

---

# Protocolo AProxyPOP

---

## 72.07 Protocolos de Comunicación Trabajo Práctico Especial

### Grupo 1

*Pablo Radnic ----- 57013*

*Juan Godfrid ----- 56609*

*Joaquin Ormachea ----- 57034*

*Constanza De Rienzo - 56659*

# Índice

<b>Índice</b>	<b>2</b>
<b>Descripción detallada de los protocolos y aplicaciones desarrolladas</b>	<b>3</b>
Administrador	3
Introducción	3
Operación Básica	3
Características de los mensajes	3
Los comandos:	3
Los argumentos:	3
Las respuestas:	4
Respuestas de estado NEGATIVO:	4
Estados	4
1. Estado Autenticación	4
Comandos	4
2. Estado Intercambio	5
Comandos	5
3. Estado Cierre	6
Pop3Filter	7
Introducción	7
Operación Básica	7
Media Types	9
Introducción	9
Operación Básica	9
<b>Problemas encontrados durante el diseño y la implementación</b>	<b>10</b>
<b>Limitaciones de la aplicación</b>	<b>11</b>
Prueba de estrés	12
<b>Posibles extensiones</b>	<b>12</b>
<b>Conclusiones</b>	<b>13</b>
<b>Ejemplos de prueba</b>	<b>13</b>
<b>Guia de instalación detallada y precisa</b>	<b>13</b>
<b>Instrucciones para la configuración</b>	<b>14</b>
<b>Ejemplos de configuración y monitoreo</b>	<b>15</b>
<b>Documento de diseño del proyecto</b>	<b>19</b>

# Descripción detallada de los protocolos y aplicaciones desarrolladas

## Administrador

### Introducción

El **objetivo** de este protocolo es estandarizar la interacción entre un usuario denominado “Administrador” con el proxy diseñado para un servicio POP3.

A partir de éste, un usuario verificado podrá comunicarse con la aplicación proxy e indicarle varias acciones a realizar sobre las configuraciones y monitoreo que tiene el mismo.

Es **requerido** para el uso de este protocolo que el cliente administrador conozca el puerto disponible para la conexión al proxy. Por defecto es 9090, pero este puede ser cambiado.

### Operación Básica

Inicialmente el cliente crea una conexión SCTP en el puerto 9090 (por defecto). En el lado del servidor se debiera encontrar una conexión escuchando en el correspondiente puerto. Cuando la conexión se establece, el servidor POP3 envía un saludo.

Ahora se encuentra en estado AUTENTICACIÓN. El cliente luego debe enviar usuario y contraseña en uno o más pasos los cuales van a ser verificados por el servidor. Si éste está permitido, se recibe un mensaje de éxito y se entra en estado INTERCAMBIO. En caso contrario, se ofrece la posibilidad de intentar de nuevo.

En estado INTERCAMBIO el cliente envía comandos hasta que la conexión se cierre o sea abortada. La conexión es cerrada al ingresar el comando QUIT.

### Características de los mensajes

Los comandos:

- Preestablecidos
- Case-sensitive
- Seguidos de n argumentos (según corresponda), separados de la orden por un (espacio)
- Longitud de hasta 10 caracteres
- Compuestas por ASCII imprimibles

Los argumentos:

- Hasta 40 caracteres de longitud
- Compuestos por ASCII imprimibles

Las respuestas:

- Formadas por indicador de **ESTADO**. Seguida de info adicional.
- Estados en **MAYÚSCULAS**:
  - “+OK”
  - “-ERR”
- Algunas son multilínea.
  - Se envía marca el final del mensaje con doble **CRLF**.

Respuestas de estado NEGATIVO:

- Órdenes no reconocidas, no implementadas, no válidas sintácticamente
- Orden enviada en estado incorrecto

Se realizan validaciones de comandos tanto desde el lado cliente como del lado servidor.

## Estados

El procedimiento para la comunicación se puede separar en distintos estados. Estos estados serán identificados por sus eventos de inicio.

### *1. Estado Autenticación*

Evento de inicio: Luego de que el cliente cree la conexión al puerto y el servidor responda con el saludo.

Descripción: Durante este estado se realiza el intercambio de usuario y contraseña para identificarse en el servidor proxy y validar que sea correcto su acceso.

