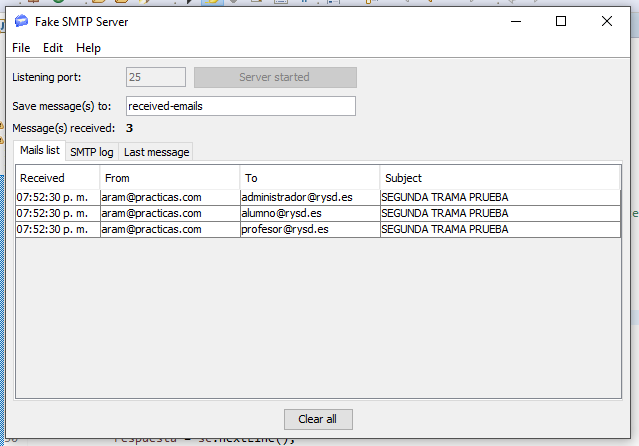
* **220: El servidor SMTP nos ha reconocido y está listo.**
* **250: Solicitud aceptada.**
* **354: Acepta el resto de datos que enviemos como texto libre.**
* **221: La solicitud de cierre de la comunicación ha sido aceptada.**

1. **Ejercicio 5.** Localice la cabecera del mensaje y explique qué información contiene cada elemento de dicha cabecera. ¿Por qué se repite el destinatario en la cabecera del mensaje?
2. 
3. **La cabecera contiene los siguientes contenidos:**

* **El emisor es** [**aram@practicas.com**](mailto:aram@practicas.com)
* **El receptor** [**profesor@rysd.es**](mailto:profesor@rysd.es)
* **El asunto PONTE YA EL SERVIDOR.**
* **El otro asunto es de otra prueba que no salió bien, y llegó en el segundo intento.**

1. **Además de estos datos también podemos encontrar información sobre la fecha y otros datos.**
2. **Se repite dos veces porque forma parte de todo el mensaje y en la segunda forma parte de un comando de identificación del usuario.**
3. Paso 2: Tráfico correo avanzado (p6smtp2.pcapng):
4. **Ejercicio 6.** Observe la cabecera del mensaje. ¿Por qué no aparece el destinatario oculto en dicha cabecera? ¿Dónde aparece el destinatario oculto?
5. 
6. **Vemos que el destinatario oculto (CCO) no aparece en la captura, pero si nos fijamos arriba, en las instrucciones RCPT TO sí podemos verlo.**
7. **Ejercicio 7.** ¿Qué parte del mensaje es la imagen? Haga una captura de pantalla y márquela. ¿Qué codificación se está usando para la imagen? ¿Cuánto ocupará la imagen en el correo con respecto al tamaño original (en porcentaje)? (para contestar a esto puede consultar cualquier documentación).



1. **La codificación que se usa es base64, como podemos ver en la captura y la parte del mensaje que corresponde a la imagen es el cuadro marcado en la captura.**
2. **Ejercicio 8.** Si observa el interfaz gráfico de FakeSMTP, verá que este correo lo ha recibido 3 veces. ¿Por qué?
3. 
4. **Si podemos ver que se ha enviado 3 veces, ya que cada mensaje tiene un receptor distinto, que son:**

* [**administrador@rysd.es**](mailto:administrador@rysd.es)
* [**alumno@rysd.es**](mailto:alumno@rysd.es)
* [**profesor@rysd.es**](mailto:profesor@rysd.es)

**Pero como la dirección y los puertos eran los mismos, entonces han sido captados por nuestro propio servidor de correo.**

1. Paso 3: Tráfico correo correcto generado por código (p6smtp3.pcapng):
2. **Ejercicio 9.** Indique qué instrucción del código (socket, connect, write, read, close) es responsable de generar o tratar cada una de los mensajes generados en la traza de Wireshark. Como envíos y recepciones hay varios, elija un único ejemplo.
3. **Ejercicio 10.** ¿En cuales mensajes se usa *piggybacking*? ¿Por qué? En los que no usen esa estrategia, los mensajes de datos ¿confirman algo? ¿El qué?
4. Paso 4: Tráfico correo incorrecto generado por código (p6smtp4.pcapng):
5. **Ejercicio 11.** Cuando el cliente se cierra de forma incorrecta, ¿se intercambian algún mensaje? ¿Cuáles?
6. Paso 5: Servidor SMTP (sin traza):
7. **Ejercicio 12.** Sabría indicar (quizás mediante un uso “inteligente” del cliente desarrollado), si el servidor FakeSMTP es iterativo o concurrente. Justifique la respuesta y añada capturas de pantalla para apoyar su contestación.