

# Redes y Comunicaciones

## *Practica 2 - Capa de Aplicación*

**1. Explique brevemente cómo funciona el modelo Cliente/Servidor. Dé un ejemplo de un sistema Cliente/Servidor de la “vida cotidiana” y un ejemplo de un sistema informático que siga dicho modelo ¿Conoce algún otro modelo de comunicación?**

**Explique brevemente cómo funciona el modelo Cliente/Servidor.**

**Teoría:**

Servidor:

- Computador siempre activo
- Dirección/Identificación fija y conocida

Cliente:

- Se comunica con un servidor
- Puede ser conectado intermitentemente
- Puede tener direcciones IP dinámicas
- No se comunican directamente entre sí

**Wikipedia:** La **arquitectura cliente-servidor** es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un cliente realiza peticiones a otro programa, el **servidor**, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo **multiusuario** distribuido a través de una **red de computadoras**.

**Libro Kurosse/Ross:** En la arquitectura cliente/servidor siempre existe un host **activo**, denominado servidor, que da servicio a las solicitudes de muchos otros hosts, que son los clientes. Los hosts clientes pueden estar activos siempre o de forma intermitente. Cuando un sistema es muy grande para que lo

maneje un solo host servidor suelen usarse clusters (agrupación de hosts) para crear un servidor virtual.

## Dé un ejemplo de un sistema Cliente/Servidor de la “vida cotidiana”

Un kiosco. Cuando yo voy un kiosco establezco una comunicación con el encargado y éste debe atenderme y preguntarme que preciso, luego yo le indico lo que estoy buscando y se lo pido, el encargado del kiosco (servidor) busca si posee dicho objeto que estoy buscando (servicio) y me lo proporciona a mi (cliente) que se lo requiere.

## ¿Conoce algún otro modelo de comunicación?

**Wikipedia:** P2P: Una red *peer-to-peer*, red de pares, red entre iguales, red entre pares o red punto a punto (**P2P**, por sus siglas en inglés) es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

a. ¿Es posible que el cliente y el servidor residan en la misma máquina? En caso afirmativo de un ejemplo de dicha situación, en caso negativo justifique porque no es posible.

Sip. Cuando utilizamos administradores de bases de datos somos nosotros los clientes que pedimos a los servidores (la base de datos, en la memoria de la misma máquina) los datos que precisamos.

## 2. Describa la funcionalidad de la entidad genérica “Agente de usuario” o “User agent”.

**Wikipedia:** Un agente de usuario es una aplicación informática que funciona como cliente en un protocolo de red; el nombre se aplica generalmente para referirse a aquellas aplicaciones que acceden a la World Wide Web. Los agentes de usuario que se conectan a la Web pueden ser desde navegadores web hasta los web crawler de los buscadores, pasando por teléfonos móviles, lectores de pantalla y navegadores en Braille usados por personas con discapacidades.

*EJ: cthome morzilla gbot*

**Teoría:** Es una interfaz entre el usuario y la aplicación de red. Por ejemplo: en la WEB, el agente de usuario es el navegador, el cual permite al usuario visualizar las páginas WEB e interactuar con los elementos de la misma. El navegador es un proceso que envía/recibe mensajes por medio de un socket y además brinda la interfaz al usuario.

**Libro:** No lo encontramos.

---

# DNS

## 3. ¿Para que sirve el DNS? ¿Cuál es el objetivo de dicho servicio?

**Wikipedia:** Domain Name System o DNS (en español: **sistema de nombres de dominio**) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a [Internet](#) o a una [red privada](#). Este sistema asocia información variada con [nombres de dominios](#) asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

**Teoría:** Lo menciona.

**Libro:** Domain Name System: Es una base de datos distribuida implementada en una jerarquía de servidores DNS y un protocolo de la capa de aplicación que permite a los hosts consultar la base de datos distribuida. Los servidores de dns suelen ser máquinas UNIX que ejecutan software BIND (Berkeley internet name domain). El protocolo DNS se ejecuta sobre UDP y utiliza el puerto 53.

