Manual de Implantación

Proxia® - Guía de Instalación

Ver. 9.7

Divisa iT
INFORMÁTICA Y TELECOMUNICACIONES

proxia® SUITE

Índice de contenidos:

1. Objetivo

El objeto del presente documento es detallar el proceso de instalación/configuración de los servidores de aplicaciones y aplicaciones que dan servicio al Portal de Salud de la Junta de Castilla y León(https://www.saludcastillayleon.es). No es objetivo de este documento describir los procesos de configuración o parametrización de la base de datos.

Este documento solamente consta de la información correspondiente a la configuración de los servidores de aplicaciones/web y de la instalación y configuración de las aplicaciones propiamente dicha, así como de todas aquellas herramientas necesarias para el funcionamiento de las mismas.[1] No es cometido de este documento el describir la instalación de los SO o aplicativos genéricos que dan soporte al portal.

Las descripciones correspondientes al proceso de configuración/instalación del software están planteadas para un sistema Linux con servidor de web/aplicaciones: Apache Httpd + Apache Tomcat, no siendo estas las únicas configuraciones soportadas por Proxia® pero si las consensuadas para esta instalación.

[1] Los procesos de configuración/instalación descritos en el documento son plantillas o guías, las cuales no son la única forma de instalación de las aplicaciones. Sobre las posibles modificaciones sobre este proceso, y la compatibilidad de las mismas, por favor pónganse en contacto con el personal técnico de DivisaiT.

2. Máquina Virtual Java

Las aplicaciones PROXIA que dan soporte al Portal de Salud de la Junta de Castilla y León se instalaran en el servidor de aplicaciones Apache Tomcat, dicho servidor necesita de una máquina virtual java Oracle JDK 1.8 para funcionar.

Se recomienda instalar la última versión **8u172** (**JDK de Oracle**) para linux de 64 bits.

3. Servidores Web/Aplicaciones

3.1 Introducción

Es necesario haber realizado previamente la instalación del usuario de base de datos que va a dar soporta a las aplicaciones que se instalarán.

El proceso de instalación de las aplicaciones Proxia® consistirá en los siguientes pasos:

1. Configuración del servidor de aplicaciones Apache Tomcat
2. Configuración del servidor Web: Apache HTTPD
3. Configuración de la conectividad entre el servidor de aplicaciones y el servidor web.
4. Instalación y configuración de las aplicaciones Proxia®

3.2 Servidor aplicaciones: Apache Tomcat

El servidor de aplicaciones en el residirán las aplicaciones que dan soporte al Portal de Salud de la Junta de Castilla y León, cliente y administración, es un contenedor de servlet Apache Tomcat versión 7.x. Se recomienda instalar la última versión Apache Tomcat de la rama 7.0.x, o al menos superior a la 7.0.42.

3.2.1 Requisitos previos

La configuración recomendada con la que arrancara el Apache Tomcat deberá definir los parámetros especificados en el Anexo 3.7.1.

Resumiendo, de forma esquemática estos parámetros son:

- La maquinar virtual java (JAVA_HOME) a utilizar.
- La memoria recomendada en un entorno de explotación para el adecuado funcionamiento de las aplicaciones es de 8GB (parámetros –Xmx y –Xms). En cuanto a la memoria de tipo PermGen se recomienda el valor de 256M.
- Opciones de rendimiento y uso de java (gcInterval, headless, timezone, server, d64, allowRestrictedHeaders)
- Opción proxy si es necesario para que la aplicación acceda a servicios http externos.
- Variables de entorno del idioma (LANG, LC_ALL, NLS_LANG)

- Variables de entorno del servidor de aplicaciones (CATALINA_HOME, CATALINA_BASE).

Los parámetros HeapDumpPath, HeapDumpOnOutOfMemoryError y los asociados a la monitorización de JMX (jmxremote) NO son obligatorios, pero recomendamos definir.

En el caso de que en el mismo servidor de aplicaciones existan otras aplicaciones distintas de las de Proxia y que requieran otras parametrizaciones distintas de las comentadas anteriormente deberá consultarse con el personal de DivisaiT para verificar posibles incompatibilidades de las mismas con Proxia.

3.3 Servidor Web: Apache HTTPD

El servidor web es necesario para el correcto funcionamiento de los **paths semánticos** de la aplicación Proxia. Esto exige la existencia de un servidor Web de front-end y por tanto una configuración especial de este servidor.

También es necesario para que este determine que peticiones serán tratadas por el servidor web o cuales se les pasara al servidor de aplicaciones.

3.3.1 Requisitos previos

Proxia está verificado contra las versiones 2.2.x y 2.4.x de Apache. En este documento suponemos que la versión instalada de Apache es una de la rama 2.4.x

```
Server version: Apache/2.4.6 (CentOS)
Server built:   Mar 12 2015 15:07:19
Server's Module Magic Number: 20120211:24
Server loaded:  APR 1.4.8, APR-UTIL 1.5.2
Compiled using: APR 1.4.8, APR-UTIL 1.5.2
Architecture:   64-bit
Server MPM:     Prefork
  threaded:     no
  forked:       yes (variable process count)
Server compiled with....
 -D APR_HAS_SENDFILE
 -D APR_HAS_MMAP
 -D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
 -D APR_USE_SYSVSEM_SERIALIZE
 -D APR_USE_PTHREAD_SERIALIZE
 -D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
 -D APR_HAS_OTHER_CHILD
 -D AP_HAVE_RELIABLE_PIPED_LOGS
 -D DYNAMIC_MODULE_LIMIT=256
 -D HTTPD_ROOT="/etc/httpd"
 -D SUEXEC_BIN="/usr/sbin/suexec"
 -D DEFAULT_PIDLOG="/run/httpd/httpd.pid"
 -D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
 -D DEFAULT_ERRORLOG="logs/error_log"
 -D AP_TYPES_CONFIG_FILE="conf/mime.types"
 -D SERVER_CONFIG_FILE="conf/httpd.conf"
```

Por temas de rendimiento es recomendable que el modo del Apache este en modo ***worker*** (Multiproceso-Multihebra) en lugar de modo ***prefork (****por defecto****)***:

```
Server version: Apache/2.4.6 (CentOS)
```

```
Server built:   Mar 12 2015 15:07:19
Server's Module Magic Number: 20120211:24
Server loaded:  APR 1.4.8, APR-UTIL 1.5.2
Compiled using: APR 1.4.8, APR-UTIL 1.5.2
Architecture:   64-bit
Server MPM:     worker
  threaded:     yes (fixed thread count)
    forked:     yes (variable process count)
Server compiled with....
 -D APR_HAS_SENDFILE
 -D APR_HAS_MMAP
 -D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
 -D APR_USE_SYSVSEM_SERIALIZE
 -D APR_USE_PTHREAD_SERIALIZE
 -D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
 -D APR_HAS_OTHER_CHILD
 -D AP_HAVE_RELIABLE_PIPED_LOGS
 -D DYNAMIC_MODULE_LIMIT=256
 -D HTTPD_ROOT="/etc/httpd"
 -D SUEXEC_BIN="/usr/sbin/suexec"
 -D DEFAULT_PIDLOG="/run/httpd/httpd.pid"
 -D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
 -D DEFAULT_ERRORLOG="logs/error_log"
 -D AP_TYPES_CONFIG_FILE="conf/mime.types"
 -D SERVER_CONFIG_FILE="conf/httpd.conf"
```

3.3.2 Configuraciones Apache HTTPD

Una vez incluido el modulo mod_jk.so en el Apache se puede empezar a realizar la integración entre Apache y Tomcat, para ello:

3.3.2.1 *workers.properties*

Para definir el conector que se encargue enviar las peticiones http que le lleguen al servidor web Apache HTTPD al servidor de aplicaciones Apache Tomcat se debe crear un nuevo archivo dentro de `/etc/httpd/conf/` llamado **`workers.properties`**.

