

SIGES

Requerimientos Técnicos y Manual de
Instalación

Índice

Introducción	4
1 Requerimientos Técnicos.....	5
1.1 Requerimientos de Hardware	5
1.2 Red	5
1.3 Requerimientos de Software	6
2 Entorno de Desarrollo	7
2.1 Introducción	7
2.2 Propósito	7
2.3 Audiencia.....	7
2.4 Requisitos para la edición del código fuente	7
2.5 Instalación de Java	7
2.6 Instalación de Netbeans.....	8
2.7 Instalación de JBoss AS.....	8
2.8 Cargar el código fuente	8
2.9 Generar la Base de Datos	9
2.10 Configuración de Parámetros.....	9
3 Procedimiento de instalación	10
3.1 Sobre la aplicación	10
3.2 Instalación de Java	10
3.3 Instalación de JBoss AS.....	10
3.4 Instalación del Motor de Base de Datos	11
3.5 Crear la base de datos para la aplicación.....	11
3.6 Crear un módulo para el conector de la Base de Datos en JBoss AS	12
3.7 Configurar el origen de datos para la aplicación en JBoss AS para la base de datos....	12
3.8 Configurar el servidor de correo electrónico	15
3.9 Configurar el dominio de autenticación.....	16
3.10 Configurar los niveles de log	16
3.11 Autorizar el acceso a la aplicación desde otros equipos.....	16
3.12 Habilitar el acceso seguro (HTTPS).....	17
3.13 Habilitar la compresión de datos	18
3.14 Iniciar el JBoss AS	18
3.15 Instalar la aplicación en el JBoss AS	18
3.16 Cambio de puertos del JBoss AS	19
3.17 Creación de Carpeta para almacenamiento de imágenes	19
3.18 Acceder a la aplicación y configuración de parámetros.....	19
3.19 Backups	21

Introducción

Este documento está orientado a técnicos en TI y posee 3 capítulos:

1. Requerimientos Técnicos: Detalla el entorno (hardware y software) necesarios para poder utilizar Siges.
2. Entorno de Desarrollo: Orientado a desarrolladores, detalla el software utilizado para desarrollar esta aplicación. Se trata de una guía para poder editar el código fuente y modificarle o agregarle funcionalidades al sistema.
3. Procedimiento de Instalación: Guía detallada con todos los pasos necesarios para poder instalar el sistema y utilizarlo. Al final de este capítulo se indican los temas a tener en cuenta para la configuración de backups.

Nota: El sistema es compatible con MySQL y PostgreSQL, en cada capítulo, en caso que sea necesario, se distinguen las diferencias en configuración e instalación dependiendo del motor de base de datos elegido.

1 Requerimientos Técnicos

El objetivo de este capítulo es detallar las necesidades adecuadas del sistema en hardware y software para desplegar la aplicación en un ambiente de producción o testing. Está dirigido al todo el personal técnico y administradores de sistemas. Se detallarán los requerimientos y especificaciones de hardware sin profundizar en detalles, los cuales quedan a cargo de cada oficina o departamento de informática.

1.1 *Requerimientos de Hardware*

Ambiente de producción con un único servidor:

- Requisitos recomendados: Memoria RAM 12 GB, espacio en disco 100 GB
- Requisitos mínimos: Memoria RAM 6 GB, espacio en disco 20 GB

Ambiente de producción para servidores separados:

Servidor de Aplicaciones:

- Requisitos recomendados: Memoria RAM 8 GB, espacio en disco 40 GB
- Requisitos mínimos: Memoria RAM 4 GB, espacio en disco 20 GB

Servidor de Base de Datos:

- Requisitos recomendados: Memoria RAM 4 GB, espacio en disco 60 GB
- Requisitos mínimos: Memoria RAM 2 GB, espacio en disco 20 GB

Se recomienda una conexión de red de 1000 Mbit/s.