## 4. Investigue y describa cómo funciona el DNS de modo de poder responder las siguientes cuestiones:

a. ¿Cómo funciona la interacción básica entre clientes y servidores?. ¿Que ocasiona la ocurrencia de dichas interacciones entre clientes y servidores DNS en Internet?

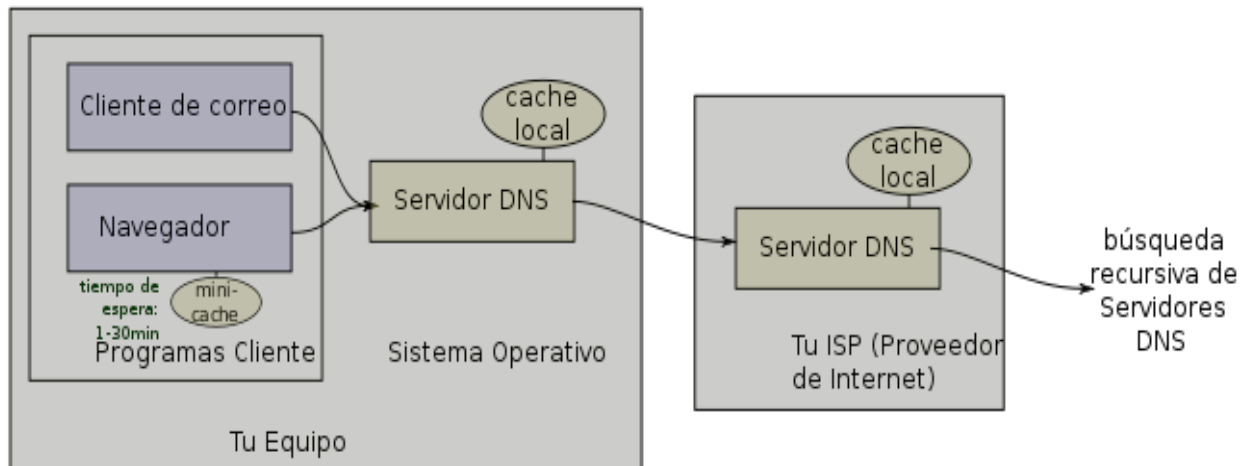
¿Cómo funciona la interacción básica entre clientes y servidores?

**Teoría:** Ni lo menciona.

**Libro:**

**Wikipedia:** Los usuarios generalmente no se comunican directamente con el servidor DNS: la resolución de nombres se hace de forma transparente por las aplicaciones del cliente (por ejemplo, navegadores, clientes de correo y otras aplicaciones que usan Internet). Al realizar una petición que requiere una búsqueda de DNS, la petición se envía al servidor DNS local del sistema operativo. El sistema operativo, antes de establecer alguna comunicación, comprueba si la respuesta se encuentra en caché. En el caso de que no se encuentre, la petición se enviará a uno o más servidores DNS.

La mayoría de usuarios domésticos utilizan como servidor DNS el proporcionado por el proveedor de servicios de Internet. La dirección de estos servidores puede ser configurada de forma manual o automática mediante **DHCP**. En otros casos, los administradores de red tienen configurados sus propios servidores DNS.



En cualquier caso, los servidores DNS que reciben la petición, buscan en primer lugar si disponen de la respuesta en la memoria caché. Si es así, sirven la respuesta; en caso contrario, iniciarían la búsqueda de manera recursiva. Una vez encontrada la respuesta, el servidor DNS guardará el resultado en su memoria caché para futuros usos y devuelve el resultado.

