



## **Trabajo Práctico - Protocolos de Comunicación**

### ***Servidor POP3***

**[24/11/2023] - Versión 1.0**

#### **Integrantes**

Martínez, Tomás - 62878

Braun, Santos - 62090

Bruzzone, Ignacio - 62038

Kean, Nathaniel - 65907

#### **Docentes**

Marcelo Fabio Garberoglio

Juan Francisco Codagnone

Sebastian Kulesz

---

## Tabla de Contenidos

Descripción de los protocolos y aplicaciones desarrolladas .....	2
Problemas encontrados en el desarrollo .....	7
Limitaciones del proyecto .....	7
Posibles extensiones .....	9
Conclusiones .....	9
Ejemplos de prueba .....	10
Guia de instalacion .....	13
Instrucciones para la configuración .....	14
Ejemplos de configuración y monitoreo .....	14
Documento de diseño del proyecto .....	15

## Descripción de los protocolos y aplicaciones desarrolladas

El proyecto tiene como objetivo implementar un servidor para el protocolo *Post Office Protocol*, o bien POP3. Para que este pueda ser usado por Mail User Agents (MUA) con el objetivo de realizar la recepción de correos electrónicos. Al mismo tiempo, se implementó un protocolo propio llamado *Server Control and Management Protocol (SCMP)* y una aplicación de usuario para poder utilizar el mismo. Este servidor puede manejar hasta 900 conexiones simultáneas.

Con respecto a la implementación del protocolo POP3, efectuamos una implementación bloqueante con mecanismo de transformación de mensajes. Más adelante se detallará los casos de uso para un uso apropiado del mismo.

Es importante destacar que hay ciertos supuestos asumidos, como por ejemplo que los usuarios ya están creados y a su vez sus casillas ya cuentan con mails de manera predeterminada. Por este motivo, el uso esperado del servidor es

conectarse a un usuario existente y navegar su casilla o bien utilizar los comandos disponibles.

## **Protocolo POP3**

Este protocolo se basa en generar una conexión entre un cliente y un servidor, que se desarrolla a través de tres fases distintas: autenticación, transacción y actualización. Inicialmente, cuando el cliente establece conexión con el servidor, es necesario que introduzca sus credenciales para acceder a su cuenta de correo. Aunque hay varios métodos de autenticación, como los comandos APOP y AUTH, el método más común implica el empleo de los comandos USER y PASS, donde se proporciona el nombre de usuario y la contraseña. Una vez que el cliente está autenticado, se inicia la fase de transacción, donde puede gestionar su correo electrónico, incluyendo la descarga, visualización y eliminación de mensajes. Al decidir terminar la sesión, el cliente pasa a la fase de actualización. En este estado, se borran los correos que el cliente eligió eliminar durante la fase de transacción y, posteriormente, se cierra la conexión. Si el cliente opta por terminar la sesión antes de autenticarse, no se entra en la fase de actualización y la conexión se corta de inmediato.

- *Funcionamiento General*

Este programa comienza parseando las opciones de línea de comandos para configurar varios aspectos del servidor, tales como los puertos IPv4 e IPv6, el directorio de almacenamiento de correos y un comando de transformación, que podría ser utilizado para procesar los correos electrónicos.

Una vez configurado, el servidor se inicia para escuchar conexiones tanto en IPv4 como en IPv6. Se configuran los sockets para permitir múltiples conexiones simultáneas y se ponen en modo de escucha, esperando conexiones entrantes. Al establecer una nueva conexión, el servidor registra

dicha actividad y envía un mensaje de bienvenida al cliente, siguiendo los protocolos estándar de comunicación.

El servidor utiliza un ciclo principal para manejar las conexiones de los clientes. Mediante el uso de `select()`, el servidor puede monitorear eficientemente múltiples sockets de cliente y responder adecuadamente a la actividad detectada en ellos. Cuando se detecta actividad en un socket, el servidor lee los datos para verificar si se ha recibido un comando POP3 completo. Estos comandos son procesados por una función dedicada que se dedica a manejar los comandos estándar de POP3 como USER, PASS, LIST, RETR y DELE.

El manejo de la autenticación y la gestión de sesiones son aspectos críticos en el servidor. Los usuarios deben autenticarse con éxito antes de que se les permita ejecutar comandos que manipulan los correos electrónicos. Esta autenticación garantiza la seguridad e integridad de las sesiones de usuario.