En este fichero definiremos las propiedades del worker.

Se ha de tener en cuenta que al estar el servidor web y el servidor de aplicaciones en diferentes servidores, la propiedad host del worker será la ip del servidor de aplicaciones.

Puede verse un ejemplo de fichero de definición del conector en el anexo 3.7.2.

3.3.2.2 *httpd.conf*

Editamos el archivo de configuración del Apache: `/etc/httpd/conf/httpd.conf` con las siguientes configuraciones: (se van indicando según el orden del fichero de principio a fin)

Se configuran unos parámetros para que en las cabeceras de las respuestas no se de información de la versión del servidor utilizado:

```
# DIVISAIT
#Desactivar los metodos: TRACE y TRACK, por temas de seguridad
TraceEnable off
# DIVISAIT

# DIVISAIT
# Para que los apaches no envien que version de apache se esta ejecutando, por temas de seguridad
ServerTokens Prod
```

```
#ServerTokens OS
# DIVISAIT
```

Habilitar la opción keepalive para reaprovechar las conexiones abiertas:

```
# DIVISAIT
KeepAlive On
#KeepAlive Off
# DIVISAIT
```

Habilitar el puerto seguro

```
Listen 443
```

En la sección de LoadModules, revisar que los siguientes módulos estén activados:

```
# DIVISAIT
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule deflate_module modules/mod_deflate.so
LoadModule disk_cache_module modules/mod_disk_cache.so
LoadModule ssl_module modules/mod_ssl.so
# DIVISAIT
```

El módulo de rewrite es necesario para el funcionamiento de los paths semánticos de la aplicación.

El modulo de deflate no es necesario aunque sí que es recomendable si no hay ningún elemento anterior al servidor web que realice las tareas de compresión de las páginas.

El modulo disk_cache no es necesario, pero se recomienda para habilitar un sistema de cacheo de los elementos estáticos de la aplicación y así evitar carga innecesaria al servidor de aplicaciones. Este módulo se vuelve más importante si es el propio apache es el que realiza la compresión de las peticiones, ya que evita la compresión repetitiva de los mismo elementos.

Configurar el Mod_JK:

```
# DIVISAIT
# Cargamos el modulo Mod_JK para que las conexion Apache-Tomcat sea posible.
LoadModule jk_module modules/mod_jk.so
# Definimos la ruta donde se encuentra el fichero con las propiedades de los workers
JkWorkersFile /etc/httpd/conf/workers.properties
# DIVISAIT
```

Definir correctamente el usuario (apache) con el que arranque el Apache HTTPD:

```
# DIVISAIT
#User daemon
#Group daemon
# El usuario/grupo con el que se debe arrancar el Apache HTTPD debe ser el usuario "apache"
User apache
Group apache
# DIVISAIT
```

Ocultar la versión de Apache:

```
# DIVISAIT
# Para que los apaches no envie que version de apache se esta ejecutando, por temas de seguridad
ServerSignature Off
#ServerSignature On
# DIVISAIT
```

Definición del NameVirtualHost y el VirtualHost del acceso a las aplicaciones del Portal en el fichero httpd-vhosts.conf

```
# DIVISAIT

NameVirtualHost *:80

NameVirtualHost *:443


<VirtualHost *:80>

    ServerAdmin admin@yourdomain.com
    ServerName www.saludcastillayleon.es
    ServerAlias *.publicaciones.saludcastillayleon.es
….


# VirtualHost SOLO para el Portal, en este ese se instala el certificado del Portal.
<VirtualHost *:443>

    ServerAdmin admin@yourdomain.com
    ServerName www.saludcastillayleon.es

…

# VirtualHost SOLO para las publicaciones, en este ese se instala el certificado wildcard
<VirtualHost *:443>

    ServerAdmin admin@yourdomain.com
    ServerName uptodate.publicaciones.saludcastillayleon.es
    ServerAlias *.publicaciones.saludcastillayleon.es
```

En los VirtualHost seguros, añadir las opciones: (en cada VH configurar el certificado concreto)

```
    SSLEngine on
    SSLOptions +ExportCertData +StdEnvVars
    SSLVerifyDepth 2

    SSLCipherSuite HIGH:MEDIUM:-SSLv2

    SSLCertificateChainFile /apache/conf/extra/sslCertificates/cacombinada.pem.crt
    SSLCACertificatePath /apache/conf/extra/sslCertificates/
    SSLCertificateFile /apache/conf/extra/sslCertificates/www.saludcastillayleon.es.2017.crt
    SSLCertificateKeyFile /apache/conf/extra/sslCertificates/www.saludcastillayleon.es.2017.RSA.key
```

Dentro del VirtualHost: Definición y compresión de los ficheros de texto. Hay que tener en cuenta que esto no es una configuración obligatoria y que si se incluye se reduce el consumo de ancho de banda de la aplicación pero se aumenta el uso de CPU del servidor donde se encuentre el servidor apache.

```
# DIVISAIT

# Compresion de los ficheros estaticos
<ifModule mod_deflate.c>
        # TIPOS DE FICHEROS QUE SE COMPRIMEN
        AddOutputFilterByType DEFLATE text/plain
        AddOutputFilterByType DEFLATE text/html
        AddOutputFilterByType DEFLATE text/css
        AddOutputFilterByType DEFLATE text/javascript
        AddOutputFilterByType DEFLATE application/javascript
        # NIVEL DE COMPRESION
        DeflateCompressionLevel 9
        # NO USAR COMPRESION EN NAVEGADORES QUE NO LO SOPORTAN
        BrowserMatch ^Mozilla/4 gzip-only-text/html
        BrowserMatch ^Mozilla/4\.0[678] no-gzip
```

```
        BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
    </ifModule>
```

