



Instituto Tecnológico de Buenos Aires

Informe POP3 Server

Protocolos de Comunicación - 1C 2023

Integrantes:

Tobias Perry 62064

Manuel Esteban Dithurbide 62057

Tomas Camilo Gay Bare 62103

Martin Hecht 62041

Abstracto

Este informe describe el desarrollo del TPE de la materia Protocolos de Comunicación - 72.07 - ITBA. El mismo consiste de un protocolo POP3 acompañado de un protocolo de monitoreo para recolectar métricas y permitir la configuración en vivo del servidor. Se tomó la decisión de implementar el servidor de manera no bloqueante y sin transformación de mensajes.

Indice:

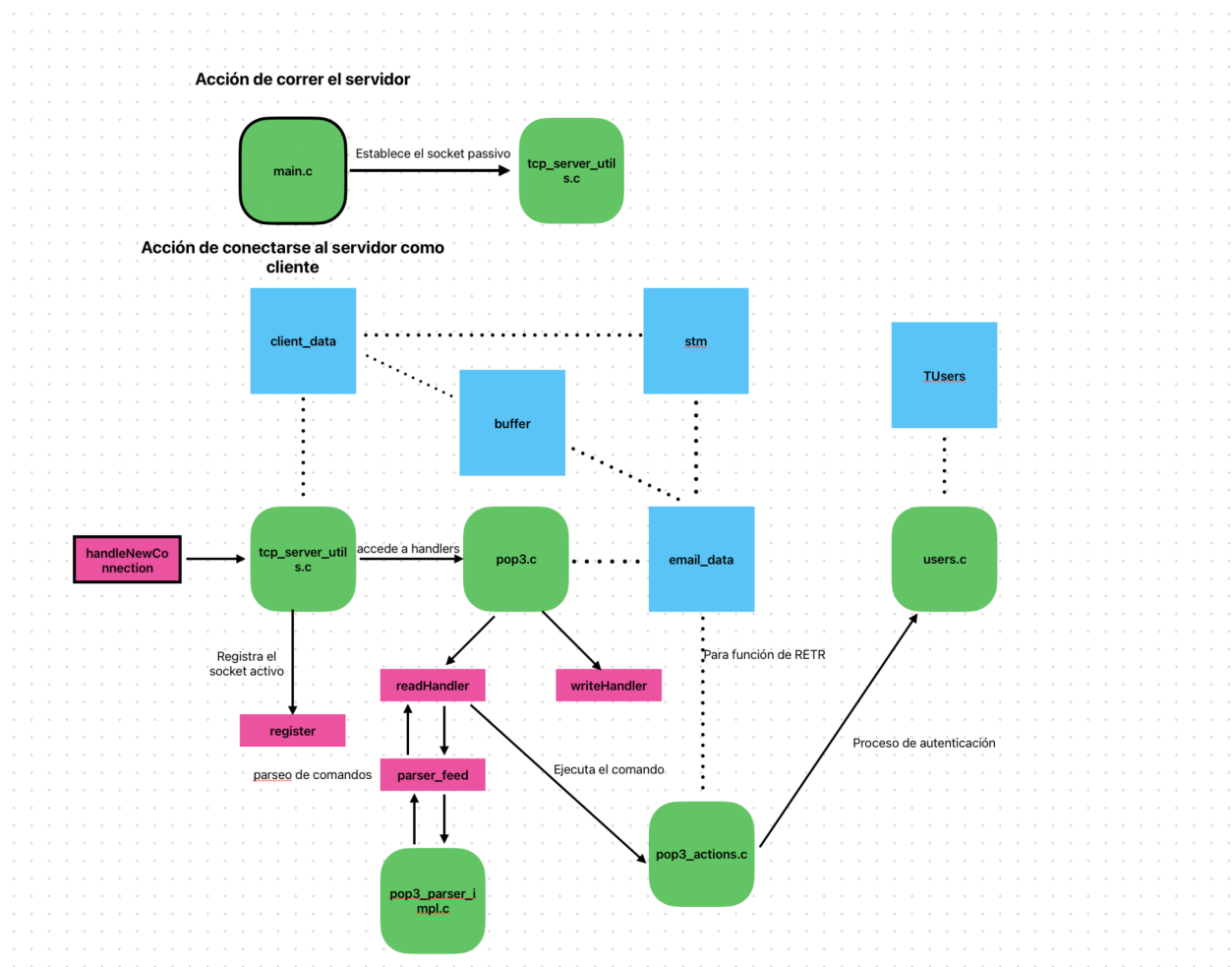
Descripción de los protocolos y aplicaciones	4
Implementación del protocolo POP3	4
Generalidades	4
Ejecución	4
Comandos disponibles	4
Ejemplos de prueba	5
Test de buffer	5
Test de bloqueo e integridad	6
Test de carga	7
Test de stty	8
Utilización de MUAs	9
Test de conexión IPv4 e IPv6	11
Protocolo de monitoreo (MGMT)	12
Generalidades	12
1 - La fase de AUTH_MGMT	12
USER msj	12
PASS msj	13
QUIT msj	14
2 - La fase de ACTIVE_MGMT	14
NOOP msj	14
HISTORIC_CONNEC msj	15
LIVE_CONNEC msj	15
BYTES_TRANS msj	16
USERS msj	16
STATUS msj	17
MAX_CONNEC msj	17
MAX_USERS msj	18
TIMEOUT msj	19
DELETE_USER msj	19
ADD_USER msj	20
RESET_USER_PASSWORD msj	20
CHANGE_PASSWORD msj	21
QUIT msj	21
CAPA msj	22
Aplicación de monitoreo	23
Generalidades	23
Ejecución	23
Comandos disponibles	23
Ejemplos de uso	24

