

## Proyecto: AG-SIGESMAN16

### Manual de Instalación y Configuración Siges

**Creación**

06-07-2023

**Versión**

3.0

**Plantilla**

SS-SOFTFACT-TE-62-ES

**Código**

AG-SIGESMAN16-MAN-INST-02-ES

**Autor**

BS

**Página**

1 de 30



## Sumario

1	Introducción.....	4
2	Propósito.....	4
3	Audiencia.....	4
4	Introducción.....	4
4.1	Requisitos.....	4
4.1.1	Requerimientos técnicos.....	4
4.1.2	Requerimientos de Software.....	4
5	Visión General de la Instalación.....	5
6	Proceso de Instalación.....	5
6.1	Instalación y configuración de base de datos.....	5
6.2	Instalación de Java.....	6
6.3	Instalación y configuración de JBOSS AS 7.1.1.....	7
6.4	Actualización de módulos JBOSS AS 7.1.1.....	8
7	Consideraciones de Seguridad.....	11
7.1	Autorizar el acceso a la aplicación desde otros equipos.....	11
7.2	Habilitar el acceso seguro (HTTPS).....	11
7.3	Habilitar comprensión de datos.....	13
8	Configuración de Parámetros.....	13
8.1	Configuración de modulo MySQL en JBOSS AS 7.1.1.....	13
8.2	Configuración de modulo PostgreSQL en JBOSS AS 7.1.1.....	14
8.3	Configuración de driver y datasource MySQL en JBOSS AS 7.1.1.....	15
8.4	Configuración de driver y datasource PostgreSQL en JBOSS AS 7.1.1.....	16
8.5	Configuración de servidor de correo electrónico.....	18
8.6	Configurar dominio de autenticación.....	19
8.7	Configurar niveles de log.....	19
8.8	Cambiar puertos del JBOSS AS.....	19
9	Instalacion y configuracion de Siges.....	20
9.1	Instalar aplicación en JBOSS AS 7.1.1.....	20
9.2	Super administrador.....	20
9.3	Usuarios por defecto en el ambiente levantado.....	21
9.4	Configuración de la aplicación.....	21
10	Respaldo y mantenimiento.....	24
11	Optimización de ambiente.....	25
11.1	Optimización de Java.....	25

11.2 Optimización de MySQL.....	26
11.3 Optimización de JBOSS.....	27
12 Entorno de desarrollo.....	29
13 Historial de cambios al documento.....	30

## 1 Introducción

El presente documento corresponde al Manual de Instalación y Configuración del sistema Siges.

## 2 Propósito

El propósito de este documento es brindar una descripción detallada de los requerimientos necesarios para la instalación y configuración de Siges.

## 3 Audiencia

Este documento está dirigido a los stakeholders del proyecto así como los equipos técnicos de ambas partes (Sofis y Agesic) y responsables de los ambientes de testing y producción de las aplicaciones.

## 4 Introducción

### 4.1 Requisitos

#### 4.1.1 Requerimientos técnicos

Este capítulo describen los requerimientos de software, para la construcción de los ambientes de pruebas y producción de Siges.

**IMPORTANTE:** El servidor donde se instalará la aplicación, debe de ser un servidor dedicado. Es decir, no se debe compartir el servidor con otra aplicación, como también se debe respetar las versiones de las tecnologías usadas ya que si no se dará error de compatibilidad.

#### 4.1.2 Requerimientos de Software

Los requerimientos de Software son:

- Sistema Operativo: Linux, preferentemente CentOS o Debian.
- Servidor de Aplicaciones: Jboss AS 7.1.1
- Servidor de Base de Datos: MySQL 5.6 o PostgreSQL 9.4
- Java: versión 1.7

## 5 Visión General de la Instalación

Tener en cuenta los requisitos, es muy importante para que la instalación sea exitosa.

## 6 Proceso de Instalación

### 6.1 Instalación y configuración de base de datos

#### MySQL Server 5.6

MySQL Server también es un producto de distribución libre (en su versión Community). Puede descargarse gratuitamente en la siguiente URL (asegurarse que se trate de la versión 5.6): <http://dev.mysql.com/downloads/mysql/>

Luego de descargar el instalador, ejecutarlo, y seguir los pasos indicados por el mismo. Como parte del proceso de instalación será necesario establecer una contraseña para el usuario root (el usuario administrador de la instalación). Se recomienda permitir que el instalador configure MySQL como servicio de forma tal que inicie automáticamente cuando se enciende el equipo.

#### PostgreSQL 9.4

PostgreSQL también es un producto de distribución libre (en su versión Community). Puede descargarse gratuitamente en la siguiente URL (asegurarse que se trate de la versión 9.4): <http://www.postgresql.org/download/>

Luego de descargar el instalador, ejecutarlo, y seguir los pasos indicados por el mismo. Como parte del proceso de instalación será necesario establecer una contraseña para el usuario root (el usuario administrador de la instalación). Se recomienda permitir que el instalador configure PostgreSQL como servicio de forma tal que inicie automáticamente cuando se enciende el equipo.

#### Crear base de datos de la aplicación.

Para crear la base de datos para la aplicación se debe iniciar sesión en el servidor MySQL o PostgreSQL (según la base elegida) usando la cuenta root y ejecutar el script **siges\_clean\_XXXX-MySQL.sql** (para MySQL) o **siges\_clean\_XXXX-PostgreSQL.sql** (para PostgreSQL) proporcionado. Puede usarse el programa mysql incluido en la instalación de MySQL, o el cliente gráfico MySQL Workbench que puede descargarse del mismo sitio donde se descargó el instalador en el caso de optar por MySQL. En el caso de PostgreSQL se puede utilizar el programa incluido en la instalación o el entorno gráfico pgAdmin.

Al finalizar la ejecución del script de creación de la base de datos se recomienda crear una cuenta de usuario específica para la aplicación SIGES, que será utilizada para la comunicación entre el servidor de aplicaciones (JBoss AS) y el servidor de bases de datos (MySQL Server o PostgreSQL Server). Si bien puede utilizarse la cuenta root no se recomienda por razones de seguridad.

En MySQL, para crear una nueva cuenta debe ejecutarse la siguiente sentencia SQL:

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL ON siges.* TO {user}@{host} IDENTIFIED BY {pass};
```

En PostgreSQL, para crear una nueva cuenta debe ejecutarse las siguientes sentencias SQL:

