



Manual de Operación

Tabla de contenido

1. [Introducción](#)
2. [Base de datos](#)
3. [Viafirma Platform](#)
4. [Auditoría](#)
5. [Correo electrónico](#)
6. [HSM](#)
7. [Monitorización del sistema](#)
 - i. [Consumo de recursos](#)
 - ii. [Espacio en disco](#)
 - iii. [Inodes libres en discos](#)
 - iv. [Consumo del disco](#)
 - v. [Gestión de Logs](#)
8. [Reinicio de los servicios](#)
9. [Actualización](#)
10. [Anexo 1 - Configuración en contexto](#)

Introducción

El presente documento pretende ser una guía de operaciones para el proyecto **Viafirma Manager**.

Esta documentación técnica está sujeta a modificaciones diarias, y alguna información o configuración avanzada podría no estar reflejada. Consulte en cualquier caso con el equipo de soporte técnico.

Si lo desea puede descargar este manual en pdf [aquí](#).

Últimos cambios

Fecha	Cambio
15/01/18	Migración a gitbook

1. Base de datos

1.1 Configuración del servicio

La conexión con la base de datos se configura en el fichero de configuración de la aplicación. Este fichero puede localizarse en dos rutas diferentes:

En tomcats que tienen activa la opción copyXML en el server.xml el fichero se encuentra en la ruta:

```
<tomcat-home>/conf/Catalina/localhost/viafirmaManager.xml
```

En tomcats que no tienen activa la opción copyXML en el server.xml el fichero se encuentra en la ruta:

```
<tomcat-home>/webapps/viafirmaManager/META-INF/context.xml
```

Los datos a configurar son:

- **username**: usuario que tiene permisos DML sobre todo el esquema y tiene visibilidad sobre los objetos del mismo (tables, constraints, sequences, etc.).
- **password**: del usuario que tiene acceso a la base de datos.
- **url**: cadena de conexión correcta para acceder al servidor de datos.
- **driverClassName**: clase del driver JDBC para la base de datos elegida.

El dialecto de la base de datos se configura en el fichero persistence.xml que se encuentra en la ruta

```
<tomcat-home>/webapps/viafirmaManager/WEB-INF/classes/META-INF/persistence.xml
```

1.2 Cuándo se consume

Siempre

1.3 Trazas de error asociadas

En el log de la aplicación se podrá observar la traza del error con el problema en la conexión con la base de datos:

```
Internal Exception: java.sql.SQLException: No more data to read from socket  
Error Code: 17410
```

```
Internal Exception:java.sql.SQLException: Closed Connection  
Error Code: 17008
```

1.4 Comprobación del Servicio

En caso de errores en la base de datos es posible que la aplicación no se pueda iniciar, y si se inicia, no se podrá realizar la autenticación en el sistema.

Si la configuración de la base de datos ha sido efectuada correctamente la aplicación podrá iniciarse e ingresar en ella.

2. Viafirma Platform

2.1 Configuración del servicio

La configuración de los datos de conexión con viafirma platform se realiza en el [contexto](#) de la aplicación Viafirma Manager.

```
<Environment description="Configuracion de Viafirma.- URL_PROVIDER ; URL_PROVIDER_WS ; API_KEY ; PASS_KEY"
  name="CONFIG_VIAFIRMA_CLIENT" override="false" type="java.lang.String"
  value="https://testservices.viafirma.com/viafirma/tokenConnector;https://testservices.viafirma.com/v
```

Los datos a configurar son:

- **URL Provider:** url de la aplicación viafirma platform.
- **URL Provider WS:** url del servicio web de la aplicación viafirma platform.
- **API KEY:** identificador de la aplicación dada de alta en viafirma manager para hacer uso de la aplicación viafirma platform.
- **PASS KEY:** contraseña para hacer uso de la aplicación.

2.2 Cuándo se consume

Al realizar la autenticación en la aplicación con certificados digitales locales.

2.3 Trazas de error asociadas

En el log de la aplicación se mostrará el problema de conexión con viafirma platform.