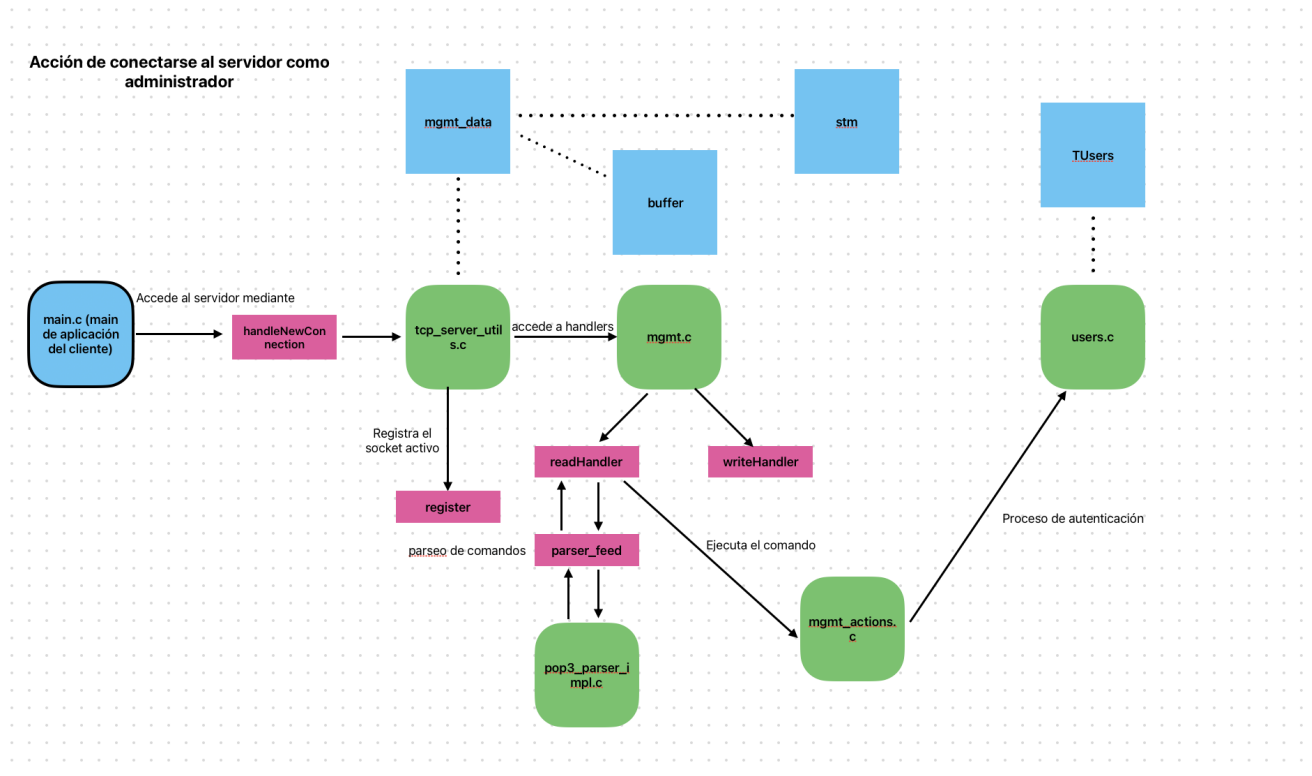
Conexiones históricas	24
Conexiones activas	24
Bytes transferidos	24
Users	24
Status	25
Máxima cantidad de conexiones	25
Máxima cantidad de usuarios	25
Ejecutar un timeout	26
Eliminar un usuario	26
Agregar un usuario	26
Resetear un usuario	26
Cambiar la contraseña del administrador	27
Guía de instalacion	27
Problemas encontrados durante el diseño y la implementación	27
Posibles Extensiones	28
Conclusiones	28

Descripción de los protocolos y aplicaciones

El proyecto cuenta con tres partes. La implementación del protocolo POP3 [RFC 1939], la implementación de un protocolo propio (MGMT) y la implementación de una aplicación para el usuario de MGMT. El sistema acepta hasta 509 conexiones simultáneas, puesto que la máxima cantidad de fds que soportamos es de hasta 1023, y contemplamos el hecho de que cada usuario le pida al mismo tiempo al servidor un mail, en cuyo caso se abrirían dos file descriptors por usuarios. $509 * 2 = 1018$ (nos dejamos un margen en caso de que en futuras implementaciones querramos abrir más sockets).

La siguiente imagen muestra gráficamente el flujo de las acciones más comunes del sistema. Las cajas azules son estructuras, las verdes son archivos, y las rosas son funciones. Las líneas punteadas indican que un archivo utiliza una estructura, y las flechas indican una acción hacia un archivo o hacia una función. Las cajas con borde son los puntos de entrada de las acciones enunciadas.





Implementación del protocolo POP3

Generalidades

Para este trabajo realizamos una implementación del servidor no bloqueante sin mecanismo de transformación de mensajes. En esta sección vamos a mostrar distintos casos de uso del servidor.

En el marco de la materia, entendemos que los usuarios que recibimos al ejecutar el servidor existen previamente y que el servidor cuenta con mails para los mismos. De este modo, el uso esperado es inicializar el servidor, conectarse con un usuario y poder aprovechar todas sus funcionalidades del servidor.

Ejecución

La aplicación deberá recibir la ip y el puerto que se desea usar para levantar el servidor; y la opción -u seguido del registro de usuarios y contraseñas que deben existir en el servidor. Se le pueden indicar al servidor todos los usuarios con sus respectivas contraseñas que se deseen. El comando queda con este formato:

```
./run <ip> <port> -u <user1>:<password1> <user2>:<password2> ...
```

Si en lugar de la ip se escribe localhost, entonces se usara la ip del host local en formato ipv6, es decir, "::FFFF:127.0.0.1".