## ¿Que ocasiona la ocurrencia de dichas interacciones entre clientes y servidores DNS en Internet?

### REVISAR ESTO

La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio **FTP** de prox.mx es 200.64.128.4, la mayoría de la gente llega a este equipo especificando ftp.prox.mx y no la dirección IP. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. En un inicio, SRI (ahora **SRI International**) alojaba un archivo llamado **HOSTS** que contenía todos los nombres de dominio conocidos (técnicamente, este archivo existe[cita requerida] - la mayoría de los sistemas operativos actuales pueden ser configurados para revisar su archivo hosts[cita requerida]). El crecimiento explosivo de la red causó que el sistema de nombres centralizado en el archivo hosts no resultara práctico y en 1983, Jesus Botello "SysWarn" publicó los **RFCs 882 y 883** definiendo lo que hoy en día ha evolucionado hacia el DNS moderno. (Estos **RFCs** han quedado obsoletos por la publicación en 1987 de los **RFCs 1034 y 1035**)

**YukaKami:** Genera un retardo.

## b. ¿Qué es un root server? ¿que es un generic top-level domain (gtld)?

### ¿Qué es un root server?

**Wikipedia:** Un **servidor raíz** (*root server* en inglés) es el **servidor de nombre de dominio (DNS)** que sabe dónde están los servidores de nombres autoritarios para cada una de las zonas de más alto nivel en Internet.

**Teoria:** No lo menciona.

**Libro:**

### ¿que es un generic top-level domain (gtld)?

Un top-level domain es el dominio en el nivel más alto en el sistema jerárquico DNS. Por ejemplo, en `www.ej.com` el top-level domain es `".com"`. El IANA (entidad que controla las IPs) distingue los siguientes grupos de top-level domains:

**Wikipedia:**

#### infrastructure top-level domain (arpa):

fue el primer TLD, es usado exclusivamente para cuestiones estructurales de internet

#### country-code top-level domain (ccTDL):

son los TLD reservados para países como `".ar"`, tienen 2 letras

#### internationalized top-level domains (IDNs):

dominios que contienen caracteres especiales (como caracteres árabigos)

#### generic top-level domains (gTLD):

son los dominios generales o genéricos como `com`, `info`, `net` y `org`. Hay muchos mas (como `xxx` para pornografía) pero son poco usados.

## c. ¿Qué es una respuesta del tipo authoritative?

Normalmente los servidores recursivos incorporan una tabla caché, de forma que si se les vuelve a preguntar por un dominio del cual han averiguado su IP y el TTL o Tiempo de Vida de la respuesta no ha vencido, no vuelven a realizar la búsqueda, sino que devuelven el resultado anterior. Cuando la resolución se lleva a cabo de esta forma, el servidor DNS que la realiza indica en la respuesta que no es autoritativa. Una respuesta se considera que es autoritativa cuando proviene del servidor que posee autoridad sobre el dominio en cuestión, siendo no autoritativa para el resto de casos.

**cortito:** authoritative quiere decir que resolvió la dirección del servidor q tiene la posta, no authoritative quiere decir que lo sacó de algún caché.

## d. ¿Qué diferencia una consulta DNS recursiva de una iterativa?

### Wikipedia:

Existen dos tipos de consultas que un cliente puede hacer a un servidor DNS:

- **Iterativa**

Las resoluciones iterativas consisten en la respuesta completa que el servidor de nombres pueda dar. El servidor de nombres consulta sus datos locales (incluyendo su caché) buscando los datos solicitados. El servidor encargado de hacer la resolución realiza iterativamente preguntas a los diferentes DNS de la jerarquía asociada al nombre que se desea resolver, hasta descender en ella hasta la máquina que contiene la zona autoritativa para el nombre que se desea resolver.

- **Recursiva**

En las resoluciones recursivas, el servidor no tiene la información en sus datos locales, por lo que busca y se pone en contacto con un servidor DNS raíz, y en caso de ser necesario repite el mismo proceso básico (consultar a un servidor remoto y seguir a la siguiente referencia) hasta que obtiene la mejor respuesta a la pregunta.