Funcionamiento:

- 1) Se abre una conexión SCTP en el puerto 9090
- 2) El servidor DEBE enviar un saludo
- 3) Ahora el cliente DEBE identificarse y autenticarse. Para esto el servidor DEBE soportar el mecanismo de autenticación USER & PASS
  - a) **USER name**: Dentro de AUTENTICACIÓN se usa para pasar el usuario. El servidor DEBE dar una respuesta positiva aunque no exista la cuenta.
  - b) **PASS password**: Utilizando los argumentos de USER y PASS, determina si el cliente puede acceder o no a las configuraciones y monitoreo.
- 4) Si el cliente es autenticado exitosamente, el servidor RESPONDE de forma positiva y el estado pasa a INTERCAMBIO.
- 5) Si la autenticación es rechazada, el servidor RESPONDE de forma negativa y el servidor PUEDE cerrar conexión.
- 6) Si el servidor no cierra la conexión, el cliente PUEDE volver a autenticarse o realizar QUIT.

Comandos

USER name

- Argumentos: name es la cadena que contiene el nombre de usuario
- Respuestas posibles:
  - “+OK”

- Funcionalidad: Permite que el cliente provea un usuario a verificar contra el servidor proxy. Siempre DEBE responder positivo para no revelar información sobre los usuarios que tienen el privilegio de acceso a las configuraciones y monitoreo.

#### PASS password

- Argumentos: password es la cadena que contiene la contraseña a utilizar.
- Respuestas posibles:
  - "+OK"
  - "-ERR"
- Funcionalidad: Permite que el cliente provea una contraseña que al pasar al proxy junto con el usuario provisto antes, se autentique como administrador del proxy. Si el comando es utilizado no seguido de la línea USER, el servidor DEBE devolver negativo. Si el proxy no autentica el usuario, el servidor DEBE devolver negativo.

## 2. Estado Intercambio

Evento de inicio: luego de que el servidor responda que el usuario fue identificado exitosamente.

Descripción: Durante este estado, el cliente va a realizar solicitudes a través de los comandos disponibles para utilizar los servicios.

#### Funcionamiento:

- 1) El cliente DEBE comunicar un comando seguido de sus correspondientes argumentos
- 2) El servidor DEBE responder con el estado de la respuesta y la información adicional que sea pertinente.
- 3) El cliente PUEDE repetir las acciones que quiera hasta utilizar el comando QUIT.

#### Comandos

##### LISTS

- Argumentos: ninguno.
- Respuestas posibles:
  - "+OK"
- Funcionalidad: El servidor DEBE listar las distintas opciones de monitoreo, junto con su número asignado para poder utilizar en el comando STATS.

##### STATS números

- Argumentos: números es un entero obligatorio que indica cuál de las estadísticas disponibles quiero solicitar.
- Respuestas posibles:
  - "+OK"
  - "-ERR" cuando no hay estadísticas disponibles o indica un número que no existe.
- Funcionalidad: a partir de este comando, el cliente PUEDE solicitar alguna de las funciones de monitoreo que el servidor provee. El servidor DEBE proveer las

##### ACTIVE [bool]

- Argumentos: bool es un parámetro opcional de 1 o 0 que indica si el filtro se activa o desactiva, si el cliente no lo envía, el servidor DEBE responder informando cuál es el estado actual.
- Respuestas posibles:
  - “+OK - Activo”
  - “+OK - Inactivo”
- Funcionalidad: el servidor DEBE cambiar la configuración del filtro del proxy si el argumento es aceptado. Por defecto, el servidor DEBE tenerlo inactivo.

#### **FILTER [cmd]**

- Argumentos: cmd es un string que indica el nombre de un comando compatible con system(3). En caso de tener un @, este carácter escapará los espacios siguientes.
- Respuestas posibles:
  - “+OK”
  - “-ERR” si no puede aplicar el filtro
- Funcionalidad: el servidor DEBE cambiar la configuración del filtro del proxy si el argumento es aceptado. Si no envía un argumento, el servidor DEBE informar al cliente, cuál es el comando actualmente.

### *3. Estado Cierre*

Evento de inicio: luego de que el cliente ingrese QUIT.

Descripción: Se cierra la conexión por parte del servidor.

# Pop3Filter

## Introducción

El **objetivo** de esta aplicación, en términos generales, es proveer un mecanismo para que múltiples clientes normalicen sus interacciones con un servidor de pop3 al ofrecer un abanico de aptitudes constantes (como por ejemplo pipelining), llevar cuenta de distintas métricas relacionadas con el flujo de información y hacer reemplazos de contenido de mensajes en forma automática.