Además, el servidor está equipado para manejar tanto la recepción de mensajes de correo electrónico como el envío de respuestas a los clientes. Esto incluye la entrega de listas de mensajes, el contenido de los mensajes y la confirmación de acciones como la eliminación de mensajes. Cuando un cliente se desconecta o emite el comando QUIT, el servidor cierra de manera adecuada el socket y actualiza su estado interno para reflejar este cambio.

Por último, el servidor mantiene un registro detallado de las conexiones, desconexiones y la cantidad de datos transmitidos. Este seguimiento es esencial para fines de monitoreo y depuración, asegurando que el servidor funcione de manera esperada.

- *Comandos Básicos*

USER: Utilizado para iniciar la sesión con un nombre de usuario. La sintaxis es `USER <username>`.

PASS: Después del comando USER, se utiliza para proporcionar la contraseña. La sintaxis es `PASS <password>`.

QUIT: Cierra la sesión y desconecta del servidor. La sintaxis es simplemente `QUIT`.

- *Comandos de Mensajes*

LIST: Enumera todos los mensajes en el buzón del usuario. Se usa después de la autenticación y no requiere argumentos adicionales.

RETR: Recupera un mensaje específico. La sintaxis es `RETR <mail_number>`, dónde `<mail_number>` es el número del mensaje.

DELE: Marca un mensaje específico para su eliminación. Se utiliza como `DELE <mail_number>`.

RSET: Desmarca todos los mensajes marcados para eliminación. Se emplea sin argumentos adicionales.

STAT: Proporciona un resumen del buzón, incluyendo el número de mensajes y el tamaño total. Se utiliza sin argumentos adicionales.

## **Protocolo SCMP**

- *Funcionamiento General*

Para brindar un mayor nivel de control al servidor, se ha desarrollado un protocolo de monitoreo y control, el cual tiene como recolectar datos y permite reconfigurar el servidor en tiempo real.

Este protocolo funciona sobre TCP, con autenticación de usuarios. Este sistema de autenticación es el mismo que fue utilizado para el protocolo POP3, únicamente difiere con respecto a la validación de si un usuario es admin.

- *Comandos*

CAPA: Muestra una lista de capacidades y comandos disponibles, que varían según si el usuario es un administrador o no.

ALL\_CONNEC: Muestra todas las conexiones actuales.

CURR\_CONNEC: Muestra las conexiones actuales del usuario.

BYTES\_TRANS: Muestra la cantidad de bytes transmitidos.

USERS: Enumera todos los usuarios.

STATUS: Muestra el estado actual del servidor.

MAX\_USERS: Establece el número máximo de usuarios. La sintaxis es `MAX_USERS <int>`.

DELETE\_USER: Elimina un usuario específico. Se utiliza como `DELETE_USER <username>`.

ADD\_USER: Añade un nuevo usuario. La sintaxis es `ADD_USER <username> <password>`.

RESET\_USER\_PASSWORD: Restablece la contraseña de un usuario. Se utiliza como `RESET_USER_PASSWORD <username>`.

CHANGE\_PASSWORD: Cambia la contraseña de un usuario. La sintaxis es `CHANGE_PASSWORD <old_password> <new_password>`.

TRANSFORM: Establece una aplicación de transformación para los correos electrónicos. Se utiliza como `TRANSFORM <transform_application>`.

## **Problemas encontrados en el desarrollo**

El primer problema encontrado durante el desarrollo del trabajo práctico fue entender bien cómo armar el funcionamiento principal del servidor, ya que lo habíamos visto de forma teórica pero en una forma mucho más simplificada. Saber bien cómo inicializar los distintos componentes y armar los listeners para manejar todas las conexiones nos llevó un tiempo hasta que lo pudimos tener funcionando correctamente.

Otro de los problemas encontrados fue el tema de byte-stuffing en los mails, que al tener mails que contienen `\r\n.\r\n` antes del final, usando thunderbird por ejemplo, que causa que el mail se lea hasta tal punto y el resto no lo lea, por eso decidimos eliminar estos de los mails aprovechando la función de transformación para elimínalos y luego aplicar la función para enviarlos al cliente correctamente.

## **Limitaciones del proyecto**

Dentro del desarrollo del servidor, nos hemos encontrado con ciertas limitaciones inherentes a las decisiones de diseño y las capacidades técnicas del sistema. Entre estas, destaca la función de transformación, cuyas restricciones merecen una mención especial en nuestro informe.

