



Protocolos de Comunicación - 72.07

# **Trabajo Práctico Especial**

## **RFC PROTOS**

Profesores:

Codagnone, Juan Francisco

Garberoglio, Marcelo Fabio

Kulesz, Sebastian

Grupo 9:

Francois, Gaston Ariel (62500)

Mentasti, José Rodolfo (62248)

Preiti Tasat, Axel Facundo (62618)

Suarez Durrels, Nicolas (62468)

## 1 Estado del informe

El presente informe presenta un protocolo de comunicación para monitoreo y configuración de un servidor pop3 y asume que el usuario se encuentra familiarizado con los conceptos distribuidos en los RFC complementarios "Post Office Protocol - Version 3" [[RFC 1939](#)] y "POP3 Extension Mechanism" [[RFC 2449](#)].

<b>1 Estado del informe.....</b>	<b>2</b>
<b>2 Introducción.....</b>	<b>3</b>
2.1 Convenciones del documento.....	3
<b>3 Parámetros y gramática general.....</b>	<b>4</b>
3.1 Esquema de solicitud.....	4
3.2 Esquema de respuesta.....	5
<b>4 Set de instrucciones.....</b>	<b>6</b>
4.1 ADD_USER.....	6
4.2 GET_MAX_MAILS.....	7
4.3 SET_MAX_MAILS.....	8
4.4 GET_MAILDIR.....	9
4.5 SET_MAILDIR.....	10
4.6 STAT_HISTORIC_CONNECTIONS.....	11
4.7 STAT_CURRENT_CONNECTIONS.....	12
4.8 STAT_BYTES_TRANSFERRED.....	13
<b>5 Transporte.....</b>	<b>14</b>
<b>6 Extensiones a futuro.....</b>	<b>14</b>
<b>7 Autores y soporte adicional.....</b>	<b>15</b>
<b>8 Referencias.....</b>	<b>15</b>

## 2 Introducción

El protocolo PROTOS es un protocolo orientado a texto y transportado mediante UDP, diseñado para el manejo de la configuración de un servidor pop3 durante su ejecución y la obtención de estadísticas de uso del servidor. En gran medida se propone el uso de este protocolo para aplicaciones clientes o programas de monitoreo de servidores pop3.

### 2.1 Convenciones del documento

Acerca de las convenciones de notación, durante el informe se hará referencia a parámetros de funciones utilizando los delimitadores "< >" para indicar el campo del argumento. En tales casos, los argumentos a utilizar no deben ser colocados entre tales símbolos. Por otro lado, la utilización de "[ ]" como delimitador se utilizará para parámetros que resultan opcionales y se aplicará la misma convención que en el caso anterior.

Una segunda aclaración proviene del uso de "/" para la muestra de dos posibles opciones. En estos casos la notación indica que dentro de ese argumento uno de los dos valores tomará el lugar efectivamente y no el otro.

Acerca de las convenciones de términos, durante el informe se hará referencia a los términos siguientes

Caracter numérico:	Cualquier ASCII entre '0' y '9'
Valor numérico:	Sucesión de caracteres numéricos.
Caracter alfanumérico:	Cualquier ASCII entre 'a' y 'z', 'A' y 'Z', sumado a los Caracteres numéricos.
Token:	Sucesión de caracteres alfanuméricos.
String:	Sucesión de caracteres alfanuméricos intercalados con espacios " " (ASCII 32), guiones bajos "_" (ASCII 95) o barras "/" (ASCII 47).

### 3 Parámetros y gramática general

La forma general de los comandos y respuestas de PROTOS se pueden modelar teniendo en cuenta los siguientes esquemas, donde cada elemento se encuentra delimitado por el separador "\n" y la sucesión de dos líneas en blanco (dos separadores de argumentos) como indicador de final de comando.

