

# Puesta en marcha y configuración del servicio de monitorización remota (MONSER)

*Este documento explica brevemente los pasos a seguir para establecer un servicio de conexión remota que permita monitorizar, desde la aplicación MONITOR, las conexiones al gestor de base de datos CTSQL en un servidor remoto.*

**BASE100**

BASE 100, S.A.  
[www.base100.com](http://www.base100.com)

## Índice

<b>1. INTRODUCCIÓN .....</b>	<b>3</b>
<b>2. CONFIGURACIÓN DEL SERVIDOR .....</b>	<b>4</b>
2.1 FICHERO DE CONFIGURACIÓN DEL MONITOR .....	4
2.2 INSTALAR, ARRANCAR Y PARAR EL SERVICIO MONSER .....	4
2.2.1 <i>Unix</i> .....	4
2.2.2 <i>Windows</i> .....	4
2.3 COMPROBACIONES.....	5
<b>3. CONFIGURACIÓN DEL CLIENTE .....</b>	<b>6</b>
3.1 FICHERO DE CONFIGURACIÓN.....	6
3.2 ACTIVAR MONITOR.....	6
3.2.1 <i>Unix</i> .....	6
3.2.2 <i>Windows</i> .....	6
<b>4. PROTOCOLO DE COMUNICACIONES .....</b>	<b>7</b>
4.1 COMANDO DE VALIDACIÓN .....	7
4.2 COMANDOS DISPONIBLES.....	7
4.3 CÓDIGOS DE RESPUESTA .....	9

© Copyright BASE 100, S.A. Todos los derechos reservados. Ninguna parte de este documento puede ser reproducida ni transmitida por medio alguno sin permiso previo por escrito del titular del copyright. Todos los productos citados en este documento son marcas registradas o marcas comerciales registradas de sus respectivos propietarios.

[NT\_Monser\_v2.1]

## **1. Introducción**

---

Para poder establecer la conexión entre el PC cliente y el servidor serán necesarias las utilidades: MONITOR y MONSER.

MONSER es un servicio TCP/IP que “escucha” conexiones en el puerto **1099** del servidor donde está instalado el SGBD CTSQL. Si se desea que “escuche” en un puerto distinto al 1099 se deberá ejecutar la aplicación MONSER pasando como primer parámetro el número de puerto deseado.

El servicio es accesible abriendo una conexión remota desde cualquier máquina que soporte TCP/IP y permitirá realizar monitorizaciones remotas de las conexiones al gestor de base de datos CTSQL.

MONITOR es una utilidad de MultiBase y Cosmos cuya finalidad es llevar un control de la actividad de los procesos CTSQL existentes en el servidor (para más información consulte el manual del CTSQL-Monitor.)

## 2. Configuración del servidor

---

### 2.1 Fichero de configuración del MONITOR

Para conectarnos desde una estación cliente tendremos que utilizar un usuario válido en el servidor. El usuario con el que nos validemos debe estar dado de alta en el fichero de configuración *monitor.ini* del servidor, en la sección GENERAL, cláusula RemoteUser.

Ejemplo de monitor.ini (servidor):

```
[GENERAL]
Server=local
RemoteUser=ctl
LogPath=/home/ctl/logs
MONRESPLOG=YES
```

La variable de entorno MONRESPLOG es opcional.

### 2.2 Instalar, arrancar y parar el servicio MONSER

#### 2.2.1 Unix

Para iniciar el servicio de monitorización remota tendremos que ejecutar el programa *monser*:

```
monser
```

Si el servicio se ha lanzado correctamente nos devolverá el PID y el puerto de conexión utilizado:

```
Starting Thread...[OK]
Monitor Service Loaded.
Process(pid) .: 7199.
Listening Port: 1099.
```

Si se ejecuta pasando como primer parámetro un número, éste será el puerto de conexión sobre el que se llevará a cabo la “escucha”:

```
monser 1100
```

```
Starting Thread...[OK]
Monitor Service Loaded.
Process(pid) .: 7199.
Listening Port: 1100.
```

Para cerrar el servicio “mataremos” el proceso haciendo un *Kill* del PID de monser.

#### 2.2.2 Windows

Para poner en marcha el servicio de monitorización remota hay que dar de alta un servicio LSTMonser y asegurarse de que existe la entrada correspondiente en el fichero *services* de Windows.

Este proceso se realiza desde la herramienta de instalación de servicios **mwlisten**:

```
mwlisten -install monser
```

A continuación deberemos configurarlo para que se active de forma automática, asociándole un usuario, que será el mismo que utilizemos en el servicio CTSQL. Para ello ejecutaremos:

```
mwlisten -d monser
```

Para borrar el servicio ejecutaremos:

```
mwlisten -remove monser
```

## 2.3 Comprobaciones

Para conectarnos a MONSER y comprobar que éste está escuchando podremos realizar las siguiente pruebas:

1. Abrir un **STREAM Remoto** de MultiBase o Cosmos al puerto 1099 (servicio MONSER), o al puerto indicado como primer parámetro del MONSER.
2. Conectarnos directamente utilizando la herramienta **TELNET**.

Por ejemplo: `C:\> Telnet <servidor> monser` (tenemos que tener la entrada "monser" en el fichero *services*.)

O bien directamente

```
C:\> Telnet <servidor> 1099
```

## 3. Configuración del cliente

---

### 3.1 Fichero de configuración

```
[GENERAL]
Server=IP del Servidor
RemotePort=1099
LogPath=c:\tmp\logs
```

Donde:

Server	Dirección Ip del servidor.
RemotePort=1099	Número de puerto donde esté “escuchando” MONSER.
LogPath	Ruta (path) en la que se guardarán los ficheros de log de las conexiones establecidas con el servidor. Si no se asigna o no se define un valor a esta variable los ficheros de los se guardarán, en el caso de Cosmos, en COSMOSDIR/logs/Ip del servidor.

