

REDES DE COMPUTADORAS

CURSO 2019

GRUPO 51

Informe - Obligatorio 2

Autores:

Tatiana Rischewski
Manuel Freire
Juan Ferrand

Supervisores:

Martín Giachino
Federico Rodriguez

15 de setiembre de 2019

Contenido

Contenido	2
Parte A - Protocolo del info server	3
Mensajes por parte del cliente	3
Protocolo de respuesta del servidor	3
Parte B - Funcionamiento del info server	4
Parte C - Funcionamiento del proxy	6
Parte D - Mejoras propuestas a la comunicación	8
Referencias	10

Parte A - Protocolo del info server

El servidor “info server” recibe pedidos utilizando el protocolo de transporte UDP [1] en el puerto 9876 y procesa de a uno a la vez secuencialmente según el orden en el que los recibe.

Mensajes por parte del cliente

El cliente es quien empieza la comunicación con un mensaje cuya estructura es:

<tag> <nombre usuario>

Descomponiendo por ítems el tag es un identificador alfanumérico único que identifica el mensaje, separado por un espacio está el nombre de usuario es la información que se desea chequear para ingresar al servidor y por último un \0.

Ejemplo de un mensaje correcto:

a01 matias\0

Ejemplo de un mensaje incorrecto:

matias

Protocolo de respuesta del servidor

El servidor tiene tres posibles respuestas para cualquier pedido, las cuales son las IPs de los dos posibles servidores a los que se puede conectar, ERROR en caso de que el usuario no esté en la base de datos y FORMATO ERROR en caso de que el formato del mensaje fuese distinto al s.

La estructura del mensaje es la siguiente:

<tag> <respuesta>

El tag es el mismo que recibió por parte del cliente e identifica el mensaje mientras que la respuesta es una de las mencionadas anteriormente. Por último se envía un espacio.

Ejemplos de respuestas a mensajes correctos:

a01 192.168.56.102

a01 ERROR

Ejemplo de respuesta a mensaje incorrecto:

a01 FORMATO ERROR

Parte B - Funcionamiento del info server

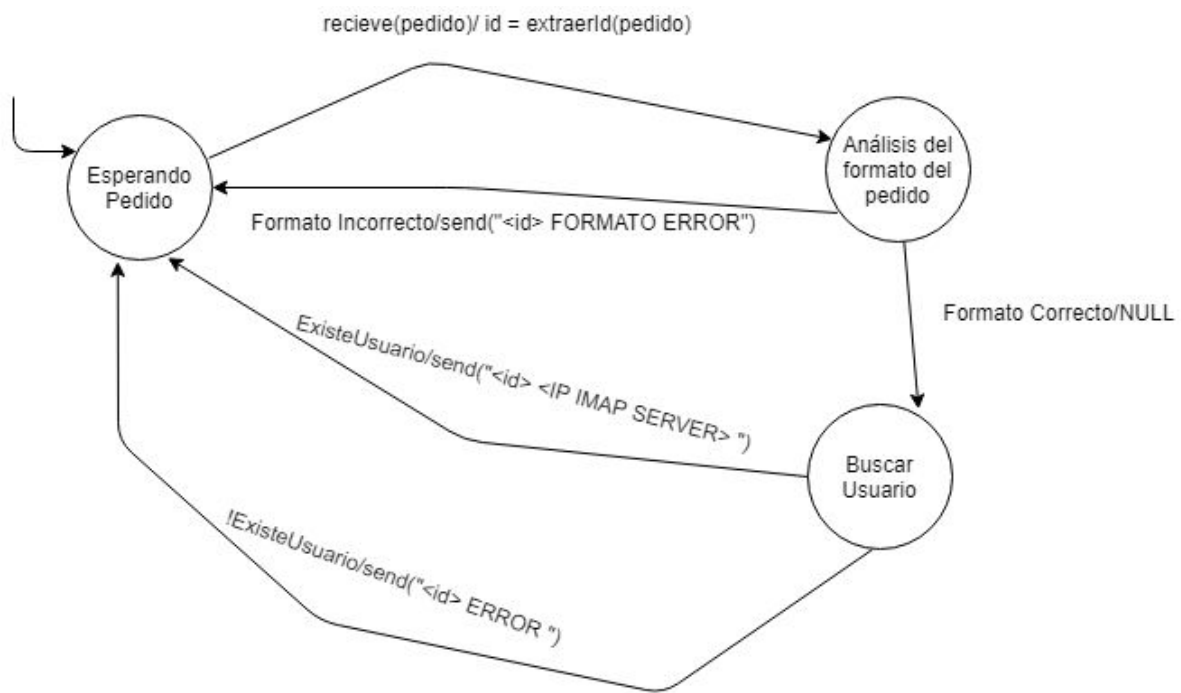


Figura 1: Info server

El info server inicializa y queda escuchando por pedidos de clientes en el puerto 9876 mediante el protocolo UDP. Cuando llega un pedido, el info server extrae el identificador del mensaje que luego se usará como identificador en la respuesta que mande. Luego intenta extraer el nombre del usuario, si detecta que falta el nombre del usuario el servidor responde con el mensaje `<id> FORMATO INCORRECTO`. Si puede extraer el nombre del usuario, busca el usuario en la lista y obtiene el IP del server IMAP. Luego crea el mensaje de respuesta con el tag del mensaje recibido y el IP que corresponde al usuario. En caso que el usuario no se encuentre, devuelve un mensaje con la palabra ERROR de la forma `<id> ERROR`.

El `info_server` es un programa relativamente sencillo que busca atender los pedidos lo más rápido posible. Para lograr una respuesta rápida, y dada la baja cantidad de usuarios que existe actualmente, se decidió por mantener todos los usuarios dentro del programa, facilitando el acceso a ellos. Considerando que todo pedido se responde muy rápido, también se optó por mantener el pedido y la generación de respuesta dentro de un hilo solo. En caso de que la cantidad de usuarios incremente, se puede modificar el programa, quitarlos del código y guardarlos en un documento aparte donde el programa podría buscarlos. Este cambio mantendrá el programa pequeño pero bajará la eficiencia en la que responde.

los pedidos. Sacar los usuarios del programa también llevaría a que trabajar con un solo thread no sea lo más eficiente y podría llevar a perder varios pedidos que lleguen mientras este intenta atender a uno.

Parte C - Funcionamiento del proxy

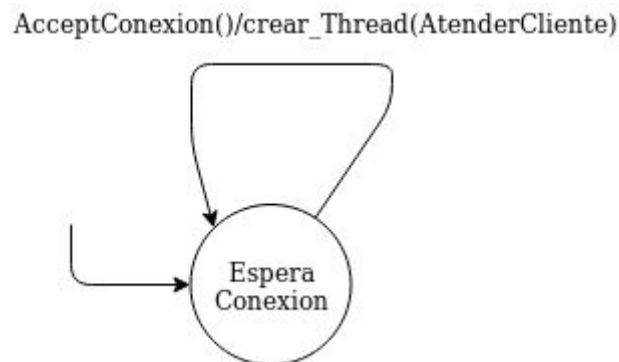


Figura 2: Main Thread

El proxy acepta conexiones TCP [2] en el puerto 3490. Por cada conexión que acepta genera un nuevo hilo para procesar la solicitud. El proxy envía el mensaje de bienvenida “* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE AUTH=PLAIN] Dovecot ready.\n”, y espera por la respuesta del cliente. En el caso de que el comando enviado por el cliente no sea un login, el proxy envía una respuesta imap adecuada. En caso de recibir el comando login con el formato incorrecto, como por ejemplo sin incluir username, se envía el mensaje imap correspondiente al cliente. En caso de recibir el comando login con el formato correcto se genera un id único para ese thread y se envía el mensaje “id username” al servidor info_server (mensaje UDP en el puerto 9876), siendo username el argumento pasado por el usuario en el comando login. Dado que la conexión entre el proxy y el info_server es de tipo UDP, el proxy tiene definido un timeout de 100 ms. Si no recibe respuesta del info_server, o si recibe una respuesta con un identificador diferente al que envió, el proxy vuelve a intentar hasta un total de 10 veces. En caso que el proxy no reciba respuesta dentro de los 10 intentos, se cierra la conexión con el cliente.