Ambiente de testing:

Para el servidor de testing se recomienda un único servidor (Srv. Aplicaciones y Base de Datos) con:

- Memoria RAM: 4 GB.
- Espacio en disco: 20 GB.
- Conexión red: no menor a 100Mbit/s.

En todos los casos es indiferente si estos requerimientos se aplican en un equipo físico o en una máquina virtual.

1.2 *Red*

Acceso Usuarios

El sistema de producción debe quedar disponible y al alcance de los usuarios que lo utilicen. Es decisión de cada organismo y personal técnico si la aplicación tiene un alcance de uso interno del organismo y/o externo, teniendo para este último, acceso desde Internet.

Las políticas de seguridad para el uso interno como externo no son detalladas en este documento y dependen de cada organismo.

Acceso personal técnico

El personal técnico (Administradores, Desarrolladores, etc.) deberá tener acceso para las siguientes tareas:

- Administración general y configuración del Sistema Operativo y aplicaciones.
- Administración de la Base de Datos.
- Administración del Servidor de Aplicaciones.

1.3 *Requerimientos de Software*

A continuación se detalla el software a ejecutar en dicho sistema.

- Sistema Operativo: preferentemente Linux (CentOS, Debian, etc.)
- Servidor de Aplicaciones: jBoss 7.1
- Servidor de Base de Datos: MySQL 5.6 o PostgreSQL 9.4
- Java: versión 1.7

2 Entorno de Desarrollo

2.1 Introducción

El código de la aplicación “Siges – Seguimiento y control de portafolios” ha sido diseñado para ser modificado, ampliado o corregido de forma accesible para cualquier persona que tenga conocimientos de programación. Está implementada con el lenguaje de programación Java y desarrollada para ser desplegada en un servidor de aplicaciones web.

2.2 Propósito

La intención de este capítulo es describir los elementos y procedimientos necesarios para modificar el código de la aplicación “Siges – Seguimiento y control de portafolios”.

2.3 Audiencia

Este capítulo está dirigido al personal técnico que realizará el mantenimiento del software.

2.4 Requisitos para la edición del código fuente

Software utilizado para el desarrollo y funcionamiento de la aplicación:

- Java JDK 1.7
- Netbeans 8.0
- Base de Datos (una de las dos):
 - MySQL 5.6. Es opcional el uso de MySQL Workbench 6.1.
 - PostgreSQL 9.4. Es opcional el uso de pgAdmin.
- JBoss-AS 7.1.1

2.5 Instalación de Java

Para compilar el código o ejecutar 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 JDK):

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Una vez descargado el instalador apropiado se 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\jdk1.7.0_25”, entonces el valor de la variable de entorno JAVA_HOME debe ser también “C:\Program Files\Java\jdk1.7.0_25”.

2.6 Instalación de Netbeans

NetBeans es un entorno de desarrollo (IDE), utilizado principalmente para el lenguaje de programación Java. Puede descargarse gratuitamente desde la siguiente URL (Asegurarse de que sea la versión Java EE):

<https://netbeans.org/downloads/>

Luego de descargarlo, ejecutar el archivo instalador siguiendo las instrucciones proporcionadas por el mismo.

2.7 Instalación de JBoss AS

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):

<http://www.jboss.org/jbossas/downloads/>

Luego de descargarlo, la instalación consiste simplemente en descomprimir el archivo descargado (un archivo comprimido de tipo ZIP) en la ubicación deseada (que será la ubicación final); dicha 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 el punto anterior, y la instalación de Java 7.

2.8 Cargar el código fuente

Una vez instalado el software antes detallado, se debe proceder a descomprimir el archivo conteniendo el código fuente. Netbeans generalmente genera una carpeta en “C:\NetBeansProjects” donde almacenará el código de los proyectos con los que se trabaje.

Dentro de la carpeta de proyectos de Netbeans genere una nueva carpeta y descomprima el archivo que contiene el código fuente denominado “SS-SIGES-Codigo_Fuente-v<<Número_de_versión>>.zip”.