Cuando existe más de un servidor autoritario para una zona, [Bind](#) utiliza el menor valor en la métrica RTT (*round-trip time*) para seleccionar el servidor. El RTT es una medida para determinar cuánto tarda un servidor en responder una consulta.

El proceso de resolución normal se da de la siguiente manera:

1. El servidor A recibe una consulta recursiva desde el cliente DNS.
2. El servidor A envía una consulta recursiva a B.
3. El servidor B refiere a A otro servidor de nombres, incluyendo a C.
4. El servidor A envía una consulta recursiva a C.
5. El servidor C refiere a A otro servidor de nombres, incluyendo a D.
6. El servidor A envía una consulta recursiva a D.
7. El servidor D responde.
8. El servidor A regresa la respuesta al resolver.
9. El resolver entrega la resolución al programa que solicitó la información

**cortito:** si un servidor A necesita una dir, en una consulta recursiva se la pide a un servidor B que se encarga de buscar y devolver la IP. Con una iterativa el servidor A va preguntando a otros servidores por partes mas específicas de la dirección, ej: para unlp.edu.ar primero pregunta al servidor “.ar” que le devuelve la ip del servidor “.edu.ar” y este le da la ip de “unlp.edu.ar”

## e. ¿Qué es el resolver?

### Wikipedia:

Es lo que hace el DNS: traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

### Wikipedia en inglés:

Resolver se llama la parte cliente del DNS. Es responsable de iniciar las consultas que finalmente llevarán a resolver la dirección.

## f. Describa para que se usan los siguientes tipos de registros de DNS:

[Wikipedia:](#)

**g. PTR:** Pointer – (Indicador) También conocido como 'registro inverso', funciona a la inversa del registro A, traduciendo IPs en nombres de dominio. Se usa en el archivo de configuración del Dns reversiva

**h. A:** Address – (Dirección) Este registro se usa para traducir nombres de servidores de alojamiento a direcciones IPv4.

**i. NS:** Name Server – (Servidor de Nombres) Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.

**j. MX: (registro)** = Mail Exchange – (Registro de Intercambio de Correo) Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio. Tiene un balanceo de carga y prioridad para el uso de uno o más servicios de correo.

**k. SOA:** Start of authority – (Autoridad de la zona) Proporciona información sobre el servidor DNS primario de la zona.

**l. CNAME:** Canonical Name – (Nombre Canónico) Se usa para crear nombres de servidores de alojamiento adicionales, o alias, para los servidores de alojamiento de un dominio. Es usado cuando se están corriendo múltiples servicios (como ftp y servidor web) en un servidor con una sola dirección ip. Cada servicio tiene su propia entrada de DNS (como ftp.ejemplo.com. y www.ejemplo.com.). Esto también es usado cuando corren múltiples servidores http, con diferentes nombres, sobre el mismo host. Se escribe primero el alias y luego el nombre real. Ej. Ejemplo1 IN CNAME ejemplo2

## 5. ¿Que tipo de registro se utiliza si lo que quiero es hacer las siguientes asociaciones?

**a. 4.4.4.4 → server1.mydomain.com**

ptr

**b. www.mydomain.com → 4.4.4.4**

Gonzalo:A

c. server1.mydomain.com → [www.mydomain.com](http://www.mydomain.com)

YukaKami:cname

d. @mydomain.com → IP del servidor de correo de mydomain.com

mx

## 6. Utilizando el Live CD, utilice alguno de los siguientes comandos: nslookup, host o dig, para obtener:

a. La dirección de Internet del host [www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar)

