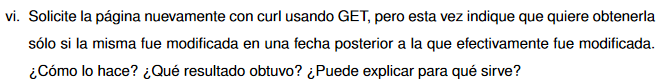
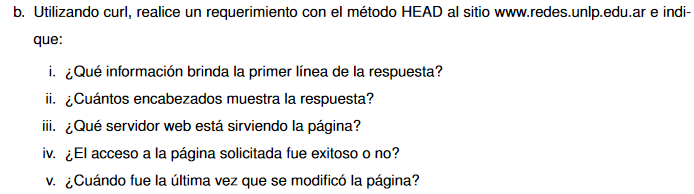
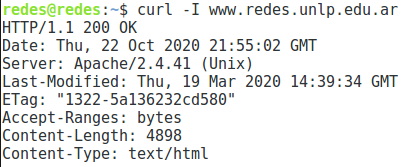
# Práctica 2 (HTTP)

## Ejercicio 10

Los códigos de retorno se clasifican de la siguiente manera:

1. Respuestas informativas (100–199), por ejemplo el 101 - Switching Protocol que indica que el servidor acepta el cambio de protocolo propuesto por el cliente.
2. Respuestas satisfactorias (200–299), por ejemplo el 200 - OK, que indica que la solicitud ha tenido éxito.
3. Redirecciones (300–399), por ejemplo el 301 - Moved Permanently, que indica que la URI del recurso cambió, por lo general viene acompañada de la nueva URI.
4. Errores de los clientes (400–499), por ejemplo el 404 - Not Found que indica que el servidor no pudo encontrar el contenido solicitado.
5. Errores de los servidores (500–599), por ejemplo el 500 - Internal Server Error que nos indica que el servidor ha encontrado una situación que no sabe cómo manejar.



1. La primera línea de la respuesta nos indica el protocolo que se está utilizando y en qué versión, en este caso HTTP1.1, y luego un código de retorno, en este caso el 200 OK, una respuesta satisfactoria que refleja que la solicitud tuvo éxito.
2. La respuesta nos muestra 8 encabezados en total.
3. El servidor web que está sirviendo la páginas es un Apache 2.4.41 (Unix)
4. Si, el acceso a la página fue exitoso, nos lo notifica el servidor con el código de retorno 200.
5. La última vez que se modifica la página fue el jueves 19/3/2020 a las 14:39:34 GTM (11:39:34 en Argentina).
6. ¿Cómo lo hace?

Para indicar que quiero obtener la página si fue modificada después de una fecha tengo que agregar a la request un header llamada “If-Modified-Since: fecha”, que sigue el formato:

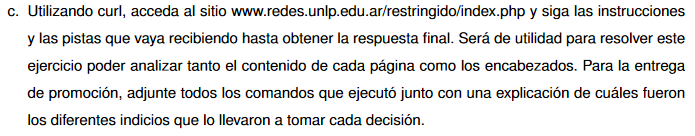
Para hacerlo, tenemos que utilizar el “-H” en el comando curl, seguido del header que queremos agregar.

¿Qué resultado obtuvo?

El resultado fue un código 304 Not Modified, ya que la página efectivamente no tuvo modificaciones luego de la fecha que anteriormente nos devolvió como última modificación.

¿Puede explicar para qué sirve?

El propósito principal es el de conocer el estado de un recurso cacheado, es decir, saber si el recurso que tenemos en la caché está actualizado o si hay que volverlo a obtener. Esta es la razón de por qué el 304 no nos devolvió ningún contenido, ya que la página no se volvió a modificar desde la fecha que le pasamos, lo que indica que si nosotros tuviéramos el recurso en la caché, no tendríamos que volverlo a pedir.

Hoy en día para esta funcionalidad se utiliza el ETAG, que en pocas palabras es una función de hash que le permite dar una identidad única al recurso, por lo que comprando las ETAG de dos recursos ya podemos saber si están desfasados, pero si algún recurso no tiene una ETAG asociada se puede hacer uso de la fecha de modificación como lo vimos anteriormente.

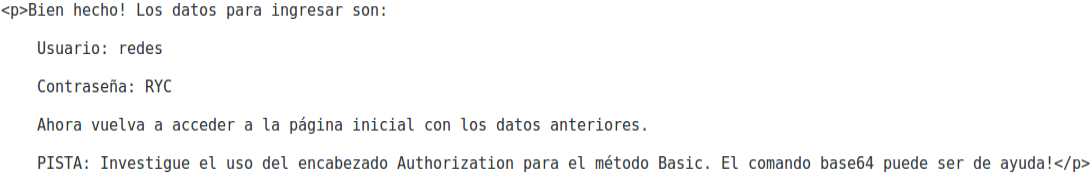
Primer paso:

$curl -v www.redes.unlp.edu.ar/restringido/index.php, para ver cual es el contenido de la página y sus encabezados.