Dentro del VirtualHost: Cacheado de elementos estáticos de la aplicación.

```
 # Además les cacheamos para que se mantenga una copia comprimida y así no se compriman
 # todas las peticiones
 <IfModule mod_disk_cache.c>
    CacheEnable disk /sanidad/css/
    CacheEnable disk /sanidad/css-sys/
    CacheEnable disk /sanidad/imagenes
    CacheEnable disk /sanidad/imagenes-sys/
    CacheEnable disk /sanidad/javaScript/
    CacheIgnoreQueryString On
    #CacheIgnoreURLSessionIdentifiers jsessionid
    CacheDefaultExpire 3600
    CacheRoot "/var/cache/mod_proxy/sanidad"
</IfModule>
```

Dentro del VirtualHost: Activación y rewrites necesarios para el acceso vía path semánticos.

```
    #Reglas Rewrite necesarias para el uso de path semanticos.
    RewriteEngine On

    RewriteLog    "/var/log/apache/logs/mod_rewrite_sanidad.log"
    RewriteLogLevel 0

    ## Para aquellos proyectos que tengan API swagger, pero se puede poner en general
    RewriteCond   %{REQUEST_URI} !^/sanidad/swagger-api
    ##
    RewriteCond %{REQUEST_URI} !^/sanidad/fonts
    RewriteCond %{REQUEST_URI} !^/sanidad/audio
    RewriteCond %{REQUEST_URI} !^/sanidad/css
    RewriteCond %{REQUEST_URI} !^/sanidad/css-sys
    RewriteCond %{REQUEST_URI} !^/sanidad/edit
    RewriteCond %{REQUEST_URI} !^/sanidad/dvFormsWeb
    RewriteCond %{REQUEST_URI} !^/sanidad/flash
    RewriteCond %{REQUEST_URI} !^/sanidad/imagenes
    RewriteCond %{REQUEST_URI} !^/sanidad/imagenes-sys
    RewriteCond %{REQUEST_URI} !^/sanidad/javaScript
    RewriteCond %{REQUEST_URI} !^/sanidad/textarea
    RewriteCond %{REQUEST_URI} !^/sanidad/error.html
    RewriteCond %{REQUEST_URI} !^/sanidad/forbidden.html
    RewriteCond %{REQUEST_URI} !^/sanidad/favicon.ico
    RewriteCond %{REQUEST_URI} !^/sanidad/cm
    RewriteCond %{REQUEST_URI} !^/sanidad-admin/
    RewriteCond %{REQUEST_URI} !^/robots.txt
    RewriteCond %{REQUEST_URI} !^/sitemap-.*
    RewriteRule ^/(.*) /sanidad/cm/semantic/$1 [PT]
    RewriteRule ^/robots.txt /sanidad/cm/robots.txt [PT]
    RewriteRule ^/(sitemap-.*) /sanidad/cm/gallery/sitemaps/$1 [PT]
    RewriteRule ^/fonts/(.*) /sanidad/fonts/$1 [PT]


    ###############################################################################
    # El apache se encarga de resolver las peticiones de paginas estaticas.
    # Esta configuración es solo valida si el Apache se encuentra en la misma
    # maquina que el Tomcat

        Alias /sanidad/audio "/tomcat/webapps/sanidad/audio"
        Alias /sanidad/css "/tomcat/webapps/sanidad/css"
        Alias /sanidad/css-sys "/tomcat/webapps/sanidad/css-sys"
        Alias /sanidad/datos "/tomcat/webapps/sanidad/datos"
        Alias /sanidad/fonts "/tomcat/webapps/sanidad/fonts"
        Alias /sanidad/edit "/tomcat/webapps/sanidad/edit"
        Alias /sanidad/estilo_css "/tomcat/webapps/sanidad/estilos_css"
        Alias /sanidad/flash "/tomcat/webapps/sanidad/flash"
        Alias /sanidad/imagenes "/tomcat/webapps/sanidad/imagenes"
        Alias /sanidad/imagenes-sys "/tomcat/webapps/sanidad/imagenes-sys"
        Alias /sanidad/javaScript "/tomcat/webapps/sanidad/javaScript"
```

```
    Alias /sanidad/textarea "/tomcat/webapps/sanidad/textarea"
    Alias /sanidad/error.jsp "/tomcat/webapps/sanidad/error.jsp"
    Alias /sanidad/forbidden.jsp "/tomcat/webapps/sanidad/forbidden.jsp"
    Alias /sanidad/error.html "/tomcat/webapps/sanidad/error.html"
    Alias /sanidad/forbidden.html "/tomcat/webapps/sanidad/forbidden.html"
    Alias /sanidad/favicon.ico "/tomcat/webapps/sanidad/favicon.ico"


    JkUnMount       /sanidad/audio/*  routerSSL
    JkUnMount       /sanidad/css/*  routerSSL
    JkUnMount       /sanidad/css-sys/*  routerSSL
    JkUnMount       /sanidad/datos/*  routerSSL
    JkUnMount       /sanidad/fonts/*  routerSSL
    JkUnMount       /sanidad/edit/*  routerSSL
    JkUnMount       /sanidad/estilos_css/*  routerSSL
    JkUnMount       /sanidad/flash/*  routerSSL
    JkUnMount       /sanidad/imagenes/*  routerSSL
    JkUnMount       /sanidad/imagenes-sys/*  routerSSL
    JkUnMount       /sanidad/javaScript/*  routerSSL
    JkUnMount       /sanidad/textarea/*  routerSSL


<Directory "/tomcat/webapps/sanidad/audio">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/css">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/css-sys">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/datos">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/fonts">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/edit">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/estilos_css">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/flash">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/imagenes">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/imagenes-sys">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/javaScript">
       Order allow,deny
       Allow from all
    </Directory>
    <Directory "/tomcat/webapps/sanidad/textarea">
       Order allow,deny
       Allow from all
    </Directory>
################################################################################
```

```
# Finalmente se montan las aplicaciones
JkMount /sanidad balancer
JkMount /sanidad/* balancer
JkMount /sanidad-admin balancer
JkMount /sanidad-admin/* balancer

</VirtualHost>

# DIVISAIT
```

La configuración del modulo de rewrite es necesaria para el funcionamiento de la aplicación. Esta configuración no se deberá tocar excepto en el poco frecuente caso de que se incluya un directorio raíz nuevo a la aplicación.

La definición del VirtualHost se puede (y se recomienda) que esté en un fichero (Ej: `vhost.conf`) a parte dentro del directorio: `/etc/httpd/conf.d/`

En tal caso se debe hacer referencia en el fichero ***httpd.conf*** del nuevo fichero:

```
Include conf.d/vhost.conf
```

Consideramos que el servicio de https vendrá dado por un elemento externo al servidor web y que por tanto no es necesaria su configuración en el fichero de configuración del Apache. En caso de ser necesario pónganse en contacto con el personal de DivisaiT para las instrucciones de cómo configurarlo.