Tras determinar a la sucesión "\n" como separador de elementos de la solicitud, se tiene un límite máximo de 128 bytes para cada argumento. En cuanto al tamaño máximo del datagrama se cuenta con la limitación de 1024 bytes como tamaño máximo. Sobre estos datos se expondrá en la sección de transporte del presente documento.

El orden de los elementos del encabezado se encuentra diagramado para no realizar lecturas innecesarias sobre los datos. En primer lugar se cuenta con el protocolo en cuestión e inmediatamente un token de sesión para la validación de la solicitud. Posteriormente se cuenta con la versión, la cual determina cómo será el resto de la solicitud y cómo se debe interpretar su contenido. En cuanto a la respuesta solo se mantiene la versión para la interpretación del contenido de la respuesta.

#### 3.1 Esquema de solicitud

A la hora de la solicitud se cuenta con el siguiente esquema.

---

PROTOS	
<versión>	Valor numérico indicando la versión
<token>	Token de verificación
<id>	Valor numérico identificador del comando
<comando>	String del comando que se desea ejecutar
<arg1>	Primer argumento del comando alfanumérico
...	
<argN>	Argumento número N del comando
	Primer línea en blanco
	Segunda línea en blanco

---

### 3.2 Esquema de respuesta

En cuanto a la respuesta esperada, su formato general cuenta con la siguiente descripción

---

PROTOS	
<versión>	Valor numérico indicando la versión
<id>	Valor numérico identificador del comando
<+/->	Identificador del estado de la respuesta
<mjs>	Mensaje de respuesta, opcional según comando
	Primer línea en blanco
	Segunda línea en blanco

---

En cuanto al mensaje de respuesta, en cada comando se indicará si el mensaje resulta obligatorio o si resulta opcional.

#### 4 Set de instrucciones

A continuación se listan los comandos disponibles del protocolo junto con ejemplos de utilización.

##### 4.1 ADD\_USER

El comando permite el agregado de un nuevo usuario al servidor pop3.

##### Esquema del comando

```
-----  
  PROTOS  
  1  
  <token>  
  <id>  
  ADD_USER  
  <usuario>  
  <contraseña>  
-----
```

##### Esquema de la respuesta

```
-----  
  PROTOS  
  1  
  <id>  
  <+/->  
  [mjs]  
-----
```

##### Argumentos del comando:

<usuario>	String indicador del nombre del usuario.
<contraseña>	String indicador de la contraseña del usuario.

##### Posible respuesta:

+	El usuario fue correctamente agregado
-	El usuario no pudo ser agregado

##### Argumentos de la respuesta:

[mjs]	Posible mensaje de respuesta en caso de fallo.
-------	--

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir al usuario como agregado. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible la adición de un nuevo usuario. No se cuenta con mensajes diferenciadores en caso de que el usuario ya esté inicializado, no se cuenta con suficiente memoria u otros posibles fallos intermedios, aunque se permite que la implementación lo indique en el campo [msj].

#### 4.2 GET\_MAX\_MAILS

El comando permite obtener el número máximo de mails a listar por cada usuario estipulado por el servidor pop3.

Esquema del comando

```
-----  
PROTOS  
1  
<token>  
<id>  
GET_MAX_MAILS  
-----
```

Esquema de la respuesta

```
-----  
PROTOS  
1  
<id>  
<+/->  
<valor>  
-----
```

Argumentos del comando:

No requiere argumentos.

Posible respuesta:

+	El comando fue correctamente procesado.
-	El comando no fue correctamente procesado.

Argumentos de la respuesta:

<valor>	Valor numérico indicador de la cantidad de mails a listar.
---------	--

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva y el argumento resulta fiable. Ante un valor de respuesta "-" se asume que resultó imposible la obtención del valor. En este caso, el servidor puede enviar la razón del error en lugar del valor.

#### 4.3 SET\_MAX\_MAILS

El comando permite modificar el número máximo de mails a listar por cada usuario estipulado por el servidor pop3.