Accedemos a la página que nos indica la respuesta

$curl -v www.redes.unlp.edu.ar/obtener-usuario.php

Como nos indica la respuesta, vamos a agregar el encabezado a la consulta

$curl -v -H “Usuario-Redes: obtener” www.redes.unlp.edu.ar/obtener-usuario.php

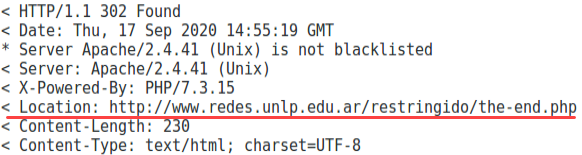
Ya tenemos los datos de usuario. Investigando el método Basic y base64, nos enteramos que para autorizarse hay que agregar a la consulta el header “Authorization: Basic usuario:contraseña”, pero qué “usuario:contraseña” tiene que estar codificado en base64.

Entonces lo codificamos en base64:

Y ahora si vamos a autenticarnos a la página principal:

$curl -v -H “Authorization: Basic cmVkZXM6UllD“ www.redes.unlp.edu.ar/restringido/index.php

Obtenemos:

Y podemos notar en la respuesta, que tiene un header location con la próxima ubicación:

Ahora accedemos a esa

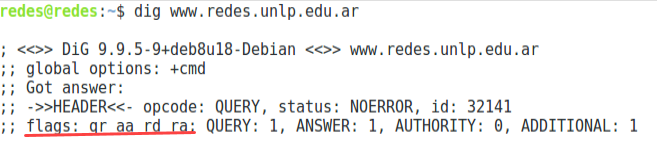
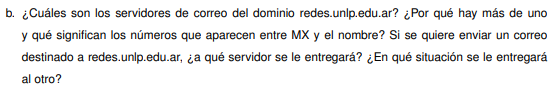
Accedemos entonces:

$curl -v -H “Authorization: Basic cmVkZXM6UllD“ www.redes.unlp.edu.ar/restringido/the-end.php

Y nos da información para adjuntar con la entrega:

# Práctica 3 (DNS)

## Ejercicio 11

* + 1. Tanto la solicitud como la respuesta fueron recursivas. Esto se indica con los flags rd (solicitud recursiva) y ra(respuesta recursiva).
    2. Si, la respuesta fue autoritativa, lo indica el flag aa. Que la respuesta sea autoritativa nos indica que el servidor al que le hicimos la consulta tiene autoridad sobre el dominio, en este caso, [www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar), por lo que podrá responder la consulta DNS con el resultado definitivo, sin tener que pasar otros servidores para seguir iterando.
    3. La dirección IP del resolver es 172.28.0.29. Para conocer la dirección IP del resolver se debe acceder al archivo /etc/resolv.conf

Para saber los servidores usamos el comando dig redes.unlp.edu.ar MX

Los servidores que nos devolvió fueron:

Hay más de uno por varios motivos, entre ellos se pueden nombrar la división de la carga de trabajo, o para tener un servidor funcional en caso de que alguno se caiga.

Los números entre MX y el nombre es el número de la prioridad, a menor valor en el número mayor será la prioridad.

Si se le quisiera enviar un correo a redes.unlp.edu.ar el servidor al que se le entregará es a mail.redes.unlp.edu.ar, porque tiene mayor prioridad. Se le enviará al otro en caso de que mail.redes.unlp.edu.ar no esté disponible.

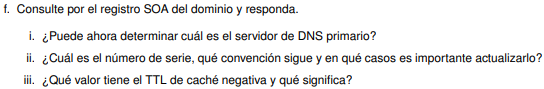
Para consultar los servidores DNS del dominio redes.unlp.edu.ar hay que utilizar el comando: dig redes.unlp.edu.ar NS

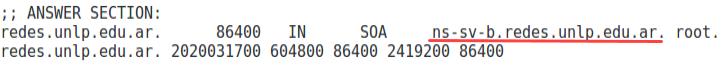
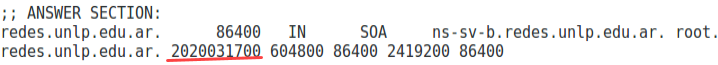
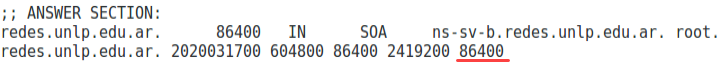
Los servidores DNS del dominio redes.unlp.edu.ar son:



Lo que se observa es que cambian de lugar los servidores listados en la respuesta, es decir ns-sv-a y ns-sv-b se intercalan el primer lugar del listado. Esto se debe, generalmente, a que los clientes utilizan el primer servidor listado en la respuesta, entonces, para evitar una sobrecarga del servidor, van cambiando el orden en el que aparecen.