## Operación Básica

Al recibir una conexión de cliente entrante, se establece una conexión con el servidor de origen y se determinan sus aptitudes. Luego se envían los pedidos entrantes del Cliente y se devuelven las respuestas del servidor de origen.

Al recibir una conexión de administrador entrante, se lo autentifica. Una vez pasada esa etapa se atienden los pedidos del administrador.

El administrador tiene la posibilidad de definir el programa de transformación. Según la selección del administrador las respuestas transformables atravesaran el programa antes de ser enviadas a los cliente.

## Decisiones de diseño

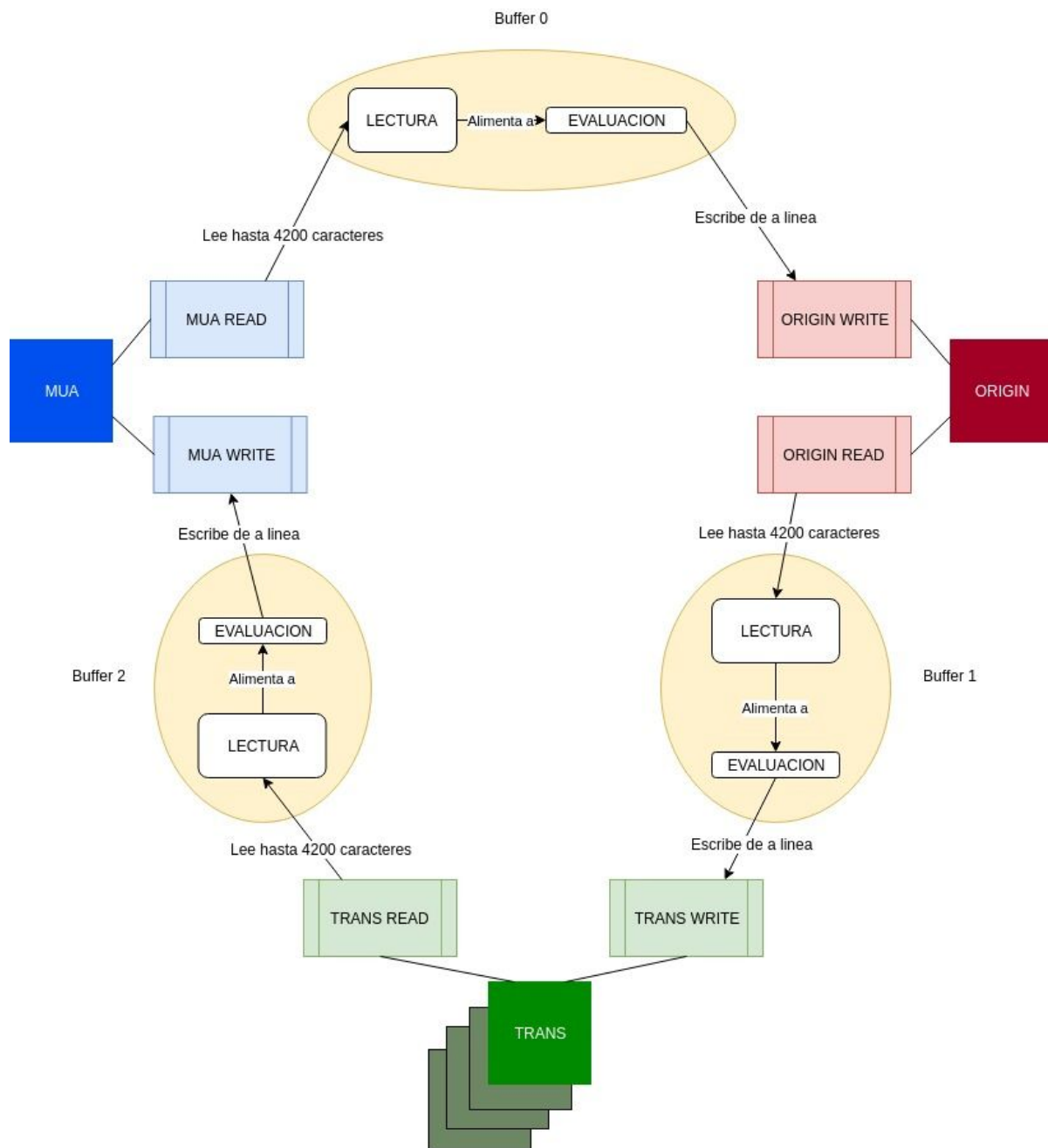


Diagrama de flujo de Pop3Filter para un cliente.