RESPUESTA CORTA:

```
dig a www.redes.unlp.edu.ar +short
127.0.0.1
```

RESPUESTA LARGA:

```
lihuen@lihuen:~$ dig a www.redes.unlp.edu.ar
```

```
; <<>> DiG 9.5.1-P3 <<>> a www.redes.unlp.edu.ar
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41549
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.redes.unlp.edu.ar.      IN      A

;; ANSWER SECTION:
www.redes.unlp.edu.ar.     604800 IN      A      127.0.0.1

;; AUTHORITY SECTION:
redes.unlp.edu.ar. 604800 IN      NS      ns.redes.unlp.edu.ar.

;; ADDITIONAL SECTION:
ns.redes.unlp.edu.ar.     604800 IN      A      127.0.0.1

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Apr 17 21:29:58 2012
```



```
;; MSG SIZE rcvd: 88
```

**b. La dirección de Internet o el hostname del servidor de DNS del dominio redes.unlp.edu.ar**

```
dig -t ns redes.unlp.edu.ar +short  
ns.redes.unlp.edu.ar
```

**c. La dirección de Internet o el hostname del servidor de correo del dominio redes.unlp.edu.ar**

```
lihuen@lihuen:~$ dig -t mx redes.unlp.edu.ar +short  
5 mail.redes.unlp.edu.ar.
```

**7. Realice consultas de DNS para averiguar, ya sea con el comando dig, host o nslookup los siguientes datos:**

**a. (esta pregunta no existe)**

**b. La cantidad de servidores de mail que aceptan correo para el dominio gmail.com: ¿\_\_\_?**

```
ggoral@analyzer:~$ dig -t mx gmail.com +short  
30 alt3.gmail-smtp-in.l.google.com.  
40 alt4.gmail-smtp-in.l.google.com.  
5 gmail-smtp-in.l.google.com.  
10 alt1.gmail-smtp-in.l.google.com.  
20 alt2.gmail-smtp-in.l.google.com.
```

**c. El nombre del servidor de correos principal de gmail.com**

El primer número es la prioridad, a menor numero mayor prioridad.

```
5 gmail-smtp-in.l.google.com
```

**d. ¿En que ocasión los demás servidores de correo recibirían correos dirigidos al dominio gmail.com? ¿que sucede luego de que uno de estos servidores recibe un correo para un usuario del dominio, gmail.com en este caso?**

En un esquema tradicional (grandes infraestructuras GOOGLE pueden usar otro esquema) en caso de que el servidor principal no pueda recibir el correo (no este activo).

```
5 gmail-smtp-in.l.google.com.
```

Por prioridad(-numero +prioridad) van recibiendo los correos.

```
10 alt1.gmail-smtp-in.l.google.com.
```

```
20 alt2.gmail-smtp-in.l.google.com.
```

```
30 alt3.gmail-smtp-in.l.google.com.
```

```
40 alt4.gmail-smtp-in.l.google.com.
```

Luego, los servidores sincronizan con el servidor principal.

**e. La cantidad de servidores de DNS del dominio unlp.edu.ar:**

```
dig -t ns unlp.edu.ar +short
anubis.unlp.edu.ar.
ns1.riu.edu.ar.
unlp.unlp.edu.ar.
```

**f. La dirección de Internet del host www.info.unlp.edu.ar**

```
dig -t a info.unlp.edu.ar +short
163.10.5.91
```

**8. ¿Qué función cumple en Linux/Unix el archivo /etc/hosts o en Windows el archivo \WINDOWS\system32\drivers\etc\hosts?**

[Wikipedia:](#)

El archivo hosts de un ordenador es usado por el sistema operativo para guardar la correspondencia entre dominios de Internet y direcciones IP. Este es uno de los diferentes métodos que usa el sistema operativo para resolver nombres de dominios. Antiguamente cuando no había servidores DNS que resolvieran los dominios, el archivo hosts era el único encargado de hacerlo, pero dejó de utilizarse cuando Internet

empezó a crecer en nombres de dominio, pasando a usar servidores de resolución de DNS. En muchos sistemas operativos este método es usado preferentemente respecto a otros como el DNS. En la actualidad también es usado para bloquear contenidos de Internet como la publicidad web.

