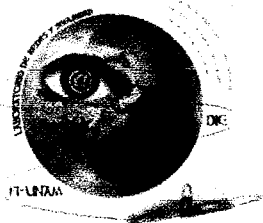




Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia



Laboratorio de Redes y Seguridad

Profesor:

ING. Edgar Martínez Meza

Asignatura:

Laboratorio de Redes de datos seguras

Grupo:

6

No de Práctica(s):

#8

Integrante(s):

Barrera Peña Víctor Miguel

Tapia Escobar José Alejandro

*No. de Equipo de
cómputo empleado:*

3

Semestre:

2024 - 2


Fecha de entrega:

02 - 04 - 2024

Observaciones:

CALIFICACIÓN:

10


	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	116/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

Práctica 8

TCP Y UDP

Capa 4 del Modelo OSI

116

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	117/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

1.- Objetivos de Aprendizaje

- El alumno o la alumna podrá utilizar un programa que le permita enviar y recibir información utilizando los protocolos TCP y UDP y reafirmando conceptos teóricos.
- El alumno o la alumna creará un socket servidor y un socket cliente

2.- Conceptos teóricos

El programa Socket

El programa socket ofrece un modo de acceder a la interfaz de los sockets sin tener que programar. Conecta la entrada/salida estándar (teclado/pantalla) con un socket cuyas características se especifican mediante parámetros al ejecutar la orden. Mediante la redirección de la entrada o la salida se puede enviar el contenido de un archivo o almacenar en un archivo la información recibida.

Los sockets pueden ser de dos tipos: UDP o TCP, que se corresponden con un servicio sin conexión, que no garantiza ni la entrega ni el orden de entrega de la información (UDP) y otro servicio que garantiza la entrega ordenada y sin errores de la información (TCP).

Además, se sabe que una aplicación puede comenzar iniciando la comunicación (enviando información) o bien puede esperar pacientemente hasta que la otra le solicite el inicio de la comunicación (espera petición).

El programa socket va a permitir imitar cualquiera de estas situaciones entre otras.

3.- Equipo y material necesario

3.1 Material del alumno o de la alumna:

- Imagen extensión BMP con calidad de una imagen fotográfica.

3.2 Equipo del Laboratorio:


- Programa socket (sock-1.1.tar.tar).

4.- Desarrollo:

Modo de trabajar

- La práctica se desarrollará en parejas.

117

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	118/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

4.1 Preparación del programa Sock

4.1.1 Abra la aplicación VirtualBox

NOTA: Antes de iniciar la máquina virtual verifique en la opción Red que se encuentre marcada la opción Habilitar adaptador de red->Conectado a: Adaptador puente (Figura No. 1)

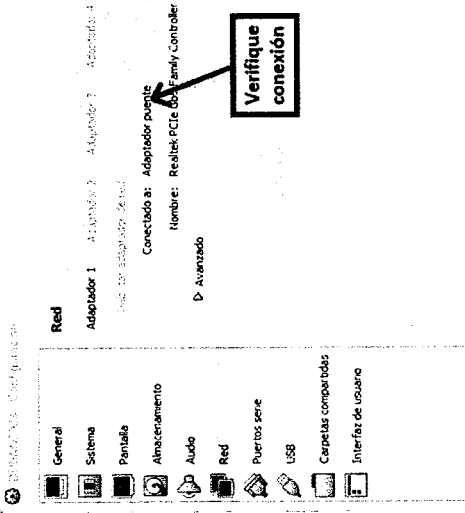


Figura No. 1. Conexión de red.

- 4.1.2 Elija la opción de cargar Linux, distribución Debian.
- 4.1.3 Inicie sesión como usuario redes.
- 4.1.4 Abra una terminal e ingrese como super usuario, para ello teclee el comando que se muestra a continuación. (Ver Figura No. 2)


	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	119/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			



Figura No. 2. Super usuario

NOTA: su significa super usuario, por lo que se emplea la misma contraseña de root

redes@debian:~\$ su

NOTA: Para realizar la práctica exitosamente debe tener instalado los paquetes ifconfig, gcc y ssh.

4.1.5 Verifique que la tarjeta de red esté debidamente configurada y que tenga asignada una dirección IP dentro del rango: 192.168.2.25-192.168.2.60. Emplee el comando ifconfig

root@debian:/home/redes# ifconfig

Anote la dirección IP 192.168.2.53


En caso de no cumplir con lo indicado en el punto 4.1.5, configure debidamente la tarjeta. Teclee:

root@debian:/home/redes# ifconfig enp0s3 192.168.2.53 netmask 255.255.255.0

NOTA: X se sustituye por una IP que se encuentre dentro del rango mencionado en el punto 4.1.5 para que esté dentro de la misma subred.