Comandos disponibles

```
USER <username>
PASS <password>
CAPA
NOOP
LIST
STAT
DELE <email number>
RETR <email number>
RSET
QUIT
```

Ejemplos de prueba

Test de buffer

Para definir el tamaño óptimo del tamaño de los buffers a utilizar realizamos distintas pruebas en Pampero. Luego de recolectar los datos como puede verse en la siguiente figura decidimos que la mejor opción en relación a memoria-tiempo seleccionar un buffer de 4KB. La decisión se basó en que llegado un punto la diferencia de tiempo no era significativa, pero sí lo era así el tamaño del buffer. El método que utilizamos fue pedirle al servidor que devuelva un mail de 1GB y observamos el tiempo que tardó.

Tiempo (segundos) vs. Buffer (Bytes)

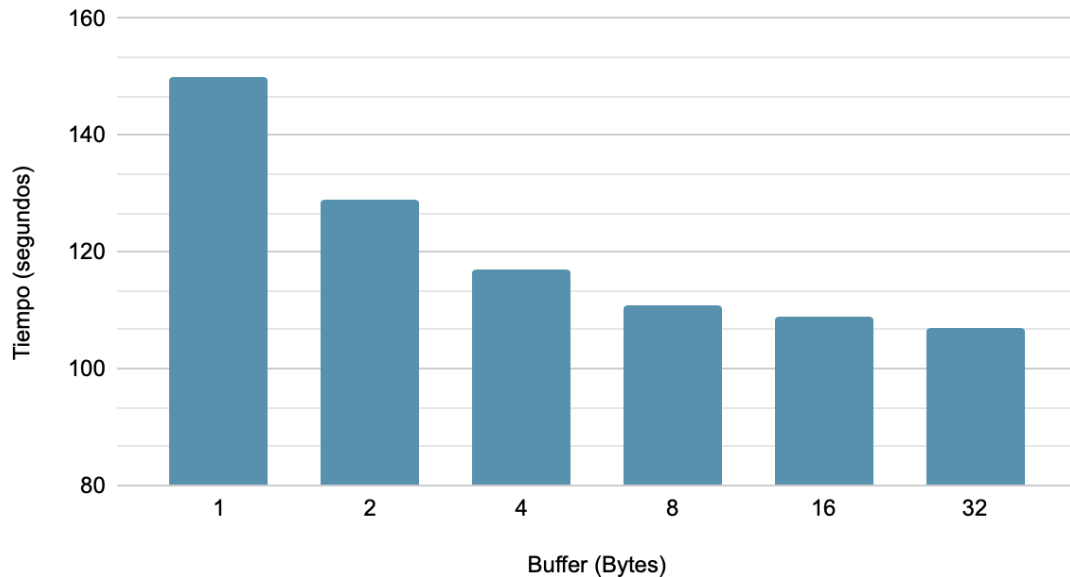


Figura 1.0: Tabla de tamaño de buffer vs tiempo de descarga

Como puede observarse en el gráfico, hicimos pruebas con 1, 2, 4, 8, 16 y 32 KB y optamos por el de 4KB.

Test de bloqueo e integridad

Para verificar el correcto funcionamiento de la aspecto no bloqueante del servidor corrimos el comando “top” en una terminal y CURL en otra. Con ello verificamos que el servidor utiliza el 100% del CPU. Imprimimos la salida en un archivo output para que la salida no dependa de la velocidad con la que se imprime en salida estándar.

La imagen muestra el servidor en la primera terminal, el top en la segunda, y el curl en la tercera, en la cual se puede ver también el tiempo que tomó. Este test se corrió con los flags fsanitize y sin optimizaciones, y es por eso que el tiempo que tardó es distinto que el expuesto en otras partes del informe.