La primera limitación de la función de transformación es el tamaño máximo de los correos electrónicos que el sistema puede procesar, establecido en aproximadamente 100 Mb. Este límite, aunque sustancial para la mayoría de los casos de uso, establece una barrera en el manejo de correos de tamaños

extremadamente grandes. Aunque este umbral se seleccionó cuidadosamente para equilibrar la capacidad y la eficiencia, es importante reconocerlo como una restricción en el sistema.

Además, hay una preocupación significativa relacionada con el uso del comando TRANSFORM por parte del administrador. Si el administrador introduce una cadena de texto inapropiada para ejecutar en una shell, podrían surgir problemas en la ejecución del comando RETR, que es fundamental para que los clientes recuperen sus correos. Esta limitación pone de relieve la importancia de que el administrador maneje este comando con extrema precaución y responsabilidad, ya que un mal uso podría afectar adversamente la funcionalidad y seguridad del servidor.

Otra medida adoptada para mejorar la gestión del sistema fue la introducción de un límite en la cantidad de comandos que pueden enviarse simultáneamente. Esta decisión, orientada a facilitar el manejo del pipelining, es crucial para controlar de manera efectiva el flujo de la aplicación y asegurar la autenticación adecuada de los comandos. Si bien esta restricción puede parecer limitante, es una parte necesaria para mantener la integridad y estabilidad operativa del servidor, equilibrando la seguridad con el rendimiento.

Estas limitaciones, aunque representan desafíos en el diseño y la operación del servidor, también demuestran nuestra prioridad con la seguridad, la estabilidad y la eficiencia. Al entender y reconocer estas restricciones, podemos trabajar hacia mejoras futuras y asegurarnos de que el sistema siga siendo robusto y confiable.



## **Posibles extensiones**

Una mejora accesible y valiosa para el servidor sería vincular el sistema de usuarios con una base de datos persistente. Esto facilita la implementación de funcionalidades adicionales, como la posibilidad de que los usuarios personalicen sus cuentas mediante comandos específicos.

Para los administradores, una extensión interesante sería el establecimiento de un sistema de control de acceso más sofisticado. Dentro de este ámbito, se podrían incorporar varias funcionalidades, como la restricción de direcciones IP específicas, la prohibición de ciertos usuarios, la implementación de políticas de seguridad para las contraseñas y la creación de roles con distintos niveles de permisos.

Otra ampliación significativa sería la integración del protocolo SMTP. Esta extensión permitiría a los usuarios no solo gestionar sus correos sino también enviar y recibir mensajes a través de internet, completando así el ciclo de funcionalidades del cliente, desde la administración de su bandeja de entrada hasta el envío y recepción de correos.

Finalmente una posible extensión de nuestra implementación puede ser incorporar los comandos opcionales definidos en el RFC como TOP; para ofrecer una manera alternativa de autenticación o bien APOP; cuyo propósito es para soportar extensiones del protocolo como los códigos de respuesta.

## **Conclusiones**

El desarrollo de un servidor que emplea el protocolo POP3 ha sido una experiencia enriquecedora y desafiante, marcada por una serie de obstáculos técnicos y logísticos que hemos logrado superar con dedicación y esfuerzo. Este proyecto no solo nos ha permitido profundizar en los intrincados detalles del

protocolo POP3, sino que también nos ha brindado la oportunidad de enfrentarnos a problemas reales y encontrar soluciones prácticas y eficientes.

Durante el proceso de desarrollo, nos enfrentamos a desafíos significativos relacionados con la gestión eficiente de las conexiones de los clientes, la implementación segura de la autenticación y la manipulación precisa de los mensajes de correo electrónico. Cada uno de estos desafíos nos obligó a pensar de manera creativa, a menudo llevándonos a explorar nuevas técnicas y estrategias. En particular, nos esforzamos por garantizar que cada aspecto del servidor fuera intuitivo y accesible, manteniendo al mismo tiempo altos estándares de seguridad y eficiencia.

A través de este proceso, hemos aprendido lecciones valiosas sobre la importancia del diseño modular y la documentación clara, lo que facilitó significativamente la adaptación y la expansión de nuestro proyecto. Además, la colaboración y la comunicación efectiva dentro de nuestro equipo fueron cruciales para resolver problemas complejos y para mantener el progreso constante del proyecto.

