

# DAJT Protocol

## 1. Introducción

El Protocolo de Oficina de Correos versión 3, o mejor conocido como *Post Office Protocol versión 3* (POP3) ha servido como un estándar adoptado frecuentemente para recuperar mensajes de correo electrónico desde un servidor remoto. Sin embargo, no ofrece una forma de editar la configuración del servidor ni monitorear métricas en tiempo real.

El DAJT es un protocolo orientado a texto basado en TCP diseñado para potenciar la funcionalidad de un servidor POP3. DAJT permite configurar y monitorear en tiempo real métricas asociadas con un servidor POP3, lo que permite a los administradores gestionar y optimizar eficientemente sus sistemas de correo electrónico.

### 1.1. Consideraciones

En este documento, cuando se define la sintaxis de un comando o respuesta, sus argumentos serán representados por su nombre clave entre símbolos '[' y ']'. Es decir, la cadena "[clave]" deberá ser reemplazada por el argumento.

Por otro lado, las palabras "DEBE", "NO DEBE", "OBLIGATORIO", "DEBERÁ", "NO DEBERÁ", "DEBERÍA", "NO DEBERÍA", "RECOMENDADO", "PUEDE" y "OPCIONAL" en este documento serán interpretadas como se describe en el RFC 2119.

## 2. Funcionamiento Básico

El servidor comienza el servicio DAJT escuchando en el puerto TCP 9090. Cuando un host cliente intenta utilizarlo, lo hace estableciendo una conexión TCP. Luego, el cliente puede enviar comandos a los cuales el servidor responderá. La transacción termina cuando la conexión se cierra o se aborta.

Hay únicamente dos estados, AUTHENTICATION y TRANSACTION. En el primero, únicamente están disponibles los comandos "AUTH" y "QUIT", y en el segundo los comandos "TTRA",

"TRAN", "BUFF", "STAT" y "QUIT". Todos ellos serán descritos más adelante en este documento. Todas las conexiones comienzan en el estado de AUTHENTICATION.

El servidor DAJT DEBE admitir múltiples conexiones de diferentes clientes simultáneamente. Las implementaciones PUEDEN admitir el encolamiento de comandos. El servidor PUEDE utilizar PIPELINING. Su uso se recomienda para sacarle provecho al comando STAT, ya que trabaja con estadísticas puntuales.

## 2.1. Formato de los comandos

Todos los comandos tienen 4 letras, la mayoría reciben 0, 1 o 2 argumentos, salvo excepciones. Los argumentos se separan con espacios. Todos los comandos finalizan con un CRLF ("\r\n").

El formato general de los comandos y respuestas se describen utilizando ABNF (RFC 2234).

Comandos DAJT:

```
command    = keyword *(SP param) CRLF ; 85 octetos
              máximo
keyword     = 4VCHAR
param      = 1*VCHAR
```

## 2.2. Formato de las respuestas

La mayoría de las respuestas son de una sola línea, salvo excepciones en las que se muestra un ">OKE" seguido de información importante para el usuario. En cualquier caso, cada línea debe terminar con un CRLF.

Las respuestas pueden ser positivas o negativas. Si son del primer tipo, comienzan con un símbolo '>', mientras que, si son del segundo, lo hacen con un '<'.

Error de comando desconocido

Un "<ERR" seguido de alguna información, puede observarse cuando se proporciona un comando inválido (es decir, cuando el comando ingresado no coincide con ninguno de los listados en la sección 6.) Esto también

incluye el caso en el que el comando enviado por el cliente está vacío.

#### Cantidad de parámetros

Para todos los comandos, si estos reciben una respuesta al ser ejecutados, este valor puede restablecerse cuando el servidor se apaga.

#### Respuestas DAJT:

```
response = status / multi-line
multi-line = status CRLF (text / number / commands)
```

#### CRLF

```
status = ok / err
ok = ">OKE" [text] CRLF
err = "<ERR [text] CRLF
number = 1*DIGIT
text = 1*schar
schar = %x21-5A / %x5C-7F
commands = *(command separator) command
command = pipeline / simple-command
pipeline = simple-command *( "|" simple-command )
simple-command = verb [options] [arguments]
[redirection]
verb = 1*ALPHA
options = *( "-" 1*ALPHA )
arguments = *word
word = 1*(ALPHA / DIGIT)
redirection = *( redirect-operator filename )
redirect-operator = "<" / ">" / "2>" / ">>" / "2>>" /
"<>"
filename = 1*(ALPHA / DIGIT / "." / "-" / "_")
separator = SP / CRLF
```

### 3. El estado AUTHENTICATION

Cuando se establece la conexión TCP, el protocolo DAJT inicia en el estado AUTHENTICATION. En este estado, el cliente solo puede utilizar los comandos "AUTH" y "QUIT", cuyo funcionamiento se explicará luego.

Los comandos disponibles en este estado son:

AUTORIZACIÓN (AUTH)

Sintaxis:

AUTH [usuario] [contraseña]

Argumentos:

Usuario y contraseña (obligatorios).

Comando válido en los estados:

AUTHORIZATION

Discusión:

Este comando debe autorizar al usuario dentro del servidor de DAJT luego de mostrar una respuesta positiva.

Es NO RECOMENDADO que los usuarios y contraseñas del protocolo DAJT sean los mismos que los de POP3.

Posibles respuestas:

ok  
err

Ejemplos:

C: AUTH usuario contraseña  
S: >OKE Autenticación exitosa!

C: AUTH usuario contraseña  
S: <ERR Autenticación inválida!

CIERRE DE CONEXIÓN DEL SERVIDOR (QUIT)

Sintaxis:

QUIT

Argumentos:

Ninguno.