En todas las etapas del proxy la lectura se lleva a cabo de la siguiente manera.

Se utiliza un buffer de lectura de 4200 bytes de tamaño.

Se utiliza un buffer de evaluación de 600 bytes de tamaño.

El buffer de evaluación se alimenta de a líneas del buffer de lectura.

La implementación se completó de esta manera debido a que en un inicio del desarrollo se determinó que hacer las evaluaciones con buffers de tamaño de línea era la forma óptima de simplificar las mismas. El buffer de lectura nos permite mantener ese comportamiento y simultáneamente hacer lecturas grandes menos costosas en tiempo.



# Media Types

## Introducción

El **objetivo** de esta aplicación es filtrar y reemplazar contenido MIME de un mensaje, según lo especificado en la lista de los MIME y el texto de reemplazo.

## Operación Básica

Inicialmente se establece una conexión con el proxy, aquí se puede especificar el mensaje de reemplazo haciendo uso de la opción -m y la lista de los MIME a filtrar, haciendo uso de la opción -M. Luego desde un administrador se establece *mediatypes* como el comando de transformación (FILTER mediatypes) y se activa (ACTIVE 1).

La lista de MIME debe estar separada por comas. Ésta va a ser validada contra una lista preexistente de los MIME válidos. En caso de tener algún MIME inválido, no se llevará a cabo la transformación.

En caso de correr el proxy y no especificar alguno de estos parámetros, se usarán los default de lista vacía y "Replaced Part".

# Problemas encontrados durante el diseño y la implementación

## **Lectura y Escritura bloqueante:**

En la etapa de diseño se planteó la problemática de cómo evitar bloquearse. Se determinó que se podía usar el syscall select filtrando los descriptors según el estado de los buffers:

Si el buffer 0 se encuentra vacío. MUA READ sino ORIGIN WRITE

Si el buffer 1 se encuentra vacío. ORIGIN READ sino TRANSFORM WRITE

Si el buffer 2 se encuentra vacío. TRANSFORM READ sino MUA WRITE.

## **Buffers:**

Durante la etapa de prueba se encontró un problema que imposibilitaba el diseño como había estado planteado. Para lograr las lecturas de a líneas se utilizaba el syscall read con tamaño 1 hasta que se leía el final de la línea. La ventaja de esto es que garantizábamos que el buffer tenga una línea completa y manteníamos el resto del contenido en los pipes para leer en la próxima iteración. Sin embargo esto probó tener un costo enorme en performance a la hora de leer mucha información.

La solución óptima para esto fue agregar otro buffer, de lectura, que garantizaba lecturas de a paquetes de hasta 4200 bytes, y permitía al buffer de línea alimentarse del buffer grande sin tener que cambiar la lógica que habíamos implementado que dependía de evaluaciones de a línea.

## **Detección de Multilíneas y Transformaciones:**

Debido al uso del servidor dovecot y la mala asunción que todos los servidores de origen se comportarían de forma igual se planteó gran parte del Proxy atado a la lógica de respuestas del servidor Dovecot.

Una vez dividido el problema nos encontramos con dos opciones. Parsear los comandos en MUA read y construir una lógica de encolamiento de comandos que estaba lejos de lo que habíamos construido en un principio. O encontrar formas para detectar las respuestas multilínea y de transformación desde READ ORIGIN esta vez de forma general.

Optamos por la segunda opción que era la que afectaba de menor manera al sistema. Esto le agrego dos limitaciones al proxy listadas en la siguiente sección.

## Limitaciones de la aplicación

El saludo inicial al servidor de origen envía el comando “CAPA” para averiguar las aptitudes del mismo. En esta etapa el proxy asume que la respuesta del servidor de origen se envía de a paquetes de por lo menos una línea. De lo contrario el comportamiento del proxy no está especificado.

Se asume que tanto el proxy como el servidor de origen respetan el protocolo de pop3 que establece una longitud máxima de línea de 512 caracteres. De lo contrario el comportamiento del proxy no está especificado.

Se asume que el servidor de origen envía las respuestas del comando CAPA en mayúsculas. De lo contrario el comportamiento del proxy no está especificado.

Se asume que el servidor de origen envía las respuestas de los comandos List y UIDL (multilineas) en el formato “[Dígitos] [String]\r\n”. De lo contrario el comportamiento del proxy no está especificado.