4.1.6 Verifique que la aplicación SSH se encuentre instalada (Active: active (running)) (Figura No. 3), para ello teclee:

root@debian:/home/redes# service sshd status

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	120/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La Impresión de este documento es una copia no controlada			

```
redes@DEBIAN2023:~  
root@DEBIAN2023:/home/redes# service sshd status  
sshd.service - OpenBSD Secure Shell server  
Loaded: loaded (/lib/systemd/system/sshd.service; enabled; vendor preset: en  
Active: active (running) since Thu 2023-06-15 09:32:39 CST; 6min ago  
Docs: man:sshd(8)  
       man:sshconfig(5)  
Process: 496 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
Main PID: 518 (sshd)  
Tasks: 1 (limit: 2954)  
Memory: 3.3M  
CPU: 23ms  
CGroup: /system.slice/sshd.service  
         └─518 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups  
  
Jun 15 09:32:38 DEBIAN2023 systemd[1]: Starting OpenBSD Secure Shell server...  
Jun 15 09:32:39 DEBIAN2023 sshd[518]: Server listening on 0.0.0.0 port 22.  
Jun 15 09:32:39 DEBIAN2023 sshd[518]: Server listening on :: port 22.  
Jun 15 09:32:39 DEBIAN2023 systemd[1]: Started OpenBSD Secure Shell server.  
  
root@DEBIAN2023:/home/redes#
```

Figura No. 3. Verificación de SSH

NOTA: En caso de que no se encuentre instalada, debe teclear el siguiente comando para instalarla (Figura No. 4):


```
root@debian:/home/redes# apt-get install ssh
```

```
redes@DEBIAN2023:~  
root@DEBIAN2023:/home/redes# apt-get install ssh
```

Figura No. 4. Instalación de SSH

4.1.7 Teclee los siguientes comandos para eliminar cualquier archivo existente cuyo nombre inicie con prac (Figura No. 5)

```
root@debian:/home/redes# rm -rf prac*  
root@debian:/home/redes# exit
```

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	121/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería	Área/Departamento: Laboratorio de Redes y Seguridad		
La impresión de este documento es una copia no controlada			

```
redes@DEBIAN2023:~  
root@DEBIAN2023:/home/redes# rm -rf prac*  
root@DEBIAN2023:/home/redes# exit  
  
Figura No. 5. Eliminación de archivos  
  
4.1.8 Salga de la cuenta de superusuario y emplee la cuenta de redes.  
  
4.1.9 Cree el subdirectorio practica dentro del directorio actual (Ver Figura No. 6)  
  
NOTA: Evite cambiarle el nombre al subdirectorio, deberá llamarse practica, sin ningún número posteriormente ni abreviatura alguna, nombres como prac8, p8, practica8, etcétera, serán inválidos.  
  
redes@debian:~$ mkdir practica  
  
redes@DEBIAN2023:~$ mkdir practica  
redes@DEBIAN2023:~$
```

Figura No. 6. Creación del subdirectorio practica

4.1.10 Copie el archivo *sock-1.1.tar* dentro del subdirectorio *practica*. (Ver figura No. 7)


```
redes@debian:~$ cp sock-1.1.tar /home/redes/practica
```

```
redes@DEBIAN2023:~  
redes@DEBIAN2023:~$ cp sock-1.1.tar /home/redes/practica/  
redes@DEBIAN2023:~$
```

Figura No. 7. Copia del archivo sock

4.1.11 Cámbiese al subdirectorio *practica* y descomprima el archivo *sock-1.1.tar*.tar (Ver Figura No. 8)

```
redes@debian:~$ cd practica  
redes@debian:~/practica$ tar xvf sock-1.1.tar.tar
```


	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	122/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

```
redes@DEBIAN2023:~/practica$ cd practica/
redes@DEBIAN2023:~/practica$ tar xvf sock-1.1.tar
sock-1.1/
sock-1.1/ChangeLog
sock-1.1/Makefile.in
sock-1.1/config.h.in
sock-1.1/configure
sock-1.1/configure.in
sock-1.1/install-sh
sock-1.1/sock.c
sock-1.1/README
sock-1.1/sock.1
sock-1.1/sock.lsm
sock-1.1/debian/
sock-1.1/debian/changelog
sock-1.1/debian/control
sock-1.1/debian/copyright
sock-1.1/debian/rules
```

Figura No. 8. Archivos en sock antes comprimidos.