```
CREATE USER {user} PASSWORD {pass};
```

```
GRANT ALL PRIVILEGES ON DATABASE siges TO {user};
```

Donde **{user}** debe ser remplazado por el nombre de usuario deseado (por ejemplo, "sigesusr"), **{pass}** por la contraseña deseada (por ejemplo, "sigespwd") y **{host}** por la dirección IP del equipo donde reside el servidor de aplicaciones JBoss AS (por ejemplo, "192.198.2.68"); si el servidor de aplicaciones JBoss AS y el servidor de bases de datos (MySQL o PostgreSQL) Server residen en el mismo equipo, puede usarse el alias "localhost".

## 6.2 Instalación de Java

Para funcionar, JBoss AS requiere que exista una instalación de Java 7 en el sistema. Si no lo hay es necesario instalar uno. Se puede obtener un instalador apropiado a partir de la siguiente URL (asegurarse que se trate de Java 7): "<http://www.oracle.com/technetwork/java/javase/downloads/index.html>".

Una vez descargado el instalador apropiado debe ejecutarlo, siguiendo las instrucciones proporcionadas por el mismo.

Tanto en el caso de que ya estuviese instalado el entorno de ejecución de Java, como en el caso de haber tenido que instalarlo, antes de continuar se debe verificar que exista una variable de entorno denominada **JAVA\_HOME**, cuyo valor sea la ruta completa hasta el directorio donde se encuentra instalado el entorno de ejecución de Java 7. La forma de configurar esta variable depende del sistema operativo en uso (por más detalles sobre cómo configurar variables de entorno del sistema operativo consultar el manual del mismo). Como ejemplo, si el directorio de instalación del entorno de ejecución de Java 7 es "C:\Program Files\Java\JRE7", entonces el valor de la variable de entorno JAVA\_HOME debe ser también "C:\Program Files\Java\JRE7".

## 6.3 Instalación y configuración de JBOSS AS 7.1.1

En este capítulo se describen los pasos para realizar la instalación y configuración de Jboss AS 7.1.1.

**Nota:** Cualquier cambio en las configuraciones de Jboss, requiere el reinicio del mismo.

### Descarga

JBoss AS es un servidor de aplicaciones de libre distribución. Puede descargarse gratuitamente desde la siguiente URL (asegurarse que se trate de JBoss AS 7.1.1): "<http://www.jboss.org/jbossas/downloads/>".

Luego de descargarlo, la instalación consiste simplemente en descomprimir el archivo descargado (un archivo comprimido de tipo ZIP o TAR-GZIP) en la ubicación deseada. Está ubicación será el llamado "directorio de instalación" del servidor de aplicaciones, y de aquí en más será denotado como "{JBOSS\_HOME}".

Para probar el funcionamiento correcto del servidor de aplicaciones se puede ejecutar el script standalone.bat (en MS Windows) o standalone.sh (en Linux) que se encuentra dentro del directorio {JBOSS\_HOME}/bin. Si la instalación es correcta, el servidor de aplicaciones JBoss 7 debería iniciar en unos pocos segundos, mostrando el mensaje "JBoss AS 7.1.1.Final 'Brontes' started". Si en lugar de ello muestra un error y no inicia, comprobar la configuración de la variable de entorno JAVA\_HOME como fue explicado en la sección Error: no se encontró el origen de la referencia.

Si va a realizar una ventana de mantenimiento como por ejemplo respaldar base de datos se debe detener el servidor de aplicaciones con el siguiente comando : desde la carpeta bin del jboss

```
./jboss-cli.sh --connect command=:shutdown
```

(2023)

Debido a otro problema que se presento durante el armado de ambiente para las nuevas versiones se agrega lo siguiente por si llegara a pasar.

Error/Exception:

```
Exception in thread "main" javax.xml.parsers.FactoryConfigurationError: Provider  
__redirected.__SAXParserFactory could not be instantiated: java.lang.reflect.InvocationTargetException.
```

...

Caused by: java.lang.reflect.InvocationTargetException

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

...

Caused by: java.lang.NullPointerException

No tiene mucho sentido copiar todo el log del servidor, por ende se copio un fragmento que permita identificar el error/exception

Esto se soluciona copiando el jar que se encuentra en la carpeta "UpdateJbossJar" en la direccion de instalacion del servidor ("{JBOSS\_HOME}").

Notaran que en esa dirección raíz del jboss ya se encuentra un jboss-modules.jar, respalden este o cambien el nombre para no perderlo y al que recién copiaron le remplazan el nombre por "jboss-modules.jar". Ya con esto deberian poder levantar el servidor y que les muestre el mensaje 'Brontes' started".

## 6.4 Actualización de módulos JBOSS AS 7.1.1

En esta sección se describen los diferentes módulos que deben actualizarse para el correcto funcionamiento de las aplicaciones. Para cada módulo se provee una tabla que contiene un listado de las librerías con la siguiente información:

- Librería: identificador de la librería en el documento
- Módulo: nombre del módulo en el JBoss.
- Group ID, Artifact ID, Versión: identificador de la librería en los repositorios maven.

Recordar que si el módulo del JBoss se llama "com.sun.faces" la librerías y el archivo "module.xml" se encuentran en el directorio "{JBOSS\_HOME}/modules/com/sun/faces/main".

- **Módulo Javassist:**

• Librería	• Módulo	• Group ID	• Artifact ID	• Versión
• 1	• org.javassist	• org.javassist	• javassist	• 3.18.1-GA

Actualizar el archivo "module.xml" del módulo "org.javassist" por el siguiente contenido

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.javassist">
  <properties>
    <property name="jboss.api" value="private"/>
  </properties>
  <resources>
    <resource-root path="javassist-3.18.1-GA.jar"/>
  </resources>
  <dependencies>
  </dependencies>
</module>
```



&lt;/module&gt;

- **Módulo Hibernate**

Librería	Módulo	Group ID	Artifact ID	Versión
2	org.hibernate	org.hibernate	hibernate-core	4.2.21.Final
3	org.hibernate	org.hibernate.common	hibernate-commons-annotations	4.0.2.Final
4	org.hibernate	org.hibernate	hibernate-entitymanager	4.2.21.Final
5	org.hibernate	org.hibernate	hibernate-infinispan	4.2.21.Final

Actualizar el archivo “**module.xml**” del módulo “**org.hibernate**” por el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.hibernate">
  <resources>
    <resource-root path="hibernate-core-4.2.21.Final.jar"/>
    <resource-root path="hibernate-commons-annotations-4.0.2.Final.jar"/>
    <resource-root path="hibernate-entitymanager-4.2.21.Final.jar"/>
    <resource-root path="hibernate-infinispan-4.2.21.Final.jar"/>
  </resources>
  <dependencies>
    <module name="asm.asm"/>
    <module name="javax.api"/>
    <module name="javax.persistence.api"/>
    <module name="javax.transaction.api"/>
    <module name="javax.validation.api"/>
    <module name="org antlr"/>
    <module name="org.apache.commons.collections"/>
    <module name="org.dom4j"/>
    <module name="org.infinispan" optional="true"/>
    <module name="org.javassist"/>
    <module name="org.jboss.as.jpa.hibernate" slot="4" optional="true"/>
    <module name="org.jboss.logging"/>
    <module name="org.hibernate.envers" services="import" optional="true"/>
  </dependencies>
</module>
```

- **Módulo Hibernate Envers**

Librería	Módulo	Group ID	Artifact ID	Versión
6	org.hibernate.envers	org.hibernate	Hibernate-envers	4.2.21.Final

Actualizar el archivo “**module.xml**” del módulo “**org.hibernate.envers**” por el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.hibernate.envers">
  <resources>
    <resource-root path="hibernate-envers-4.2.21.Final.jar"/>
  </resources>
  <dependencies>
    <module name="org.hibernate"/>
    <module name="org.jboss.logging"/>
  </dependencies>
</module>
```

```
<module name="org.dom4j"/>
<module name="javax.api"/>
<module name="javax.persistence.api"/>
<module name="javax.transaction.api"/>
<module name="org.javassist"/>
</dependencies>
</module>
```