Se asume que el servidor de origen envía las respuestas del comando CAPA en mayúsculas. De lo contrario el comportamiento del proxy no está especificado.

Como una forma de simplificar el diseño de control del administrador se eligió utilizar buffers de tamaño fijo para enviar los mensajes de comandos y respuesta entre el cliente y el proxy. En algunos casos se eligió un tamaño de 64 cuando se sabía que dichos buffers no iban a extenderse mucho y MAX\_FILTER\_BUFFER (256) en casos como el comando de filtración donde el mensaje podía llegar a extenderse más. Adicionalmente, se asumió que no iba a haber más de 10 métricas por lo cual la validación del argumento del comando STATS lo compara con un dígito.

## Prueba de estrés

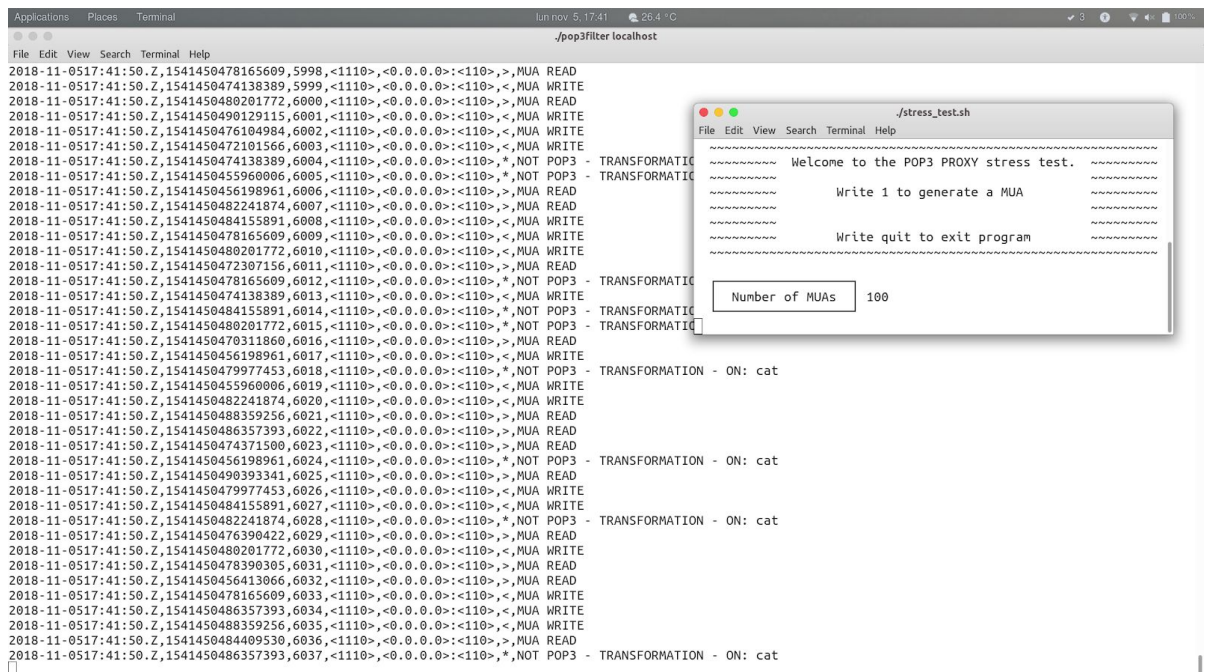
El proxy fue estresado con el programa *stress\_test*, una GUI para generar clientes MUA que hacen requests al servidor constantemente.

Se llegó a probar con 10 clientes autenticados, hasta 100 clientes sin autenticar y 2 admins.

Entendiendo por autenticados. Clientes que pasaron su usuario y contraseña; y están solicitando iterativamente sus mails.

Entendiendo por sin autenticar. Clientes que solicitan iterativamente CAPA al servidor.

