



Instituto Tecnológico  
de Buenos Aires

72.07 Protocolos de Comunicación  
Trabajo Práctico Especial

Servidor POP3

2do Cuatrimestre 2024

Integrantes:

Sergio Smirnoff - Legajo: 62256

Patrick Luca Torlaschi - Legajo: 62273

Santiago Joaquín Nartallo Galvagno - Legajo: 62208

Profesores:

Valerio Ramos, Pedro Martín

Garberoglio, Marcelo Fabio

Stupenengo Faus, Hugo Javier

Axt Roberto Oscar, Roberto Oscar

Codagnone, Juan F.

Kulesz, Sebastian

Mizrahi, Thomas

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Descripción de los protocolos implementados</b>	<b>3</b>
2.1 POP3	3
2.2 Protocolo de Monitoreo	4
<b>3. Dificultades encontradas</b>	<b>5</b>
<b>4. Limitaciones de la aplicación</b>	<b>6</b>
4.1. Limitaciones del POP3	6
4.2. Limitaciones del monitor del server	6
<b>5. Posibles extensiones</b>	<b>7</b>
<b>6. Conclusiones</b>	<b>8</b>
<b>7. Instructivo de instalación</b>	<b>9</b>
<b>8. Pruebas de Uso</b>	<b>10</b>
<b>9. Referencias</b>	<b>11</b>
10.1 Normativas	11
10.2 Documentación	11

# **1. Introducción**

Para el trabajo práctico especial de la materia Protocolos de Comunicación se nos solicitó hacer una implementación de un servidor POP3 basado en el RFC 1939, con soporte para autenticación de usuario/contraseña según el RFC 2449, para poder ser usado por Mail User Agents para recibir correos electrónicos. A su vez se pidió desarrollar un protocolo de monitoreo de métricas del servidor puesto en funcionamiento para que el cliente pueda utilizar.

A continuación, se desarrollará todo lo vinculado a la implementación del práctico. Se harán referencias a los componentes a implementar, dificultades encontradas al momento de desarrollar el servidor, limitaciones del proyecto, posibles extensiones a futuro para la implementación y, por último, una conclusión.

## 2. Descripción de los protocolos implementados

### 2.1 POP3

El protocolo POP3 (Post Office Protocol versión 3), definido en el RFC 1939, es un protocolo utilizado para la recuperación de correos electrónicos desde un servidor remoto. Su funcionamiento se basa en un modelo cliente-servidor en el que el cliente descarga los mensajes almacenados en el servidor de correo. El cliente inicia la conexión con el servidor en el puerto 110 y se autentica mediante un nombre de usuario y contraseña (vinculado al soporte pedido para RFC 2449). Una vez verificadas las credenciales, el cliente puede interactuar con los mensajes almacenados en el servidor.

El protocolo se divide en varias fases. Durante la fase de autorización, el cliente se autentica. Luego, en la fase de transacción, el cliente puede listar, recuperar y eliminar mensajes. Entre los comandos más utilizados en esta fase se encuentran **USER** y **PASS** para la autenticación, **STAT** para obtener el número de mensajes y su tamaño total, **LIST** para listar los mensajes disponibles, **RETR** para descargar un mensaje completo y **DELE** para marcar un mensaje para su eliminación. Finalmente, el comando **QUIT** cierra la sesión y aplica las eliminaciones pendientes. En la fase de actualización, los mensajes marcados como eliminados se borran del servidor al finalizar la sesión.

A lo largo de una sesión POP3, el protocolo transita por tres estados principales: **AUTHENTICATION**, **TRANSACTION** y **UPDATE**. El estado de **AUTHENTICATION** es el primero en el que el cliente se conecta al servidor y proporciona sus credenciales (nombre de usuario y contraseña) para verificar su identidad. Durante esta fase, el cliente no podrá interactuar con los correos electrónicos, o al menos hasta que la autenticación haya sido exitosa. Una vez autenticado, el servidor transita al estado de **TRANSACTION**, donde el cliente puede listar, recuperar, eliminar o marcar mensajes en el servidor. En esta fase, las operaciones sobre los correos son temporales. Los cambios no se aplican de forma definitiva hasta que la sesión finaliza. Finalmente, al cerrar la sesión con el comando **QUIT**, el servidor pasa al estado de **UPDATE**, momento en el cual se aplican los cambios que hayan quedado pendientes, como la eliminación de los mensajes marcados para ser borrados, por ejemplo. En este estado, el servidor confirma que los mensajes eliminados ya no están disponibles y se cierra la conexión de manera definitiva.

