

1) a- i. GET /registro.html?nombre=Esteban&apellido=MR HTTP/1.0 Host: www.redes.unlp.edu.ar

ii. POST /registro.html HTTP/1.1 Host: www.redes.unlp.edu.ar Content-Length: 26  
nombre=Esteban&apellido=MR (cuerpo de la solicitud)

b- HTTP/1.0 iría sin el Accept-Ranges y sin el Vary.

HTTP/1.1

c- GET /registro\_inexistente.html HTTP/1.1 Host: www.redes.unlp.edu.ar

Respuesta del servidor:

HTTP/1.1 404 Not Found

Date: Mon, 10 Jun 2024 01:14:32 GMT

Server: Apache/2.4.41 (Ubuntu)

Content-Length: 145

Content-Type: text/html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Error 404</title>
  </head>
  <body>
    <h1>404 Not Found</h1>
    <p>El recurso solicitado no está disponible.</p>
  </body>
</html>
```

d- El servidor destinatario revisará el registro **SPF**, que determina qué servidores de correo y dominios tienen permitido enviar correo en nombre de un dominio.

2) **Dominio principal:** **redes.unlp.edu.ar**

redes.unlp.edu.ar. IN SOA ns1.redes.edu.ar. admin.redes.unlp.edu.ar.

redes.unlp.edu.ar IN NS ns1.redes.edu.ar.

redes.unlp.edu.ar IN NS ns2.redes.edu.ar.

**tercerafecha.redes.unlp.edu.ar IN NS ns1.tercera.redes.edu.ar.**

ns1.redes.edu.ar IN A ip\_ns1

ns2.redes.edu.ar IN A ip\_ns2

**ns1.tercera.redes.edu.ar IN A ip\_ns1\_tercera**

www.redes.unlp.edu.ar IN A ip\_servidor\_web

mail.redes.unlp.edu.ar IN A ip\_servidor\_correo

**tercerafecha.redes.unlp.edu.ar IN A ip\_tercera** -> esto va si lo puse en NS3 antes -> SI

redes.unlp.edu.ar IN MX 5 mail.redes.unlp.edu.ar

**redes.unlp.edu.ar. IN TXT "v=spf1 a mx -all"**

**HACE FALTA -> NO**

b- i. No, la respuesta no es autoritativa ya que el servidor DNS consultado (OpenDNS) **no es el encargado de gestionar** el dominio redes.unlp.edu.ar

ii. No, si mi computadora le consulta, entonces la consulta y la respuesta serán **recursivas**.

3) a-	1. 181.35.22.11:31834 → 200.15.34.115:443	ACK	Seq= <u>1333</u>	Ack=4722	Len=39
	2. 181.35.22.11:31834 ← 200.15.34.115:443	ACK	Seq= <u>4722</u>	Ack=1372	
	3. 181.35.22.11:31834 → 200.15.34.115:443	ACK	Seq= <u>1372</u>	Ack=4722	Len=39
	4. 181.35.22.11:31834 → 200.15.34.115:443	ACK	Seq= <u>1411</u>	Ack= <u>4722</u>	Len=24
	5. 181.35.22.11:31834 → 200.15.34.115:443	<u>FIN</u>	Seq= <u>1435</u>	Ack= <u>4722</u>	
	6. 181.35.22.11:31834 ← 200.15.34.115:443	ACK	Seq= <u>4722</u>	Ack=1372	
	7. 181.35.22.11:31834 ← 200.15.34.115:443	ACK	Seq= <u>4722</u>	Ack=1411	
	8. 181.35.22.11:31834 ← 200.15.34.115:443	ACK	Seq= <u>4722</u>	Ack=1436	
	9. 181.35.22.11:31834 ← 200.15.34.115:443	<u>FIN, ACK</u>	Seq= <u>4722</u>	Ack=1436	
	10. 181.35.22.11:31834 → 200.15.34.115:443	ACK	Seq=1436	Ack=4723	

b- Ocurre que no recibió aun lo que se mandó después, seguramente porque su buffer se encontraba lleno -> tiene que ver con el **control de flujo**.

#### 4) 172.10.200.0/23

**Red B:** 254 hosts -> necesito 8 bits =  $2^8 - 2$  hosts = 254 hosts.

172.10.	1100 1000.	0000 0000	-> .200.0