4.1.12 Sitúese dentro del subdirectorio sock-1.1 y ejecute la orden `./configure` con la que el programa quedará preparado para su compilación y montaje. (Ver Figura No.9)

```
redes@debian:~/practica$ cd sock-1.1
redes@debian:~/practica/sock-1.1$ ./configure
```


	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	123/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

```
redes@DEBIAN2023:~/practica/sock-1.1$
redes@DEBIAN2023:~/practica/sock-1.1$ ./configure
creating cache ./config.cache
checking for gcc... gcc
checking whether the C compiler (gcc) works... yes
checking whether the C compiler (gcc) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking whether warnings should be enabled... yes
checking for a BSD compatible install... /usr/bin/install -c
checking for gethostbyname in -lresolv... yes
checking for socket in -lsocket... no
checking for gethostbyname in -lnsl... yes
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for pid_t... yes
checking return type of signal handlers... void
updating cache ./config.cache
creating Makefile
creating config.h
redes@DEBIAN2023:~/practica/sock-1.1$
```

Figura No. 9. Configuración de archivos y creación de un "Makefile"

4.1.13 Compile el programa. Ahora ya se dispone del programa sock ejecutable. (Ver figura No. 10)

```
redes@debian:~/practica/sock-1.1$ make
```

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	124/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

```
redes@DEBIAN2023: ~/practica/sock-1.1$ make
gcc -g -O2 -Wall -W -Wno-parentheses -Wstrict-prototypes -Wno-unused -lnsl
-lresolv sock.c -o sock
sock.c: In function 'main':
sock.c:461:37: warning: pointer targets in passing argument 3 of 'accept' di
ffer in signedness [-Wpointer-sign]
461 | int ns = accept(sk, sa_incoming, &i);
    |                                     ^
    |                                     |
    |                                     int *
In file included from sock.c:18:
/usr/include/x86_64-linux-gnu/sys/socket.h:233:28: note: expected 'socklen_t'
but argument is of type 'int *'
233 | socklen_t restrict addr_len);
    |               ^~~~~~
```

Figura No. 10. Compilación de archivos

4.2 Clientes TCP


4.2.1 Observe qué sucede cuando un navegador se dirige a un servidor de web y le solicita una página. En el shell teclee lo siguiente y después de pulsar la tecla "ENTER", escriba el texto GET / HTTP/1.0 Finalice presionando dos veces "ENTER" (Ver figura No. 11).

```
redes@debian:~/practica/sock-1.1$ ./sock -e www.fi-b.unam.mx:80
GET / HTTP/1.0
```

```
redes@DEBIAN2023: ~/practica/sock-1.1$ ./sock -e www.fi-b.unam.mx:80
GET / HTTP/1.0
```

Figura No. 11. Socket hacia www.fi-b.unam.mx

Con esto se está conectando al servidor www.fi-b.unam.mx (que es el servidor web de la DIE) al puerto 80, que es donde se encuentra este servicio habitualmente (well-known port) y se utiliza el protocolo TCP. Lo que se está haciendo es crear un socket en nuestra computadora. Ese socket, que actúa como cliente, lo conectamos al servidor de web de la DIE y le solicitamos que nos envíe el contenido de su página web inicial. La conexión iniciada por el programa sock se realiza al puerto 80 del servidor www.fi-b.unam.mx y dura sólo lo indispensable hasta que se entrega la página web solicitada. Es importante destacar que la respuesta del servidor contiene una información del protocolo HTTP (o cabecera) a la que

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	125/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

sigue, después de una línea en blanco, el código HTML de la página solicitada. Tras enviar esa información el servidor cierra la conexión, con lo cual la ejecución de la orden sock finaliza.

4.2.2 En la terminal teclee lo siguiente:

```
redes@debian:~/practica/sock-1.1$ ./sock :22
```

Deberá obtener como resultado algo similar a: (Ver figura No. 12).

```
redes@DEBIAN2023: ~/practica/sock-1.1$ Q  x
redes@DEBIAN2023: ~/practica/sock-1.1$ ./sock :22
SSH-2.0-OpenSSH_8.4p1 Debian-5+deb11u1
```

Figura No. 12. Socket usando el puerto 22

Observará que el programa no finaliza, para que lo haga pulse las teclas <CTRL>+<C>.

En este ejercicio se está conectando con el servidor SSH local que se está ejecutando en la misma computadora desde el que ejecuta la orden. Esto es así porque al no especificar un servidor y sólo un puerto (22) se entiende que nos referimos a la computadora local.

El servidor SSH comienza enviando una cadena que identifica la versión del programa, y eso es lo que obtenemos como resultado.