Los errores más comunes son:

- La url de viafirma platform no es correcta o no es visible.

```
javax.servlet.ServletException: org.viafirma.cliente.exception.InternalException: El servidor de autenticac:
org.openid4java.discovery.yadis.YadisException: 0x706: GET failed on https://testservices.viafirma.com/viaf:
```

- El api key o contraseña indicados no son correctos, o la aplicación está desactivada.

```
javax.xml.ws.soap.SOAPFaultException: Application unavailable. This application is not allowed to access to
```

2.4 Comprobación del Servicio

Se puede comprobar el estado de la conexión con viafirma platform al acceder a la aplicación Viafirma Manager mediante usuario y contraseña. En el caso de que no esté correctamente configurada la conexión con Platform se mostrará un mensaje de alerta advirtiéndonos de ello en la pantalla principal.

3. Auditoría

3.1 Configuración del servicio

La configuración de los datos de auditoría se realiza por parte de un usuario administrador de sistemas en el [contexto](#) de la aplicación.

Deben indicarse los siguientes campos:

- **VIAFIRMA_LOG_DIRECTORIES**: Lista de directorios de logs separados por punto y coma (;)

3.2 Cuándo se consume

Cada vez que se realiza el parseo de los logs de auditoría en una tarea programada o cada vez que se inicia la aplicación.

3.3 Trazas de error asociadas

En el caso de que no esté bien configurada la ruta de auditoría o no sea accesible, visualizaremos la siguiente traza en el log.

```
2018-01-17 10:52:08 INFO com.viafirma.manager.listener.StartupServletContextListener.init(235) - Iniciando
2018-01-17 10:52:09 INFO com.viafirma.manager.listener.StartupServletContextListener.init(237) - Finalizado
2018-01-17 10:52:09 WARN com.viafirma.manager.threads.ThreadLogParser.getDirectories(351) - C:\tmp\viafirma
2018-01-17 10:52:09 WARN com.viafirma.manager.threads.ThreadLogParser.getDirectories(351) - C:\tmp\viafirma
2018-01-17 10:52:09 INFO com.viafirma.manager.threads.ThreadLogParser.processLog(160) - No hay ficheros qu
```

3.4 Comprobación del Servicio

Verificaremos si se ha realizado correctamente el parseo de los logs de auditoría si comprobamos que existan datos realizando una búsqueda avanzada de firma y autenticación, o que existan datos en el módulo de estadísticas.

Estas dos secciones se nutren de los datos que se parsean desde los ficheros de logs configurados en la carpeta de auditoría.

4. Correo electrónico

4.1 Configuración del servicio

La configuración de los datos del servidor de correo electrónico se realiza por parte de un usuario administrador de sistemas en el [contexto](#) de la aplicación.

Los datos a rellenar son:

```
FROM_ADDRESS
HOST_SMTP
SMTP_USER
SMTP_PASS
```

4.2 Cuándo se consume

En el envío de las notificaciones de caducidad de los certificados.

4.3 Trazas de error asociadas

Los posibles problemas que pueden ocurrir son:

- Problemas con la conexión al host. En el log se muestra un error del tipo

```
ERROR com.viafirma.manager.util.EmailUtilities.sendEmail(444) - No se puede enviar el email a 'inbox@viafirma.com'
Could not connect to SMTP host: HOST, port: PORT (java.net.ConnectException: Connection timed out: connect)
ConnectException: Connection timed out: connect
javax.mail.MessagingException: Could not connect to SMTP host: 192.168.10.224, port: 25 (java.net.ConnectException: Connection timed out: connect)
```

Puede probarse que el servicio esté operativo realizando desde el terminal un telnet al host y puerto del servidor de correo:

```
telnet HOST PORT
```

- Problemas con el usuario y contraseña. En el log se muestra un error del tipo

```
535 5.7.8 Error: authentication failed: authentication failure
ERROR com.viafirma.manager.util.EmailUtilities.sendEmail(444) -
```

```
No se puede enviar el email a 'test@correo.com'. null. AuthenticationFailedException: javax.mail.Authen
```

4.4 Comprobación del Servicio

Cuando está a punto de caducar algún certificado de los añadidos en alguna aplicación, se enviará un correo a la dirección configurada.

5 HSM

5.1 Configuración del servicio

Debe incluirse el parámetro `-Djava.library.path=/usr/safenet/lunaclient/jsp/lib` (o la ruta en concreto en la que se hayan instalado dichas librerías) en el arranque de Tomcat para que se carguen las librerías de acceso al HSM.