Esquema del comando

```
-----  
PROTOS  
1  
<token>  
<id>  
SET_MAX_MAILS  
<valor>
```

Esquema de la respuesta

```
-----  
PROTOS  
1  
<id>  
<+/->  
[mjs]
```

Argumentos del comando:

<valor> Valor numérico esperado a ser el nuevo máximo.

Posible respuesta:

+ Se pudo modificar el máximo de mails a  
listar.  
- No se pudo modificar el máximo de mails a  
listar.

Argumentos de la respuesta:

[mjs] Posible mensaje de respuesta en caso de fallo.



Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir que el máximo fue modificado. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible la modificación del máximo. No se cuenta con mensajes diferenciadores del error ocurrido, aunque se permite que la implementación lo indique en el campo [msj].

#### 4.4 GET\_MAILDIR

El comando permite obtener el path del servidor donde se pretende acceder a los mails de los usuarios. En otras palabras, se espera recibir el maildir para todos los usuarios del servidor.

Esquema del comando

```
-----
  PROTOS
  1
  <token>
  <id>
  GET_MAILDIR
-----
```

Esquema de la respuesta

```
-----
  PROTOS
  1
  <id>
  <+/->
  <path>
-----
```

Argumentos del comando:

No requiere argumentos.

Posible respuesta:

+	Se pudo obtener el path al maildir.
-	No se pudo obtener el path al maildir.

Argumentos de la respuesta:

<path>	String del path al maildir del servidor.
--------	--

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir que el valor resulta el path al maildir. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible obtención del path. En dicho caso, el servidor puede enviar un mensaje indicando el error en lugar del path.

#### 4.5 SET\_MAILDIR

El comando permite modificar el path del servidor donde se pretende acceder a los mails de los usuarios. En otras palabras, se espera modificar el maildir para todos los usuarios del servidor.

Esquema del comando

```
-----  
PROTOS  
1  
<token>  
<id>  
SET_MAILDIR  
<maildir>  
  
-----
```

Esquema de la respuesta

```
-----  
PROTOS  
1  
<id>  
<+/->  
[mjs]  
  
-----
```

Argumentos del comando:

<maildir>           String indicador del path al maildir.

Posible respuesta:

+                   Se pudo modificar el path al maildir.  
-                   No se pudo modificar el path al maildir.

Argumentos de la respuesta:

[mjs]               Posible mensaje de respuesta en caso de fallo.

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir que el valor del maildir fue modificado. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible la modificación del path. No se cuenta con mensajes diferenciadores del error ocurrido, aunque se permite que la implementación lo indique en el campo [msj].

#### 4.6 STAT\_HISTORIC\_CONNECTIONS

El comando permite obtener el número de conexiones/sesiones de usuario que se realizaron desde que el servidor pop3 se encuentra en ejecución.

Esquema del comando

```
-----  
  PROTOS  
  1  
  <token>  
  <id>  
  STAT_HISTORIC_CONNECTIONS  
-----
```

Esquema de la respuesta

```
-----  
  PROTOS  
  1  
  <id>  
  <+/->  
  <valor>  
-----
```

Argumentos del comando:

No requiere argumentos.

Posible respuesta:

+	Se pudo obtener el número de conexiones.
-	No se pudo obtener el número de conexiones

Argumentos de la respuesta:

<valor>	Valor numérico de la cantidad de conexiones.
---------	--

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir que el valor resulta del número de conexiones históricas. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible obtención del número de conexiones. El servidor puede enviar un mensaje indicando la razón del error en lugar del valor.

#### 4.7 STAT\_CURRENT\_CONNECTIONS

El comando permite obtener el número de sesiones de usuario activas al momento del comando.

Esquema del comando

```
-----  
  PROTOS  
  1  
  <token>  
  <id>  
  STAT_CURRENT_CONNECTIONS
```

Esquema de la respuesta

```
-----  
  PROTOS  
  1  
  <id>  
  <+/->  
  <valor>
```

Argumentos del comando:

No requiere argumentos.