A continuación, genere un nuevo grupo de proyectos en Netbeans. Para esto diríjase a File→Project Groups... . Se desplegará una nueva ventana “Manage Groups” en donde seleccionará el botón “New Group...”. Escriba el nombre del grupo y seleccione la opción “Folder of Projects”. Luego con el botón “Browse...” seleccione la ubicación del código fuente. Por ejemplo: “C:\NetBeansProjects\SS-SIGES”. Por último presione el botón “Create Group”. De esta forma ya tiene el código a disposición para poder trabajar

En función del motor de base de datos elegido se deberá modificar el archivo persistence.xml que se encuentra dentro de la carpeta “META-INF”,

Para MySql se deberá descomentar la siguiente línea:

```
<property name="hibernate.dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
```


Para PostgreSQL se deberá descomentar la siguiente línea:

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.PostgreSQLDialect"/>
```

2.9 Generar la Base de Datos

Para MySQL:

Para crear la base de datos para la aplicación se debe iniciar sesión en el servidor MySQL usando la cuenta root y ejecutar el script "<ruta_del_proyecto>\SS-SIGES\SS-SIGES-data\src\main\SQLScripts\Siges_X.Y-MySQL.sql" proporcionado en el código fuente. 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.

Para PostgreSQL:

Para crear la base de datos para la aplicación se debe iniciar sesión en el servidor PostgreSQL usando la cuenta root y ejecutar el script "<ruta_del_proyecto>\SS-SIGES\SS-SIGES-data\src\main\SQLScripts\Siges_X.Y-PostgreSQL.backup" proporcionado en el código fuente. Puede usarse el programa psql incluido en la instalación de PostgreSQL, o el cliente gráfico pgAdmin que puede descargarse del mismo sitio donde se descargó el instalador.

2.10 Configuración de Parámetros

Para profundizar más en la configuración de parámetros (Java, JBoss, MySQL, PostgreSQL, Base de Datos, Mail, etc.), consultar el siguiente capítulo de este documento.

3 Procedimiento de instalación

En este capítulo se detallan los pasos necesarios para realizar la instalación del sistema. En muchos de las secciones de este capítulo se hace referencia a variables del sistema a configurar. Para configurar estas variables es necesario loguarse al sistema con un usuario con rol de PMO Transversal. Es por esto, que se debe seguir esta guía paso a paso, pero las variables de cada paso se deben configurar una vez esté instalado el sistema. En el punto 3.2 explica cómo ingresar al sistema y configurar estas variables.

3.1 Sobre la aplicación

La aplicación se ejecuta sobre un servidor de aplicaciones **JBoss AS versión 7.1.1** y almacena sus datos en una base de datos **MySQL Server 5.6 o PostgreSQL 9.4**. Es necesario instalar dichos productos antes de proceder a instalar la aplicación en sí misma, como se explica en este documento. Adicionalmente, es necesario contar con un entorno de ejecución **Java (JRE) versión 7** instalado en el sistema.

A continuación se explica la instalación de cada uno de estos componentes.

3.2 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 se 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”.

3.3 Instalación de JBoss AS

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):

<http://www.jboss.org/jbossas/downloads/>

Luego de descargarlo, la instalación consiste simplemente en descomprimir el archivo descargado (un archivo comprimido de tipo ZIP) en la ubicación deseada (que será la ubicación final); dicha 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 el punto anterior, y la instalación de Java 7.

Configurar los valores de Xmx y Xms, para esto editar el archivo standalone.conf en linux o standalone.conf.bat en Windows que se encuentra en la carpeta **{JBoss_HOME}/bin** y modificar la siguiente línea:

```
JAVA_OPTS="-Xms10240m -Xmx10240m -XX:MaxPermSize=512m -  
Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -  
Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
```

por los valores correctos. Los valores correctos dependen de la memoria RAM y del CPU del servidor donde se está realizando la instalación. Se recomienda como mínimo 2024m. En el ejemplo anterior está seteado en 10 GB (para un equipo con 12 GB de memoria RAM).

3.4 Instalación del Motor de Base de Datos

El sistema es compatible con MySQL o PostgreSQL, a continuación se explica cómo instalar estos motores 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.

3.5 Crear la base de datos para 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 **sigesXXXX-**

MySQL.sql (para MySQL) o **Siges_XXXX-PostgreSQL.backup** (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:

```
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”.