La configuración de HSM se realiza por parte de un usuario administrador de sistemas en el [contexto](#) de la aplicación.

LOAD_HSM_INIT_APPLICATION: Si la siguiente variable tiene valor 'true', se cargaran los HSM configurados en el arranque de la aplicación, en caso contrario, solo cuando se usen.

5.2 Cuándo se consume

Este servicio es consumido al ser listados los certificados en HSM.

5.3 Trazas de error asociadas

El log de la aplicación indicará el problema en la conexión con el HSM configurado.

- No está instalada correctamente la librería de LunaClient

Al intentar añadir un nuevo repositorio de certificados de tipo HSM desde la configuración de la aplicación muestra el siguiente error en la traza.

```
Caused by: com.safenetinc.luna.LunaException: Failed to load LunaAPI and LunaAPI_64 libraries
    at com.safenetinc.luna.LunaAPI.<clinit>(LunaAPI.java:180)
    ... 116 more
Caused by: java.lang.UnsatisfiedLinkError: no LunaAPI_64 in java.library.path
    at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1864)
    at java.lang.Runtime.loadLibrary0(Runtime.java:870)
    at java.lang.System.loadLibrary(System.java:1122)
    at com.safenetinc.luna.LunaAPI.<clinit>(LunaAPI.java:178)
```

- Error al conectar con el slot

Si se produce este error significa que no nos estamos conectando al slot correcto del HSM.

```
2018-01-17 12:14:30 INFO    org.viafirma.plugin.keystore.SafeNetLunaHSMPugin.init(146) - Login into HSM. Some
2018-01-17 12:14:30 INFO    org.viafirma.plugin.keystore.SafeNetLunaHSMPugin.init(160) - Login failure.
2018-01-17 12:14:30 INFO    org.viafirma.plugin.keystore.SafeNetLunaHSMPugin.initCacert(258) - Loading KeySto
2018-01-17 12:14:30 ERROR   org.viafirma.plugin.keystore.SafeNetLunaHSMPugin.init(195) - Unable to load keys
2018-01-17 12:14:30 ERROR   org.viafirma.nucleo.Nucleo.runWithProperties(344) - java.io.IOException: Unable to
```

5.4 Comprobación del Servicio

Se puede validar este servicio en el momento en el que se listen los certificados de tipo Keystore HSM.

6 Monitorización del sistema

En este capítulo se detallará cómo realizar el monitoreo del sistema operativo y cómo interpretar estos datos.

La monitorización del sistema está compuesta por:

- [Consumo de recursos](#)
- [Espacio en disco](#)
- [Inodes libres en disco](#)
- [Consumo del disco](#)
- [Gestion de Logs](#)

6.1 Consumo de recursos

En este apartado veremos qué comando es necesario para comprobar el espacio utilizado en el disco duro del servidor.

El comando a utilizar es **top** cuando ejecutamos el comando tenemos que prestar especial atención a los siguientes datos:

- **Load average**, el valor de este dato no debería ser superior a 1, en el caso que lo sea, podemos deducir que hay algo en los servidores que no está funcionando correctamente.

```
1. userviaf@medviafclu02:~ (ssh)
top - 05:01:53 up 55 days, 6:19, 3 users, load average: 0.00, 0.02, 0.24
Tasks: 149 total, 1 running, 148 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.2%us, 0.1%sy, 0.0%ni, 99.6%id, 0.1%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 14100760k total, 11245612k used, 2855148k free, 2340424k buffers
Swap: 4194300k total, 58380k used, 4135920k free, 4181620k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
16221 userviaf  20   0 10.5g 1.5g 12m  S   1.0  11.5   4:33.22 java
27310 userviaf  20   0 5824m 167m 4672  S   0.7   1.2   27:52.03 java
  10 root      20   0   0     0     0   S   0.3   0.0   39:35.79 rcu_sched
16923 userviaf  20   0 15084 1252  924  R   0.3   0.0    0:00.02 top
   1 root      20   0 19416  860  760  S   0.0   0.0    0:01.80 init
   2 root      20   0   0     0     0   S   0.0   0.0    0:01.69 kthreadd
   3 root      20   0   0     0     0   S   0.0   0.0    0:32.23 ksoftirqd/0
   5 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kworker/0:0H
   7 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kworker/u:0H
   8 root      RT   0   0     0     0   S   0.0   0.0    0:00.91 migration/0
   9 root      20   0   0     0     0   S   0.0   0.0    0:00.00 rcu_bh
  11 root      RT   0   0     0     0   S   0.0   0.0    0:26.05 watchdog/0
  12 root      RT   0   0     0     0   S   0.0   0.0    0:20.06 watchdog/1
  13 root      20   0   0     0     0   S   0.0   0.0    0:23.47 ksoftirqd/1
  14 root      RT   0   0     0     0   S   0.0   0.0    0:01.21 migration/1
  16 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kworker/1:0H
  17 root      RT   0   0     0     0   S   0.0   0.0    0:16.11 watchdog/2
  18 root      20   0   0     0     0   S   0.0   0.0    0:21.80 ksoftirqd/2
  19 root      RT   0   0     0     0   S   0.0   0.0    0:00.31 migration/2
  20 root      20   0   0     0     0   S   0.0   0.0    0:00.00 kworker/2:0
  21 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kworker/2:0H
  22 root      RT   0   0     0     0   S   0.0   0.0    0:16.03 watchdog/3
  23 root      20   0   0     0     0   S   0.0   0.0    0:18.65 ksoftirqd/3
  24 root      RT   0   0     0     0   S   0.0   0.0    0:00.32 migration/3
  25 root      20   0   0     0     0   S   0.0   0.0    0:00.00 kworker/3:0
  26 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kworker/3:0H
  27 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 cpuset
  28 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 khelper
  29 root      20   0   0     0     0   S   0.0   0.0    0:00.00 kdevtmpfs
  30 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 netns
  31 root      20   0   0     0     0   S   0.0   0.0    0:00.55 bdi-default
  32 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kintegrityd
  33 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 crypto
  34 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kblockd
  35 root      20   0   0     0     0   S   0.0   0.0   28:32.13 kworker/3:1
  36 root      20   0   0     0     0   S   0.0   0.0   27:17.80 kworker/2:1
  38 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 ata_sff
  39 root      20   0   0     0     0   S   0.0   0.0    0:00.00 khubd
  40 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 md
  41 root      20   0   0     0     0   S   0.0   0.0    0:04.58 khungtaskd
  42 root      20   0   0     0     0   S   0.0   0.0   21:28.00 kswapd0
  43 root      25   5   0     0     0   S   0.0   0.0    0:00.00 ksmd
  44 root      20   0   0     0     0   S   0.0   0.0    0:00.00 fsnotify_mark
  55 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kthrotld
  58 root      0  -20   0     0     0   S   0.0   0.0    0:00.00 kpsmoused
```

NOTA: hay situaciones en la que el servidor tiene una gran carga de trabajo y el valor puede ser superior a lo indicado, pero sólo momentáneamente, pasado un tiempo el load average debe de estar estabilizado.