- **Módulo Java Server Faces**

Librería	Módulo	Group ID	Artifact ID	Versión
7	com.sun.jsf-impl	com.sun.faces	jsf-impl	2.1.21
8	javax.faces.api	com.sun.faces	jsf-api	2.1.21

Actualizar el archivo “**module.xml**” del módulo “**com.sun.jsf-impl**” por el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.sun.jsf-impl">
  <properties>
    <property name="jboss.api" value="private"/>
  </properties>
  <dependencies>
    <module name="javax.faces.api"/>
    <module name="javaee.api"/>
    <module name="javax.servlet.jstl.api"/>
    <module name="org.apache.xerces" services="import"/>
    <module name="org.apache.xalan" services="import"/>
  </dependencies>
  <resources>
    <resource-root path="jsf-impl-2.1.21.jar"/>
  </resources>
```

Actualizar el archivo “module.xml” del módulo “javax.faces.api” por el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="javax.faces.api">
  <dependencies>
    <module name="javax.el.api" export="true"/>
    <module name="javax.servlet.api" export="true"/>
    <module name="javax.servlet.jsp.api" export="true"/>
    <module name="javax.servlet.jstl.api" export="true"/>
    <module name="javax.validation.api" export="true"/>
  </dependencies>
  <resources>
    <resource-root path="jsf-api-2.1.21.jar"/>
  </resources>
</module>
```

## 7 Consideraciones de Seguridad

### 7.1 Autorizar el acceso a la aplicación desde otros equipos

Por defecto, el servidor de aplicaciones JBoss AS limita el acceso a las aplicaciones que contiene, permitiendo acceder a ellas únicamente desde el propio equipo donde reside (localhost). Normalmente, los usuarios acceden a las aplicaciones desde otros equipos de la red. Para permitir este tipo de acceso, se debe continuar editando el archivo **standalone.xml**. En este archivo, se debe buscar la sección **<interfaces>**, y dentro de ella modificar la configuración de la interfaz cuyo nombre es **"public"**, cambiando el valor "127.0.0.1" por **"0.0.0.0"**, debiendo resultar algo similar a lo siguiente (solo la línea que está en negrita fue modificada):

```
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address:0.0.0.0}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="${jboss.bind.address.unsecure:127.0.0.1}"/>
  </interface>
</interfaces>
```

### 7.2 Habilitar el acceso seguro (HTTPS)

La aplicación está diseñada para aceptar conexiones solo por HTTPS (HTTP + SSL/TLS). Sin embargo, por defecto el servidor de aplicaciones JBoss AS no acepta dicho tipo de conexiones. Para habilitarlo se deberá contar con un keystore (un repositorio de certificados digitales) que contendrá únicamente al certificado digital que entregará el servidor de aplicaciones a los usuarios (en realidad, a los navegadores web de los mismos) cada vez que inicien una comunicación nueva. Para esto se tienen dos casos: cuando se cuenta con un certificado digital emitido por una autoridad certificadora reconocida (CA), y cuando no se tiene dicho certificado. En ambos casos se creará un keystore nuevo, llamado **siges.jks**, para asegurar que no se utilice otro certificado no deseado.

En el caso de que se cuente con un certificado digital reconocido, éste debe ser importado al keystore con el siguiente comando:

```
keytool -v -importkeystore -srckeystore {archivo} -srcstoretype PKCS12 -destkeystore siges.jks -deststoretype JKS
```

**Nota:** reemplazar **{archivo}** por el nombre (incluyendo la ruta completa si no está en el directorio actual) del archivo que contiene el certificado digital.

Al ejecutar este comando el programa keytool solicitará la contraseña del certificado, que debe haber sido proporcionada por el emisor del mismo y la contraseña del keystore, que debe ser la misma que la del certificado.

En el caso de que no se cuente con un certificado digital reconocido, se puede crear uno propio, aunque se debe tener en cuenta que todos los usuarios verán un mensaje de advertencia, al acceder a la aplicación, indicándoles que el servidor utiliza un certificado digital que no es de confianza. Para crear un nuevo keystore conteniendo el nuevo certificado digital, se debe ejecutar el siguiente comando:

```
keytool -genkeypair -keystore siges.jks -alias siges
```

Al ejecutar este comando el programa keytool solicitará primero el ingreso de la contraseña para el keystore (habrá que ingresarla dos veces) y luego el ingreso de la información que será incluida en el certificado. Los únicos datos obligatorios son el primero ("nombre", ingresar "Agesic Siges ") y el último ("país", ingresar "UY"), en el resto de los casos se puede dejar en blanco.

Finalmente el programa keytool solicitará confirmación, a lo que se debe responder "sí" (o "yes" si está en inglés). Para terminar, el programa keytool solicitará la nueva contraseña para el certificado, que debe ser la misma que se usó para el keystore.

**Nota:** si al intentar ejecutar alguno de los comandos anteriores se observa un mensaje indicando que no se reconoce al comando keytool como un programa válido, es porque el directorio de programas de Java no está en la lista de directorios donde el sistema operativo busca programas. En este caso se puede ejecutar el mismo comando especificando la ruta completa al programa keytool (que está en el directorio bin de la instalación del JRE), o modificar la variable de entorno PATH (tanto en MS Windows como en Linux) para incluir el directorio bin de la instalación del JRE.

Una vez creado el keystore, con cualquiera de los dos casos, se debe indicarle al servidor de aplicaciones JBoss que permita las conexiones seguras, y que utilice el keystore creado anteriormente, para lo cual se debe continuar editando el archivo **standalone.xml**. En este archivo se debe buscar el subsistema llamado **"urn:jboss:domain:web:1.1"** y dentro de él remplazar la definición del conector HTTP (la línea que comienza con "<connector name='http' ...>" por el siguiente texto (quitar la línea mencionada y poner en su lugar las siguientes):

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"
  redirect-port="8443"/>
```

```
<connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https"
    secure="true">
  <ssl name="ssl" key-alias="{alias}" password="{pass}"
    certificate-key-file="{jboss.server.config.dir}/siges.jks"
    protocol="TLSv1"/>
</connector>
```

**Nota:** remplazar **{alias}** por el alias del certificado dentro del keystore (si no se conoce, puede ejecutarse el siguiente comando: `keytool -list -keystore siges.jks`) **{pass}** por la contraseña del keystore y del certificado (que deben coincidir)

## 7.3 Habilitar comprensión de datos

Dado que la aplicación puede transferir mucha información hacia los clientes, es recomendable habilitar la compresión para acelerar la transferencia. Para hacerlo, se debe editar el archivo **standalone.xml** y copiar al principio, justo debajo de las extensiones (luego del tag de cierre `</extensions>`) el siguiente texto:

```
<system-properties>
  <property name="org.apache.coyote.http11.Http11Protocol.COMPRESSION"
    value="on"/>
  <property name="org.apache.coyote.http11.Http11Protocol.COMPRESSION_MIME_TYPES"
    value="text/javascript,text/css,text/html,text/xml,text/json"/>
</system-properties>
```

# 8 Configuración de Parámetros

## 8.1 Configuración de modulo MySQL en JBOSS AS 7.1.1

Para que el servidor de aplicaciones JBoss AS pueda conectarse al servidor de bases de datos MySQL Server, es necesario crear un módulo con la configuración apropiada. Para hacerlo se debe acceder a la carpeta **modules** de JBoss AS (**{JBOSS\_HOME}/modules**) y crear la siguiente estructura de directorios: **com/mysql/main**. Luego, en la carpeta **main**, se deben agregar el archivo correspondiente al driver jdbc de mysql y el archivo de configuración **module.xml**.

Para obtener el driver jdbc de mysql debe descargarse del siguiente enlace: ["https://dev.mysql.com/downloads/connector/j/5.0.html"](https://dev.mysql.com/downloads/connector/j/5.0.html). Una vez descargado el archivo con nombre "mysql-connector-java-XXX.tar.gz" o "mysql-connector-java-XXX.zip" (XXX corresponde a la versión), se debe descomprimir mismo y copiar el archivo con nombre "mysql-connector-java-XXX.jar" al directorio **"{JBOSS\_HOME}/modules/com/mysql/main"**.

Por último, el contenido del archivo **module.xml** es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
  <resources>
    <resource-root path="mysql-connector-java-XXX.jar"/>
  </resources>
</module>
```

**Importante:**

Se uso la version del mysql-connector-java.jar que vino descargada por defecto en el proyecto (5.1.18)

Verificar que el archivo mysql-connector-java-XXX.jar al que hace referencia el archivo module.xml tenga el mismo nombre.

## 8.2 Configuración de modulo PostgreSQL en JBOSS AS 7.1.1

Para que el servidor de aplicaciones JBoss AS pueda conectarse al servidor de bases de datos PostgreSQL Server, es necesario crear un módulo con la configuración apropiada.

Para hacerlo se debe acceder a la carpeta modules de JBoss AS ({JBOSS\_HOME}/modules) y crear la siguiente estructura de directorios: **org/postgresql/main**. Luego, en la carpeta **main**, se deben agregar el archivo correspondiente al driver jdbc de postgres y el archivo de configuración "**module.xml**".

Para obtener el driver jdbc de postgres debe descargarse la versión 4.1 (es la definida para Java 7) del siguiente enlace: "<https://jdbc.postgresql.org/download.html>". Un vez descargado el archivo con nombre "**postgresql-XXX.jar**" (XXX puede variar según la versión) se debe copiar el mismo al directorio "{JBOSS\_HOME}/modules/org/postgresql/main".

Por último, el contenido del archivo **module.xml** es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
  <resources>
    <resource-root path="postgresql-XXX.jar"/>
  </resources>
</module>
```

Verificar que el archivo **postgresql-XXX.jar** al que hace referencia el archivo **module.xml** tenga el mismo nombre.

### 8.3 Configuración de driver y datasource MySQL en JBOSS AS 7.1.1

Para que la aplicación pueda conectarse a la base de datos MySQL Server a través del servidor de aplicaciones JBoss AS, es necesario definir un origen de datos en este último. Para hacerlo, se debe abrir el archivo **standalone.xml** que se encuentra en el directorio de configuración de JBoss AS (**{JBOSS\_HOME}/standalone/configuration/**) y buscar el subsistema cuyo nombre es **"urn:jboss:domain:datasources:1.0"**; dentro de este subsistema, identificar la sección **<datasources>** y dentro de ella copiar el siguiente texto:

```
<datasource jta="true" jndi-name="java:jboss/datasources/MySqlDS" pool-name="MySqlDS"
enabled="true">
  <connection-url>jdbc:mysql://{host}:{port}/siges</connection-url>
  <driver>com.mysql</driver>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
  </pool>
  <security>
    <user-name>{user}</user-name>
    <password>{pass}</password>
  </security>
  <validation>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    <background-validation>true</background-validation>
    <background-validation-millis>2000</background-validation-millis>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>

<datasource jta="false" jndi-name="java:jboss/datasources/MySqlDS2" pool-name="MySqlDS2"
enabled="true">
  <connection-url>jdbc:mysql://{host}:{port}/siges</connection-url>
  <driver>com.mysql</driver>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
  </pool>
  <security>
    <user-name>{user}</user-name>
    <password>{pass}</password>
  </security>
  <validation>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    <background-validation>true</background-validation>
    <background-validation-millis>2000</background-validation-millis>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
```

En donde:

- **{host}**: corresponde a la dirección IP o nombre del host donde se encuentra el servidor de base de datos MySQL
- **{port}**: puerto del servidor de base de datos MySQL utilizado.
- **{user}**: usuario utilizado por Jboss para conectarse a la base de datos
- **{pass}**: contraseña del usuario **{user}** para conectarse a la base de datos

Esto agregará dos datasources, uno con el atributo `jta="true"` y otro con `jta="false"`. El primero de estos datasources es utilizado normalmente por la aplicación para operaciones que requieren un manejo transaccional. El segundo datasource es utilizado para el envío de notificaciones.

Para la configuración del driver, debe agregarse dentro del mismo subsistema en la sección **<drivers>** el siguiente contenido:

```
<driver name="com.mysql" module="com.mysql">
  <xa-datasource-class>com.mysql.main.mysql-connector-java</xa-datasource-class>
</driver>
```

Notar que el driver especificado, hace referencia al módulo MySQL instalado a través del atributo `module="com.mysql"`. Luego el driver es referenciado con su nombre (`name="com.mysql"`) por el datasource en la etiqueta `<driver>com.mysql</driver>`.

## 8.4 Configuración de driver y datasource PostgreSQL en JBOSS AS

### 7.1.1

Para que la aplicación pueda conectarse a la base de datos MySQL Server a través del servidor de aplicaciones JBoss AS, es necesario definir un origen de datos en este último. Para hacerlo, se debe abrir el archivo **standalone.xml** que se encuentra en el directorio de configuración de JBoss AS (**{JBOSS\_HOME}/standalone/configuration/**) y buscar el subsistema cuyo nombre es **"urn:jboss:domain:datasources:1.0"**; dentro de este subsistema, identificar la sección **<datasources>** y dentro de ella copiar el siguiente texto:

```
<datasource jta="true" jndi-name="java:jboss/datasources/MySqlDS" pool-
name="MySqlDS" enabled="true">
  <connection-url>jdbc:postgresql://{host}:{port}/{database}?currentSchema={schema}</connection-
url>
  <driver>org.postgresql</driver>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
  </pool>
  <security>
    <user-name>{user}</user-name>
    <password>{pass}</password>
  </security>
  <validation>
    <validate-on-match>>false</validate-on-match>
    <background-validation>>false</background-validation>
  </validation>
  <statement>
```



```

    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>

<datasource jta="false" jndi-name="java:jboss/datasources/MySQLDS2" pool-
name="MySQLDS2" enabled="true">
  <connection-url>jdbc:postgresql://{host}:{port}/{database}?currentSchema={schema}</connection-
url>
  <driver>org.postgresql</driver>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
  </pool>
  <security>
    <user-name>{user}</user-name>
    <password>{pass}</password>
  </security>
  <validation>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    <background-validation>true</background-validation>
    <background-validation-millis>2000</background-validation-millis>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>

```

Esto agregará dos datasource, uno con el atributo jta="true" y otro con jta="false". El primero de estos datasources es utilizado normalmente por la aplicación para operaciones que requieren un manejo transaccional. El segundo datasources es utilizado para el envío de notificaciones. En donde:

- **{host}**: corresponde a la dirección IP o nombre del host donde se encuentra el servidor de base de datos PostgreSQL
- **{port}**: puerto del servidor de base de datos PostgreSQL utilizado.
- **{user}**: usuario utilizado por Jboss para conectarse a la base de datos
- **{pass}**: contraseña del usuario **{user}** para conectarse a la base de datos
- **{database}**: nombre de la base de datos utilizada en el servidor de base de datos
- **{schema}**: esquema de la base de datos **{database}** utilizada.

Para la configuración del driver, debe agregarse dentro del mismo subsistema en la sección **<drivers>** el siguiente contenido:

```

<driver name="org.postgresql" module="org.postgresql">
  <driver-class>org.postgresql.Driver</driver-class>
</driver>

```

Notar que el driver especificado, hace referencia al módulo PostgreSQL instalado a través del atributo *module="org.postgresql"*. Luego el driver es referenciado con su nombre (*name="org.postgresql"*) por el datasource en la etiqueta *<driver>org.postgresql</driver>*.

## 8.5 Configuración de servidor de correo electrónico

Para que la aplicación Siges pueda enviar correos electrónicos es necesario definir un servidor SMTP en el servidor de aplicaciones JBoss AS. Para hacerlo, se debe continuar editando el archivo **standalone.xml**. En este archivo, se debe buscar la sección **<socket-binding-group>** (normalmente se encuentra al final del archivo) e incluir dentro de ella el siguiente texto:

```
<outbound-socket-binding name="mail-siges-smtp">
  <remote-destination host="{host}" port="{port}" />
</outbound-socket-binding>
```

En donde **{host}** y **{port}** corresponden a la dirección IP o nombre del host, y puerto respectivamente del servidor SMTP utilizado.

A continuación se debe configurar la dirección de correo utilizada agregando dentro del subsistema "urn:jboss:domain:mail:1.0" el siguiente contenido:

```
<mail-session jndi-name="java:jboss/mail/SigesMail"from="{from}">
  <smtp-server ssl="true" outbound-socket-binding-ref="mail-siges-smtp">
    <login name="{user}" password="{pass}" />
  </smtp-server>
</mail-session>
```

En donde **{user}** y **{pass}** corresponden al usuario y contraseña respectivamente, de la dirección de correo utilizada para enviar los correos. Por otro lado, el valor de la dirección **{form}** corresponde a la dirección de correo que figura como remitente en los correos.

**Nota:** **{user}** y **{from}** pueden ser distintos.

### Actualización del módulo de mail

Debido a un error. Jboss AS 7.1.1 no soporta el protocolo TLS para el servicio SMTP. Para resolver esto, existe una actualización del módulo de correo **no oficial** para poder utilizar TLS. En caso que se desee aplicar esta solución, se debe descomprimir el archivo "**jboss-mail.zip**" facilitado por la empresa Sofis Solutions, en el directorio **{JBoss\_HOME}/modules/org/jboss/as/mail/main/**. Luego se debe actualizar el subsistema para el envío de correo "urn:jboss:domain:mail:1.0" editando el archivo **standalone.xml** cambiando el nombre del subsistema "urn:jboss:domain:mail:1.0" por "urn:jboss:domain:mail:1.1". Esto permite utilizar el atributo **tls="true"** en la etiqueta "<smtp-server>" de la siguiente forma:

```
<mail-session jndi-name="java:jboss/mail/SigesMail"from="{from}">
  <smtp-server tls="true" outbound-socket-binding-ref="mail-siges-smtp">
    <login name="{user}" password="{pass}" />
  </smtp-server>
</mail-session>
```

## 8.6 Configurar dominio de autenticación

La aplicación delega al servidor de aplicaciones JBoss la autenticación de los usuarios que intentan acceder a ella (validar nombres de usuario y contraseña). Para que el servidor de aplicaciones JBoss AS pueda encargarse de esta tarea, debe ser configurado un dominio de autenticación. Para hacerlo, se debe continuar editando el archivo **standalone.xml**. En este archivo, se debe buscar el subsistema “**urn:jboss:domain:security:1.1**” y copiar dentro de él el siguiente texto:

```
<security-domain name="form-auth" cache-type="default">
  <authentication>
    <login-module code="com.sofis.web.seguridad.SimpleUsernamePasswordLoginModule"
      flag="required"/>
  </authentication>
</security-domain>
```

## 8.7 Configurar niveles de log

Está previsto que la aplicación escriba sus propios mensajes de log en el archivo de log del servidor de aplicaciones. El nivel de detalle de los mensajes registrados es configurable. Para hacerlo, se debe continuar editando el archivo **standalone.xml**. En este archivo, se debe buscar el subsistema “**urn:jboss:domain:logging:1.1**” y copiar dentro de él el siguiente texto:

```
<logger category="siges">
  <level name="ALL"/>
</logger>
<logger category="com.sofis">
  <level name="ALL"/>
</logger>
```

Para reducir el nivel de log, puede remplazarse “ALL” por uno de “DEBUG”, “INFO”, “WARN”, “ERROR” (de mayor a menor).

## 8.8 Cambiar puertos del JBOSS AS

Por defecto, los puertos usados por JBoss AS para atender conexiones son el 8080 para HTTP y 8443 para HTTPS. Si se desea cambiar esto, se debe hacer lo siguiente:

- Para cambiar el puerto HTTP: editar el archivo **standalone.xml** y buscar la línea que comienza con “**<socket-binding name="http"**”, remplazando en ella 8080 por el puerto deseado.
- Para cambiar el puerto HTTPS: editar el archivo **standalone.xml** y buscar la línea que comienza con “**<socket-binding name="https"**”, remplazando en ella 8443 por el puerto deseado. Luego,

buscar la línea que comienza con "<connector name='http'" y remplazar en ella 8443 por el puerto deseado, que debe ser el mismo que el indicado anteriormente.

## 9 Instalacion y configuracion de Siges

### 9.1 Instalar aplicación en JBOSS AS 7.1.1

Para instalar la aplicación en el servidor de aplicaciones JBoss AS, solo se debe copiar el archivo SS-SIGES-ear-x.x.x.ear proporcionado por Sofis Solutions a la carpeta {JBOSS\_HOME}/standalone/deployments. Tras unos pocos segundos en dicha carpeta aparecerá un nuevo archivo llamado SS-SIGES-ear-x.x.x.ear.deployed, que indica que la aplicación se ha instalado correctamente; si no es así, o el nombre del archivo es SS-SIGES-ear-x.x.x.ear.failed, significa que ocurrió un error y debe examinarse el archivo de log ({JBOSS\_HOME}/standalone/log/server.log) para intentar determinar la causa del error.

Si para instalar aplicacion se presenta el siguiente caso:

ERROR [org.jboss.msc.service.fail] Failed to start service jboss.deployment.subunit

Caused by: org.jboss.msc.service.ServiceNotFoundException: Service service jboss.ejb.default-resource-adapter-name-service not found

Agregar el siguiente contenido al standalone.xml en el subsistema xmlns="urn:jboss:domain:ejb3:1.2"

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
  <mdb>
    <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
  </mdb>
```

### 9.2 Super administrador

Si ya ejecuto en base de datos el script brindado por sofis de nombre "siges\_clean\_x.x.x-MySQL.sql" el siguiente paso no es necesario, pero es bueno saberlo.