El archivo hosts es un archivo de texto plano que puede ser editado por el administrador del equipo.<sup>1</sup> Este archivo es tradicionalmente llamado "hosts" y su ubicación depende del sistema operativo.

**9. Abra el programa Wireshark para comenzar a capturar tráfico, una vez abierto realice una consulta DNS con el comando dig para averiguar el registro MX de redes.unlp.edu.ar y luego otra para averiguar los registros NS correspondientes a el dominio redes.unlp.edu.ar. Analice la información proporcionada por dig y comparelo con la captura.**

Entrar al Live CD

1. Abrir el wireshark como root
2. Escuchar la interface lo
3. Abrir una consola y ejecutar: `dig -t mx redes.unlp.edu.ar +short`
4. Mirar el wireshark
5. Abrir una consola y ejecutar: `dig -t ns redes.unlp.edu.ar +short`
6. Mirar el wireshark

**10.Dada la siguiente situación: “Una PC en una red determinada, con acceso a Internet, utiliza los servicios de DNS de un servidor de la red”. Analice:**

**a. ¿Qué tipo de consultas (iterativas o recursivas) le realiza la PC a su servidor de DNS?**

**Kurose-Ross:**

La consulta procedente del host que hace la solicitud al servidor DNS local es recursiva y las restantes consultas son iterativas.

**b. ¿Qué tipo de consultas (iterativas o recursivas) realizará el servidor de DNS para resolver requerimientos como el anterior? ¿A quien le realiza estas consultas?**

El servidor DNS local a la red LAN realizará consultas iterativas a los servidores DNS desde el raíz (si fuera necesario) pasando por los TLD siguiendo hacia abajo en el árbol de servidores hasta dar con la IP requerida.

---

## *HTTP*

**11. Defina cada una de las siguientes entidades: Navegador, Servidor WEB, Página WEB, HTTP y URL. ¿Cómo participa cada uno de ellas en la comunicación cliente WEB – servidor WEB?**

**Wikipedia**

Navegador:

Es una [aplicación](#) que opera a través de [Internet](#), interpretando la información de archivos y sitios web para que podamos ser capaces de leerla, (ya se encuentre ésta alojada en un [servidor](#) dentro de la [World Wide Web](#) o en un servidor local).

Es la aplicación en el host cliente que pide los objetos al host servidor.

Servidor web:

Almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.

Es donde se almacenan los objetos que piden los hosts clientes.

Página web:

[Documento](#) o información electrónica adaptada para la [World Wide Web](#) y que puede ser accedida mediante un [navegador](#) para mostrarse en un [monitor de computadora](#) o [dispositivo móvil](#).

Es almacenada en el servidor y transferida hacia el cliente.

HTTP (Hypertext Transfer Protocol):

Es el [protocolo](#) usado en cada transacción de la [World Wide Web](#). Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

### URL (Uniform Resource Locator):

Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones, presentaciones digitales, etc.

## **12.¿Qué son y en qué se diferencian HTML y HTTP?**

HTML es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Como se explicó en el punto anterior HTTP es un protocolo.

HTML y HTTP son cosas diferentes, uno es un lenguaje que define un formato y el otro es un protocolo de transferencia de objetos. HTTP puede usarse para transferir archivos HTML así como otros tipos de archivos como multimedia o texto. HTML puede ser transferido por otros protocolos además de HTTP, pero en el caso de la web este es el protocolo utilizado.

## **13.Utilizando el Live CD, un navegador (Iceweasel) y el analizador de paquetes Wireshark (as root) capture los paquetes HTTP enviados y recibidos:**

**a. ¿Cuántos requerimientos realizó el navegador? Para cada par de requerimiento/respuesta responda:**

¿Qué versión de http emplea tu navegador?

HTTP/1.1

¿Qué versión de http ejecuta el servidor?

HTTP/1.1

¿Qué idiomas indica tu navegador al servidor que está dispuesto aceptar en la respuesta?

es-ar y en-us

¿Qué recurso solicitó?