## 2.2 Protocolo de Monitoreo

Adicionalmente, se implementó un servicio de monitoreo que permite extraer las estadísticas sobre el servidor POP3 y, a su vez, permite cambiar propiedades/configuraciones del mismo servidor.

En esta versión del protocolo se pueden consultar las siguientes métricas vinculadas al servidor POP3:

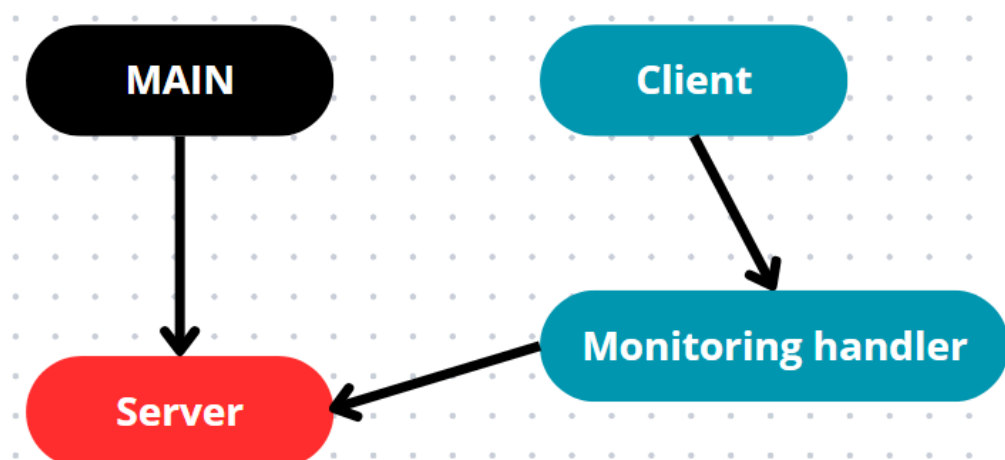
1. Cantidad total de mensajes
2. Cantidad total de bytes enviados en el servidor POP3
3. Cantidad de usuarios en línea en el servidor
4. Cantidad histórica de usuarios en el servidor

Todas estas métricas pueden ser visibles a través del comando **METR**.

Asimismo, se puede modificar la configuración en tiempo de ejecución de las siguiente propiedades:

1. Transformación (comando **TRAN**)
2. Host (comando **HOST**)
3. Basedir (comando **DIRR**)
4. Puerto (comando **PORT**)
5. IP (comando **IPV6**)

Por último, hay dos comandos adicionales. **INFO** le permite al cliente solicitar la información del servidor. Por otro lado, **QUIT** le permite al usuario desconectarse del monitoring client.



Imágen 1: Diagrama estructural del funcionamiento del cliente de monitoreo

### **3. Dificultades encontradas durante el desarrollo e implementación de los protocolos**

A lo largo del desarrollo del trabajo práctico, nos hemos encontrado con algunas dificultades. Como por ejemplo, durante la implementación de la función *receive* se descubrió que el `‘/r/n’` en algunas ocasiones lo recibía entero, y en otras lo recibía en dos partes, haciendo que se perdiera la funcionalidad de la misma. No logramos identificar un patrón con respecto a los casos en los que sucedía esta situación. De todas formas, se pudo arreglar.

Otro desafío con el que nos encontramos, que probablemente haya sido de las mayores dificultades surgidas en el práctico, fue la conversión del servidor a uno con un sistema no bloqueante. El procesamiento de archivos en el modo no bloqueante fue el problema principal ya que teníamos que encontrar una forma que no se bloquee y en simultáneo poder soportar archivos de un peso grande mientras que el servidor pueda seguir funcionando adecuadamente.

## 4. Limitaciones de la aplicación

### 4.1. Limitaciones del POP3

Una de las principales limitaciones de esta implementación de POP3 es la dependencia de una estructura predefinida de directorios para el almacenamiento de los correos electrónicos, que sigue un esquema basado en un maildir por usuario. Esto implica que los directorios específicos para cada usuario deben estar creados previamente antes de ejecutar el servidor. Si estos directorios no están configurados correctamente o si se añade un nuevo usuario sin actualizar manualmente la estructura, el sistema no podrá manejar los mensajes de manera adecuada. Esta rigidez puede dificultar la escalabilidad o la integración dinámica de nuevos usuarios, limitando su flexibilidad en entornos donde la administración automatizada de usuarios es un requisito.