Los usuarios del tipo Super Administrador son creados directamente en la base de datos utilizando las siguientes sentencias SQL:

```
insert ss_usuario ( usu_administrador, usu_cambio_estado_desc, usu_cod, usu_correo_electronico,
usu_cuenta_bloqueada, usu_descripcion, usu_direccion, usu_fecha_password, usu_fecha_uuid,
usu_foto, usu_intentos_fallidos, usu_nro_doc, usu_oficina_por_defecto, usu_origen, usu_password,
usu_primer_apellido, usu_primer_nombre, usu_registrado, usu_segundo_apellido,
```

```
usu_segundo_nombre, usu_telefono, usu_celular, usu_user_code, usu_uuid_des, usu_version,  
usu_vigente, usu_aprob_facturas, usu_ult_usuario, usu_ult_mod, usu_ult_origen, usu_area_org, usu_token,  
usu_token_act) values  
('${admin_pk}', NULL, '${admin_cod}', '${admin_mail}', '0', NULL, NULL, '2014-11-11 00:00:00', NULL,  
NULL, NULL, NULL, NULL, NULL, '${admin_pass}', 'Admin', 'Usuario', NULL, NULL, NULL, NULL, NULL,  
'0', NULL, '1', '1', NULL, 'admin', now(), NULL, NULL, NULL, NULL);
```

```
insert ss_usu_ofi_roles(usu_ofi_roles_origen, usu_ofi_roles_user_code, usu_ofi_roles_oficina,  
usu_ofi_roles_rol, usu_ofi_roles_usuario, usu_ofi_roles_usu_area, usu_ofi_roles_activo, version)  
values  
('instalacion', '0', NULL, '0', '${admin_pk}', NULL, '1', '0');
```

**En donde:**

- **\${admin\_pk}**: clave primaria del usuario super administrador
- **\${admin\_cod}**: código del usuario super administrador
- **\${admin\_mail}**: correo del usuario super administrador
- **\${admin\_pass}**: md5 de la contraseña del usuario super administrador

### 9.3 Usuarios por defecto en el ambiente levantado

A continuación se detallan el conjunto de usuarios por defecto que vienen con el sistema, estos usuarios podrán ser usados la primera vez que se inicie/monte el sistema.

Los usuarios creados en este dump son: (rol: usuario / contraseña)

- Administrador: admin / admin
- Usuario: usuario@siges.com / siges
- PMOF: usuario.pmot@siges.com / siges
- PMOT: usuario.pmot@siges.com / siges

### 9.4 Configuración de la aplicación

La aplicación tiene configuraciones que aplican a nivel de cada organismo y otras que aplican para todos los organismos. En el caso de las configuraciones a nivel de organismo, éstas pueden ser modificadas a través de la aplicación por un usuario con rol PMOT en la pantalla “Administración → Configuración → General”. Por otro lado, las configuraciones globales sólo pueden ser modificadas directamente desde la base de datos.

Los parámetros de configuración a nivel de organismo son:

- **ADJUNTO\_MODIFICA\_PRESUPUESTO:** Adjunto puede modificar presupuesto
- **ALCANCE\_INDICE\_Límite\_AMARILLO:** Límite semáforo amarillo para Alcance
- **ALCANCE\_INDICE\_Límite\_ROJO:** Límite semáforo rojo para Alcance
- **CALIDAD\_GERENTE\_MODIFICA:** Permitir al Gerente modificar ítems calidad
- **CALIDAD\_Límite\_AMARILLO:** semáforo Límite amarillo calidad
- **CALIDAD\_Límite\_ROJO:** semáforo Límite rojo calidad
- **CON\_CORREO:** true/false: habilitar envío de correos. En caso de estar activado, se debe configurar el servicio de mail definido en la sección Error: no se encontró el origen de la referencia.
- **COSTO\_ACTUAL\_Límite\_AMARILLO:** Límite amarillo del semáforo de costo
- **COSTO\_ACTUAL\_Límite\_ROJO:** Límite rojo del semáforo de costo
- **DOCUMENTO\_PORCENTAJE\_Límite\_AMARILLO:** Límite amarillo del semáforo de documentos
- **DOCUMENTO\_PORCENTAJE\_Límite\_ROJO:** Límite rojo del semáforo de documentos
- **ENVIO\_NOTIFICACIONES:** Determina si se envían notificaciones
- **ESTADO\_INICIO\_Límite\_AMARILLO:** semáforo estado Inicio amarillo
- **ESTADO\_INICIO\_Límite\_ROJO:** semáforo estado Inicio rojo
- **ESTADO\_Planificación\_Límite\_AMARILLO:** semáforo estado Planificación amarillo
- **ESTADO\_Planificación\_Límite\_ROJO:** semáforo estado Planificación rojo
- **FILTRO\_INICIO\_POR\_AREAS:** true/false: Agrupar resultado inicio por áreas
- **FOLDER\_MEDIA:** Carpeta donde se almacenan los archivos Multimedia. Verificar que el directorio exista y el usuario con el que se ejecuta el servidor de aplicaciones tenga permisos de lectura y escritura sobre este directorio.
- **KEYSTORE\_ORG\_ALIAS:** Alias del certificado del organismo en el keystore configurado para consumir servicios de la PGE. Verificar que el certificado con el alias definido exista en el keystore configurado.
- **KEYSTORE\_ORG\_PASS:** Contraseña del certificado del organismo en el keystore configurado para consumir servicios de la PGE.
- **KEYSTORE\_ORG\_PATH:** Keystore que contiene el certificado del organismo utilizado para consumir servicio de la PGE. Verificar que el path exista y el usuario con el que se ejecuta el servidor de aplicaciones tiene permisos de lectura sobre este archivo.
- **KEYSTORE\_SSL\_ALIAS:** Alias del certificado utilizado para la conexión SSL de la PGE

- **KEYSTORE\_SSL\_PASS:** Contraseña del certificado en keystore utilizado para la conexión SSL de la PGE
- **KEYSTORE\_SSL\_PATH:** Keystore utilizado para la conexión SSL de la PGE. Verificar que el path exista y el usuario con el que se ejecuta el servidor de aplicaciones tiene permisos de lectura sobre este archivo.
- **MAIL\_DEBUG:** true/false: habilitar debug del envío de mail
- **MAIL\_ENCODING:** Encoding de los mails a enviar
- **MAIL\_FROM:** Dirección desde donde se envían los mails
- **MAIL\_TLS:** true/false: Habilitar configuración TLS en envío de mail
- **PRODUCTO\_INDICE\_Límite\_AMARILLO:** Límite semáforo amarillo para índice de Productos
- **PRODUCTO\_INDICE\_Límite\_ROJO:** Límite semáforo rojo para índice de Productos
- **PROYECTO\_GANTT\_PERIODO:** true/false: habilitar configuración de período de proyecto en entregables
- **RIESGO\_INDICE\_Límite\_AMARILLO:** Límite amarillo del índice de los de riesgos
- **RIESGO\_INDICE\_Límite\_ROJO:** Límite rojo del índice de los de riesgos
- **RIESGO\_TIEMPO\_Límite\_AMARILLO:** Límite amarillo del tiempo de los riesgos
- **RIESGO\_TIEMPO\_Límite\_ROJO:** Límite rojo del tiempo de los de riesgos
- **STS\_EMITOR:** Emisor del Security Token Service de la PGE (Agesic)
- **STS\_POLITICA:** Política del Security Token Service de la PGE ("urn:std:15")
- **STS\_URL:** URL del Security Token Service de la PGE
- **TAMANIO\_MAX\_ARCHIVO\_DOCUMENTO:** tamaño máximo de archivos de documentos soportado
- **TAMANIO\_MAX\_ARCHIVO\_MULTIMEDIA:** Tamaño máximo de los archivos multimedia
- **TAMANIO\_MAX\_LOGO\_ORGANISMO:** Tamaño máximo en bytes del logo del Organismo
- **TRUSTSTORE\_SSL\_PASS:** Contraseña del trustore utilizado para consumir servicios de la PGE via SSL
- **TRUSTSTORE\_SSL\_PATH:** Trustore utilizado para consumir servicios de la PGE via SSL. Verificar que el path exista y el usuario con el que se ejecuta el servidor de aplicaciones tiene permisos de lectura sobre este archivo.
- **VISUALIZADOR\_EXPORTACION\_POR\_PGE:** true/false: realizar la exportación al Visualizador por la PGE

