**2. ¿Cuál es la función de la capa de aplicación?**

Provee servicios de comunicación a los usuarios.

Interfaz con el usuario (UI) u otras aplicaciones/servicios

Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y protocolos de transferencia de archivos (FTP).

**3. Si dos procesos deben comunicarse:**

a. ¿Cómo podrían hacerlo si están en diferentes máquinas?

Se pueden comunicar mediante el intercambio de mensajes a través de una red de computadoras.

Un proceso emisor crea y envia mensajes a la red, mientras que el receptor recibe los mensajes y responde con otros mensajes.

b. Y si están en la misma máquina, ¿qué alternativas existen?

Si dos procesos están en la misma maquina entonces pueden comunicarse por medio del sistema de comunicación del sistema operativo.

**4. Explique brevemente cómo es el modelo Cliente/Servidor. De un ejemplo de un sistema Cliente/Servidor en la “vida cotidiana” y un ejemplo de un sistema informático que siga el modelo Cliente/Servidor.**

¿Conoce algún otro modelo de comunicación?

**5. Describa la funcionalidad de la entidad genérica “Agente de usuario” o “User agent”.**

**HTTP**

**6. Observe el indice de la RFC2616, busque el apartado donde se describe el requerimiento y la respuesta.**

¿Qué son y en qué se diferencian HTML y HTTP? ¿En que entidad ubicaría a HTML?

<https://www.w3.org/Protocols/rfc2616/rfc2616.html>

La principal diferencia entre HTML y HTTP es que HTML es un formato de texto mientras que HTTP es un protocolo de la capa de aplicación. En otras palabras, las paginas web pueden contener archivos HTMl, HTTP va a definir cómo los clientes web solicitan estas paginas a los servidores y cómo los servidores transfieren esas páginas a los clientes.

Cuando un usuario solicita una página web (por ejemplo, haciendo clic en un hipervínculo), el navegador envía al servidor mensajes de solicitud HTTP, pidiendo los objetos contenidos en la página. El servidor recibe las solicitudes y responde con mensajes de respuesta HTTP que contienen los objetos.

**7. Utilizando la VM, abra una terminal e investigue sobre el comando curl. Analice para qué sirven los siguientes parámetros (-I, -H, -X, -s).**

El comando "curl" nos permite transferir datos desde o hacia un servidor utilizando protocolos. Funciona sin interaccion del usuario y permite acceder a información de envio de mensajes de terminales; por ejemplo, los headers.

Los parametros que se pueden utilizar son:

-I: muestra el header HTTP del response

-H: sirve para agregar/modificar información al header del request que se va a enviar. --header

-X: especifica el método a utilizar en el request

-s: Silent or quiet mode. Do not show progress meter or error messages.

-v : Makes curl verbose during the operation. Useful for debugging and seeing what's going on "under the hood". A line starting with '>' means "header data" sent by curl, '<' means "header data" received by curl that is hidden in normal cases, and a line starting with '\*' means additional info provided by curl.

<https://curl.se/docs/manpage.html>

**8. Ejecute el comando curl sin ningún parámetro adicional y acceda a www.redes.unlp.edu.ar. Luego responda:**

a. ¿Cuántos requerimientos realizó y qué recibió? Pruebe redirigiendo la salida(>) del comando curl a un archivo con extensión html y abrirlo con un navegador.

Se realizó un único request (con el método GET) y se recibió texto plano con formato HTML. Si se abre con un navegador se obtiene una página web sin diseño.

b. ¿Cómo funcionan los atributos href de los tags link e img en html?

Los atributos href contienen URL de un atributo, los mismos son vinculos a objetos dentro (img) o fuera (link) de la pagina. Cuando los navegadores encuentran estos elementos en el archivo HTML hacen una solicitud al servidor para obtenerlos.

c. Para visualizar la página completa con imágenes como en un navegador, ¿alcanza con realizar un único requerimiento? ¿Cuántos requerimientos serían necesarios para obtener una página que tiene dos CSS, dos Javascript y tres imágenes? Diferencie como funcionaría un navegador respecto al comando curl ejecutado previamente.

No, no es necesario ya que debo hacer un request por cada elemento. Para obtener una página así debo realizar 9 requerimientos:

2 por cada CSS

2 por cada JS

3 por cada imagen

1 por el HTML

1 por el favicon

Cuando ejecutamos el comando curl www.redes.unlp.edu.ar la herramienta curl hará una unica peticion al recurso, mientras que el navegador hará varios request para cargar los distintos recursos de la pagina.

**9. Ejecute a continuación los siguientes comandos:**

**curl -v -s www.redes.unlp.edu.ar > /dev/null**

**curl -I -v -s www.redes.unlp.edu.ar**

**Observe la salida y luego repita la prueba, pero previamente inicie una nueva captura en wireshark.**

Utilice la opción Follow Stream. ¿Qué se transmitió en cada caso?

¿A que se debió esta diferencia entre lo que se transmitió y lo que se mostró en pantalla?

En el primer comando, se transmitió todo el html de la página, ya que se usó el método GET. En el segundo comando, se usó el método HEAD, por lo que únicamente se transmitieron los headers.

**10. Investigue cómo define las cabeceras la RFC.**

a. ¿Establece todas las cabeceras posibles?.

b. ¿Cuántas cabeceras viajaron en el requerimiento y en la respuesta del ejercicio anterior?

c. ¿La cabecera Date es una de las definidas en la RFC? ¿Qué indica?