Si mientras ejecutamos el comando top pulsamos la tecla **1** podremos ver el porcentaje de uso de las CPUs que se están utilizando.

```

top - 05:02:04 up 55 days, 6:19, 3 users, load average: 0.00, 0.02, 0.23
Tasks: 148 total, 1 running, 147 sleeping, 0 stopped, 0 zombie
Cpu0 : 0.7%us, 0.3%sy, 0.0%ni, 99.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1 : 0.7%us, 0.3%sy, 0.0%ni, 99.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2 : 0.0%us, 0.0%sy, 0.0%ni, 100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3 : 0.0%us, 0.0%sy, 0.0%ni, 100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 14100760k total, 11245596k used, 2855164k free, 2340540k buffers
Swap: 4194300k total, 58380k used, 4135920k free, 4181624k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
16221	userviaf	20	0	10.5g	1.5g	12m	S	0.7	11.5	4:33.61	java
27310	userviaf	20	0	5824m	167m	4672	S	0.3	1.2	27:52.08	java
1	root	20	0	19416	860	760	S	0.0	0.0	0:01.80	init
2	root	20	0	0	0	0	S	0.0	0.0	0:01.69	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:32.23	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/u:0H
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.91	migration/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	S	0.0	0.0	39:35.79	rcu_sched
11	root	RT	0	0	0	0	S	0.0	0.0	0:26.05	watchdog/0
12	root	RT	0	0	0	0	S	0.0	0.0	0:20.06	watchdog/1
13	root	20	0	0	0	0	S	0.0	0.0	0:23.47	ksoftirqd/1
14	root	RT	0	0	0	0	S	0.0	0.0	0:01.21	migration/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
17	root	RT	0	0	0	0	S	0.0	0.0	0:16.11	watchdog/2
18	root	20	0	0	0	0	S	0.0	0.0	0:21.80	ksoftirqd/2
19	root	RT	0	0	0	0	S	0.0	0.0	0:00.31	migration/2
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/2:0
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/2:0H
22	root	RT	0	0	0	0	S	0.0	0.0	0:16.03	watchdog/3
23	root	20	0	0	0	0	S	0.0	0.0	0:18.65	ksoftirqd/3
24	root	RT	0	0	0	0	S	0.0	0.0	0:00.32	migration/3
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/3:0
26	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/3:0H
27	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	cpuset
28	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
29	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
30	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
31	root	20	0	0	0	0	S	0.0	0.0	0:00.55	bdi-default
32	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
33	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	crypto
34	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kblockd
35	root	20	0	0	0	0	S	0.0	0.0	28:32.13	kworker/3:1
36	root	20	0	0	0	0	S	0.0	0.0	27:17.80	kworker/2:1
38	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	ata_sff
39	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khubd
40	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	md
41	root	20	0	0	0	0	S	0.0	0.0	0:04.58	khungtaskd
42	root	20	0	0	0	0	S	0.0	0.0	21:28.00	kswapd0
43	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
44	root	20	0	0	0	0	S	0.0	0.0	0:00.00	fsnotify_mark

6.2 Espacio en disco

En el presente apartado se mostrará cómo ver el estado actual del almacenamiento de los servidores. Para ver el espacio ocupado y disponible del disco tendríamos que ejecutar el siguiente comando:

```
df -h
```

El resultado del comando sería algo como lo que se muestra en la siguiente imagen.

```
[userviaf@medviafclu02 ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg_00-lv_root  9,8G    1,2G    8,1G   13% /
tmpfs                     6,8G         0    6,8G   0% /dev/shm
/dev/sda1                  477M    180M    268M   41% /boot
/dev/mapper/vg_00-lv_home  3,9G      15M    3,6G   1% /home
/dev/mapper/vg_00-lv_opt   9,8G    8,1G    1,3G   87% /opt
/dev/mapper/vg_00-lv_tmp   9,8G     73M    9,2G   1% /tmp
/dev/mapper/vg_00-lv_usr   9,8G    2,3G    7,0G   25% /usr
/dev/mapper/vg_00-lv_var   3,9G    928M    2,7G   26% /var
/dev/mapper/vg_00-lv_var_log 9,8G     82M    9,2G   1% /var/log
/dev/sdb                   1,1T    967G    158G   87% /cluster
[userviaf@medviafclu02 ~]$
```

Con el resultado del comando podemos observar las diferentes particiones que posee el servidor, así como el espacio total de cada partición/disco duro y el porcentaje de espacio ocupado en cada una de ellas.

6.3 Inodes libres en discos

A continuación se describe cómo utilizar el comando que nos permitirá ver los IOS libres en el disco duro. Esto nos permitirá saber la cantidad de ficheros que podemos escribir en una unidad, por ejemplo, si nuestra unidad permite 100 inodes y 1TB de almacenamiento significa que solo podemos escribir 100 ficheros, aunque sean de 1KB y tengamos disponible el resto de la unidad.

El comando es el siguiente:

`df -ih`