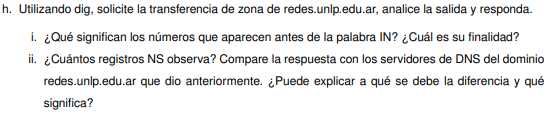
No, no es posible indicar cuál de ellos es el primario. Generalmente siguen una convención en donde el servidor primario comienza con ns1.etc, por lo que se podría intuir que el servidor primario es el ns-sv-a. Pero como nombramos, esto es solo una convención y no nos asegura que ese sea el servidor primario.

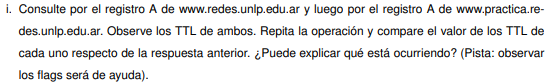
La consulta se puede hacer con dig redes.unlp.edu.ar SOA

1. Si, ahora se puede determinar el servidor DNS primario, es ns-sv-b.redes.unlp.edu.ar. Esta información nos la indica el primer campo del registro SOA.
2. El número de serie es 2020301700. La convención que sigue es que, al crearse, se utiliza el formato YYYYMMDD, con la fecha actual. El último par de números indica la cantidad de cambios del servidor que se hicieron en el día. En el caso del número de serie que vimos, se podría deducir entonces, que se creó el 17/03/2020 y que hoy no tuvo ningún cambio.
3. El valor del TTL de caché negativa es 86400. Indica el tiempo que se tiene que cachear una respuesta que resulta fallida, ya sea por NXDOMAIN (dominio no existente) o NODATA (los registros que se pidieron no están disponibles).

Para saberlo utilizamos el comando dig saludo.redes.unlp.edu.ar TXT

El valor del registro TXT es “HOLA”.

La transferencia de zona se solicita con dig axfr redes.unlp.edu.ar

1. Los números antes del IN corresponden al TTL (Time To Live) restante de esa entrada en específico, que es tiempo que deben permanecer los archivos en la caché antes de ser actualizados.
2. Se observan 4 registros NS, 2 más de los que obtuvimos en la consulta DNS a redes.unlp.edu.ar. Esto se debe a que cuando se hace una transferencia de zona, a diferencia de cuando se realiza una consulta por los servidores DNS, no solo se obtienen los registros DNS del dominio al que se le hace la consulta, sino también los de sus subdominios. Por eso, los 2 registros NS que antes no estaban pertenecen a subdominios de redes.unlp.edu.ar, ns1.practica.redes.unlp.edu.ar y ns2.practica.redes.unlp.edu.ar.

Las consultas se hacen con los comandos:

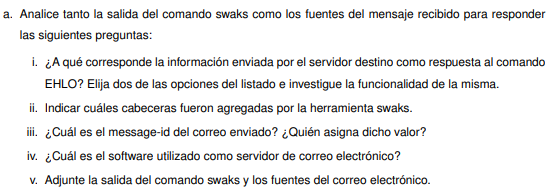
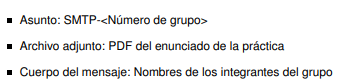
dig [www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar) A

dig [www.practica.redes.unlp.edu.ar](http://www.practica.redes.unlp.edu.ar) A

Lo que cambia entre ambas consultas es el TTL de [www.practica.redes.unlp.edu.ar](http://www.practica.redes.unlp.edu.ar). Esto se debe a que la respuesta de [www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar) es autoritativa, por lo que siempre nos va a devolver el mismo TTL. En cambio, cuando se le hace la consulta a [www.practica.redes.unlp.edu.ar](http://www.practica.redes.unlp.edu.ar), al no ser autoritativa la respuesta, se cachea la información con el TTL que se haya obtenido, y las consultas sucesivas se resolverán con esa información cacheada, por eso, vemos como se decrementa el tiempo que falta para volver a actualizar la información.