En conclusión, el desarrollo de este servidor POP3 ha sido una experiencia muy valiosa, caracterizada tanto por sus desafíos como por sus triunfos. Nos ha permitido crecer como desarrolladores y como equipo, brindándonos una comprensión más profunda de lo que implica construir una infraestructura de comunicación segura y confiable.

## **Ejemplos de prueba**

### **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. Realizamos este test con telnet y con CURL para probar si funciona también y nos dieron los resultados esperados. En las siguientes capturas de pantalla puede verse cómo en ambos casos se corren los comandos y el funcionamiento es el esperado.

```
bruzo@DESKTOP-KTU0F08:~$ stty -icanon && telnet localhost 1110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 mail.itba.net v4.20 server ready
USER user1
PASS +OK User name accepted, send PASS command
pass1
RETR 1
+OK Logged in
Subject: Welcome to the service

Welcome to our email service, user1!

We're excited to have you on board.

Best regards,
The Email Service Team

.
```

```
bruzo@DESKTOP-KTU0F08:~$ stty -icanon && curl pop3://user1:pass1@localhost:1110/1
Subject: Welcome to the service

Welcome to our email service, user1!

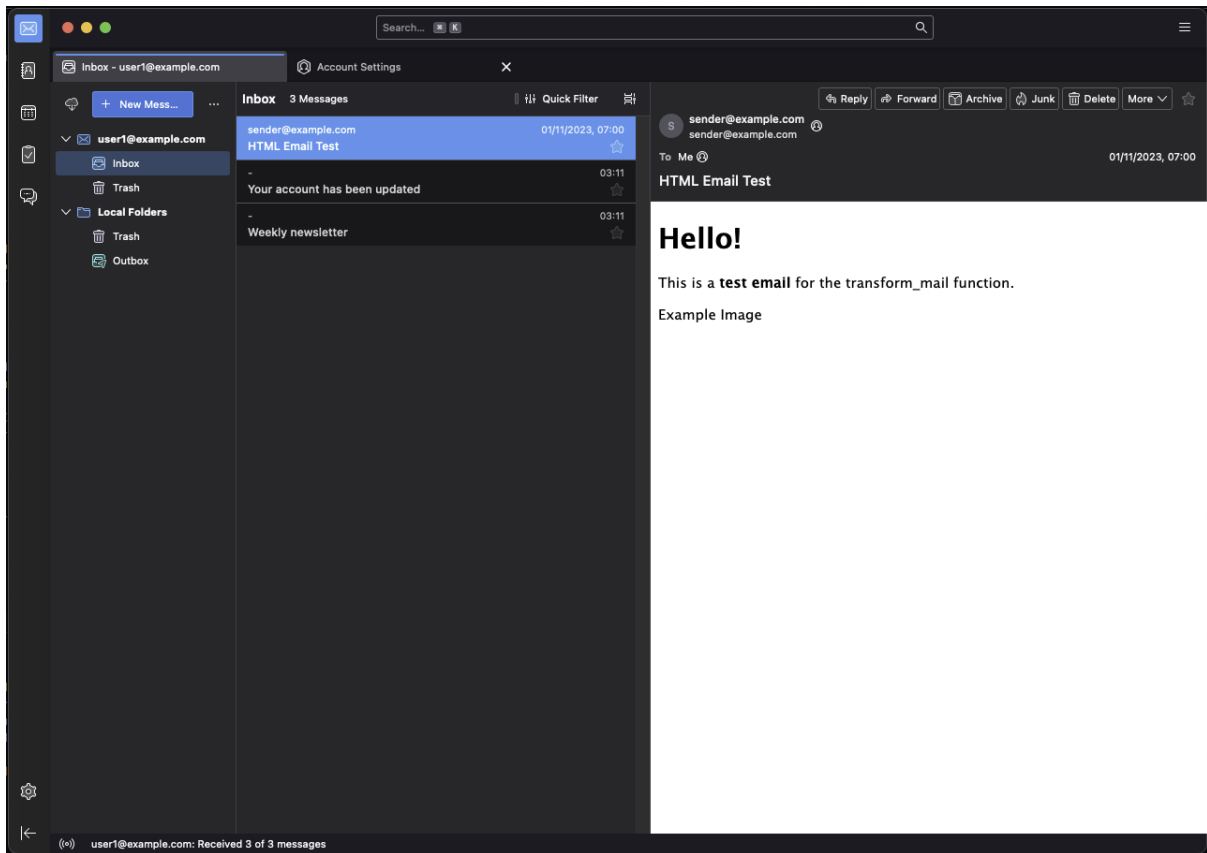
We're excited to have you on board.

Best regards,
The Email Service Team
```

*test stty en la consola*

## Utilización de MUAs

Nos conectamos a Thunderbird, el objetivo de esto era poder ver nuestros mails usando un MUA así podíamos observar si nuestro protocolo se alinea correctamente con el RFC 1939 lo cual pudo hacerlo sin problema.