Para cualquier estrés mayor a este el comportamiento del Proxy no está especificado



```

Applications  Places  Terminal
File Edit View Search Terminal Help
/pop3filter localhost
2018-11-0517:41:50.Z,1541450478165609,5998,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450474138389,5999,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450480201772,6000,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450490129115,6001,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450476104984,6002,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450472101566,6003,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450474138389,6004,<1110>,<0.0.0.0>:<110>,>,MUA POP3 - TRANSFORMATION
2018-11-0517:41:50.Z,1541450455960006,6005,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION
2018-11-0517:41:50.Z,1541450456198961,6006,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450482241874,6007,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450484155891,6008,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450478165609,6009,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450480201772,6010,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450472307156,6011,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450478165609,6012,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION
2018-11-0517:41:50.Z,1541450474138389,6013,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450484155891,6014,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION
2018-11-0517:41:50.Z,1541450480201772,6015,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION
2018-11-0517:41:50.Z,1541450470311860,6016,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450456198961,6017,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450479977453,6018,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION - ON: cat
2018-11-0517:41:50.Z,1541450455960006,6019,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450482241874,6020,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450480359256,6021,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450486357393,6022,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450474371500,6023,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450456198961,6024,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION - ON: cat
2018-11-0517:41:50.Z,1541450490393341,6025,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450479977453,6026,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450484155891,6027,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450482241874,6028,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION - ON: cat
2018-11-0517:41:50.Z,1541450476390422,6029,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450480201772,6030,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450478390305,6031,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450456413066,6032,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450478165609,6033,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450486357393,6034,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450488359256,6035,<1110>,<0.0.0.0>:<110>,<,MUA WRITE
2018-11-0517:41:50.Z,1541450484409530,6036,<1110>,<0.0.0.0>:<110>,>,MUA READ
2018-11-0517:41:50.Z,1541450486357393,6037,<1110>,<0.0.0.0>:<110>,>,MUA *NOT POP3 - TRANSFORMATION - ON: cat

```

## Posibles extensiones

Siendo esta la versión inicial se puede extender en ciertos aspectos.

- Implementar más métricas y dar mayor detalle sobre estas.
- Mejorar la autenticación de usuario para que se encripte la contraseña y esta no se envíe de forma legible.
- Implementar pipelining en el administrador.
- Implementar comando para bloquear/desbloquear los MIME desde el administrador.
- Implementar un sistema de timeout.

## Conclusiones

La realización del trabajo resultó un desafío, teniéndose que en varias instancias superar obstáculos de difícil resolución. Esto nos llevó a tener que pensar soluciones creativas que no siempre se encontraron rápido.

Además podemos ver en retrospectiva que se tomaron varias decisiones de diseño en etapas iniciales, donde no se tenía tanto conocimiento, que luego de estar más adentrados en los temas se hubiera preferido hacer de otra manera.

## Ejemplos de prueba

Para ejecutar el pop3filter se realiza lo siguiente:

```
$ ./pop3filter 127.0.0.1
```

El mismo se puede ejecutar con opciones:

```
$ ./pop3filter -p 1111 -o 9091 -l ::1 -L 0.0.0.0 127.0.0.1
```

El stripmime se puede ejecutar de forma integrada con el proxy, en tal caso se debería conectar con el proxy, especificando si así se deseara una lista de los MIME a filtrar y el texto a reemplazar y luego pedir un mail para que este sea filtrado.

También es posible ejecutarlo de manera *standalone*. Para realizar esto se debe tener las variables FILTER\_MEDIAS y FILTER\_MSG seteadas:

```
$ export FILTER_MEDIAS=image/png
$ export FILTER_MSG=Parte reemplazada
```

Y luego se podría correr:

```
$ FILTER_MEDIAS=text/plain ./stripmime < mail_a_filtrar
```

## Guia de instalación

Referirse al archivo README.md en la carpeta raíz del repositorio.

# Instrucciones para la configuración

Se encuentran dos usuarios administradores:

- username: juan ; password: juan
- username: marcelo ; password: marcelo

La configuración se realiza a través de la aplicación pop3ctl. En esta aplicación se cuenta con 6 (seis) comandos:

- HELP
- LISTS
- STATS
- FILTER
- ACTIVE
- QUIT

De estos comandos todos menos HELP y QUIT sirven para cambiar la configuración del proxy o para realizar el monitoreo.

FILTER se usa para cambiar el comando de transformación. Éste puede recibir un argumento (*cmd*) en el caso de querer cambiar el comando actual por *cmd*, o bien puede consultarse sin ningún argumento, lo cual informará del comando de transformación actual.

De forma similar se puede utilizar el comando ACTIVE, el cual puede activar/desactivar la transformación si se pasa con un argumento que puede ser 0 o 1, o informar el estado actual de la transformación si se lo utiliza sin argumento.

Finalmente, los comandos LISTS y STATS sirven para el monitoreo. LISTS lista los posibles monitoreos a consultar y STATS *number* accede al monitoreo correspondiente a *number*.

