



# Laboratorio #2

# Protocolos de Comunicación en Redes

*Redes de Computación 2*

**Estudiante:**

Alexis H. Segales R.

**Fecha:**

8 sept 2024

---

## Index

<b>Index.....</b>	<b>2</b>
<b>Actividad #1 - Diseñar protocolo de comunicación.....</b>	<b>3</b>
Ejemplo del Protocolo Capioso.....	3
<b>Actividad #2 - Transferencia de archivos usando TCP y UDP.....</b>	<b>6</b>
TCP.....	6
UDP.....	6
<b>Actividad #3 - Configurar acceso remoto.....</b>	<b>8</b>

---

---

## Actividad #1 - Diseñar protocolo de comunicación

Utilizando estas simples características, diseñar un protocolo de comunicación para enviar y recibir mensajes:

- **Dirección:** ¿Cómo se identificarán el emisor y el receptor?

Siendo objetivo, se requiere de un punto de inicio y un punto de llegada, siendo estos la dirección IPv4 y el puerto.

También se necesitara que los usuarios tengan sus llaves pública y privada.

- **Formato de datos:** ¿Cómo se estructurará el mensaje (por ejemplo, encabezado con longitud del mensaje, contenido real del mensaje)?
  - Encabezado
    - Dirección de partida (IPv4 y puerto)
    - Dirección de destino (IPv4 y puerto)
    - Longitud del mensaje
    - Llave pública del receptor
  - Cuerpo
    - Mensaje cifrado
- **Verificación de errores:** ¿Cómo se asegurarán de que el mensaje llegue intacto (por ejemplo, agregando una suma de comprobación para verificación Checksum)?

Se adiciona un header para el checksum del mensaje ya cifrado. Y se utilizara este para validar el mensaje descifrado.

- **Confirmación:** ¿Cómo sabrá el emisor que el mensaje se recibió correctamente? ACK

Un paquete que maneja igualmente el punto de partida y de destino mas un campo de confirmación ACK.

Este protocolo tiene el nombre de protocolo **Capioso**.

### Ejemplo del Protocolo Capioso

---

---

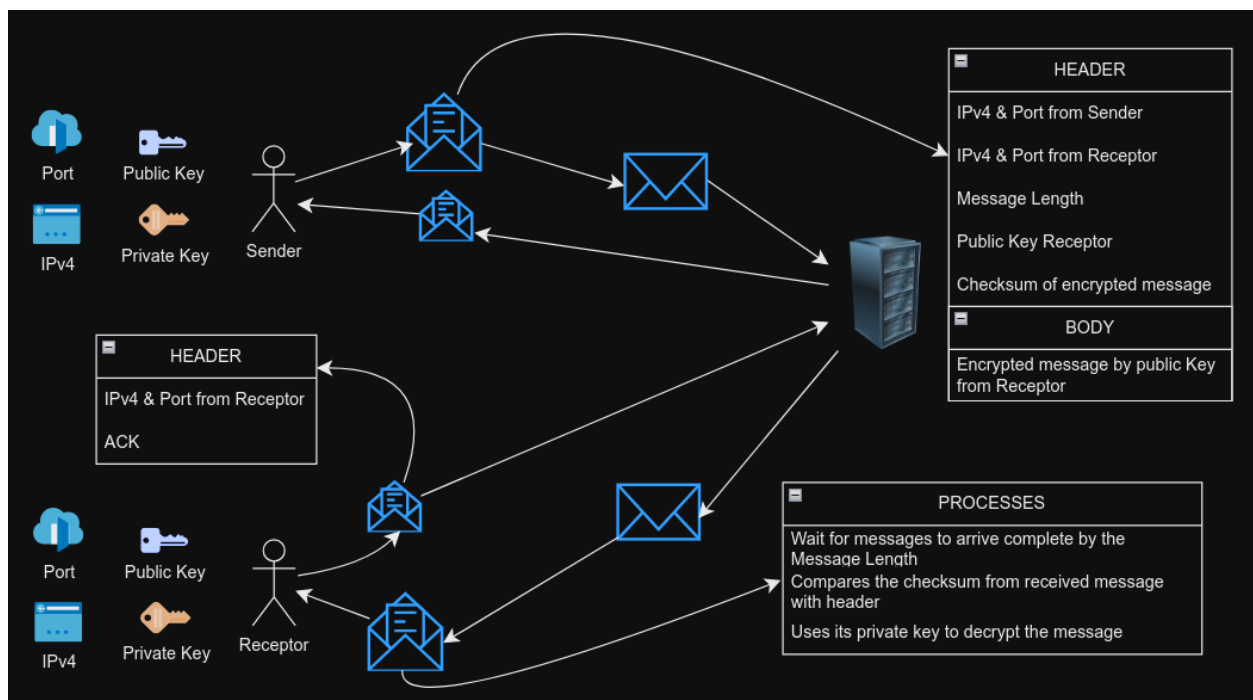
- **Preparación y Envío del Mensaje por el Dispositivo A:**