3.3.3 Configuraciones Apache Tomcat

Una vez definido el worker y su puerto en el Apache HTTPD podremos configurar el conecto o conectores en el Apache Tomcat el fichero `/opt/apache-tomcat-7.0.73/conf/server.xml`

```
      <Connector     port="8020"     protocol="AJP/1.3"     redirectPort="8021"     maxSpareThreads="300"
minSpareThreads="150" maxThreads="1200" maxKeepAliveRequests="10000" keepAl

iveTimeout="15000" />

       <Connector  port="8021"  protocol="AJP/1.3"  scheme="https"  secure="true"  maxSpareThreads="300"
minSpareThreads="150" maxThreads="1200" maxKeepAliveRequests="1000

0" keepAliveTimeout="15000" />
```

Finalmente, para que todos los cambios realizados tengan efecto reiniciar el Apache Tomcat y el Apache HTTPD.

3.4 Aplicaciones Proxia ®

Proxia ® está formada por dos aplicaciones *cliente* y *administración*, se instalarán ambas en el servidor de aplicaciones Apache Tomcat.

Para instalar las aplicaciones se puede hacer o bien descomprimiéndolas como usuario tomcat en el directorio `/opt/apache-tomcat-7.0.73/webapps/` o bien utilizando la aplicación (en caso de existir) *manager* del Apache Tomcat

La aplicación cliente se debe instalar en todos los nodos (servidores de aplicaciones) que existan en la arquitectura en la que se va a desplegar la solución.

La aplicación de administración, recomendamos que solo este instalada en un nodo, ya que no es una aplicación que requiera alta disponibilidad.

Es importante que todos los nodos donde se instalen las aplicaciones estén sincronizados temporalmente (ej: mediante NTP). Esto es especialmente importante si la aplicación de administración está instalada en varias maquinas.

3.4.1 Configuración contexto de aplicación

Una vez instaladas las aplicaciones, es necesario configurar/revisar la configuración del contexto en el fichero context.xml de cada una de las aplicaciones.

Una configuración básica que se realiza en estos ficheros es la configuración de las conexiones entre la base de datos y la aplicación.

Un ejemplo de dicha configuración es la siguiente:

3.4.1.1 Pooles de conexiones para la aplicación de cliente

En la aplicación cliente deben existir los siguientes pooles de conexiones:

- jdbc/sanidad.mainDS: Es el pool de conexiones principal de la aplicación.
- jdbc/sanidad.searchDS: Es el pool utilizando para las operaciones de búsquedas.
- jdbc/sanidad.hpResourcesDS: Es el pool de conexiones utilizado para obtener imágenes o recursos almacenados en base de datos.
- jdbc/sanidad.syncDS: Es el pool de conexiones para operaciones de sincronización de ciertos elementos de los datos del portal.

Seguidamente se especifica un ejemplo de un fichero de configuración. Los tamaños de los pooles (maxActive) están considerados con valores relativos con lo que normalmente se ve en sistemas en explotación. La parametrización de los mismos es muy dependiente del nº de accesos de los usuarios, de la forma en que naveguen los mismos, y de las tasas de acierto de la caché, por lo que se recomienda que se vayan ajustando con datos del entorno de explotación.[2]

```
/opt/apache-tomcat-7.0.73/webapps/sanidad/META-INF/context.xml
<?xml version='1.0' encoding='utf-8'?>
<!-- The contents of this file will be loaded for each web application -->
<Context sessionCookiePath="/" sessionCookieName="JSESSIONID">

    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>


                <Resource auth="Container" name="jdbc/sanidad.mainDS"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID" username="demo" password="demo"
maxActive="20" initialSize="10" maxIdle="3" minIdle="1" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000"/>
```

[2] El tamaño del pool de sync no depende en la mayoría de los casos del acceso de los usuarios, por lo que un tamaño inicial de explotación de 10 conexiones puede ser más que suficiente.

```
        <Resource auth="Container" name="jdbc/sanidad.searchDS" factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID"
username="demo" password="demo" maxActive="8" initialSize="4" maxIdle="3" minIdle="1" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000"/>

        <Resource auth="Container" name="jdbc/sanidad.hpResourceDS"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID" username="demo" password="demo"
maxActive="8" initialSize="4" maxIdle="3" minIdle="1" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000"/>

        <Resource auth="Container" name="jdbc/sanidad.syncDS" factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID"
username="demo" password="demo" maxActive="5" initialSize="2" maxIdle="2" minIdle="1" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000"/>

</Context>
```

Y en la aplicación de administración deben existir los siguientes pooles

- jdbc/sanidad-admin.mainDS: Es el pool de conexiones principal de la aplicación.
- jdbc/sanidad-admin.searchDS: Es el pool utilizando para las operaciones de búsquedas.
- jdbc/sanidad-admin.regeneratorDS: Es el pool de conexiones utilizado para operaciones de regeneración de contenidos.
- jdbc/sanidad.mainDS: Es el pool de conexiones utilizado para probar las conexiones del componente de consultas SQL que se utilizará en cliente (misma nomenclatura que el pool principal de conexiones del cliente).

Seguidamente se especifica un ejemplo de un fichero de configuración. Los tamaños de los pooles (maxActive) están considerados con valores relativos con lo que normalmente se ve en sistemas en explotación. La parametrización de los mismos es muy dependiente del nº de accesos de los usuarios administradores por lo que se recomienda que se vayan ajustando con datos del entorno de explotación[3]

```
/opt/apache-tomcat-7.0.73/webapps/sanidad-admin/META-INF/context.xml
<?xml version='1.0' encoding='utf-8'?>
<!-- The contents of this file will be loaded for each web application -->
<Context sessionCookiePath="/" sessionCookieName="JSESSIONID">

        <!-- Default set of monitored resources -->
        <WatchedResource>WEB-INF/web.xml</WatchedResource>

        <Resource auth="Container" name="jdbc/sanidad-admin.mainDS"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID" username="demo" password="demo"
maxActive="25" initialSize="0" maxIdle="5" minIdle="0" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000" maxWait="2000"/>

        <Resource auth="Container" name="jdbc/sanidad-admin.searchDS"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID" username="demo" password="demo"
maxActive="15" initialSize="0" maxIdle="5" minIdle="0" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000" maxWait="2000"/>


        <Resource auth="Container" name="jdbc/sanidad-admin.regeneratorDS"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID" username="demo" password="demo"
```

[3] El valor del tamaño del jdbc/lsanidad-client.mainDS no depende especialmente del uso que se le de a la herramienta por lo que el tamaño especificado aquí (maxActive=5) es correcto para un sistema en explotación.

```
maxActive="15" initialSize="0" maxIdle="1" minIdle="0" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000" maxWait="2000"/>

        <Resource auth="Container" name="jdbc/sanidad.mainDS" factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin://@HOST:PORT/SID"
username="demo" password="demo" maxActive="5" initialSize="0" maxIdle="2" minIdle="0" validationQuery="SELECT 1 FROM DUAL"
timeBetweenEvictionRunsMillis="1000" maxWait="2000"/>

</Context>
```

Para que los pooles de datos configurados funcionen se deberá añadir en el directorio *lib* del Apache Tomcat la librería: ***ojdbc8.jar***

3.4.1.2 *Firewall entre Aplicaciones y Base de Datos*

En el caso de que los servidores de aplicaciones y la base de datos estén separados por algún elemento de red que pueda cortar las comunicaciones entre ellos, por ejemplo, un firewall, se debe tener las siguientes consideraciones al configurar la comunicación entre ambos.