### 3.2 Activar MONITOR

Desde el PC cliente se ejecutará la aplicación MONITOR. Será en este punto donde nos pedirá un usuario de validación. El usuario que se indique debe ser un usuario existente en el servidor y, además, debe ser uno de los valores de la variable RemoteUser del fichero *monitor.ini* del servidor.

El monitor del servidor se puede activar desde la aplicación MONITOR del cliente.

#### 3.2.1 Unix

Para activar el monitor en un sistema UNIX/LINUX.

```
monitor -start
```

Para más información consulte el manual del CTSQL-Monitor en Unix.

#### 3.2.2 Windows

Para arrancar MONITOR en Windows ejecute la aplicación monitor.exe con el usuario que arranca el servicio.

Para más información consulte el manual del CTSQL-Monitor en Windows.

## 4. Protocolo de comunicaciones

---

Es posible monitorizar las conexiones en un servidor remoto sin utilizar la aplicación MONITOR en el equipo cliente.

Para ello, se ha implementado un protocolo de comunicación que permite la validación en el servidor, la consulta de sesiones remotas, activar/desactivar el monitor remoto, etc.

Este protocolo de comunicación se podrá utilizar desde una aplicación como “telnet” o desde una aplicación de terceros que implemente el protocolo.

### 4.1 Comando de validación

Un vez conectados al servicio tendremos que validarnos para poder hacer uso de los comandos de administración. Para validarnos emplearemos el comando VALIDATE:

```
VALIDATE user passwd \r\n
```

Tendremos que introducir un nombre de usuario y una palabra clave (password) válidos en la máquina servidor y que estén contemplados en el fichero de configuración *monitor.ini*, en la sección GENERAL, cláusula RemoteUser:

```
[General]
...
RemoteUser = ctl, dctl
...
```

Una vez validados nos aparecerá la respuesta del servidor:

```
200 User Authorized. \r\n
```

Si la validación es errónea el servidor denegará el servicio, cerrando la conexión:

```
-201 User not authorized.
```

**NOTA:** Si estamos utilizando la herramienta TELNET no aparecerán los caracteres `\r\n` [chr(13) + chr(10)], pero desde una aplicación programada por nosotros sí deberemos tenerlos en cuenta.

### 4.2 Comandos disponibles

Una vez validados por el servidor podremos ejecutar una serie de comandos de administración para interrogar sobre las sesiones activas y actuar sobre ellas. Los comandos disponibles se pueden consultar ejecutando el comando HELP:

Por ejemplo:

```
VALIDATE user passwd \r\n
200 User Authorized. \r\n
HELP
270+ Help command.
270 HELP - Monitor Remote Commands.
270 -----
270 VALIDATE user passwd
270 STATUS - START - STOP
270 SESSIONS - LIST - GET num [param]
270 WHOAMI - KILL nun
```

Como podemos observar en el ejemplo anterior, todas las líneas de respuesta a los comandos van precedidas por un número de 3 cifras. La primera línea indica si va sola o no, ya que tendrá un signo “+” detrás del número en caso de que le sigan más líneas. Para detectar que ya no sigue más información utilizaremos la línea con el punto, es decir, nos encontraremos con líneas de información hasta “\r\n.\r\n”.

Las respuestas de error devuelven un número negativo de tres cifras:

```
LISTT
-501 Unknown Command (listt).
GET 0
-231 No Session Info.
SESSIONS
230 0 active sessions.
```

La lista de comandos soportados es la siguiente:

VALIDATE user passwd	Permite validarnos con el servicio y tener acceso a los demás comandos. Si la validación es errónea cierra la conexión.
STATUS	Informa del estado del servidor, si está START o STOP, es decir, si podemos abrir conexiones o no.
START	Comando para activar el servidor.
STOP	Comando para cerrar el servidor. No se puede cerrar si hay sesiones conectadas.
SESSIONS	Indica el número de sesiones.
LIST	Lista las sesiones activas.
GET num [param]	Proporciona información de una sesión, pudiendo especificar la información que deseemos.
WHOAMI	Permite identificar sesiones provenientes de la máquina desde la que nos hemos conectado.



KILL num	Permite "matar" una sesión remota.
QUIT	Desconecta la conexión al MONSER.

### 4.3 Códigos de respuesta

#### Códigos de respuesta simples

```
200 User Authorized. (VALIDATE)
210 Monitor STARTED. (START)
220 Monitor STOPED. (STOP)
230 %d active sessions. (SESSIONS)
240 %s -> información básica de sesión (LIST)
270 %s -> línea de ayuda. (HELP)
280 %s -> línea de información de sesión. (GET)
290 %d killed. (KILL)
300 %d - %s %s ->línea de respuesta a WHOAMI ( id - clienthostname(ip)
user )
310 bye. (QUIT)
```

#### Códigos de respuesta complejos (respuesta de más de una línea)

```
240+ %d active sessions. -> acompañado de códigos 240 de sesiones
(LIST)
270+ Help command. (HELP)
280+ Session Info. primera línea de GET.
300+ Session List. primera línea de WHOAMI, si hay más de una.
```

#### Códigos de respuesta de errores

```
-201 User Not Authorized.
-202 Validate first please. "VALIDATE user passwd".
-211 Unable to START.
-221 Unable to STOP.
-231 No Session Info.
-291 Unable to kill %d Session.
-301 No session matches to you.
-311 bye unable disconnect.
-501 Unknown Command (%s).
-502 %s. ->otros posibles errores
```

**NOTA:** %s: texto; %d: caracteres con contenido numérico.