3.6 Crear un módulo para el conector de la Base de Datos en JBoss AS

Con MySQL:

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 copiar los archivos **mysql-connector-java.jar** y **module.xml** proporcionados.

Con PostgreSQL:

Para que el servidor de aplicaciones JBoss AS pueda conectarse al servidor de bases de datos PostgreSQL, 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 copiar los archivos **postgresql-9.3-1103.jdbc3.jar** y **module.xml** proporcionados.

3.7 Configurar el origen de datos para la aplicación en JBoss AS para la base de datos

Para que la aplicación pueda conectarse a la base de datos (MySQL Server o PostgreSQL) 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>**. En el archivo **standalone.xml** proporcionado están los dos bloques necesarios, uno para MySQL y otro para PostgreSQL (cada bloque tiene 2 datasources). Ahí mismo se indica qué bloque deberá eliminarse según el motor de base de datos elegido. De todas formas, a continuación se indica cuál es el bloque válido para cada motor de base de datos.

Para MySQL:

```
<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>
    <background-validation>true</background-validation>
    <background-validation-millis>2000</background-validation-millis>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
```

```
<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>
    <background-validation>true</background-validation>
    <background-validation-millis>2000</background-validation-millis>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
  </validation>
  <statement>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
```

Para PostgreSQL:

```
<datasource jta="true" jndi-name="java:jboss/datasources/MySqlDS" pool-
name="MySqlDS" enabled="true" use-java-context="true" use-ccm="false">
  <connection-
url>jdbc:postgresql://{host}:{port}/siges?currentSchema=siges</connection-url>
  <driver>org.postgresql</driver>
  <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" use-java-context="true" use-ccm="false">
    <connection-
url>jdbc:postgresql://{host}:{port}/siges?currentSchema=siges</connection-url>
    <driver>org.postgresql</driver>
    <security>
        <user-name>{user}</user-name>
        <password>{pass}</password>
    </security>
    <validation>
        <background-validation>true</background-validation>
        <background-validation-millis>2000</background-validation-millis>
        <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    </validation>
    <statement>
        <share-prepared-statements>true</share-prepared-statements>
    </statement>
</datasource>

```

Nota: En ambos bloques remplazar “{host}” y “{port}” por la dirección IP y número de puerto donde escucha el servidor de bases de datos (MySQL o PostgreSQL), si el servidor de bases de datos reside en el mismo equipo que el servidor de aplicaciones JBoss AS puede usarse “localhost”; el puerto por defecto, a menos que se haya especificado otra cosa durante la instalación es “3306” para MySQL o “5432” para PostgreSQL). También remplazar “{user}” y “{pass}” por el nombre de usuario y la contraseña especificadas al crear la cuenta de usuario en el punto 2.5.

A continuación, dentro del mismo subsistema, ir hasta la sección **<drivers>** y copiar el siguiente texto (tanto para MySQL como para PostgreSQL):

```

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

```

Luego de realizar estos cambios, guardar y cerrar el archivo.

3.8 Configurar el servidor de correo electrónico

Siges utiliza jBoss para el envío de correos electrónicos. Por más información consultar el siguiente link: <https://docs.jboss.org/author/display/AS71/Mail+Subsystem>

Es importante configurar una cuenta de correo por la que Siges envíe correos para crear usuarios, restablecer contraseñas, enviar solicitudes (por ejemplo iniciar un proyecto, finalizar un proyecto, pasarlo a ejecución, etc) y enviar notificaciones automáticas a sus usuarios. Cada servidor de correo tiene sus características por lo que la configuración en Siges podrá variar dependiendo de las características del servidor de la cuenta de correo que se desee utilizar.