4.3 Servidor TCP


Los programas pueden esperar pacientemente a que se les solicite algo antes de enviar alguna información. Éste es el comportamiento de muchos servidores. Utilizando el programa sock va a crear un servidor cuya única función es esperar a que un cliente se conecte y luego conecta la entrada y salida estándar con ese cliente.

4.3.1 Para crear un socket servidor, teclee lo siguiente en el shell:

NOTA: **PUERTO deberá sustituirse por un número que esté dentro del rango de puertos 1024-65535.**

```
redes@debian:~/practica/sock-1.1$ ./sock -le :PUERTO
```

4.3.2 Ahora, abra un nuevo shell, sitúese en el subdirectorio sock-1.1 y ejecute la siguiente orden: (Ver figura No. 13).

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	126/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

NOTA: PUERTO deberá sustituirse por un número que esté dentro del rango de puertos 1024-65535 e igual al del punto 4.3.1

redes@debian:~/practica/sock-1.1\$./sock -e :PUERTO


	redes@DEBIAN2023:~/practica/sock-1.1\$./sock -le :PUERTO	Q	≡	x
	redes@DEBIAN2023:~/practica/sock-1.1\$./sock -e :PUERTO	Q	≡	x

Figura No. 13. Creación de un socket servidor y de un socket cliente

4.3.3 Escriba en el Shell cliente y después teclee "ENTER" observe los que sucede en el Shell servidor. Seguidamente escriba en el Shell servidor, ¿qué sucede en el Shell cliente? (Ver figura No. 14).

Por el protocolo STP nos garantizan que lleguen los mensajes, lo que se escribe en servidor o en el cliente se recibe, mientras que UDP no garantiza la llegada de los paquetes, en STP hay una comunicación cliente-servidor



	redes@DEBIAN2023:~/practica/sock-1.1\$./sock -le :PUERTO	Q	≡	x
	redes@DEBIAN2023:~/practica/sock-1.1\$./sock -e :PUERTO	Q	≡	x

Figura No. 14. Comunicación entre terminales

Salga con CTRL + C

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	127/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

La orden del punto 4.3.2 es equivalente a: telnet localhost PUERTO

El parámetro -l hace que la aplicación configure el socket en modo escucha (listen) y acepte peticiones. Por tanto, en el punto 4.3.1 ha puesto en marcha, en su computadora, un servidor que escucha en el puerto seleccionado Mientras que las órdenes de los pasos 4.3.2 y 4.3.3 han arrancado clientes TCP que se han conectado a ese puerto.

4.3.4 En un shell, sitúese en el subdirectorio sock-1.1 y cree un socket servidor tecleando lo siguiente:


NOTA: PUERTO deberá sustituirse por un número que esté dentro del rango de puertos 1024-65535.

redes@debian:~/practica/sock-1.1\$./sock -l :PUERTO -d ls

4.3.5 Ahora, en otro shell, sitúese en el subdirectorio sock-1.1 y cree un socket cliente ejecutando la orden: (Ver figura No. 15).

NOTA: PUERTO deberá sustituirse por un número que esté dentro del rango de puertos 1024-65535 e igual al del punto 4.3.4

redes@debian:~/practica/sock-1.1\$./sock -e :PUERTO

	Manual de prácticas del Laboratorio de Redes de Datos Seguras			Código:	MADO-31
				Versión:	06
				Página	128/479
				Sección ISO	8.3
				Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad			
La impresión de este documento es una copia no controlada					

redes@DEBIAN2023:~/practica/sock-1.1\$

redes@DEBIAN2023:~/practica/sock-1.1\$./sock -l :PUERTO -d 1s

redes@DEBIAN2023:~/practica/sock-1.1\$

redes@DEBIAN2023:~/practica/sock-1.1\$./sock -e :PUERTO

```
Changelog
config.cache
config.h
config.h.in
config.log
config.status
configure
configure.in
debian
install-sh
Makefile
Makefile.in
README
sock
sock.1
sock.c
sock.lsm
sock.lsm
```


redes@DEBIAN2023:~/practica/sock-1.1\$

Figura No. 15. Creación de un socket servidor y cliente

4.3.6 Observe lo que sucede.

En este experimento se ha construido un "miniservidor". Lo que hace el programa es esperar la conexión de un usuario al puerto indicado y cuando el cliente se conecta (mediante la orden sock o el programa telnet) entonces ejecuta la orden ls que lista el contenido del directorio y lo envía a través del socket. Una vez finalizada la orden ls el servidor corta la conexión del cliente telnet, pero sigue escuchando en el puerto para atender nuevas peticiones de otros clientes.

Si se sustituye la orden 'ls' por la orden 'date' en el punto 4.3.4 tendrá un miniservidor de fecha y hora.