Posible respuesta:

+	Se pudo obtener el número de conexiones.
-	No se pudo obtener el número de conexiones

Argumentos de la respuesta:

<valor>	Valor numérico de la cantidad de conexiones.
---------	--

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir que el valor resulta del número de conexiones actuales. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible obtención del número de conexiones. El servidor puede enviar un mensaje indicando la razón del error en lugar del valor.

#### 4.8 STAT\_BYTES\_TRANSFERRED

El comando permite obtener el número de bytes transferidos desde que el servidor se encuentra en ejecución, es decir la cantidad de bytes que salieron del servidor.

Esquema del comando

```
-----  
  PROTOS  
  1  
  <token>  
  <id>  
  STAT_BYTES_TRANSFERRED
```

Esquema de la respuesta

```
-----  
  PROTOS  
  1  
  <id>  
  <+/->  
  <valor>
```

Argumentos del comando:

No requiere argumentos.

Posible respuesta:

+        Se pudo obtener el número de bytes transferidos.  
-        No se pudo obtener el número de bytes  
         transferidos.

Argumentos de la respuesta:

<valor>    Valor numérico de la cantidad de bytes  
         transferidos.

Interpretación de los resultados:

Ante un valor de "+" la respuesta resulta inequívocamente positiva, se puede asumir que el valor resulta del número de bytes transferidos. Ante un valor de respuesta "-" solo se puede asumir que resultó imposible obtención del número de bytes. El servidor puede enviar un mensaje indicando la razón del error en lugar del valor.

## 5 Transporte

El protocolo PROTOS asume que los mensajes se transmitirán como datagramas mediante UDP [[RFC-768](#)] en el puerto UDP 1024 (decimal). Los mensajes transportados están restringidos a 1024 bytes (sin contar los encabezados UDP). Los mensajes más largos no se fragmentan sino que se envían con el máximo disponible, en caso de superar ese máximo el comando fallará.

Resulta destacable que las consultas enviadas mediante UDP pueden perderse y, por lo tanto, se requiere una estrategia de retransmisión o aviso al usuario. El cliente y el servidor pueden y se recomienda implementar un sistema de detección de timeout y considerar la utilización del argumento <id> para diferenciar los comandos que fueron perdidos y los que fueron correctamente recibidos.

## 6 Extensiones a futuro

El protocolo PROTOS nace como un protocolo de monitoreo y configuración, por lo que se promueve la extensión del protocolo para añadir más funcionalidades de gestión sobre el servidor. En este sentido la adopción de contar con la versión en el encabezado de la solicitud y de la respuesta brindan la posibilidad de incluso modificar los comandos existentes al momento. Un mejor manejo de usuarios y códigos de error para parametrización de errores pueden resultar extensiones naturales del protocolo.

## 7 Autores y soporte adicional

En caso de querer contactar a los autores del protocolo.

Axel Facundo Preiti Tasat  
Instituto Tecnológico de Buenos Aires  
[apreititasat@itba.edu.ar](mailto:apreititasat@itba.edu.ar)

Gaston Ariel Francois  
Instituto Tecnológico de Buenos Aires  
[gfrancois@itba.edu.ar](mailto:gfrancois@itba.edu.ar)

Jose Rodolfo Mentasti  
Instituto Tecnológico de Buenos Aires  
[jmentasti@itba.edu.ar](mailto:jmentasti@itba.edu.ar)

Nicolas Suarez Durells  
Instituto Tecnológico de Buenos Aires  
[nsuarezdurrells@itba.edu.ar](mailto:nsuarezdurrells@itba.edu.ar)

## 8 Referencias

- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol -- Version 3", STD 53, [RFC 1939](#), May 1996.
- [RFC2449] R. Gellens, C. Newman, L. Lundblade, "POP3 Extension Mechanism" [RFC 2449](#), November 1998
- [RFC0768] J. Postel, "User Datagram Protocol", [RFC 0768](#), August 1980.