*thunderbird usando nuestro protocolo*

## Test ipv4 e ipv6

Para este test nos conectamos usando telnet a la aplicación primero es con ipv4 entonces usamos el socket pasivo 1110 y luego para ipv6 que usamos el socket pasivo 9090.

```
bruzo@DESKTOP-KTU0F08:~/TP_Protos/src/maildir/user1/cur$ telnet localhost 1110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 mail.itba.net v4.20 server ready
```

*conexión ipv4 al servidor*

```
bruzo@DESKTOP-KTU0F08:~/TP_Protos/src/maildir/user1/cur$ telnet ::1 9090
Trying ::1...
Connected to ::1.
Escape character is '^]'.
+OK POP3 mail.itba.net v4.20 server ready
|
```

*conexión con ipv6 al servidor*

## Guía de instalación

Para compilar el proyecto, será necesario la utilización de GCC y Make.

Como primer paso, deberá abrir una terminal y ejecutar el siguiente comando con la ruta al directorio donde se encuentra alojado el proyecto.

```
cd <ruta_al_proyecto>
```

A continuación, ejecutar el siguiente comando el cual generará el ejecutable necesario para correr el proyecto (Compilado con gcc).

```
make
```

Alternativamente, para compilar el proyecto con clang, se debería correr el siguiente comando

```
make CC=clang
```

Finalmente, se contará con el ejecutable `./pop3d` que corresponde al servidor que se puede correr desde la terminal.

## Instrucciones para la configuración

Para empezar se puede configurar el servidor con distintos flags al correrlo por primera vez. proporcionamos los argumentos **-p**, **-P**, **-d** y **-t**. Estas opciones sirven para la configuración inicial del servidor. **-p** se usa para determinar el puerto para ipv4 que por default es el 1110. Luego, **-P** se usa para determinar el puerto para ipv6 que por default es el 9090. También, tenemos el flag **-d** que sirve para establecer el directorio de los correos. Por último, tenemos el **-t** que sirve para especificar el comando de transformación que se quiere usar para los mails.

```
./pop3d -p 110 -P 6262 -t wc -d /var/maildir
```

Este es un ejemplo de uso donde se usará el directorio `/var/maildir` para guardar los mails. Para conectarse debe usar el puerto 110 para ipv4 o el puerto 6262 para ipv6. Al intentar ver un mail el protocolo le enviará el resultado de aplicarle el comando `wc` al mismo.

## Ejemplos de configuración y monitoreo

El sistema de monitoreo y control cuenta con una serie de comandos que permite hacer un manejo de usuarios y permiten ver el estado interno del servidor. Para poder acceder a estos comandos, se usa un usuario especial el cual inicia sesión como cualquier otro usuario, su usuario es **default** y su contraseña es **default**.

A continuación algunos ejemplos:

Si desea agregar un usuario con username **aeinstein** y contraseña **physics** lo puede hacer utilizando el siguiente comando como administrador.

```
ADD_USER aeinstein physics
```

Análogamente podemos remover usuarios con la utilización del siguiente comando.

```
DELETE_USER username
```

Si desea cambiar la cantidad de cuentas de usuario con las que cuenta su servidor, puede hacerlo utilizando el siguiente comando

```
MAX_USERS 500
```

Si quiere consultar las estadísticas del servidor en tiempo real, como la cantidad total de conexiones, la cantidad de conexiones ocurriendo en este momento o bien la cantidad de bytes transmitidas, puede hacerlo utilizando el siguiente comando.

```
STATUS
```

## **Documento de diseño del proyecto**

Esta sección presenta la arquitectura y diseño del servidor. Este ha sido diseñado con un enfoque en la seguridad, la eficiencia y la escalabilidad, proporcionando una base sólida para la gestión de correos electrónicos.