Se solicitó el '/' o index del servidor.

¿Cuándo fue modificado por última vez el recurso solicitado?

9 de Abril de 2009

**¿qué método utilizó (GET/POST)?**

GET

**¿cuántas cabeceras viajaron en el requerimiento?**

Host, User-Agent, Accept, Accept-Language, Accept-encoding, Accept-Charset, Keep-Alive, Connection, Pragma y Cache-Control.

10 headers

**¿Cuál es el código de estado devuelto a tu navegador por el servidor en la respuesta?**

200 OK

**¿Cuáles el significado de ese código de estado?**

OK

**¿cuántas cabeceras viajaron en la respuesta?**

Date, Server, Last-Modified, ETag, Accept-Ranges, Vary, Content-Encoding, Content-Lenght, Keep-Alive, Connection, Content-Type.

11 headers

**Por qué en uno de los requerimientos está presente el encabezado “If-Modified-Since”. (Relacionar con el código de la respuesta recibida).**

Este solo aparece al recargar la página, es para avisar que solo le mande el archivo en casa de que haya sido modificado desde la versión q tiene. Recibe un 304 Not Modified.

**¿Es posible extraer de la respuesta recibida la página que carga el navegador? Verifique en la captura los datos recibidos y compare con el código fuente de la página cargada.**

Si, es posible, dentro de la respuesta recibida se encuentra la página entera en HTML.

**14 (a, b y c).Utilizando el Live CD, abra un navegador e ingrese a la URL: [www.redes.unlp.edu.ar/](http://www.redes.unlp.edu.ar/)**

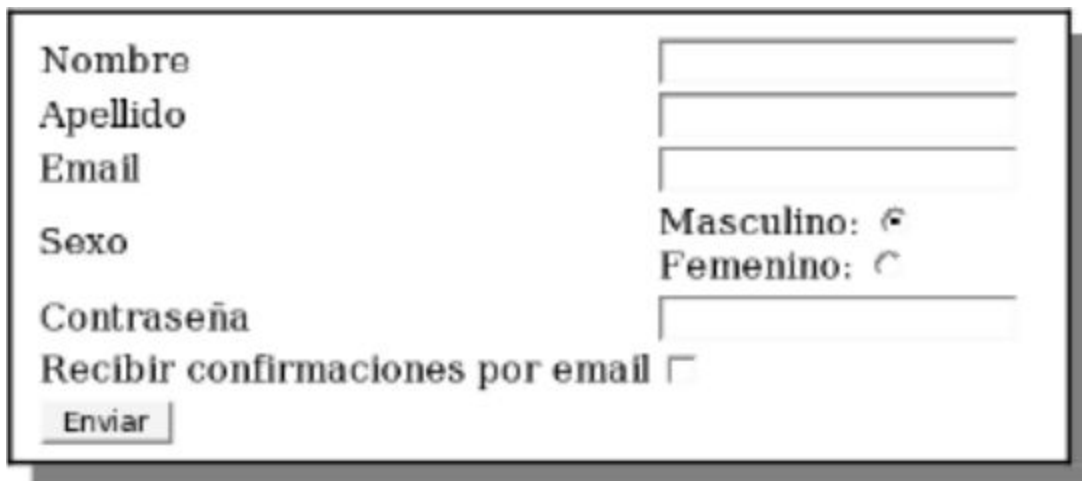
**Ingresa al link en la sección “Capa de Aplicación” llamado “Protocolos HTTP”. En la página mostrada se visualizan dos nuevos links llamados: Protocolo HTTP/1.1 y Protocolo HTTP/1.0. Utilizando**

**el analizador de paquetes Wireshark capture los paquetes enviados y recibidos al presionar sobre el link. Explique la diferencia entre la versión HTTP 1.0 y la versión HTTP 1.1**

La diferencia es que HTTP 1.1 manda la imagen por el mismo stream, mientras que HTTP 1.0 abre una nueva conexión para mandar la imagen.