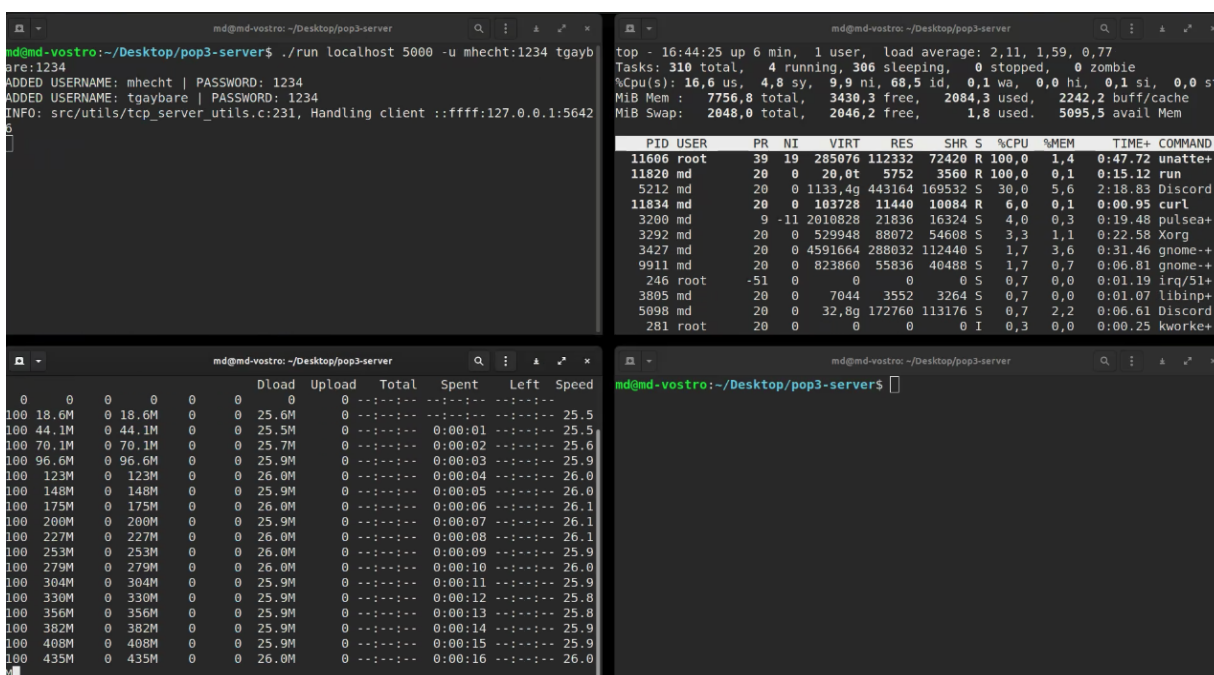


Figura 1.1: Imagen de uso de CPU del servidor en descarga

Luego corrimos el comando “diff” entre el mail que trajo el RETR de nuestro servidor y el archivo original y comprobamos que eran iguales, con la salvedad del enter que agregamos nosotros al finalizar el comando.

```
md@md-vostro: ~/Desktop/pop3-server
md@md-vostro:~/Desktop/pop3-server$ diff output src/mail/mhecht/cur/file2
7185966d7185965
<
md@md-vostro:~/Desktop/pop3-server$ tail output
utyNDWd+60g2q0voGNK7oRD3vAzGwN+j1Wxs03oTI/w2Y4cLubEUNIZ0vRcrY9sIuWXf+xroQtZ
EuNIQf8M4hrCKx1/dSPunw97yioyT6YWo/azoSq5v+MzjoGYHSTb96fj8gvfK7KRv6Jt4pxsk/4f
1Ezn8RIHm870Ht+gzx2g9zVMSqx7fG62oMrF90WpbI/VbIqdWSWZAJ1YIwe8D0w3FLHU87PsnBfD
hI2uDx9zbCnS452gJgKLrH7VrORigXxLFZjmD0KUupfAYNRRG34i9dk/k884Y2v7en1+PBmtQoT/
46/cYRoFWZ+VL+PDMdSiFb89bz1b6cN3v19l9RfBDIVbH1EBG4BBINQTVk4EdiGEzmXP5KbYlrcS
9sIo26fSFapsQ6zKad3N8cqNzYwkLlviA5us0ULf2QgbeXo8PQIdmxsc/XRLTifMuK2KqcYDVEfN
j5ALg/tVztiliHE+yBKVfSH1F2sC9FifU7if09YTA4YQ5navXQMbyVW0Bv0+ufI78TzLsb7vxi2s
iQKkRkqb4oMFMqsyGoN3AVSaRm5lx5Vy0zkogDrEW6iRq1/l1ibd6pugwd25GyoJc6tl+pwJNwb7
rUcuIO26w45AR23GKJq6uf18yF5A83MUjofnPGWo5MnFnMRUALHZVceDQLw1iDnd+T5XQ==
md@md-vostro:~/Desktop/pop3-server$ tail src/mail/mhecht/cur/file2
JYKVt95CbZBfPBwUvdQqpalJEGCyQ5L3zW0hjEKRoxiUEQs+wYmIQVRQeqeoUjACfaleptus4rTX
utyNDWd+60g2q0voGNK7oRD3vAzGwN+j1Wxs03oTI/w2Y4cLubEUNIZ0vRcrY9sIuWXf+xroQtZ
EuNIQf8M4hrCKx1/dSPunw97yioyT6YWo/azoSq5v+MzjoGYHSTb96fj8gvfK7KRv6Jt4pxsk/4f
1Ezn8RIHm870Ht+gzx2g9zVMSqx7fG62oMrF90WpbI/VbIqdWSWZAJ1YIwe8D0w3FLHU87PsnBfD
hI2uDx9zbCnS452gJgKLrH7VrORigXxLFZjmD0KUupfAYNRRG34i9dk/k884Y2v7en1+PBmtQoT/
46/cYRoFWZ+VL+PDMdSiFb89bz1b6cN3v19l9RfBDIVbH1EBG4BBINQTVk4EdiGEzmXP5KbYlrcS
9sIo26fSFapsQ6zKad3N8cqNzYwkLlviA5us0ULf2QgbeXo8PQIdmxsc/XRLTifMuK2KqcYDVEfN
j5ALg/tVztiliHE+yBKVfSH1F2sC9FifU7if09YTA4YQ5navXQMbyVW0Bv0+ufI78TzLsb7vxi2s
iQKkRkqb4oMFMqsyGoN3AVSaRm5lx5Vy0zkogDrEW6iRq1/l1ibd6pugwd25GyoJc6tl+pwJNwb7
rUcuIO26w45AR23GKJq6uf18yF5A83MUjofnPGWo5MnFnMRUALHZVceDQLw1iDnd+T5XQ==
md@md-vostro:~/Desktop/pop3-server$
```

Figura 1.2: Diferencias entre el archivo original y el descargado por el servidor

Test de carga

Para evaluar la performance del servidor realizamos algunos tests. Para el primero creamos un mail de 1GB y corrimos en distintas terminales varios usuarios para ver como variaba el tiempo de devolución del mail entero a medida que aumentamos los usuarios. Se muestran los datos de test abajo, donde se testean hasta diez usuarios concurrentes.