En el servidor de aplicaciones se deben configurar las conexiones JDBC para que sean capaces de detectar aquellas conexiones caídas. Para ello se debe especificar el parámetro ENABLE=BROKEN en la definición de la conexión JDBC.

Así un ejemplo de la url a utilizar para utilizar en una conexión JDBC será el siguiente:

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=BROKEN)(address=(host=192.168.2.32)(protocol=tcp)(port=1521))(connect_data=(sid=EXA
MPLE)))
```

Además, se deberá definir un tiempo bajo (siempre menor que el tiempo de inactividad que establezca el firewall) del keepalive para las conexiones TCP a nivel de sistema operativo.

En un sistema operativo de tipo Linux se modificará mediante el comando:

```
echo 1200 > /proc/sys/net/ipv4/tcp_keepalive_time
```

En la base de datos se deberá especificar también un tiempo de comprobación para ver si las conexiones siguen activas. Esto se consigue especificando el parámetro sqlnet.expire_time que especifica el tiempo en minutos en que se manda un paquete para mantener la conexión viva.

Se deberá especificar un tiempo menor que el tiempo de inactividad que establezca el firewall (ej: 20 minutos).

Así, por ejemplo, estableceremos en el fichero sqlnet.ora del servidor la línea:

```
SQLNET.EXPIRE_TIME=20
```

3.4.2 Configuración Proxia

Configuraciones especificas de las aplicaciones, es necesario que se revisen/configuren con los datos correctos:

3.4.2.1 *Configuración de logs*

Para todo el sistema de logs de la aplicación se utiliza log4j en su versión 1.2.x (véase http://logging.apache.org/log4j/1.2/)

La configuración de logs de las aplicaciones de cliente y administración se puede encontrar en los siguientes ficheros:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad/WEB-INF/etc/log4jConfiguration.xml
/opt/apache-tomcat-7.0.73/webapps/sanidad-admin/WEB-INF/etc/log4jConfiguration.xml
```

Los ficheros de logs que genera la aplicación siguen un estructura genérica a partir de un directorio base (en los ejemplos que siguen el directorio base es el /opt/apache-tomcat-7.0.73/logs/sanidad/ para la aplicación cliente).

- En el directorio principal se guardan ficheros denominados debug.log con rotación que contienen información genérica de la aplicación.
- En el subdirectorio stats se guardan las estadísticas generadas por proxia relativa a los accesos de los usuarios. Estas estadísticas son procesadas por el programa statisticLoader
- En el subdirectorio errors se guardan todos los logs de error generados por la aplicación. Estos ficheros son rotados en función del día y comprimidos con bz2. El proceso de rotado no borra información por lo que aquí se conserva un histórico de los errores producidos en la herramienta.
- En el subdirectorio profiling se guarda información de rendimiento de la aplicación. Como en el caso anterior se rotan diariamente y se comprimen con bz2.
- En el subdirectorio cache se guarda información del sistema de cache de disco. Como en el caso anterior se rotan diariamente y se comprimen con bz2.

Los path donde se escribirán los logs se pueden encontrar de forma general en los parámetros File y FileNamePattern del fichero de configuración. Ej:

```
<param name="File" value="/opt/apache-tomcat-7.0.73/logs/sanidad/debug.log" />
```

3.4.2.2 *Configuración servicio de envío de correos*

El fichero de configuración del acceso al servidor SMTP que ofrece el servicio de relay para el envío de correos por parte de la aplicación se encuentra en la siguiente ruta: WEB-INF/etc/sendMail.xml. Se encontrará tanto en la aplicación de cliente como en la de administración.

En él se define el host que hara de relay y la dirección de correo electrónico que se utilizará como remitente de los correos que envíe la aplicación.

Un ejemplo del contenido de este fichero es el siguiente

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mailConfig SYSTEM "http://www.divisait.com/mail.dtd">
<mailConfig>
    <senderName>Proxia Admin</senderName>
    <senderMail>somebody@somewhere.com</senderMail>
    <mailServer>17.0.0.1 </mailServer>
</mailConfig>
```

3.4.2.3 *Configuración de la cache de disco*

La configuración de la cache de disco de la aplicación cliente[4] se puede encontrar en el fichero:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad/WEB-INF/etc/classcache.xml
```

[4] La herramienta de administración no dispone de caché de disco.

Algunos parámetros que se pueden configurar en la misma son los siguientes:

- Activación de la caché de disco

```
activated="true"
```

- Base del path del directorio donde se escribirá la caché de disco

```
baseDir="/WEB-INF/cache"
```

- Si el path anterior es absoluto o no (por defecto y si no se especifica este parámetro será relativo al context-root de la aplicación, aunque empiece por /)

```
absolutePath="false"
```

- Tamaño máximo de la caché en KB (en el ejemplo siguiente está configurado a 8GB)

```
<size maxSize="8388608"
```

La caché de disco, en el caso de path no absoluto, se ubica en directorios de la forma:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad/WEB-INF/etc/cache_1429875276209
```

El espacio máximo ocupado por este directorio será el configurado anteriormente.

3.4.2.4 *Parámetros del acceso a base de datos*

La configuración de los parámetros que determinan el acceso a la base de datos para determinadas operaciones se encuentra en:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad-admin/WEB-INF/etc/rdbmsPlugin.properties
```

En este fichero se puede configurar los nombres de los tablespaces del usuario de base de datos, en concreto se pueden especificar los nombres de los tablespaces de datos, índices, búsquedas, lobs y mviews:

```
dataTableSpace          =POSA_TSD1
indexTableSpace         =POSA_TSI1
iotTableSpace           =POSA_TSC1
lobTableSpace           =POSA_TSL1
mviewTablespace         =POSA_TSD1
```

3.4.2.5 *Configuración del Planificador de Tareas*

La configuración del planificador de tareas de la aplicación, herramienta de administración, se puede encontrar en el fichero:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad-admin/WEB-INF/classes/quartz.properties
```

Debe utilizará el dataSource main de la aplicación de administración, por lo que se debe configurar en la siguiente línea.