```
[userviaf@medviafclu02 ~]$ df -ih
Filesystem          Inodes  IUsed  IFree  IUse% Mounted on
/dev/mapper/vg_00-lv_root
                    640K    22K   619K    4% /
tmpfs                1,7M      1   1,7M    1% /dev/shm
/dev/sda1            126K     65   125K    1% /boot
/dev/mapper/vg_00-lv_home
                    256K    362   256K    1% /home
/dev/mapper/vg_00-lv_opt
                    640K    24K   617K    4% /opt
/dev/mapper/vg_00-lv_tmp
                    640K    415   640K    1% /tmp
/dev/mapper/vg_00-lv_usr
                    640K    80K   561K   13% /usr
/dev/mapper/vg_00-lv_var
                    256K    2,5K   254K    1% /var
/dev/mapper/vg_00-lv_var_log
                    640K    135   640K    1% /var/log
/dev/sdb             281M   242M    40M   87% /cluster
[userviaf@medviafclu02 ~]$
```

6.4 Consumo del disco

También podemos ver el estado actual de la lectura y escritura de todas las unidades que están disponible en las máquinas ejecutando el siguiente comando:

```
iotstat -xtc
```

El resultado de ejecutar el comando mencionado es el siguiente:

```
[userviaf@medviafclu02 ~]$
[userviaf@medviafclu02 ~]$ iostat -xtc
Linux 3.8.13-98.2.2.el6uek.x86_64 (medviafclu02.colombiamovil.corp) 17/06/16 _x86_64_ (4 CPU)

17/06/16 05:09:54
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           7,45    0,00   2,17   1,77    0,00   88,61

Device:            rrqm/s   wrqm/s     r/s     w/s  rsec/s   wsec/s  avgrq-sz  avgqu-sz   await  svctm   %util
sda                 0,08     2,95     0,74     5,62   62,22  1156,22   191,57     0,02     2,65   0,65   0,42
sdb                 0,00    39,29    78,65    53,37  630,00   737,98    10,36     0,27     2,07   0,61   8,01
fd0                 0,00     0,00     0,00     0,00    0,00    0,00     8,00     0,00    26,83  26,83   0,00
dm-0                 0,00     0,00     0,09     0,04    3,59    0,30    30,05     0,00     3,71   0,86   0,01
dm-1                 0,00     0,00     0,09     0,11    0,68    0,90     8,00     0,01    41,38   0,34   0,01
dm-2                 0,00     0,00     0,24     0,10    6,68    0,87    22,56     0,00     4,12   0,74   0,02
dm-3                 0,00     0,00     0,05     0,13    6,31    4,37    60,25     0,00     1,49   0,70   0,01
dm-4                 0,00     0,00     0,01     0,05    1,08    0,42    27,04     0,00     1,34   0,63   0,00
dm-5                 0,00     0,00     0,00     0,31    0,05    3,92    12,90     0,00     0,80   0,28   0,01
dm-6                 0,00     0,00     0,00     0,12    0,02    0,95     8,20     0,00     0,43   0,32   0,00
dm-7                 0,00     0,00     0,34     7,72   43,37  1144,45   147,28     0,02     2,26   0,46   0,37

[userviaf@medviafclu02 ~]$ █
```

Si añadimos un 1 al comando (`iotstat -xtc 1`) se ejecutará el comando cada segundo.

A continuación el significado de cada columna:

- **r/s**: Lecturas por segundo
- **w/s**: Escrituras por segundo
- **kr/s**: Kbytes leídos por segundo
- **kw/s**: Kbytes escritos por segundo
- **Wait**: Número medio de transacciones que están en espera de servicio (longitud de cola)
- **Actv**: Número medio de transacciones que están siendo gestionadas de manera activa
- **svc_t**: Tiempo medio de servicio (en milisegundos)
- **%w**: Porcentaje de tiempo durante el cual la cola no está vacía
- **%b**: Porcentaje de tiempo durante el cual el disco está ocupado

6.5 Gestión de Logs

¿Dónde se configura?

El fichero de configuración se llama `log4j.properties`, y por defecto se encuentra en la siguiente ruta:

```
<tomcat-home>/webapps/viafirmaManager/WEB-INF/classes/log4j.properties
```

Por el contrario, si hemos configurado un directorio externo para la configuración de la aplicación, dicho fichero se encontrará en la carpeta externa configurada.

¿Dónde se escriben los ficheros de log?

En el fichero de configuración `log4j.properties` se define el lugar de escritura de los ficheros de log.

Por defecto está definida la ruta:

```
<tomcat-home>/logs/vmanager.log
```

Se puede modificar cambiando el valor en la variable **`log4j.appender.R.File`** dentro del fichero de configuración.

Rotación de los logs

Por defecto, esta configurada la rotación de logs cada 5MB, almacenando un máximo de 10 logs. Esto puede modificarse cambiando las siguientes propiedades:

- `log4j.appender.R.MaxFileSize`: Tamaño máximo de los ficheros de log.
- `log4j.appender.R.MaxBackupIndex`: Número máximo de ficheros a almacenar.

7 Reinicio de los servicios

Parar el tomcat

La forma de actuar para **parar** el Tomcat es la siguiente:

1. Ejecutar el comando: `PATH_TOMCAT/bin/catalina.sh stop`
2. Comprobar que ha finalizado el proceso Tomcat ejecutando el siguiente comando: `ps - aux | grep tomcat` . El resultado del comando anterior debe ser similar al siguiente en el caso de que aún exista un proceso Tomcat ejecutándose:

```
1001 8777 0.5 4.3 934320 129628 ? Ss 12:37 0:18 /usr/lib/jvm/java-6-sun-1.6.0.22/bin/java -
agentlib:jdwp=transport=dt_socket,suspend=y,address=localhost:45632 -
Dcatalina.base=/home/user/java/workspacehelios/.metadata/.plugins/org.eclipse.wst.server.core/tmp3 -
Dcatalina.home=/home/user/java/apache-tomcat-5.5.29 -
Dwtp.deploy=/home/user/java/workspacehelios/.metadata/.plugins/org.eclipse.wst.server.core/tmp3/wtpwebapps -
Djava.endorsed.dirs=/home/user/java/apache-tomcat-5.5.29/common/endorsed -Dfile.encoding=UTF-8 -classpath
/home/user/java/apache-tomcat-5.5.29/bin/bootstrap.jar:/usr/lib/jvm/java-6-sun-1.6.0.22/lib/tools.jar
org.apache.catalina.startup.Bootstrap start
```

```
1001 9587 0.0 0.0 3340 856 pts/0 S+ 13:35 0:00 grep --color=auto tomcat
```

3. En caso de que no se haya parado el proceso Tomcat, tendremos que pararlo manualmente.

Para ello localizaremos en el código anterior el PID del proceso (el que se encuentra en negrita e itálica) y ejecutaremos el siguiente comando: ***kill -9 PID***, donde PID es el número anteriormente mencionado.

Arrancar el tomcat

La forma de actuar para **arrancar** el Tomcat es la siguiente:

1. Ejecutar el comando: `PATH_TOMCAT/bin/catalina.sh start`

NOTA: Mediante una sencilla modificación del `catalina.sh`, se puede ejecutar `catalina.sh stop -force` (opción modificadora `force`) que, en principio, se encarga de matar el proceso Java si se hubiese podido quedar latente.

8 Actualizaciones

El proceso de actualización del sistema se compone de los siguientes pasos:

1. Parar el tomcat

[Ver punto Parar el tomcat de este manual](#)

2. Actualización de la base de datos

Para las versiones que requieran la actualización de base de datos será necesario ejecutar los scripts de actualización.

3. Eliminación de la versión actual

Eliminación de la carpeta viafirmaManager y del fichero viafirmaManager.xml que se encuentran en la ruta /tomcat-home/webapps.

```
rm -rf /<tomcat-home>/webapps/viafirmaManager*
```

4. Despliegue del nuevo war

Copiar el fichero viafirmaManager.war en la ruta /tomcat-home/webapps.

5. Arrancar el tomcat

[Ver punto Arrancar el tomcat de este manual](#)

6. Actualizar la configuración

En las versiones que requieran cambios de configuración será necesario la realización de los mismos.

9. Anexo 1 - Configuración en contexto

En tomcats que tienen activa la opción copyXML en el server.xml el fichero se encuentra en la ruta:

```
/<tomcat-home>/conf/Catalina/localhost/viafirmaManager.xml
```

En tomcats que no tienen activa la opción copyXML en el server.xml el fichero se encuentra en la ruta:

```
/<tomcat-home>/webapps/viafirmaManager/META-INF/context.xml
```

Si por el contrario, hemos configurado un directorio externo para la configuración de la aplicación, dicho fichero se encontrará en la carpeta externa configurada, el cual se llama **viafirmaManagerConfig.properties**.