Tiempo de descarga de un mail de 1GB

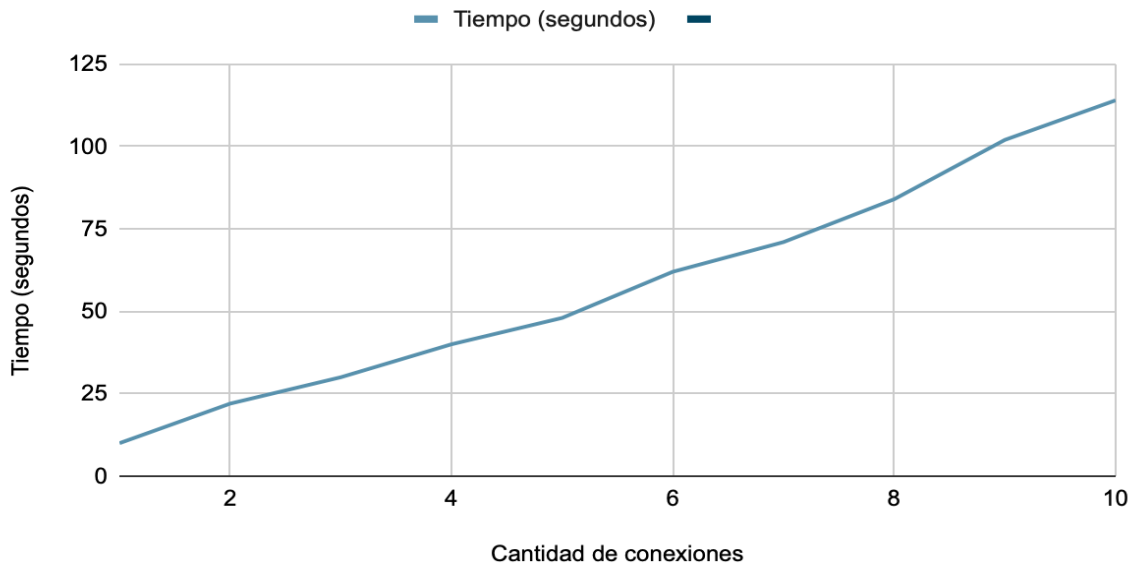


Figura 1.3: Tiempo de descarga vs cantidad de descargas en simultáneo

Como puede verse en el gráfico, el tiempo de respuesta es lineal para los primeros 10 usuarios.

Intentamos usar JMeter para poder realizar un test de carga más completo pero no lo logramos, así que lo tomamos como un punto para testear en una futura implementación.

De todos modos, nos aseguramos de poder correr 500 usuarios con un test de carga más simple con un “for” que abra 509 conexiones al servidor. Luego, realizando el test con 1000 conexiones, el archivo de logs mostró que la mitad de las conexiones no pudieron realizarse porque se había alcanzado la máxima cantidad de file descriptors dispuesta.

Test de stty

Realizamos un test donde deshabilitamos el formato canónico del sistema operativo para que el servidor reciba caracter a caracter lo que le enviemos y corroborar que funcione. Realizamos este test con netcat y con CURL para probar si funciona también y los resultados fueron positivos. En las siguientes capturas de pantalla puede verse cómo en ambos casos se corren los comandos y el funcionamiento es el esperado.

```
tobi@DESKTOP-DM35EIE:/mnt/c/Users/tobia/OneDrive/Documents/GitHub/pop3-server$ stty -icanon && nc -C localhost 5000
+OK POP3 server ready v1.0
USER mhecht
+OK USER
PASS wrongpass
-ERR
PASS pass
+OK PASS
RETR 1
+OK 17 octets
este es el mail 1
.

tobi@DESKTOP-DM35EIE:/mnt/c/Users/tobia/OneDrive/Documents/GitHub/pop3-server$ stty -icanon && curl pop3://mhecht:pass@localhost:5000/1
este es el mail 1
```

Figura 1.4: Test de stty en consola

Utilización de MUAs

Nos conectamos a Mozilla Thunderbird para probar nuestra implementación del protocolo. Fue muy útil porque nos ayudó a seguir mejor los lineamientos del RFC 1939. Si bien fue difícil finalmente logramos realizar la conexión.

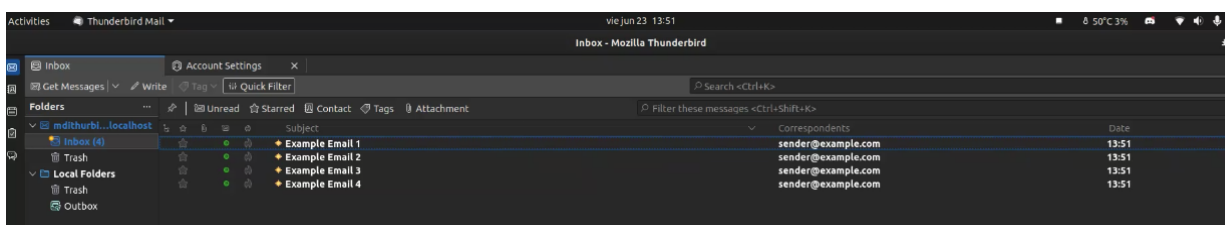


Figura 1.5: Casilla de mail de Thunderbird

En la imagen superior se ve como Thunderbird nos permite visualizar nuestros emails. En la siguiente imagen se puede ver el log del servidor cuando

Thunderbird se conecta. Se puede ver cómo corre los comandos CAPA, STAT, LIST, RETR, DELE y QUIT, y como intenta correr UIDL, que es un comando que no implementamos, pero al ser opcional para el protocolo, no lo requiere.

```
INFO: Concurrent Connections: 1
INFO: New connection from socket 4
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command CAPA executed
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command USER executed
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command PASS executed
INFO: Socket 4 - mdithurbide logged in
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command STAT executed
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command LIST executed
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command UIDL not found
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command RETR executed
INFO: Socket 5 - finished reading email
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command DELE executed
INFO: Socket 4 - email 1 deleted
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command RETR executed
INFO: Socket 6 - finished reading email
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command DELE executed
INFO: Socket 4 - email 2 deleted
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command RETR executed
INFO: Socket 7 - finished reading email
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command DELE executed
INFO: Socket 4 - email 3 deleted
INFO: Socket 4 - reading
INFO: Socket 4 - analyzing command
INFO: Socket 4 - command QUIT executed
INFO: Concurrent Connections: 0
INFO: Socket 4 - deleting emails
INFO: Socket 4 - unregistering file descriptor
INFO: Socket 4 - freeing memory
```

Figura 1.6: Logs del servidor luego de usar Thunderbird

Test de conexión IPv4 e IPv6

Para verificar el soporte de conexión para IPv4 y IPv6 abrimos el socket pasivo del servidor con direcciones IPv4 e IPv6 respectivamente sin problemas. Luego para verificar que el cliente pueda conectarse de ambos

modos corrimos el comando netcat con los flags -4 y -6 para forzarlo a correr en el formato IP que requeríamos.