Para que la aplicación 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>
```

Nota: reemplazar "{host}" y "{port}" por la dirección IP o el nombre y el puerto del servidor SMTP que será usado para enviar correos electrónicos.

A continuación, se debe buscar el subsistema "urn:jboss:domain:mail:1.0", y dentro de él copiar el siguiente texto:

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

Notas:

1. El "from="{dirección mail from}" en la sección "mail-session" no siempre es necesario dependiendo de las características del servidor de correo.
2. Reemplazar {dirección mail from} por la dirección del correo que utilizará siges para enviar mails y "{user}" y "{pass}" por el nombre de usuario y la contraseña en el servidor de correo SMTP que el servidor de aplicaciones JBoss debe utilizar para conectarse cada vez que la aplicación requiera el envío de un mensaje de correo electrónico. Normalmente, el nombre de usuario tendrá el formato de una dirección de correo electrónico (por ejemplo, siges@agesic.gub.uy).

Luego de realizar estos cambios, guardar y cerrar el archivo. Es necesario reiniciar jBoss luego de la modificación de standalone.xml.

Notas: El sistema posee tres parámetros que son necesario configurar, esto se hace ingresando al sistema como PMO Transversal en el menú Administración – Configuración - General (es necesario hacer un clic sobre el botón "buscar"):

1. General cambiar valor del parámetro "CON_CORREO" a "true".
2. En el parámetro "MAIL_FROM" ponerle la casilla de correo que va a utilizar el sistema para enviar mails.

3. En caso que el servidor de correo electrónico necesite que el envío se realice utilizando TLS además de la configuración en el `standalone.xml` es necesario modificar el parámetro "MAIL_TLS" en "true". En este caso, además en el `standalone.xml` asegurarse de haber incluido el parámetro `ssl="true"` en: `<mail-session jndi-name="java:jboss/mail/SigesMail" debug="true" from=[mail_from] ssl="true">`

3.9 Configurar el 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"/>
    </login-module>
  </authentication>
</security-domain>
```

3.10 Configurar los 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).

3.11 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>

```

Luego de realizar estos cambios, guardar y cerrar el archivo.

3.12 *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 reemplazar 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: reemplazar {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).

3.13 *Habilitar la compresión 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>
```

3.14 *Iniciar el JBoss AS*

Para iniciar el servidor de aplicaciones JBoss AS se debe ejecutar el script **standalone.bat** (en MS Windows) o **standalone.sh** (en Linux) que se encuentran en la carpeta **{JBoss_HOME}/bin**, de la misma manera que se hizo al finalizar la sección 2.3.

3.15 *Instalar la aplicación en el JBoss AS*

En el paquete de archivos hay una carpeta “EARMySQL” con el .ear para MySQL o “EARPostgreSQL” con el .ear para PostgreSQL. Se deberá tomar el .ear correspondiente según el motor de base de datos elegido.

Para instalar (“deployar”) la aplicación en el servidor de aplicaciones JBoss AS, solo se debe copiar el archivo **SS-SIGES-ear-1.0-SNAPSHOT.ear** proporcionado a la carpeta **{JBoss_HOME}/standalone/deployments**. Tras unos pocos segundos en dicha carpeta aparecerá un nuevo archivo llamado **SS-SIGES-ear-1.0-SNAPSHOT.ear.deployed**, que indica que la aplicación se ha instalado correctamente; si no es así, o el nombre del archivo es SS-SIGES-

ear-1.0-SNAPSHOT.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.

3.16 Cambio de 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””, reemplazando 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””, reemplazando en ella 8443 por el puerto deseado. Luego, buscar la línea que comienza con “<connector name=“http”” y reemplazar en ella 8443 por el puerto deseado, que debe ser el mismo que el indicado anteriormente.