Otra restricción importante es que la ejecución de los binarios del servidor POP3 depende de que el usuario se encuentre parado en la carpeta del componente que quiera correr. Esto introduce una fragilidad en el proceso de despliegue, ya que cualquier intento de ejecutar la aplicación desde un contexto diferente puede provocar errores relacionados con rutas relativas o la incapacidad de acceder a algunos recursos críticos. Este diseño afecta la portabilidad y complica la integración con sistemas externos o el poder generar/utilizar scripts de automatización que puedan invocar el servidor desde otras ubicaciones. Esta limitación puede ser un inconveniente significativo, requiriendo medidas adicionales para manejar las rutas correctamente o asegurar un entorno preconfigurado.

### 4.2. Limitaciones del monitor del server

Otra limitación relevante se encuentra en el diseño inicial del cliente para monitorear el servidor. En un principio, se implementó un único comando llamado **STAT** para obtener todas las estadísticas relacionadas con el servidor, incluyendo métricas de rendimiento, uso y estado de los usuarios. Sin embargo, esta solución no se alinea con las buenas prácticas del protocolo definido. Agrupar toda la información en un solo comando puede dificultar la extensibilidad y claridad del protocolo, además de sobrecargar la comunicación entre cliente y servidor al enviar grandes volúmenes de datos incluso cuando solo se requiere una parte específica de las estadísticas. Es por esto que se decidió manejar con un grupo de comandos que cumplen las funcionalidades descritas en la sección previa 2.2 de este informe.

## 5. Posibles extensiones

Para el protocolo POP3, una posible extensión sería incluir un comando de búsqueda como SEARCH, que permita filtrar mensajes en el servidor basándose en criterios como asunto, remitente o fecha, evitando la necesidad de descargar todos los correos y mejorando la experiencia del usuario.

Otra extensión relevante sería habilitar acceso concurrente para un mismo usuario, implementando un sistema de bloqueo optimista de mensajes que permita a múltiples sesiones operar de manera eficiente sin conflictos. Esto es especialmente útil en escenarios donde un usuario accede al correo desde varios dispositivos. Adicionalmente, se podría incluir un comando como REPORT, que permita al cliente solicitar un resumen de actividad reciente, proporcionando información sobre los mensajes recuperados o eliminados en sesiones previas, lo cual sería útil para usuarios que necesitan un mayor control sobre sus interacciones con el servidor.

En cuanto al cliente de monitoreo, sería valioso implementar un sistema de notificaciones en tiempo real que permita al cliente recibir alertas automáticas sobre eventos específicos, como la conexión o desconexión de usuarios, cambios en el tráfico del servidor o problemas críticos. Esto mejoraría la capacidad de respuesta frente a incidentes y optimizaría la administración del sistema. Asimismo, se podrían extender las capacidades de configuración para incluir opciones avanzadas, como modificar el tamaño máximo permitido para mensajes, ajustar tiempos de espera por inactividad o activar o desactivar comandos específicos del protocolo POP3, ofreciendo una mayor personalización según las necesidades del entorno.

Otra extensión útil para el cliente sería agregar comandos que permitan acceder a fragmentos de logs del servidor en tiempo real. Esto facilitará el diagnóstico de problemas o el seguimiento de eventos recientes sin necesidad de buscar manualmente en los archivos de registro. Finalmente, también podría ser de muy buena ayuda implementar la consulta de métricas históricas por períodos específicos, como días o semanas, para poder hacer un seguimiento completo del rendimiento y uso del servidor a lo largo del tiempo, mejorando así las capacidades de análisis y planificación a futuro.



## 6. Conclusiones

En la implementación del servidor POP3 y el sistema de monitoreo, se ha logrado desarrollar una solución funcional que cumple con los requisitos fundamentales para la gestión de correos electrónicos y el monitoreo de las operaciones del servidor. A lo largo del proyecto, se abordaron conceptos clave del protocolo POP3, aplicando sus principios para construir una herramienta capaz de gestionar mensajes de manera eficiente. Asimismo, se implementó un sistema de monitoreo que permite extraer métricas críticas del servidor, aportando valor para la administración y el análisis del rendimiento.

El desarrollo de estas funcionalidades enfrentó desafíos técnicos significativos, como la implementación de un servidor no bloqueante y la gestión precisa de los mensajes en un entorno concurrente. A pesar de las limitaciones identificadas, como la dependencia de estructuras de directorios predefinidas y ciertos aspectos del diseño inicial del cliente de monitoreo, la solución final ofrece una base sólida y extensible. Este proyecto no solo permitió poner en práctica conocimientos adquiridos en clase, sino también explorar nuevas técnicas y herramientas, proporcionando una plataforma funcional que puede ser mejorada y expandida en futuras iteraciones para satisfacer necesidades más complejas.