Protocolo de monitoreo (MGMT)

Generalidades

- Para agregarle utilidad al protocolo pop3, se acompaña con un protocolo de monitoreo (mgmt), que tiene como objetivo la recolección de datos y la configuración del servidor.
- El protocolo está implementado sobre TCP con autenticación de usuarios.
- Utiliza la misma estructura de usuarios que el pop3 pero validando con usuario con privilegios de administrador.
- El protocolo recolecta los siguientes datos:
 - Conexiones activa
 - Conexiones históricas
 - Manejo de usuarios (ABM)
 - Bytes enviados\recibidos
- El protocolo permite configurar los siguientes parámetros:
 - Máxima cantidad de usuarios
 - Máxima cantidad de conexiones
 - Timeout
- Por default corre en el puerto 6000

1 - La fase de AUTH_MGMT

USER msj

Argumentos:

- un nombre de usuario (requerido) que hace referencia al administrador.

Restricciones:

- Solo puede darse en la fase de AUTH_MGMT después de que el servidor envía el saludo inicial.

Comentario:

- Para que el usuario pueda autenticarse primero debe enviar el nombre de usuario con el comando USER. En caso de recibir respuesta afirmativa del servidor, debe enviar la contraseña con el comando PASS.
- En caso de recibir respuesta de error (-ERR) el usuario puede enviar nuevamente el usuario o cerrar la conexión con QUIT.

Respuestas posibles:

+OK USER
-ERR

Ejemplos:

```
C: USER admin
S: +OK USER

C: USER notAdmin
S: -ERR
```

PASS msj

Argumentos:

- Una contraseña (requerida) que hace referencia a la contraseña del nombre de usuario previamente enviado.

Restricciones:

- Solo puede darse en la fase de AUTH_MGMT después de el comando USER con respuesta afirmativa.

Comentario:

- Una vez que el cliente envió el comando USER, el servidor guarda dicho usuario hasta la invocación de PASS. Luego de que se envía la contraseña se validan las credenciales.

Respuestas posibles:

+OK PASS
-ERR

Ejemplos:

```
C: USER admin
S: +OK USER
```

```
C: PASS admin
S: +OK PASS

C: USER admin
S: +OK USER
C: PASS 1234
S: -ERR
```

QUIT msj

Argumentos:

- Ninguno.

Restricciones:

- Ninguna

Respuestas posibles:

+OK

Ejemplo:

```
C: QUIT
S: +OK GOODBYE
```

2 - La fase de ACTIVE_MGMT

NOOP msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El comando representa (NO-Operation). El servidor simplemente devuelve una respuesta afirmativa.

Respuestas posibles:

+OK

Ejemplo:

```
C: NOOP
S: +OK NOOP
```

HISTORIC_CONNEC msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El servidor responde una respuesta afirmativa acompañada de la cantidad de conexiones históricas (totales) que tuvo el servidor.

Respuestas posibles:

+OK

Ejemplo:

```
C: HISTORIC_CONNEC
S: +OK HISTORIC_CONNEC
   HISTORIC CONNECTIONS: 2
```

LIVE_CONNEC msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El servidor responde una respuesta afirmativa acompañada de la cantidad de conexiones activas que tiene el servidor en el momento.

Respuestas posibles:

+OK

Ejemplo:


```
C: LIVE_CONNEC
S: +OK LIVE_CONENC
   LIVE CONNECTIONS: 1
```

BYTES_TRANS msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El servidor responde una respuesta afirmativa acompañada de la cantidad de bytes recibidos al igual que la cantidad de bytes transferidos.

Respuestas posibles:

+OK

Ejemplo:

```
C: BYTES_TRANS
S: +OK BYTES_TRANS
   BYTES SENT = 133
   BYTES RECEIVED = 97
```

USERS msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El servidor responde una respuesta afirmativa acompañada con la cantidad de usuarios y lista de usuarios registrados en el servidor. El usuario admin es considerado un usuario más pero con privilegios de administrador que le permiten ingresar al MGMT.

Respuestas posibles:

+OK

Ejemplo:

```
C: USERS
S: +OK USERS
   USER COUNT: 2
   USER LIST:
     admin
     user1
```

STATUS msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El comando status es un comando que combina todos los anteriores de AUTH_MGMT mostrando todas las estadísticas del servidor y las configuraciones.

Respuestas posibles:

+OK

Ejemplo:

```
C: STATUS
S: +OK STATUS
   BYTES SENT = 133
   BYTES RECEIVED = 97
   LIVE CONNECTIONS: 1
   HISTORIC CONNECTIONS: 1

   MAX USERS: 2
   MAX CONNECTIONS: 5
```

MAX_CONNEC msj

Argumentos:

- La nueva cantidad de conexiones (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El comando configura la cantidad máxima de conexiones. En caso de que la cantidad de conexiones ingresada sea invalida el servidor devuelve un mensaje de error.

Respuestas posibles:

- +OK
- ERR

Ejemplos:

```
C: MAX_CONNEC 5
S: +OK MAX_CONENC CHANGED

C: MAX_CONNEC -2
S: -ERR
```

MAX_USERS msj

Argumentos:

- La nueva cantidad de usuarios máximos (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El comando configura la cantidad máxima de usuarios. Como los cambios se ejecutan en vivo, la nueva cantidad de usuarios ingresada tiene que ser mayor o igual a la cantidad de usuarios que existen actualmente en el servidor.

Respuestas posibles:

- +OK
- ERR

Ejemplos:

```
C: MAX_USERS 5
S: +OK MAX_USERS
    MAX USERS CHANGED SUCCESSFULLY
```

```
C: MAX_USERS -5
S: -ERR
```

TIMEOUT msj

Argumentos:

- La cantidad de tiempo del timeout (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- Debido a que el comando duerme el servidor una cantidad de tiempo determinada por el usuario, el servidor envía su respuesta afirmativa luego de que transcurra el lapso de tiempo.
- Durante el timeout el servidor queda inhabilitado completamente sin funcionalidad alguna.