	Manual de prácticas del Laboratorio de Redes de Datos Seguras			Código:	MADO-31
				Versión:	06
				Página	129/479
				Sección ISO	8.3
				Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad			
La impresión de este documento es una copia no controlada					

4.4 El protocolo UDP

Del mismo modo que en los ejemplos anteriores ha utilizado el protocolo TCP, ahora va a ver cómo se puede enviar información mediante el protocolo UDP. Para ello mantendrá los dos shells que tiene abiertos.

4.4.1 En un shell cree un socket servidor tecleando lo siguiente:

NOTA: **PUERTO deberá sustituirse por un número que esté dentro del rango de puertos 1024-65535.**

redes@debian:~/practica/sock-1.1\$./sock -ul :PUERTO

4.4.2 Y en otro shell ejecute la orden: (Ver Figura No. 16).

NOTA: **PUERTO deberá sustituirse por un número que esté dentro del rango de puertos 1024-65535 e igual al del punto 4.4.1**

redes@debian:~/practica/sock-1.1\$./sock -u :PUERTO


redes@DEBIAN2023:~/practica/sock-1.1\$./sock -ul :PUERTO

redes@DEBIAN2023:~/practica/sock-1.1\$

redes@DEBIAN2023:~/practica/sock-1.1\$./sock -u :PUERTO

Figura No. 16. Socket servidor y cliente.

4.4.3 Escriba en el Shell cliente y después del ENTER observe lo que sucede en el Shell servidor. (Ver figura No. 17). Realice la prueba del shell servidor hacia el cliente.

	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	136/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

redes@DEBIAN2023:~/practica/sock-1.1

```
redes@DEBIAN2023:~/practica/sock-1.1$ cd practica/sock-1.1/
redes@DEBIAN2023:~/practica/sock-1.1$ exit
logout
Connection to 192.168.2.56 closed.
redes@DEBIAN2023:~/practica/sock-1.1$
```

Figura No. 28. Cierre de la conexión por ssh.

4.5.13 Cierre sesión.

5-Questionario

- De acuerdo con lo visto en el desarrollo de la práctica ¿qué diferencias sustanciales existen entre TCP y UDP?


La transmisión de información usando TCP es mas ordenada, requiere un constante monitoreo para el correcto envío, recepción, orden y errores de info enviada. Para UDP la confirmación no se realiza el envío de información no se garantiza ni la entrega ni el orden en que se reciben las tramas

- ¿Por qué la conexión iniciada por el socket al servidor sólo dura lo necesario para recibir la información requerida?

Para optimizar el desempeño, evitar el desperdicio de recursos

- Mencione algunos ejemplos de los usos de TCP y UDP

Comunicación por texto, transferencia de archivos, acceso a pagina web, video conferencias, etc.


	Manual de prácticas del Laboratorio de Redes de Datos Seguras	Código:	MADO-31
		Versión:	06
		Página	137/479
		Sección ISO	8.3
		Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad	
La impresión de este documento es una copia no controlada			

6.- Conclusiones.

Revise los objetivos planteados al inicio de la práctica y concluya.

Tapia: Se pudo realizar la transmisión de información utilizando los protocolos UDP y TCP y las diferencias al enviar los paquetes con estos protocolos

Barrios: Se logró realizar la práctica incluso llorando más allá y enlazarlo con máquinas externas logrando obtener el funcionamiento de chat con sockets.

	Manual de prácticas del Laboratorio de Redes de Datos Seguras		Código:	MADO-31
			Versión:	06
			Página	138/479
			Sección ISO	8.3
			Fecha de emisión	11 de agosto de 2023
Facultad de Ingeniería		Área/Departamento: Laboratorio de Redes y Seguridad		
La impresión de este documento es una copia no controlada				

PRÁCTICA 8
TCP y UDP
Cuestionario Previo

1. Mencione al menos 2 funciones de la capa de transporte del Modelo OSI
2. Mencione algunos protocolos de transporte (no incluya TCP ni UDP).
3. ¿Qué es el protocolo de transporte TCP?
4. ¿Qué es el protocolo de transporte UDP?
5. Dibuje un datagrama UDP.
6. Dibuje un segmento TCP.
7. ¿Qué es un socket y qué se necesita para crearlo?
8. ¿Para qué se usa el comando apt-get install gcc o apt install gcc?
9. ¿Para qué se usa el comando apt-get install ssh o apt install ssh?
10. ¿Qué es un puerto?
11. ¿Cuáles son los rangos de puertos existentes?
12. ¿Qué rangos de puertos pueden utilizarse para establecer comunicaciones?