# Ejemplos de configuración y monitoreo

## I. Conexión default

Precondiciones:

- Tener escuchando con una conexión SCTP en el puerto 9090 y dirección 127.0.0.1

Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username:
```

## II. Conexión en otro puerto

Precondiciones:

- Tener escuchando con una conexión SCTP en el puerto 9898 y dirección 127.0.0.1

Se ejecuta:

```
$ ./pop3ctl -p 9898
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username:
```

## III. La autenticación (éxito)

Precondiciones:

- Mismas que en el caso I

Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE

Login succesful
```

## IV. La autenticación (falla)

Precondiciones:

- Mismas que en el caso I

Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
```

```
Please enter your password: foo
-ERR - FAILED. Try again or QUIT.
```

```
Login failed. Do you wish to retry? [Y/n]:
```

## V. El comando HELP

Precondiciones:

- Mismas que en caso I

Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE

Login succesful
HELP
LISTS ----- Lists the possible monitoring options along
with its corresponding number
STATS number --- Returns the monitoring information
corresponding to number (number between 0 and 9)
ACTIVE [BOOL] -- With no BOOL, returns the current
transformation filter's status
                  With BOOL, activates (1) or deactivates (0)
the current filter
FILTER [CMD]---- With no CMD, returns current transformation
filter
                  With CMD, returns current transformation
filter
                  @ escapes all blank spaces after ir
QUIT ----- Closes the connection
HELP ----- Lists the possible input options
```

## VI. El comando LISTS

Precondiciones:

- Mismas que en caso I

Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE
```



```

Login succesful
LISTS
+OK -List:
1 - Connected MUAs
2 - Connected admins
3 - Historical MUA accesses
4 - Total transferred bytes

```

## VII. El comando STATS *number*

Precondiciones:

- Mismas que en caso I

Se ejecuta:

```

$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE

Login succesful
STATS number (donde number es un número del 1 al 4)
+OK junto con un mensaje correspondiente a la métrica
seleccionada y la métrica.

```

## VIII. El comando ACTIVE [BOOL]

Precondiciones:

- Mismas que en caso I

Se ejecuta:

```

$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE

Login succesful
ACTIVE
+OK - Transformation is: Active
ACTIVE 0
+OK - SUCCESS. Transformation is: Inactive
ACTIVE
+OK - Transformation is: Inactive

```

## IX. El comando FILTER [CMD]

### Precondiciones:

- Mismas que en caso I

### Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE

Login succesful
FILTER
+OK - Current transformation: cat
FILTER foo
+OK - SUCCESS. Current transformation: foo
```