```
# Definicion del jobDS (utilizando jndi)
org.quartz.dataSource.jobDS.jndiURL = /jdbc/sanidad-admin/mainDS
```

3.4.2.6 *Configuración de la comunicación JMX*

Las aplicaciones Proxia que existan en diferentes servidores necesitan comunicarse vía JMX, para ello se configuran los puertos JMX en el fichero:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad/WEB-INF/etc/jmx-cache.xml
```

Ej:

```
<ports min="11097" max="11197"/>
```

Es necesario que existen conectividad (en los puertos jmx configurados) entre las maquinas donde estén instaladas todas las aplicaciones, tanto clientes como administraciones.

También es necesario la comunicación para la caché de objetos que se define en el fichero objectcache.xml:

```
   <cacheManagerPeerListenerFactory
class="com.divisait.dv2ee.controller.objectcache.RMIObjectCacheManagerPeerListenerFactory"
        properties="hostName=${server.ip},minport=1300,maxport=1349,socketTimeoutMillis=120000"
        propertySeparator="," />
```

En este caso se debería reservar el rango de puerto 1300 hasta el 1349 para la comunicación entre las distintas caches de objetos.

En ambos ficheros se define la variable ${server.ip} que es sustituida en tiempo de ejecución por la aplicación por la primera IP interna que tenga la máquina. Se puede sustituir este variable por un valor literal con la IP o hostaname de la máquina que se quiera utilizar para las comunicaciones entre máquinas.

Se recomienda el uso de IP para que las comunicaciones sean más rápidas. En caso de utilizar el hostname el resto de máquinas deben resolver este hostname a la maquina donde se encuentra la aplicación que se está configurando.

3.4.2.7 *Configuración especificas del entorno*

Las aplicaciones en función del entorno donde se instalen (producción, preproducción, etc) se deben parametrizar con unas configuraciones u otras dependiendo de dicho entorno.

Proxia para esta situación tiene un fichero específico en el que configurar los parámetros necesarios:

```
/opt/apache-tomcat-7.0.73/webapps/sanidad/WEB-INF/etc/config.properties
```

Actualmente en dicho fichero se deben configurar los siguientes aspectos:

```
secure.server = https://www.saludcastillayleon.es
non.secure.server = http://www.saludcastillayleon.es

keypassfile = /WEB-INF/etc/sanidad.key
```

Las dos primeras entradas identifican cuales van a ser los dominios base de acceso a la aplicación. Estas entradas son utilizadas para generar las urls internas absolutas que genere la aplicación.

La entrada keypassfile identifica un fichero de 32 bytes que contiene la clave con la que se encriptan las claves de los usuarios de la herramienta. Este fichero debe tener el mismo contenido en todas las aplicaciones que se conecten a un mismo usuario de BBDD de Proxia, es decir al mismo sistema de usuarios. Por lo tanto, la

aplicación de administración y cliente deben tener le mismo fichero de clave. Si se pierde este fichero no se pueden recuperar las claves de los usuarios y por tanto es necesario resetear todas las claves de los usuarios. Si se modifica, o se utiliza otro, no se podrá autenticar ningún usuario, y las aplicaciones no serán capaces de arrancar.

3.4.2.1 *Licencia Proxia©*

Todas las aplicaciones desarrolladas con Proxia deben contar con un fichero de licencia proporcionado por DivisaiT. El fichero proporcionado se debe situar dentro de la aplicación en la ruta WEB-INF/license.proxia. Para que la verificación de la licencia se pueda realizar la aplicación debe poder acceder vía HTTP, de forma permanente, a https://www.divisait.com.

Si la aplicación no está licenciada correctamente dejará de dar servicio.

3.4.2.2 *Aplicaciones de apoyo*

Para la habilitación de algunas funcionalidades que soporta Proxia se necesita tener instaladas en el servidor de aplicaciones determinadas aplicaciones cuyas rutas serán configuradas en el fichero binaries-paths.properties que se encuentra en el directorio WEB-INF/classes de las aplicaciones de cliente y administración.

Un ejemplo del contenido de este fichero es el siguiente:

```
# Este fichero contiene paths a los ejecutables que son invocados por la aplicación
# para realizar cierto tipo de operaciones.
# Debe ser modificado en cada instalación para apuntar al fichero real.

ffmpeg = /usr/local/bin/ffmpeg
pdf2swf = /usr/local/bin/pdf2swf
swfrender = /usr/local/bin/swfrender
```

Los ejecutables anteriormente citados son utilizados por el proceso de conversión de PDF's a formato EBook.

Las versiones actualmente soportadas de estos binarios son las siguientes:

- ffmpeg versión 2.2.1
- swftools versión 0.9.2

3.5 Instalación Aplicaciones Externas

3.5.1 StatisticLoader

Esta aplicación se encarga de realizar el proceso de carga de las diferentes tipos de estadísticas de acceso a la aplicación.

Esta aplicación no es requerida para el funcionamiento de Proxia, pero si que lo es si se quieren utilizar los informes de estadísticas que proporciona la herramienta de administración de Proxia.

En este apartado suponemos que el directorio de logs configurado en los ficheros log4jConfiguration es el siguiente: `/opt/apache-tomcat-7.0.73/logs/sanidad/stats`

La herramienta StatisticLoader se instalara en el directorio: `/appserver/StatisticLoader/`

Esta instalación se supone bajo equipamiento y aplicaciones base (servidor de aplicaciones y base de datos) ya instaladas y configuradas para un correcto funcionamiento con Proxia®.

Prerrequisitos:

- Oracle JDK 1.8
- ANT 1.9.x
- El usuario que realice la instalación tenga acceso y permisos de lectura sobre los logs.
- La base de datos debe ser accesible desde la maquina en que se realice el proceso de carga de las estadísticas, en general se realizará desde cada una de los nodos en el que está instaladas las aplicaciones clientes.

Una vez que se cumplen los requisitos, definiremos los diferentes casos en los que se usara la herramienta de carga de estadísticas:

Caso genérico

En primer lugar echaremos un vistazo al directorio `/appserver/StatisticLoader`

```
StatisticLoader$ ls
run.sh
build-load.xml
run-anteriores.sh
etc
lib
output
log
```

Como se puede ver tenemos varios scripts .sh, las tareas del ant (ficheros xml) que usaran los script y los directorios de configuración (/etc) de las librerías(/lib) y de logs(/log).

Los ficheros que necesitaremos modificar para caso genérico serán los siguientes:

- **jdbc.xml**: fichero que se encuentra dentro de directorio de configuración (directorio etc) en el que especificaremos la secuencia de conexión con la base de datos.
- **run.sh:** script principal que se encuentra en el directorio raíz de la aplicación y que se encargara de cargar las estadísticas diarias.

  Se establecerán y exportaran las variables de entorno necesarias para el java, ant y la variable NLS_LANG que debe tomar el valor correspondiente a la base de datos.