Comando válido en los estados:

AUTHORIZATION, TRANSACTION

Discusión:

Este comando debe cerrar la conexión luego de mostrar una respuesta positiva.

Posibles respuestas:

ok

Ejemplos:

C: QUIT  
S: >OKE

#### 4. El estado TRANSACTION

Luego de que el usuario se autorice, se pasa al estado TRANSACTION. Cualquier comando que sólo pertenezca a este estado y se utilice en AUTHORIZATION, mostrará un mensaje de error.

MOSTRAR EL NOMBRE DEL TRANSFORMADOR ACTUAL O ASIGNAR UNO (TRAN)

Sintaxis:  
TRAN [commands]

Argumentos:  
comandos (opcional) que se ejecutan para hacer la transformación.

Comando válido en los estados:  
TRANSACTION

Discusión:  
Este comando establece un nuevo transformador en caso que se le pase uno por argumento, o muestra el transformador actual en caso de que no se haya enviado ninguno.

Posibles respuestas:  
ok [secuencia de comandos]

Ejemplos:  
C: TRAN /usr/bin/tr a-z A-Z  
S: >OKE  
C: TRAN  
S: >OKE  
S: /usr/bin/tr a-z A-Z

HABILITACIÓN Y DESHABILITACIÓN DE LAS TRANSFORMACIONES (TTRA)

Sintaxis:  
TTRA [número]

Argumentos:

Número (opcional): 1 para activar las transformaciones, 0 para desactivarlas, sin argumentos para mostrar si están activadas o no.

Comando válido en los estados:

TRANSACTION

Discusión:

Este comando DEBE habilitar o deshabilitar las transformaciones globalmente y mostrar una respuesta positiva en su uso correcto. En caso de ingresar algo distinto a 0 o 1, se mostrará un mensaje de error. El uso de TTRA necesita haber definido una transformación antes, se mostrará un mensaje de error si no es el caso.

Posibles respuestas:

ok [0/1]  
err

Ejemplos:

```
C: TRAN /usr/bin/tr a-z A-Z
S: >OKE
C: TTRA 1
S: >OKE
C: TTRA
S: >OKE
S: >1
C: TTRA 0
S: >OKE
C: TTRA
S: >OKE
S: >0
C: TTRA 2
S: <ERR Uso inválido!

C: TTRA 1
S: <ERR No se definió una transformación!
```

MOSTRAR O MODIFICAR EL TAMAÑO DEL BUFFER (BUFF)

Sintaxis:

BUFF [nuevo tamaño del buffer]

Argumentos:

tamaño del buffer (opcional) es un número representado en bytes entre 128B y 1MB (mínimo y máximo OPCIONAL y RECOMENDADO).

Comando válido en los estados:

TRANSACTION

Discusión:

Este comando DEBE mostrar o modificar el tamaño del buffer en su uso correcto. Para modificar el tamaño del buffer se debe ingresar un entero como argumento, mientras que si únicamente se quiere visualizar el tamaño actual, se debe ejecutar el comando sin argumentos.

Posibles respuestas:

ok [tamaño del buffer]  
err

Ejemplos:

```
C: BUFF 1234
S: >OKE
C: BUFF
S: >OKE
S: 1234

C: BUFF 2
S: <ERR Tamaño inválido!

C: BUFF 2 a
S: <ERR Ruido detectado: a.

C: BUFF aaaa
S: <ERR Número inválido.
```

MOSTRAR ESTADÍSTICAS (STAT)

Sintaxis:

STAT [letra]

Argumentos:

letra que representa la estadística que se quiere visualizar.

Se DEBE utilizar la letra 'b' para mostrar el total de bytes enviados, 'c' para mostrar las conexiones actuales y 't' para mostrar las conexiones totales.

Discusión:

Este comando DEBE mostrar la estadística pedida en su uso correcto.

En caso de ingresar más de un carácter, o ingresar uno que no sea alguno de los detallados previamente, DEBERÍA mostrar un mensaje de error.

Comando válido en los estados:

TRANSACTION

Posibles respuestas:

número

err

Ejemplos:

C: STAT b

S: 500

C: STAT c

S: 2

C: STAT t

S: 3

C: STAT a

S: <ERR Uso incorrecto.

C: STAT hola

S: <ERR Uso incorrecto.

CIERRE DE CONEXIÓN DEL SERVIDOR (QUIT)

Definido en la sección 3.

## 5. Extensión del protocolo

Este protocolo puede extenderse en el futuro agregando más comandos a cada uno de los estados. Todos los nuevos comandos deben seguir la sintaxis definida en la sección 2.1. de este documento y sus respectivas respuestas deben seguir la sintaxis definida en la sección 2.2.



## 6. Resumen de los comandos del protocolo DAJT

Comando (y cantidad de argumentos)	Estados válidos
AUTH (2)	AUTHENTICATION
TTRA (1)	TRANSACTION
TRAN (*)	TRANSACTION
BUFF (0 o 1)	TRANSACTION
STAT (1)	TRANSACTION
QUIT (0)	AMBOS

\*dependerá del transformador y sus argumentos

## 7. Referencias

[RFC 1939] Myers, J. and M. Rose, "Post Office Protocol -- Version 3", STD 53, **RFC 1939**, May 1996.

[RFC 2234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", **RFC 2234**, November 1997.

[RFC 2449] R. Gellens, C. Newman y L. Lundblade. POP3 Extension Mechanism. **RFC 2449**, November 1998.

## 8. Autores

- Abril Vilamowski 62495
- David Wischñevsky 62494
- Jonathan Daniel Liu 62533
- Tomás Santiago Marengo 61587