- Dirección y Puerto:
  - Emisor: Dispositivo A
  - Dirección IP: 192.168.1.2
  - Puerto: 678
  - Receptor: Dispositivo B
  - Dirección IP: 192.168.1.3
- Mensaje Original: "Hola B"
- Cifrado del Mensaje:
  - El dispositivo A cifra el mensaje "Hola B" usando la clave pública del dispositivo B.
- Generación del Paquete:
  - Header:
    - Dirección de Destino: 192.168.1.3
    - Puerto de Destino: 678
    - Longitud del Mensaje: Tamaño del mensaje cifrado en bytes
    - Checksum del Mensaje Cifrado: Calculado sobre el mensaje cifrado
  - Body:
    - Mensaje Cifrado: El mensaje "Hola B" cifrado con la clave pública del dispositivo B
- Envío: El dispositivo A envía el paquete al dispositivo B.

- **Recepción y Verificación del Mensaje por el Dispositivo B:**

- Recepción del Paquete:
    - El dispositivo B recibe el paquete del dispositivo A.
  - Verificación:
    - Checksum: El dispositivo B calcula el checksum del mensaje cifrado recibido y lo compara con el checksum incluido en el header del paquete.
    - Si coinciden:
      - El mensaje cifrado no ha sido alterado durante la transmisión.
-

- El dispositivo B descifra el mensaje cifrado usando su clave privada para obtener "Hola B".
- Confirmación (ACK):
  - El dispositivo B envía un paquete de confirmación (ACK) al dispositivo A indicando que el mensaje fue recibido correctamente.
  - Header del ACK:
    - Dirección de Destino: 192.168.1.2
    - Puerto de Destino: 678
    - ACK: Indica la recepción exitosa del mensaje.
- **Recepción del ACK por el Dispositivo A:**
  - Recepción del Paquete ACK:
    - El dispositivo A recibe el ACK del dispositivo B.
  - Confirmación:
    - El dispositivo A confirma que el mensaje fue recibido correctamente y puede proceder con cualquier acción subsiguiente.



---

## Actividad #2 - Transferencia de archivos usando TCP y UDP

### TCP

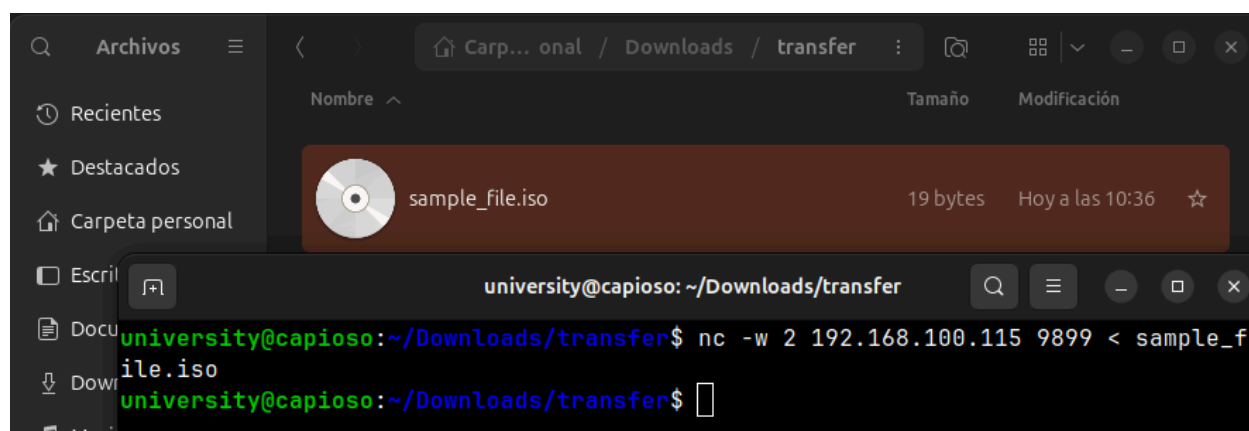
Configuración en Windows (Receptor)

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ahseg>ncat -l 9899 > sample_file.iso

C:\Users\ahseg>
```

Configuración en Linux (Emisor)




### UDP


La primera comparacion que puedo hacer es que netcat no requiere de una aclaración o versionamiento entre servidor y cliente.

De hecho netcat solo se activa como un escucha si es un cliente, y si trabaja como servidor solo se necesita de un simple comando.

---





[Home](#)[Compare Versions](#)



Catapult Clients


CentOS 6 & 7






Catapult Server


CentOS 6 & 7






Catapult Clients


Ubuntu 16.04 LTS






Catapult Server


Ubuntu 16.04 LTS






Catapult Control


Linux (Docker)






Catapult Clients & Server


Windows 64-bit







Catapult Client Only

Windows 64-bit





Catapult Control



## Actividad #3 - Configurar acceso remoto

Preferí crear un VM con Linux Server, que viene con SSH:

```

Ubuntu Server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
capibara@capiososerver:~$ sudo systemctl status ssh
[sudo] password for capibara:
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: inactive (dead)
TriggeredBy: ● ssh.socket
             Docs: man:sshd(8)
                  man:sshd_config(5)
capibara@capiososerver:~$ sudo systemctl start ssh
capibara@capiososerver:~$ sudo systemctl status ssh
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Sun 2024-09-08 14:59:55 UTC; 1s ago
TriggeredBy: ● ssh.socket
             Docs: man:sshd(8)
                  man:sshd_config(5)
   Process: 1270 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1271 (sshd)
    Tasks: 1 (limit: 2276)
   Memory: 2.0M (peak: 2.2M)
      CPU: 16ms
   CGroup: /system.slice/ssh.service
           └─1271 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 08 14:59:55 capiososerver systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Sep 08 14:59:55 capiososerver sshd[1271]: Server listening on :: port 22.
Sep 08 14:59:55 capiososerver systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
capibara@capiososerver:~$ _

```

Entonces luego de asegurarme que el servicio este activo, tenia que revisar el firewall.

```

Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
capibara@capiososerver:~$ sudo ufw status
Status: inactive
capibara@capiososerver:~$ sudo ufw enable
Firewall is active and enabled on system startup
capibara@capiososerver:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
capibara@capiososerver:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)

capibara@capiososerver:~$

```



Información en el servidor Linux:

```
Ubuntu Server (Fresh Install)
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

capibara@capiososerver:~$ ls -la
total 32
drwxr-x--- 4 capibara capibara 4096 Sep  8 15:04 .
drwxr-xr-x 3 root     root     4096 Sep  8 14:58 ..
-rw----- 1 capibara capibara 227 Sep  8 15:04 .bash_history
-rw-r--r-- 1 capibara capibara 220 Mar 31 08:41 .bash_logout
-rw-r--r-- 1 capibara capibara 3771 Mar 31 08:41 .bashrc
drwx----- 2 capibara capibara 4096 Sep  8 14:59 .cache
-rw-r--r-- 1 capibara capibara 807 Mar 31 08:41 .profile
drwx----- 2 capibara capibara 4096 Sep  8 14:58 .ssh
-rw-r--r-- 1 capibara capibara  0 Sep  8 14:59 .sudo_as_admin_successful
capibara@capiososerver:~$ _
```

Información tras ejecutar ssh desde Windows:

```
C:\Users\ahseg>ssh capibara@192.168.100.175
The authenticity of host '192.168.100.175 (192.168.100.175)' can't be established.
ECDSA key fingerprint is SHA256:SyQJB/KLwgTBAwvBR+2Lhjh7648QKcEbMjff18ptnDt4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.175' (ECDSA) to the list of known hosts.
capibara@192.168.100.175's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Sun Sep  8 03:56:30 PM UTC 2024

System load:            0.0
Usage of /:              46.2% of 8.19GB
Memory usage:           14%
Swap usage:              0%
Processes:              117
Users logged in:         1
IPv4 address for enp0s3: 10.0.2.15
IPv4 address for enp0s8: 192.168.100.175
IPv6 address for enp0s8: 2800:cd0:ad11:ef00:a00:27ff:fe50:9a02

Expanded Security Maintenance for Applications is not enabled.

18 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

capibara@capiososerver:~$ _
```

```
capibara@capiososerver: ~  
capibara@capiososerver:~$ ls -la  
total 32  
drwxr-x--- 4 capibara capibara 4096 Sep  8 15:04 .  
drwxr-xr-x 3 root      root      4096 Sep  8 14:58 ..  
-rw----- 1 capibara capibara  227 Sep  8 15:04 .bash_history  
-rw-r--r-- 1 capibara capibara  220 Mar 31 08:41 .bash_logout  
-rw-r--r-- 1 capibara capibara 3771 Mar 31 08:41 .bashrc  
drwx----- 2 capibara capibara 4096 Sep  8 14:59 .cache  
-rw-r--r-- 1 capibara capibara  807 Mar 31 08:41 .profile  
drwx----- 2 capibara capibara 4096 Sep  8 14:58 .ssh  
-rw-r--r-- 1 capibara capibara    0 Sep  8 14:59 .sudo_as_admin_successful  
capibara@capiososerver:~$
```

Como dato adicional, intente conectarme usando IPv6 sin embargo no dio ningún resultado, si funciona con IPv4