Se ejecutan las tarea del ant que se encargue de cargar las estadísticas del día anterior al que se ejecute.

```
#!/bin/sh
#
# Script que llama a la tarea del ANT encargada de cargar las estadÃ-icas
LANG=es_ES.UTF-8
LC_ALL=es_ES.UTF-8
NLS_LANG=SPANISH_SPAIN.UTF8
JAVA_HOME=/usr/java/latest/
ANT_OPTS="-Xms512M -Xmx512M -XX:MaxPermSize=128M"
ANT_HOME=/usr/share/ant
PATH=$PATH:$JAVA_HOME/bin:$ANT_HOME/bin

export PATH JAVA_HOME ANT_HOME ORACLE_HOME NLS_LANG LANG LC_ALL LD_LIBRARY_PATH ANT_OPTS
# Para asegurarnos pasamos al directorio donde esta el build.xml
cd /appserver/StatisticLoader
# Y ejecutamos la tarea del ant
ant -v -f build-load.xml load
```

- **build-load.xml**: tarea del ANT para la carga de estadísticas. Las campos a tener en cuenta y que es muy posible que haya que modificar son los siguientes:

  - Directorio donde se encuentren las librerías de la aplicación:

```
<!-- Las librerías del core cliente, normalmente se tomará el directorio de instalación de la
aplicación cliente) [*]-->
<property name="core-lib" value=" /opt/apache-tomcat-7.0.73/webapps/sanidad/WEB-INF/lib/"/>
```

  - Directorio donde se encuentren los archivos de logs, este será el que se haya puesto en el fichero *log4jConfiguration.xml* del directorio WEB-INF/etc de la aplicación cliente:

```
<!-- Directorio donde se encuentran los archivos de logs [*]-->
<property name="logs" value="/opt/apache-tomcat-7.0.73/logs/sanidad/stats"/>
```

  - Definición de los nombres de los ficheros de logs:

```
<!-- Nombre del fichero de datos de acceso para la carga -->
<property name="httpd-dat" value="httpd.dat"/>
<!-- Nombre del fichero de datos de sesiones para la carga -->
<property name="session-dat" value="session.dat"/>
<!-- Nombre base de los ficheros de logs de recursos para la carga -->
<property name="resources-dat" value="httpd-resources.dat"/>
<!-- Nombre base de los ficheros de logs de mappings para la carga -->
<property name="mappings-dat" value="httpd-mappings.dat"/>
<!-- Nombre base de los ficheros de logs de servicios para la carga -->
<property name="services-dat" value="httpd-services.dat"/>
<!-- Nombre base de los ficheros de logs de busquedas para la carga-->
<property name="searchs-dat" value="httpd-searchs.dat"/>
```

    En el caso de que los nombres de los ficheros se hayan modificado en el fichero *log4jConfiguration.xml* y no coincidan con los definidos en el *build-load.xml*, se debe modificar el nombre definido en la propiedad value por el correcto.

  - El modo en el que se va a cargar las estadísticas, cuando se cargan las estadísticas de un día por primera vez el modo será “add”.

```
<!-- El modo en el que se va a cargar las estadísticas, pueden ser de tres tipos:
          add: añade los datos aunque ya existiesen registros para ese día.
        overwrite: añade o actualiza los registros especificados.
         skip: Inserta los registros especificados, si ya existen registros no hace nada.
-->

    <property name="mode" value="add"/>
```

- Pagina web de prueba para forzado de rotado de logs:

```
<!-- Página web de prueba para forzado de rotado de logs [*]-->
<property name="web-test-page" value=”http://sanidad.local/”>
```

- Si queremos quitar los referals[5] y además la aplicación respondiese por varias urls distintas:

```
<!-- La url del sitio donde esta publicada la aplicación, sirve para al cargar las estadísticas
quitar los referals que provengan de el mismo.
Si la aplicación respondiese en varios nombres se debe modificar la tarea load-referals
añadiéndole tantas etiquetas referral como sitios distintos responde la aplicación
[*]-->
<property name="selfSite"  value=" http://sanidad.local/"/>
<!-- Eliminamos los referal del mismo sitio -->
<referal pattern="${selfSite}"/>
<!-- Si se quieren quitar referal de otros sitios (o de otros nombres del mismo sitio se deben
añadir tags referal con los distintos sitios -->
```

Una vez que los ficheros les tenemos preparados la tarea de cargar las estadísticas diarias se añadirá al crontab:

```
[tomcat@]/crontab -e
#sanidad  CARGA DE ESTADISTICAS
0 3 * * * /appserver/StatisticLoader/run.sh > /appserver/StatisticLoader/salidas.txt
#FIN sanidad
```

de tal manera que todos los días a las 3:00 de la madrugada se ejecutara el *run.sh*, el resultado que se iría mostrando por pantalla lo almacenaremos en principio en el fichero salidas.txt hasta que veamos que el proceso funciona correctamente, una vez que dicho funcionamiento sea correcto sería bueno que lo redireccionaremos al fichero especial: `> /dev/null 2>&1`

Si existiera más de un nodo, es decir más de una maquina, habría que realizar la instalación de las estadísticas en todas las maquinas que se tuviesen, ya que las estadísticas totales serían las sumas de las estadísticas que hubiesen en todas la maquinas. Para evitar posibles problemas de acceso concurrente a las tablas de estadísticas se deberá programar la carga de cada nodo a horas distintas.

Caso genérico carga días anteriores

El caso genérico seria el habitual, pero puede ocurrir el caso de que las estadísticas de unos días no se hayan cargado por lo que el proceso anteriormente explicado no funcionaria para poder cargar las estadísticas, ya que la tarea del crontab solo carga las estadísticas del día anterior. Cuando tengamos este problema usaremos los ficheros:

[5] URLs externas que referencian a nuestro sitio web

run-anteriores.sh: básicamente es igual que el run.sh del caso genérico exceptuando la tarea del ant en la cual se especificará el día del que se quiere recuperar las estadísticas, si son varios días habrá una tarea por cada día.

```
#!/bin/sh
#
# Script que llama a la tarea del ANT encargada de cargar las estadÃ-icas
LANG=es_ES.UTF-8
LC_ALL=es_ES.UTF-8
NLS_LANG=SPANISH_SPAIN.UTF8
JAVA_HOME=/usr/java/latest/
ANT_OPTS="-Xms512M -Xmx512M -XX:MaxPermSize=128M"
ANT_HOME=/usr/share/ant
PATH=$PATH:$JAVA_HOME/bin:$ANT_HOME/bin

export PATH JAVA_HOME ANT_HOME ORACLE_HOME NLS_LANG LANG LC_ALL LD_LIBRARY_PATH ANT_OPTS
# Para asegurarnos pasamos al directorio donde esta el build.xml
cd /appserver/StatisticLoader
# Y ejecutamos la tarea del ant
# Ejemplo:
ant -v -f build-load.xml -DYESTERDAY=2015-06-17 load
```

El proceso a realizar será manual, lanzando el script run-anteriores.sh

3.6 Recomendaciones de mantenimiento

3.6.1 Rotado y Compresión de Logs

Aplicaciones

Las aplicaciones generan una serie de logs en los directorios:

```
/opt/apache-tomcat-7.0.73/logs/sanidad
/opt/apache-tomcat-7.0.73/logs/sanidad-admin
```