Respuestas posibles:

```
+OK
-ERR
```

Ejemplos:

```
C: TIMEOUT 5
S: +OK TIMEOUT COMPLETED

C: TIMEOUT -1
S: -ERR
```

DELETE_USER msj

Argumentos:

- El nombre de usuario del que se desea eliminar (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- Como el servidor tiene simplemente 1 solo administrador (nombre de usuario: admin) este mismo no puede ser eliminado.

Respuestas posibles:

+OK
-ERR

Ejemplos:

```
C: DELETE_USER user1
S: +OK DELETE_USER
   USER DELETED SUCCESSFULLY

C: DELETE_USER admin
S: -ERR ADMIN CANNOT BE DELETED

C: DELETE_USER aaaaaa
S: -ERR USER NOT FOUND
```

ADD_USER msj

Argumentos:

- El nombre de usuario del que se desea agregar (requerido) y la contraseña (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Respuestas posibles:

+OK

Ejemplos:

```
C: ADD_USER user password
S: +OK ADD_USER
   USER ADDED SUCCESSFULLY

C: ADD_USER admin admin
S: -ERR USER ALREADY EXISTS

C: ADD_USER user1 pass1
S: -ERR MAX USERS REACHED
```

RESET_USER_PASSWORD msj

Argumentos:

- El nombre de usuario del que se desea resetear (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El comando permite resetear la contraseña de un usuario. La nueva contraseña es equivalente al nombre del usuario.

Respuestas posibles:

+OK

Ejemplos:

```
C: RESET_USER_PASSWORD user1
S: +OK RESET_USER_PASSWORD
  PASSWORD RESET SUCCESSFULLY

C: RESET_USER_PASSWORD aaaa
S: -ERR USER NOT FOUND
```

CHANGE_PASSWORD msj

Argumentos:

- El nombre de usuario del que se desea resetear (requerido).

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Comentario:

- El comando permite resetear la contraseña de un usuario. La nueva contraseña es equivalente al nombre del usuario.

Respuestas posibles:

+OK

Ejemplos:

```
C: RESET_USER_PASSWORD user1
S: +OK RESET_USER_PASSWORD
  PASSWORD RESET SUCCESSFULLY

C: RESET_USER_PASSWORD aaaa
S: -ERR USER NOT FOUND
```

QUIT msj

Argumentos:

- Ninguno.

Restricciones:

- Ninguna

Respuestas posibles:

+OK

Ejemplo:

```
C: QUIT
S: +OK GOODBYE
```

CAPA msj

Argumentos:

- Ninguno.

Restricciones:

- Solo puede darse en la fase de ACTIVE_MGMT.

Respuestas posibles:

+OK

Ejemplo:

```
C: CAPA
S: +OK CAPA
   NOOP
   QUIT
   HISTORIC_CONEC
   LIVE_CONEC
   BYTES_TRANS
   USERS
   STATUS
   MAX_USERS <int>
   DELETE_USER <username>
   ADD_USER <username> <password>
   RESET_USER_PASSWORD <username>
   CHANGE_PASSWORD <oldPassword> <newPassword>
```

Aplicación de monitoreo

Generalidades

- La aplicación de monitoreo crea un socket para conectarse con el protocolo de monitoreo (mgmt) a través de TCP. Luego la aplicación envía los comandos que desee el usuario e imprime el resultado que le devuelve el servidor.
- La aplicación se ejecuta desde la línea de comandos.

Ejecución

La aplicación deberá recibir la versión del protocolo, por si hay futuras implementaciones; el usuario y su contraseña, puesto que la información que muestra el protocolo es sensible para los administradores del servidor; y por último el comando que se desea ejecutar del siguiente modo:

```
./admin_monitor <version> <username> <password> <command>
```

Comandos disponibles

```
-HELP  
-HISTORIC_CONNEC  
-LIVE_CONNEC  
-BYTES_TRANS  
-USERS  
-STATUS  
-MAX_USERS <maxUsers>  
-MAX_CONNECTIONS <maxConnections>  
-TIMEOUT <seconds>  
-DELETE_USER <username>  
-ADD_USER <username> <password>  
-RESET_USER_PASSWORD <username>  
-CHANGE_PASSWORD <oldPassword> <newPassword>
```

Ejemplos de uso

La siguiente sección muestra distintos casos de uso implementados por el servidor. Los siguientes cuadros muestran extractos de código implementados al utilizar nuestra aplicación del cliente.

Conexiones históricas

```
./admin_monitor 1.0 admin admin -HISTORIC_CONNEC  
  
+OK Management server ready  
+OK USER  
+OK PASS  
+OK HISTORIC_CONEC  
HISTORIC CONNECTIONS = 4  
+OK GOODBYE
```

Conexiones activas

```
./admin_monitor 1.0 admin admin -LIVE_CONNEC  
  
+OK Management server ready  
+OK USER  
+OK PASS  
+OK LIVE_CONEC  
LIVE CONNECTIONS = 5  
+OK GOODBYE
```

Bytes transferidos

```
./admin_monitor 1.0 admin admin -BYTES_TRANS  
  
+OK Management server ready  
+OK USER  
+OK PASS  
+OK BYTES_TRANS  
BYTES SENT = 435  
BYTES RECEIVED = 178  
+OK GOODBYE
```

Users

```
./admin_monitor 1.0 admin admin -USERS  
  
+OK Management server ready  
+OK USER  
+OK PASS  
+OK USERS  
USER COUNT: 2  
USER LIST:
```

```
admin
user1

+OK GOODBYE
```