## X. El comando QUIT

### Precondiciones:

- Mismas que en caso I

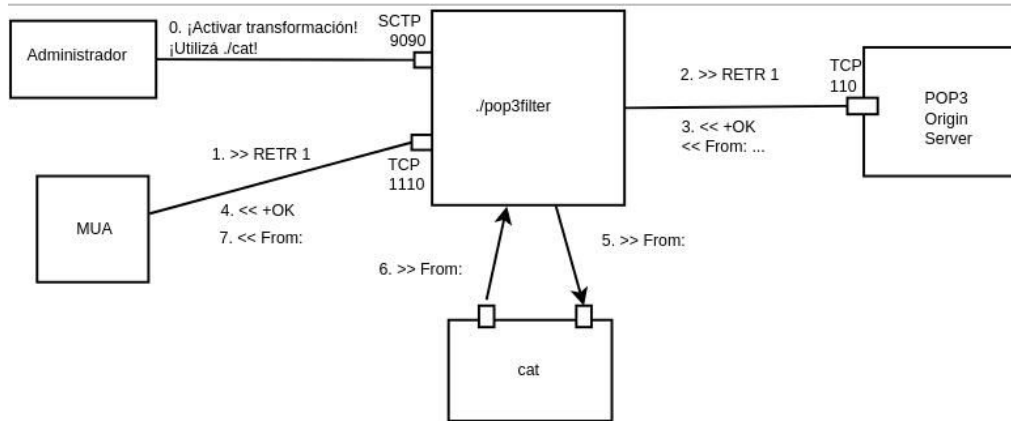
### Se ejecuta:

```
$ ./pop3ctl
Hello! Starting connection...
+OK Greetings Earthlings. Now log in.
Please enter your username: juan
Please enter your password: juan
+OK - Entering EXCHANGE

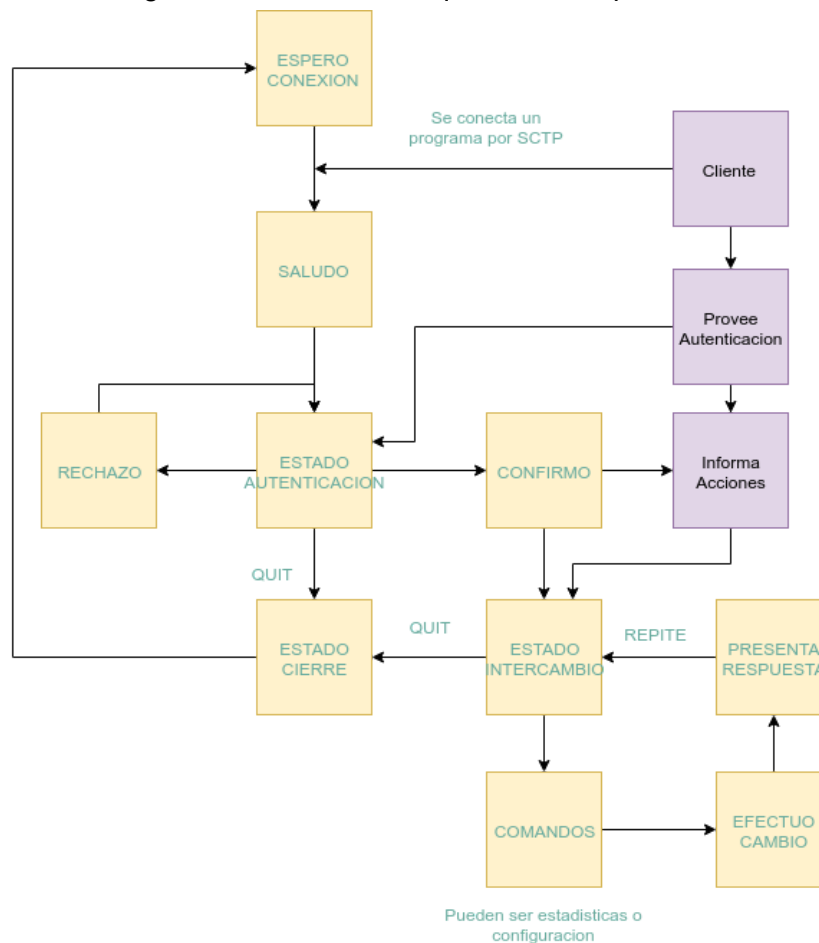
Login succesful
QUIT
+OK - Quitting
Goodbye, hope to see you soon!
```

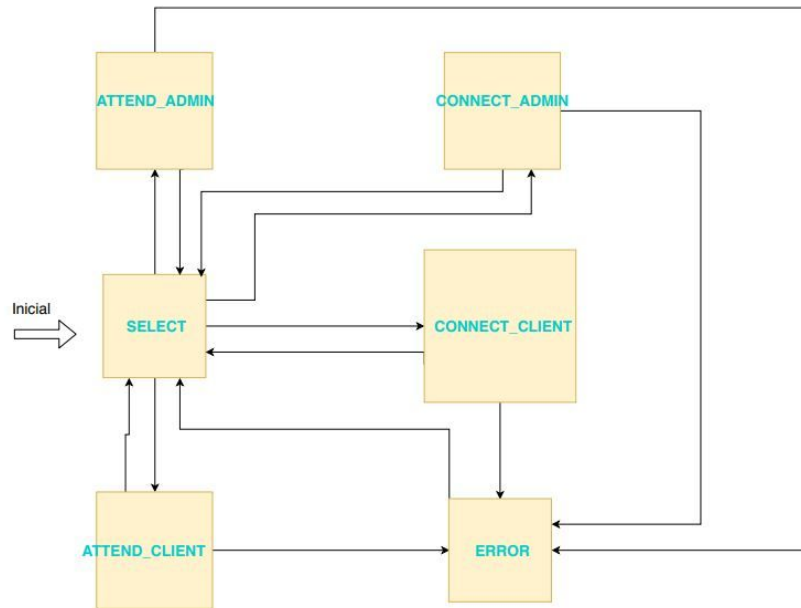
# Documento de diseño del proyecto

De forma general se puede utilizar el diagrama visto en clase para entender el proyecto.



En éste se puede ver la interacción entre las distintas secciones del trabajo. Luego, los siguientes diagramas informan el esquema inicial planteado:





### Estados internos de ATTEND\_CLIENT