Entre los logs que van a crecer y de los que por tanto hay que programar una política de movimiento/compresión para que no se quede el servidor sin espacio:

`/opt/apache-tomcat-7.0.73/logs/sanidad/stats` [Rotan automáticamente cada día][Utilizados para la carga de estadísticas de acceso al Portal][No se deben borrar, aunque sí almacenar en otro sitio, ya que son los ficheros donde se registran los datos de acceso a la aplicación para las estadísticas]

`/opt/apache-tomcat-7.0.73/logs/sanidad/profiling` [Rotan y se Comprimen automáticamente cada día][Se recomienda mantener una copia de los últimos 2 meses]

`/opt/apache-tomcat-7.0.73/logs/sanidad/cache` [Rotan y se Comprimen automáticamente cada día][Se recomienda mantener una copia de los últimos 2 meses]

`/opt/apache-tomcat-7.0.73/logs/sanidad/errors` [Rotan y se Comprimen automáticamente cada día][Se recomienda mantener siempre una copia de ellos, o con un periodo de rotado muy largo, un año]

`/opt/apache-tomcat-7.0.73/logs/sanidad-admin/profiling` [Rotan y se Comprimen automáticamente cada día][Se recomienda mantener una copia de los últimos 2 meses]

`/opt/apache-tomcat-7.0.73/logs/sanidad-admin/errors` [Rotan y se Comprimen automáticamente cada día]][Se recomienda mantener siempre una copia de ellos, o con un periodo de rotado muy largo, un año]

Apache Tomcat

El rotado de logs del Apache Tomcat, en concreto de los ficheros de tipo: `localhost_access_log.`, es automático con la configuración que viene por defecto en el fichero `server.xml`:

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log." suffix=".txt"
               pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />
```

Apache HTTPD

Se deberá especificar el rotado de logs del apache en el fichero `/etc/logrotate.d/httpd.`

3.6.2 URL de monitorización Balanceador

Las herramientas de Proxia, tanto cliente como administración, disponen de una URL especial que permite realizar una monitorización muy básica del estado de la aplicación.

La URL de monitorización para la aplicación cliente será

http://sanidad.local/sanidad/cm/server-status

Las peticiones a esta página devuelve el siguiente resultado si todo es correcto

```
test_ok
test_datasource.mainDS = ok
test_container_free_disk_space = ok
```

La primera línea estará presente si la aplicación está respondiendo. La segunda línea toma el valor de OK si el pool de datos mainDS está pudiendo acceder a la base de datos.

3.7 Anexos

3.7.1 Configuración por defecto con la que arrancara el Apache Tomcat

```
#!/bin/sh
JAVA_HOME=/usr/java/latest/

JAVA_OPTS="-d64 -server
-Duser.timezone=Europe/Paris
-Djava.awt.headless=true
-Xms16G -Xmx16G -Xss2048k
-Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dhttp.proxyHost=proxy.mydomain.es -Dhttp.proxyPort=8080
-XX:HeapDumpPath=/opt/apache-tomcat-7.0.73/logs/
-XX:+HeapDumpOnOutOfMemoryError
-Dsun.net.http.allowRestrictedHeaders=true
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=1200
-Dcom.sun.management.jmxremote.password.file=/opt/apache-tomcat-7.0.73/conf/jmxremote.password
-Dcom.sun.management.jmxremote.access.file=/opt/apache-tomcat-7.0.73/conf/jmxremote.access
```

```
-Dcom.sun.management.jmxremote.ssl=false
-Dsun.java2d.cmm=sun.java2d.cmm.kcms.KcmsServiceProvider
-Dfile.encoding=UTF-8
-Djava.rmi.server.hostname=IP_SERVIDOR"

PATH=$PATH:$JAVA_HOME/bin
LANG=es_ES.UTF-8
LC_ALL=es_ES.UTF-8
NLS_LANG=SPANISH_SPAIN.UTF8
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/apr/lib
export PATH JAVA_HOME JAVA_OPTS LANG LC_ALL NLS_LANG LD_LIBRARY_PATH

CATALINA_HOME=/opt/apache-tomcat-7.0.73
CATALINA_BASE=/opt/apache-tomcat-7.0.73
export CATALINA_HOME CATALINA_BASE
```

Ejemplo de los dos ficheros necesarios para la configuración de jmxremote con autenticación:

jmxremote.password

```
sanidadMonitor  JMXSanidad
sanidadControl  JMXSanidad
```

jmxremote.access

```
sanidadMonitor   readonly
sanidadControl   readwrite
```

3.7.2 Definición de conector Apache-Tomcat

```
# Conector para el tomcat presente en el sistema
# Al estar el servidor web y el servidor de aplicaciones en diferentes servidores, el valor del parametro host sera la IP del servidor de apliaciones, en el caso de que estuvieran en el mismo servidor dicha variable tendria el valor‘localhost’

worker.list=balancer

worker.tomcat1.type=ajp13
worker.tomcat1.host=IP-nodo1
worker.tomcat1.port=8010
worker.tomcat1.ping_mode=A
# Es recomedable que connection_pool_size=ThreadsPerChild
worker.tomcat1.connection_pool_size=25
# connection_pool_minsize = connection_pool_size / 2
worker.tomcat1.connection_pool_minsize=12
# socket_keepalive si existe un firewall entre Apache y Tomcat
worker.tomcat1.socket_keepalive=True
worker.tomcat1.socket_timeout=300
worker.tomcat1.lbfactor=10

worker.tomcat2.type=ajp13
worker.tomcat2.host= IP-nodo2
worker.tomcat2.port=8010
worker.tomcat2.ping_mode=A
# Es recomedable que connection_pool_size=ThreadsPerChild
worker.tomcat2.connection_pool_size=25
# connection_pool_minsize = connection_pool_size / 2
worker.tomcat2.connection_pool_minsize=12
# socket_keepalive si existe un firewall entre Apache y Tomcat
worker.tomcat2.socket_keepalive=True
worker.tomcat2.socket_timeout=300
worker.tomcat2.lbfactor=10

worker.tomcat3.type=ajp13
worker.tomcat3.host= IP-nodo3
worker.tomcat3.port=8010
worker.tomcat3.ping_mode=A
# Es recomedable que connection_pool_size=ThreadsPerChild
```

```
worker.tomcat3.connection_pool_size=25
# connection_pool_minsize = connection_pool_size / 2
worker.tomcat3.connection_pool_minsize=12
# socket_keepalive si existe un firewall entre Apache y Tomcat
worker.tomcat3.socket_keepalive=True
worker.tomcat3.socket_timeout=300
worker.tomcat3.lbfactor=10


worker.balancer.type=lb
worker.balancer.sticky_session=true
worker.balancer.balance_workers=tomcat1,tomcat2,tomcat3
```