Status

```
./admin_monitor 1.0 admin admin -STATUS

+OK Management server ready
+OK USER
+OK PASS
+OK STATUS
BYTES SENT = 611
BYTES RECEIVED = 253
LIVE CONNECTIONS = 8
HISTORIC CONNECTIONS = 8
TOTAL USERS = 2

MAX USERS = 2
MAX CONNECTIONS = 5
+OK GOODBYE
```

Máxima cantidad de conexiones

```
./admin_monitor 1.0 admin admin -MAX_CONNECTIONS 2

+OK Management server ready
+OK USER
+OK PASS
+OK MAX_CONNECT CHANGED
+OK GOODBYE
```

Máxima cantidad de usuarios

```
./admin_monitor 1.0 admin admin -MAX_USERS 5

+OK Management server ready
+OK USER
+OK PASS
+OK MAX_USERS
MAX USERS CHANGED SUCCESSFULLY
+OK GOODBYE
```

Ejecutar un timeout

```
./admin_monitor 1.0 admin admin -TIMEOUT 10
```

```
+OK Management server ready  
+OK USER  
+OK PASS  
+OK TIMEOUT COMPLETED  
+OK GOODBYE
```

Eliminar un usuario

```
./admin_monitor 1.0 admin admin -DELETE_USER user1
```

```
+OK Management server ready  
+OK USER  
+OK PASS  
+OK DELETE_USER  
USER DELETED SUCCESSFULLY  
+OK GOODBYE
```

Agregar un usuario

```
./admin_monitor 1.0 admin admin -ADD_USER user2 pass2
```

```
+OK Management server ready  
+OK USER  
+OK PASS  
+OK ADD_USER  
USER ADDED SUCCESSFULLY  
+OK GOODBYE
```

Resetear un usuario

```
./admin_monitor 1.0 admin admin -RESET_USER_PASSWORD user2
```

```
+OK Management server ready  
+OK USER  
+OK PASS  
+OK RESET_USER_PASSWORD  
PASSWORD RESET SUCCESSFULLY  
+OK GOODBYE
```

Cambiar la contraseña del administrador

```
./admin_monitor 1.0 admin admin -CHANGE_PASSWORD admin newpass  
  
+OK Management server ready  
+OK USER  
+OK PASS  
+OK CHANGE_PASSWORD  
PASSWORD CHANGED SUCCESSFULLY  
+OK GOODBYE
```

Guia de instalacion

Requerimientos:

- GCC
- Make

Para compilar el proyecto se puede correr el comando make all en el directorio del proyecto (pop3-server).

Una vez compilado se generan dos archivos ejecutables ./run que corresponde al servidor y un ejecutable ./admin_monitor que corresponde a la aplicación de monitoreo.

En el README.md en la raíz de proyecto se pueden encontrar más detalles.

Problemas encontrados durante el diseño y la implementación

Durante el proceso de diseño y la implementación del proyecto, nos encontramos con varios problemas que afectaron tanto la calidad como la eficiencia del proyecto. A continuación, se detallan los desafíos más significativos encontrados:

- Implementación del RETR: Al implementar la funcionalidad de RETR en el servidor pop3, tuvimos dificultades al diseñar una implementación que no sea bloqueante. Para solucionar este obstáculo, dimos un paso atrás y nos planteamos exactamente cómo debería funcionar la coordinación de los select en conjunto con la máquina de estados. Luego, a partir de esta base desarrollamos un código que sigue con lo planteado. Nos sirvió mucho consultar en clase por el diseño, puesto que el docente nos ayudó a razonar fuera del código para entender bien la coreografía del selector, y del fd del mail.

- Compatibilidad con curl: En una instancia inicial, el servidor no contaba con la capacidad de que los usuarios se conecten a través del comando “curl”. Nos dimos cuenta que esto se debía a que existían pequeñas diferencias entre nuestro formato de respuesta comparado con el estándar de un servidor pop3. Para esto, leímos nuevamente el RFC y modificamos nuestras respuestas para que cumpla con el mismo.
- Monitoreo UDP: Inicialmente teníamos la idea de implementar un protocolo de monitoreo UDP. Sin embargo, al comenzar la implementación nos dimos cuenta que esto implicaba la consideración de múltiples posibles problemas como por ejemplo el orden de recepción de los paquetes. Finalmente nos decidimos por una implementación TCP, ya que no tenía estos problemas y además contábamos con la experiencia del servidor pop3.

Posibles Extensiones

Una posible extensión que no sería difícil implementar, sería integrar el sistema de usuarios a una base de datos no volátil. Acompañado esto se pueden implementar comandos para que los usuarios puedan configurar su propia cuenta.

Otra extensión que se puede implementar para el administrador es el control de acceso. En esta línea se pueden implementar varias funcionalidades. El ban de IPs, el ban de usuarios, políticas en las contraseñas, roles con diferentes permisos, etc.

Otra posible extensión es la integración de SMTP. Esto permitirá al usuario poder enviar y recibir mails a través de internet. Con esta extensión el servidor ya completa el flujo del cliente, desde el manejo de la casilla hasta el transporte de los mails.

Conclusiones

En conclusión el trabajo resultó ser un gran desafío que puso a prueba nuestras capacidades como desarrolladores. Fue gratificante el hecho de implementar nuestro primer servidor, interactuar con el mismo y que sea compatible con Mail User Agents. El desarrollo nos llevó muchos días, así que haber empezado con anticipación fue muy positivo. Nos dividimos bien las tareas en el grupo, pero también programamos juntos varias veces, sobre todo para resolver problemas que iban surgiendo.

Podemos decir que este trabajo nos enseñó mucho acerca de cómo afrontar nuevas tecnologías y cómo utilizar y entender código ajeno a través de los patches. Tuvimos varios momentos en donde nos estancamos y aprendimos a mantener la cabeza fría, aprovechar el tiempo e integrantes eficientemente para resolver nuestras dudas y problemas. Sin duda fue de los desafíos más grandes que afrontamos hasta ahora en la carrera y sentimos que aprendimos mucho de cómo manejarnos en situaciones desfavorables.