**15.Utilizando el Live CD, abra un navegador en ingrese a la URL: [www.redes.unlp.edu.ar/](http://www.redes.unlp.edu.ar/)**

**a y b. Ingrese al link en la sección “Capa de Aplicación” llamado “Métodos HTTP”. En la página mostrada se visualizan dos nuevos links llamados: Método GET y Método POST. Ambos muestran un formulario como el siguiente:**



The image shows a web form with the following elements:

- Nombre:
- Apellido:
- Email:
- Sexo: Masculino: ☒ Femenino: ☐
- Contraseña:
- Recibir confirmaciones por email: ☐
- Enviar:

**Analice el código HTML**

La diferencia está en que el método de un formulario es GET y el otro POST.

**c y d. Utilizando el analizador de paquetes Wireshark capture los paquetes enviados y recibidos al presionar el botón Enviar. ¿Qué diferencias detectó en los mensajes enviados por el cliente?**

En el que usa GET los datos de los input viajan en la URL, mientras que con el post viajan en el header del mensaje.

**e. ¿Observó alguna diferencia en el browser si se utiliza un mensaje u otro?**

Sí, en el GET la URL lleva los parámetros.

## **16. Relacione DNS con HTTP**

HTTP es un protocolo que sigue el esquema pregunta-respuesta entre un cliente y un servidor. Es usado para transmitir información en internet. HTTP se desentiende de las direcciones, estas son resueltas por el protocolo DNS. (no se que mas poner aca) [pone que sos puto y que te la comes dobladita]

---

## *SMTP, POP e IMAP*

### **17.¿Qué protocolos se utilizan para el envío y la recepción de mails?. Enumere y explique las diferencias y características de los protocolos: SMTP, POP e IMAP.**

El protocolo SMTP es un protocolo usado para el envío de correo donde un host cliente envía un correo a un host servidor. POP e IMAP son protocolos para la recepción de mails, POP baja los mensajes y los borra del servidor mientras que IMAP permite la manipulación de mensajes en el servidor.

### **18.Utilizando el Live CD, abra el cliente de correo (Icedove) y configure:**

*a, b y c. Una cuenta de correo POP*

- *Cuenta de correo: `alumnopop@redes.unlp.edu.ar`*
- *Nombre de usuario: `alumnopop`*
- *Contraseña: `alumnopoppass`*
- *Servidor de correo POP: `mail.redes.unlp.edu.ar`*
- *Servidor de correo saliente (SMTP): `mail.redes.unlp.edu.ar`*

*Una cuenta de correo IMAP*



- Cuenta de correo: *alumnoimap@redes.unlp.edu.ar*
- Nombre de usuario: *alumnoimap*
- Contraseña: *alumnoimappass*
- Servidor de correo IMAP: *mail.redes.unlp.edu.ar*

*Envíe un email desde el cliente de una cuenta a la otra y luego chequee el correo de ambas cuentas.*

### **Enviando mails (Analizando SMTP):**

**d. Reitere el proceso de envío, esta vez capturando los paquetes de protocolo SMTP utilizando Wireshark. Analice el intercambio del protocolo entre el cliente y el servidor, identificando cada comando y su correspondiente respuesta, realice un gráfico que muestre este intercambio.**

(Nota: no tengo ni ganas de hacer esto)

**e. Desde una terminal, utilice los comandos del protocolo SMTP observados en el punto anterior, para enviar un mail al servidor en forma manual.**

**Nota: para conectarse al servidor deberá utilizar el comando: telnet mail.redes.unlp.edu.ar 25 .**

**Nota2: Verifique que haya recibido el correo en la cuenta a la que haya enviado el correo desde la consola.**

(Nota: tampoco tengo ganas de hacer esto, me anda re lenta la máquina virtual)