Cuando recibe una respuesta que corresponde al pedido realizado, el proxy verifica la respuesta que le envió el info_server. En caso de ser ERROR el proxy manda el mensaje que corresponde al cliente (indicando que fue rechazado el nombre de usuario o la contraseña) y espera por el siguiente comando del cliente. El proxy nunca recibe la respuesta FORMATO ERROR porque siempre le envía mensajes al info server con el formato correcto. En otro caso, significa que el nombre de usuario ingresado se corresponde a uno de los usuarios en alguno de los servidores imap y el info_server le responde con la dirección ip del servidor.

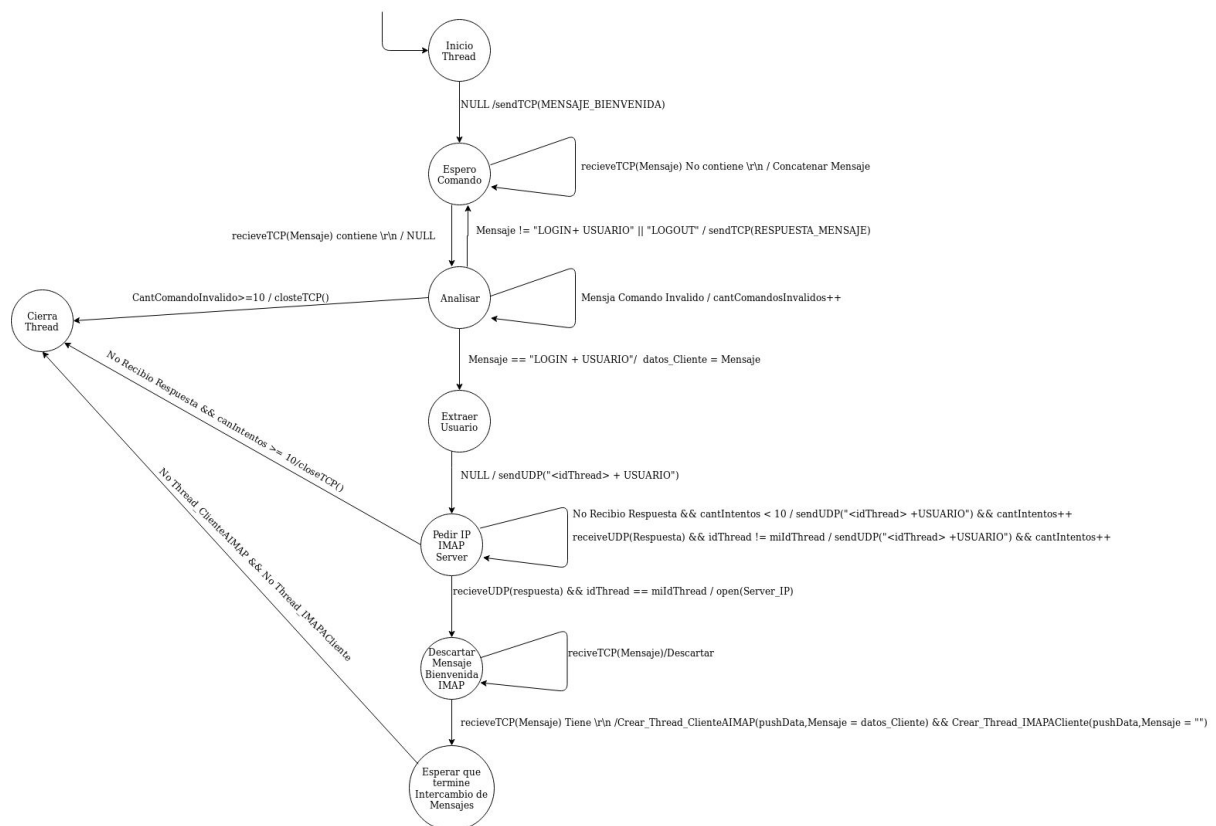


Figura 3: Atender cliente

Luego el proxy establece una conexión TCP con el servidor imap correspondiente. Recibe y descarta el mensaje de bienvenida enviado por el servidor imap [3] y genera dos nuevos threads: uno para enviar todos los mensajes que recibe del imap al cliente, y otro para enviar todos los mensajes que recibe del cliente al imap. Ambos threads continúan funcionando hasta que alguno de los dos (el imap o el cliente) cierre la conexión. En dicho caso ambos threads terminan y se cierran los sockets correspondientes.