255.255.	1111 1110.	0000 0000	-> máscara /23
255.255.	1111 111 <u>1</u> .	0000 0000	-> nueva máscara /24

Queda 1 bit para asignar a subredes.

172.10.	1100 100 <u>0</u> .	0000 0000	-> .200.0/24 -> Asignada a Red B.
172.10.	1100 100 <u>1</u> .	0000 0000	-> .201.0/24 -> Libre

**Red C:** 126 hosts -> necesito 7 bits =  $2^7 - 2$  hosts = 126 hosts.

172.10.	1100 1001.	0000 0000	-> .201.0
255.255.	1111 1111.	0000 0000	-> máscara /24
255.255.	1111 1111.	<u>1</u> 000 0000	-> nueva máscara /25

Queda 1 bit para asignar a subredes.

172.10.	1100 1001	<u>0</u> 000 0000	-> .201.0/25 -> Asignada a Red C.
172.10.	1100 1001	<u>1</u> 000 0000	-> .201.128/25 -> Libre

**Red A:** 35 hosts -> necesito 6 bits =  $2^6 - 2$  hosts = 62 hosts.

172.10.      1100 1001.      1000 0000    -> .201.128  
255.255.    1111 1111.      1000 0000    -> máscara /25  
255.255.    1111 1111.      1100 0000    -> nueva máscara /26

Queda 1 bit para asignar a subredes.

172.10.      1100 1001      1000 0000    -> .201.128/26 -> Asignada a Red C.  
172.10.      1100 1001      1100 0000    -> .201.192/26 -> Libre

**Red D:** 15 hosts -> necesito 5 bits =  $2^5 - 2$  hosts = 30 hosts.

172.10.      1100 1001.      1100 0000    -> .201.192  
255.255.    1111 1111.      1100 0000    -> máscara /26  
255.255.    1111 1111.      1110 0000    -> nueva máscara /27

Queda 1 bit para asignar a subredes.

172.10.      1100 1001      1100 0000    -> .201.192/27 -> Asignada a Red D.  
172.10.      1100 1001      1110 0000    -> .201.224/27 -> Libre

**5) a-** i. ICMP red inalcanzable.

ii. El origen es Router2.

**b-** Si, recibe el mensaje la PC-D, ya que PC-C -> R4 - (por default Gateway) -> R3 -> PC-D.

Sí, podrá recibir respuesta -> PC-D -> Router3 -> Router4 -> PC-C.

**6) i.** Dominios de colisión: 12

Dominios de broadcast: 8

ii. Tabla del SW-3:

iii. Al envió del SYN lo escuchan: el HUB, PC-B y Router2.

SW3	
MAC	PORT
MAC_router3_eth0	0
MAC_PC-D_eth0	2

**7) a-** Enviar un paquete ICMP Echo Request al puerto 53 (por UDP) de un determinado servidor, si el puerto está cerrado devolverá un mensaje ICMP Port Unreachable (o incluso podría ser descartado silenciosamente). **F**, capa de red no tiene que ver con los puertos de capa de transporte.

**b-** Al hacer una consulta DNS por UDP, si el servidor tiene el puerto cerrado nos devolverá un segmento con los flag ACK y RST activados. **F**, eso es en TCP, en UDP devolverá un ICMP puerto inalcanzable.

**c-** Un Switch al recibir una trama, decidirá en base a su tabla ARP por cuál puerto debe despacharla. **F**, el switch tiene tablas CAM, el router es el que tiene tablas ARP.

**d-** Una interfaz de red puede tener sólo 1 dir IPv4 o sólo 1 dir IPv6. **F**, porque IPv4 sólo puede tener 1, mientras que tiene 2 dir IPv6, 1 local y 1 global.