- **VISUALIZADOR.PUBLICARSERVICIO\_MTOM:** true/false: habilitar consumo del servicio del Visualizador a con MTOM
- **VISUALIZADOR.PUBLICARSERVICIO\_NAMESPACE:** namespace utilizado para consumir servicios del visualizador a través de la PGE.

Las configuraciones globales sólo pueden ser actualizadas con el siguiente comando SQL:

```
update ss_configuraciones set cnf_valor = '{valor}' where cnf_codigo = {código} and cnf_org_pk is null;
```

En donde **{código}** y **{valor}** corresponden al código y valor de la configuración que se desea modificar.

Actualmente la única configuración global es:

- **DOCUMENTOS\_DIR:** Directorio donde se almacenarán los archivos de los documentos. Verificar que el directorio exista y el usuario con el que se ejecuta el servidor de aplicaciones tiene permisos de lectura sobre este archivo.

## 10 Respaldo y mantenimiento

En caso de que se desee hacer un respaldo de forma periódica de un ambiente de producción, se recomienda respaldar los siguientes datos:

- Directorio **{JBOSS\_HOME}/bin**
- Directorio **{JBOSS\_HOME}/modules/standalone/configuration/**
- Directorio **{JBOSS\_HOME}/modules/standalone/deployments/**
- Directorio **{JBOSS\_HOME}/modules/standalone/log**
- Directorio configurado para almacenar los archivos de los documentos (parámetro de configuración global **"DOCUMENTOS\_DIR"**)
- Keystores configurados para los organismos en el parámetro **"KEYSTORE\_ORG\_PATH"**.
- Keystores configurados para los organismos en el parámetro **"KEYSTORE\_SSL\_PATH"**.
- Truststore configurados para los organismos en el parámetro **"TRUSTSTORE\_SSL\_PATH"**.
- Directorios donde los organismos almacenan sus contenidos multimedia. Se especifica en el parámetro **"FOLDER\_MEDIA"** a nivel organismo.
- Dump de la base de datos:

Para el caso de Mysql ejecutar el siguiente script : `$ mysqldump -u user -p exampledb > respaldo_exampledb.sql`



Para el caso de Postgresql : `pg_dump --file=$ARCHIVO.dmp -Fc -v -b ${BASE}`

Para tareas programadas, se sugiere el reinicio de forma periódica del servidor de aplicaciones eliminando el contenido de los directorios `"{JBOSS_HOME}/modules/standalone/data/"` y `"{JBOSS_HOME}/modules/standalone/tmp"`.

## 11 Optimización de ambiente

### 11.1 Optimización de Java

La optimización de Java se realiza agregando parámetros a la variable de entorno **JAVA\_OPTS** definida en el archivo de configuración `{JBOSS_HOME}/bin/standalone.conf` en Linux o en `{JBOSS_HOME}/bin/standalone.conf.bat` para el caso de windows.

Los parámetros de **JAVA\_OPTS** que se definen son en la sección de archivo **standalone.conf**:

```
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -
Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000"
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=JBOSS_MODULES_SYSTEM_PKGS -
Djava.awt.headless=true"
    JAVA_OPTS="$JAVA_OPTS -Djboss.server.default.config=standalone.xml"
else
    echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
fi
```

Los parámetros de java que se utilizan comúnmente para servidores son:

- **Memoria y performance:**
  - `-server` : activa características de la JVM para servidores.
  - `-d64` : para sistemas de 64bits permite utilizar más de 4GB como tamaño del heap. En algunos sistemas este parámetro es innecesario, pero se recomienda definirlo explícitamente.
  - `-Xms` : cantidad de memoria reservada para el heap al arranque del procesos. Se recomienda establecer este parámetro en  $\frac{3}{4}$  de la memoria física disponible del servidor. Por ejemplo, para un servidor de 12GB de RAM, el parámetro sería `"-Xms9G"`

- -Xmx: cantidad máxima de memoria reservada para el heap. Se recomienda establecer este parámetro en  $\frac{3}{4}$  de la memoria física disponible del servidor. Por ejemplo, para un servidor de 12GB de RAM, el parámetro sería "-Xms9G".
- -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled: activa algoritmo concurrente para garbage collector de la JVM
- -XX:+UseCompressedOops : activa la compresión de direcciones.
- **Activar log del Garbage Collector:**
  - -XX:+PrintGCDateStamps -verbose:gc -XX:+PrintGCDetails -Xloggc:{gc} :
    - activa el log del Garbage Collector y lo guarda en el archivo con path {gc}
  - -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M
- **Dump de memoria si ocurre un error "OutOfMemoryError":**
  - -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath={hprof}`date`.hprof
    - {hprof}: corresponde al path donde se generará el dump de memoria

Para un servidor con 12 GB de RAM, 8 cores y de 64-bits, si se definen todos estos valores, la sección del archivo **standalone.conf** debería ser:

```
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-server -d64 -Xms9G -Xmx9G -XX:+UseConcMarkSweepGC -XX:
+CMSParallelRemarkEnabled -XX:+UseCompressedOops -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -XX:+PrintGCDateStamps -verbose:gc -XX:+PrintGCDetails -
Xloggc:{gc} -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath={hprof}`date`.hprof"
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBoss_MODULES_SYSTEM_PKGS -
Djava.awt.headless=true"
    JAVA_OPTS="$JAVA_OPTS -Djboss.server.default.config=standalone.xml"
else
    echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
```

## 11.2 Optimización de MySQL

Para optimizar el rendimiento de MySQL, se recomienda activar el cache estableciendo los siguientes valores en función de la cantidad de RAM y cantidad de cores disponibles. Los parámetros a definir son:

- **query\_cache\_type = 1** . Activa el cache.
- **max\_allowed\_packet = 16777216**
- **query\_cache\_size = 64m** por cada 1GB de RAM. Por ejemplo, para 4GB de RAM el valor es 256m
- **key\_buffer\_size = 32m** por cada 1GB de RAM. Por ejemplo, para 4GB de RAM el valor es 128m
- **sort\_buffer\_size = 1m** por cada 1GB de RAM. Por ejemplo, para 4GB de RAM el valor es 4m

- **read\_buffer\_size** = 1m por cada 1GB de RAM. Por ejemplo, para 4GB de RAM el valor es 4m
- **read\_rnd\_buffer\_size** = 1m por cada 1GB de RAM. Por ejemplo, para 4GB de RAM el valor es 4m
- **join\_buffer\_size** = 1m por cada 1GB de RAM. Por ejemplo, para 4GB de RAM el valor es 4m
- **thread\_cache\_size** = 64
- **max\_connections** = Cantidad máxima de conexiones. Por ejemplo, si Siges utiliza 2 datasources con configurados en el Jboss que tienen definido un max-pool-size de 150 cada una, la cantidad máxima de conexiones debe ser como mínimo 301 (se reserva 1 conexión para un usuario root)
- **query\_cache\_limit** = 1m
- **innodb\_buffer\_pool\_size** = Un 70% de la memoria RAM disponible. Para un servidor de 4GB de RAM, el valor al 70% es "2867m".