El sistema se divide en dos componentes principales: el servidor POP3 y el sistema de gestión del servidor (SCMP). Cada uno de estos componentes atienden diferentes aspectos del sistema general, trabajando en conjunto para proporcionar una experiencia completa y segura.

Con respecto al servidor pOP3, este es la piedra angular del sistema, encargado de gestionar las comunicaciones de correo electrónico. Su núcleo funcional reside en el parser de POP3, que interpreta y procesa los comandos enviados por los clientes de correo.

### *Parser de POP3*

El parser, inicializado por `pop3_parser_init`, es responsable de analizar comandos como `USER`, `PASS`, `LIST` y `RETR`. El sistema de autenticación incorporado en el parser, ejemplificado por las funciones `authenticate_user` y `authenticate_password`, es fundamental para garantizar que solo los usuarios autorizados accedan a sus correos.

### *Server Control Management Protocol (SCMP)*

Este sistema proporciona un conjunto de comandos que permiten a los administradores supervisar y controlar el servidor POP3 de manera mas precisa, pudiendo cambiar la configuración en tiempo real.

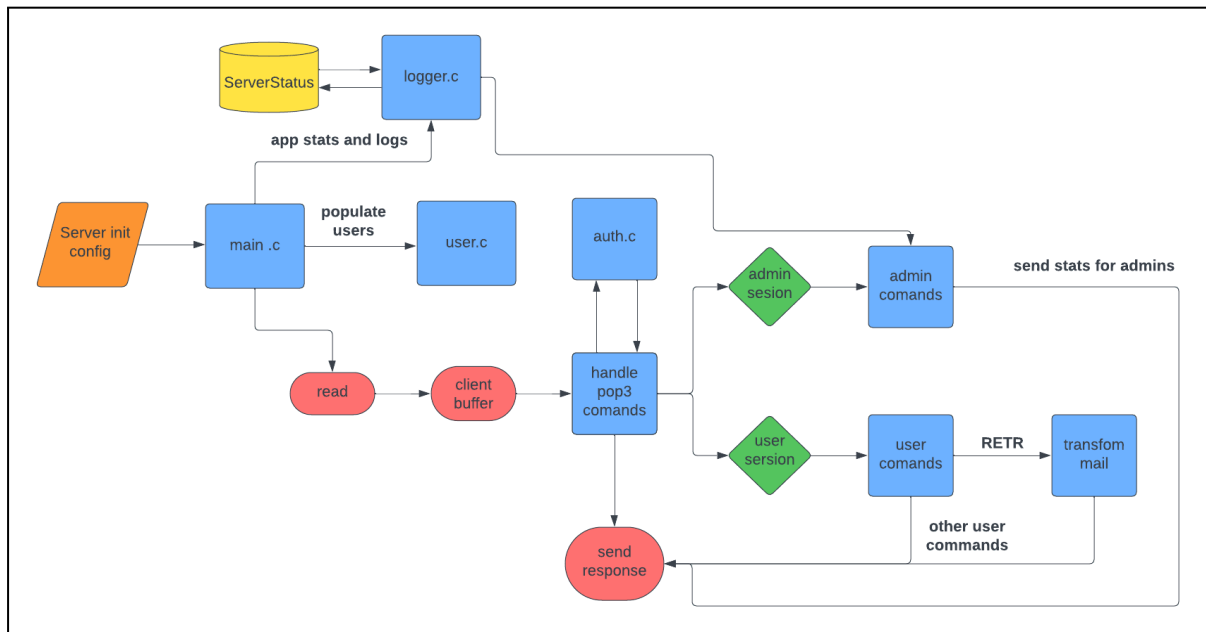
Los comandos administrativos, como `handle_all_connec_command` y `handle_curr_connec_command`, ofrecen información vital sobre el estado del servidor, como el número de conexiones totales y actuales. Además, funciones como `handle_bytes_trans_command` y `handle_status_command` brindan datos sobre el tráfico de datos en el servidor.

Un aspecto crucial del sistema de gestión es la administración de usuarios. A través de comandos como `handle_users_command`, `handle_add_user_command`, `handle_delete_user_command`, `handle_reset_user_password_command`, y `handle_change_password_command`, los administradores pueden gestionar



cuentas de usuario, ajustar configuraciones de seguridad y mantener la integridad del sistema.

A continuación se puede observar un diagrama donde se muestra como están conectados los diferentes componentes que hacen al servidor.



*Diagrama de flujo del servidor*