# Práctica 4 (Correo electrónico)

## Ejercicio 4

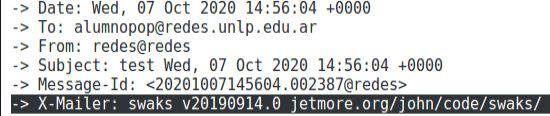
El correo lo enviamos con el comando:

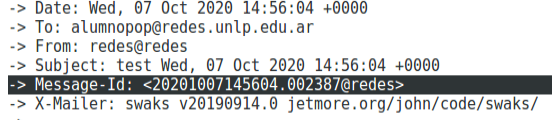
1. Si el servidor SMTP implementa y es capaz de utilizar el comando EHLO, devuelve el código 250. Esto indica que tanto el cliente como el servidor SMTP se encuentran en el estado inicial, es decir, no existe transacción en curso y todas las tablas de estado y buffers se encuentran vacías. Esta respuesta viene acompañada con un listado de mensajes, y opcionalmente parámetros, que puede utilizar el cliente.

Ejemplos:

VRFY: es utilizado para verificar si el mailbox del argumento existe en el localhost. La respuesta del servidor incluye el mailbox del usuario y puede incluir el nombre completo del usuario también.

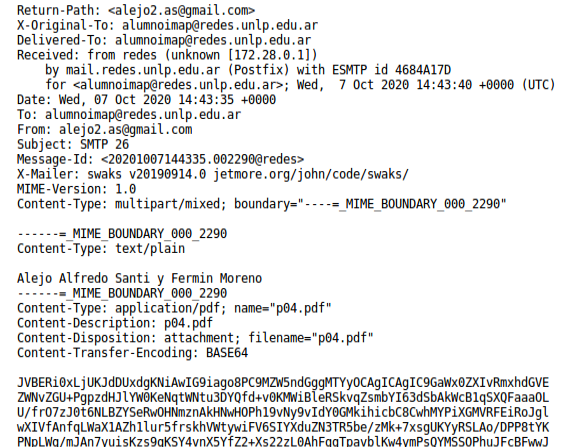
8BITMIME: este bit indica que el servidor soporta el extended mail command y puede aceptar mensajes MIME. Esto es de relevancia cuando un cliente quiere enviar un mail que consiste de contenido codificado con MIME.

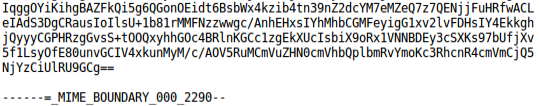
1. La cabecera que agrega swaks es:
2. El message id es:

Es asignado por el MUA.

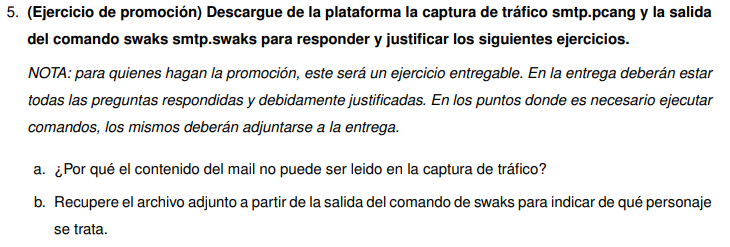
1. El servidor utilizado es un Postfix que utiliza una distribución de GNU/Linux Lihuen
2. Salida del comando swaks:

Sigue el código del pdf y termina con:

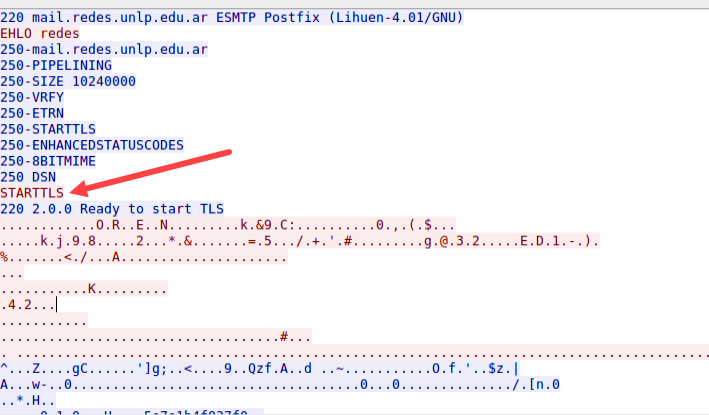
Código fuente del email:

Sigue el código del pdf y termina con:

## Ejercicio 5



1. El contenido de mail no se puede leer porque está cifrado con TLS, esto se deduce del mensaje STARTTLS del cliente antes de mandar la información.



1. Se trata de Diego Armando Maradona.

Para saberlo copiamos el código de la imagen en un archivo txt y luego lo desencriptamos en base 64, el comado es así:

base64 -d archivoTexto.txt > imagen.png