## 7. Instructivo de instalación

### Compilación:

Ejecutar el comando `make clean all` desde la carpeta `TP-PROTOS-2C2024` Los binarios se generarán en la carpeta `./bin`

### Ejecución:

Los siguientes son los argumentos posibles para la ejecución del servidor `./bin/server` o se puede correr con argumentos (hacer `./bin/server -h` para obtener más información sobre los argumentos posibles)

Por ejemplo:

```
./bin/server -T /usr/bin/sed:'s/e/x/g'
```

En caso de que sean transformaciones, aceptamos un argumento que tiene que estar separado por un ':' No se aceptan flags.

### Para el client:

(Si se utilizó la opción `-P`, se debe usar ese puerto) Por default es el 1111 `./bin/client` [dirección IPV6 o IPV4] [puerto de management]

## 8. Pruebas de Uso

```
ss-pc@smirnoff-Lin-PC ~/Documents/Facultad/72.07-ProtocolosComunicacion/TP-Protos-2C2024 <merging>
$ nc localhost 1110 -c
+OK POP3 server ready
USER user1
+OK User accepted, password needed
PASS pass
ERR Password incorrect
USER user1
+OK User accepted, password needed
PASS pass1
+OK Password accepted
STAT
+OK 1 messages (324 octets)
LIST
+OK 1 messages (324 octets)
1 324
LIST 1
1 324
RETR 1
+OK 324 octets
Date: 9/12/2018\nFrom: Emisor <juliana@bancomarta.com>
To: Receptor <receptor@gmail.com>
Subject: Asunto del Correo

Buenos días les envío este correo para poder informarles que su cuota del banco Marta esta inapaga por favor acceda a este link para poder regularizar su situacion: https://www.youtube.com/watch?v=dQw4w9WgXcQ
QUIT
+OK Goodbye
```

*Ejemplo de uso 1.1: Autenticación de usuario y RETR de un correo electrónico*

```
ss-pc@smirnoff-Lin-PC ~/Documents/Facultad/72.07-ProtocolosComunicacion/TP-Protos-2C2024 <merging>
$ ./bin/server --default
Server started on port 1110 and for managers on port 1111, accepting IPv4 and IPv6 connections...
Client connected.
Started file transfer process with PID: 19077
File transfer complete.
Client disconnected.
```

*Ejemplo de uso 1.2: El estado del servidor con el escenario previamente descrito*

```
ss-pc@smirnoff-Lin-PC ~/Documents/Facultad/72.07-ProtocolosComunicacion/TP-Protos-2C2024 <merging>
$ nc localhost 1110 -c
+OK POP3 server ready
USER user1
+OK User accepted, password needed
PASS pass1
+OK Password accepted
RETR 1
+OK 324 octets
Date: 9/12/2018\nFrom: Emisor <juliana@bancomarta.com>
To: Rxcxptor <rxcxptor@gmail.com>
Subject: Asunto del Correo

Buxnos días lxs xnvio xstx corrxo para podxr informarlx qux su cuota dxl banco Marta xsta inapaga por favor accda a xstx link para podxr rxgularizar su situacion: https://www.youtubx.com/watch?v=dQw4w9WgXcQ
QUIT
+OK Goodbye
```

*Ejemplo de uso 2.1: Muestra del funcionamiento de las transformaciones*

```
ss-pc@smirnoff-Lin-PC ~/Documents/Facultad/72.07-ProtocolosComunicacion/TP-Protos-2C2024 <merging>
$ ./bin/server --default -T /usr/bin/sed:'s/e/x/g'
original: /usr/bin/sed:s/e/x/g
Server started on port 1110 and for managers on port 1111, accepting IPv4 and IPv6 connections...
Client connected.
Started file transfer process with PID: 19767
File transfer complete.
Client disconnected.
^Csignal 2, cleaning up and exiting
poll failed: Interrupted system call
```

*Ejemplo de uso 2.2: El estado del servidor con el escenario previamente descrito*

## 9. Referencias

### 10.1 Normativas

- [1] [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.
- [2] [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] [RFC2449] Gellens, R., Newman, C., and L. Lundblade, "POP3 Extension Mechanism", RFC 2449, November 1998.
- [4] [RFC3778] Taft, E., Pravetz, J., Zilles, S., and L. Masinter, "The application/pdf Media Type", RFC 3778, DOI 10.17487/RFC3778, May 2004, <<http://www.rfc-editor.org/info/rfc3778>>.

### 10.2 Documentación

- [1] <https://spamassassin.apache.org/>
- [2] <http://www.pc-tools.net/unix/renattach/>
- [3] <https://pubs.opengroup.org/onlinepubs/9699919799/nframe.html>