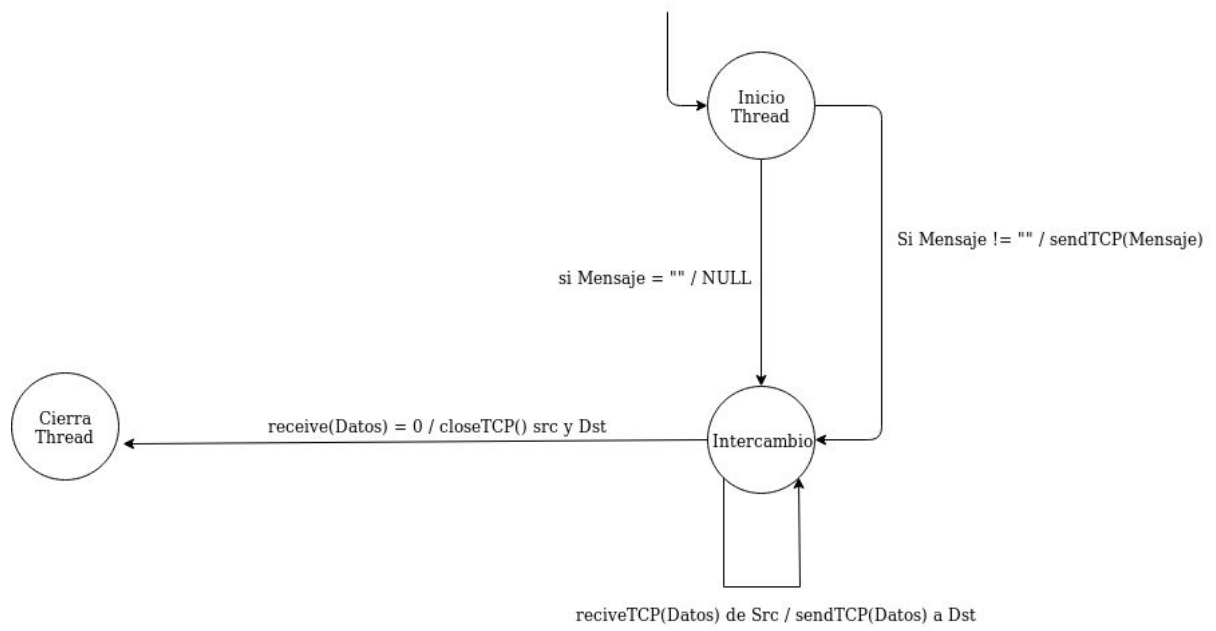


Figura 4: Push data

Parte D - Mejoras propuestas a la comunicación

La comunicación podría mejorarse utilizando un control de pérdidas que permita asegurar que el mensaje llegará. Así se evita que pudiesen darse casos en el que al intentar varias veces sin respuesta el proxy define que el info server está caído y cierra la conexión con el cliente. También se podría incorporar un sistema de cacheo en el proxy para no tener que consultar siempre al servidor lo cual en caso de escalar se provocaría una gran masa de pedidos al mismo que no atiende de forma concurrente lo que generaría delay.

Referencias

- [1] Internet Engineering Tasking Force. (1980, 28 agosto). RFC 768 - User Datagram Control. Recuperado 15 septiembre, 2019, de <https://tools.ietf.org/html/rfc768>
- [2] Internet Engineering Tasking Force. (1981, septiembre). RFC 793 - Transmission Control Protocol. Recuperado 15 septiembre, 2019, de <https://tools.ietf.org/html/rfc793>
- [3] Internet Engineering Tasking Force. (2003, marzo). RFC 3501 - INTERNET MESSAGE ACCESS PROTOCOL. Recuperado 15 septiembre, 2019, de <https://tools.ietf.org/html/rfc3501>