**Nota:** tomar en cuenta que estas configuraciones corresponden a un servidor MySQL dedicado. Es decir, únicamente tiene instalado MySQL server.

## 11.3 Optimización de JBOSS

A nivel de jboss las principales recomendaciones para mejorar la performance de la aplicación consisten en:

- Incrementar el pool de EJBs cambiando el valor "max-pool-size" (en este caso de 20 a 100) en el archivo **{JBOSS\_HOME}/standalone/configuration/standalone.xml** en el subsistema **"urn:jboss:domain:ejb3:1.2"** de la siguiente forma:

```
<pools>
  <bean-instance-pools>
    <strict-max-pool name="slsb-strict-max-pool" max-pool-size="100" instance-acquisition-
timeout="5" instance-acquisition-timeout-unit="MINUTES"/>
    <strict-max-pool name="mdb-strict-max-pool" max-pool-size="20" instance-acquisition-
timeout="5" instance-acquisition-timeout-unit="MINUTES"/>
  </bean-instance-pools>
</pools>
```

- Incrementar el timeout de transacciones (por ejemplo, a 3600 segundos) para prevenir errores en el procesamiento cuando la base de datos demora demasiado en una consulta. Para esto se define en el **{JBOSS\_HOME}/standalone/configuration/standalone.xml** el subsistema **"urn:jboss:domain:transactions:1.3"** :

```
<subsystem xmlns="urn:jboss:domain:transactions:1.3">
  <core-environment>
    <process-id>
      <uuid/>
```

```

        </process-id>
    </core-environment>
    <recovery-environment socket-binding="txn-recovery-environment" status-socket-binding="txn-
status-manager"/>
    <coordinator-environment default-timeout="3600"/>
</subsystem>

```

- Incrementar pool mínimo (min-pool-size) y máximo (max-pool-size) de conexiones a la base de datos, por 50 y 200 respectivamente. Para esto, se debe cambiar en los datasources definidos en el archivo **{JBOSS\_HOME}/standalone/configuration/standalone.xml**:

```

<datasource jta="true" jndi-name="java:jboss/datasources/MySqlDS"                                pool-
name="MySqlDS" enabled="true">
  <connection-url>jdbc:mysql://{host}:{port}/{database}</connection-url>
  <driver>com.mysql</driver>
  <pool>
    <min-pool-size>50</min-pool-size>
    <max-pool-size>200</max-pool-size>
  </pool>
  <security>
    <user-name>{user}</user-name>
    <password>{pass}</password>
  </security>
  <validation>
    <validate-on-match>false</validate-on-match>
    <background-validation>false</background-validation>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>

```

```

<datasource jta="false" jndi-name="java:jboss/datasources/MySqlDS2"                            pool-
name="MySqlDS2" enabled="true">
  <connection-url>jdbc:mysql://{host}:{port}/{database}</connection-url>
  <driver>com.mysql</driver>
  <pool>
    <min-pool-size>50</min-pool-size>
    <max-pool-size>200</max-pool-size>
  </pool>
  <security>
    <user-name>{user}</user-name>
    <password>{pass}</password>
  </security>
  <validation>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    <background-validation>true</background-validation>
    <background-validation-millis>2000</background-validation-millis>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>

```

**Nota:** tomar en cuenta que al incrementar el pool de conexiones a la base de datos, debe configurarse el parámetro “max\_connections” de MySQL para permitir la cantidad de conexiones definidas en el pool. Por ejemplo, si se definen 2 datasources con 200 conexiones como máximo, el valor de “max\_connections” debe ser 401 (reservando 1 conexión para el usuario root).

- Incrementar pool de conexiones http y cantidad de threads de procesamiento editando en el archivo **{JBoss\_HOME}/standalone/configuration/standalone.xml** los subsistemas **urn:jboss:domain:threads:1.1** y **urn:jboss:domain:web:1.1**. Por ejemplo, para soportar 200 usuarios concurrentes la configuración sería.

```
<subsystem xmlns="urn:jboss:domain:threads:1.1">
  <bounded-queue-thread-pool name="http-executor" blocking="true">
    <core-threads count="200" />
    <queue-length count="1000" />
    <max-threads count="300" />
    <keepalive-time time="5" unit="seconds"/>
  </bounded-queue-thread-pool>
```

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="true">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"
    executor="http-executor" enable-lookups="false"
    max-connections="300" max-post-size="2048" max-save-post-size="4096">
```

## 12 Entorno de desarrollo

Esta sección describe el software necesario para la instalación y configuración para crear el ambiente de desarrollo de Siges.

### Requisitos

- Java JDK 1.7
- Netbeans versión >= 8.0
- Base de Datos (una de las dos):
  - MySQL Server 5.6 . Es opcional el uso de MySQL Workbench 6.1.
  - PostgreSQL Server 9.4. Es opcional el uso de PgAdmin.
- Jboss AS 7.1.1
- **Maven versión >= 3.x**

## 13 Historial de cambios al documento

Versión	Fecha	Autor	Detalle
1.0		SS	Versión original
1.1		BS	<ul style="list-style-type: none"> <li>- Actualización del formato del documento.</li> <li>- Reestructura del documento en general</li> <li>- Incorporación de la actualización de módulos de Jboss 7.1.1 debido a bugs del servidor de aplicaciones.</li> <li>- Sección de optimización con recomendaciones</li> <li>- La sección "Backup" se cambia por "Respaldo y mantenimiento" para incluir información recomendada para mantenimiento del servidor de aplicaciones.</li> </ul>
2.0		BS	<ul style="list-style-type: none"> <li>- Corrección del driver postgres</li> <li>- Agregado cómo ingresar usuario Super Administrador</li> <li>- Agregado tuning de conexiones del Jboss.</li> </ul>
2.1	20-06-2017	BS	<ul style="list-style-type: none"> <li>- Corrección "thread_cache_size: 64m" por "thread_cache_size: 64"</li> </ul>
3.0	06-07-2023	SS	<ul style="list-style-type: none"> <li>- Configuración de driver mysql</li> <li>- Detener servidor Jboss</li> <li>- Solución Exception FactoryConfigurationError</li> <li>- Dump base de datos</li> <li>- Nota de IMPORTANTE (Compatibilidad)</li> </ul>