3.17 Creación de Carpeta para almacenamiento de imágenes

El módulo de Multimedia de un proyecto gestiona tres cosas: videos, cámaras webs e imágenes. Los videos deben ser publicados en youtube.com (o similar), luego en Siges se ingresa la URL. Las cámaras web, de forma análoga, también son URLs. Las imágenes se deben almacenar en una carpeta compartida dentro de la infraestructura donde está alojado el sistema (podría ser el servidor de Aplicaciones o de Base de datos) pero como condición para que esto funcione correctamente es que el usuario que inicia JBoss debe tener permisos sobre esta carpeta.

Si bien esta carpeta puede estar en diferentes lugares, se recomienda crearla dentro del servidor de aplicaciones en: "[ruta_jboss]/standalone/siges/media_files/"

En un sistema Linux para modificar el usuario que accede a una carpeta se debe ejecutar lo siguiente:

```
chown {usuario} {[ruta_jboss]/standalone/siges/media_files/}
```

Nota: sustituir {usuario} por el nombre del usuario que ejecuta el JBoss y {[ruta_jboss]/standalone/siges/media_files/} por la ruta a la carpeta creada.

Luego, en un sistema Linux para modificar los permisos se debe ejecutar lo siguiente:

```
chmod u+rw {[ruta_jboss]/standalone/siges/media_files/}
```

Nota: sustituir {[ruta_jboss]/standalone/siges/media_files/} por la ruta a la carpeta creada

Finalmente, es necesario iniciar sesión en el sistema con el rol usuario “PMO Transversal”, ir al menú Administración – Configuración – General y hacer un clic en “Buscar”. En el parámetro “FOLDER_MEDIA” modificar el valor y poner el path de la carpeta creada para almacenar las fotos (ejemplo: "[ruta_jboss]/standalone/siges/media_files/").

3.18 Acceder a la aplicación y configuración de parámetros

Para acceder a la aplicación se debe iniciar un navegador web (Internet Explorer, Mozilla Firefox, Google Chrome) y apuntarlo a cualquiera de las siguientes URLs:

http://{host}:{port}/SS-SIGES-web

Nota: remplazar {host} para dirección IP o el nombre del equipo donde reside el servidor de aplicaciones JBoss, y {port} por el número de puerto por el cual atiende conexiones (por defecto es 8080). Si todo funciona correctamente, debería verse la pantalla inicial de la aplicación Siges, solicitando el nombre de usuario y la contraseña; si no es así, se debe mirar el log del servidor para intentar detectar posibles errores.

Utilizar las siguientes credenciales para acceder:

- Usuario: admin
- Contraseña: admin

Luego de ingresar al sistema se podrá cambiar la contraseña.

Si todos los pasos anteriores fueron ejecutados en forma exitosa ya se podría acceder al sistema utilizando un explorador Web. Tener en cuenta que es un sistema “multi-organización”, por lo que con el usuario “admin”, se estará accediendo al menú de superadministrador. En este menú, solamente se pueden crear organismos y usuarios para cada organismo, no es posible desde acá modificar la información de cada organismo, entre ellas las variables necesarias para que funcione el correo.

Cuando se crea un usuario, el sistema le genera una contraseña y se la manda por correo. El problema es que en este momento recién se creó el organismo y el envío de correo por defecto está deshabilitado. El sistema creará igualmente el usuario pero el correo nunca llegará, por lo que no sabremos la contraseña y no podremos ingresar con ese usuario al organismo para habilitar el uso del correo. Para solucionar esto, los pasos a seguir son los siguientes:

- Desde el usuario admin, crear un nuevo organismo.
- Desde el usuario admin, sobre el organismo creado, dar de alta un nuevo usuario. Asegurarse que el correo sea correcto, ingresar el nombre y el apellido, seleccionar el área (al crear el organismo se creó un área con el mismo nombre por defecto) y fundamentalmente ponerle rol de “PMO Transversal” para el organismo. En este momento, el sistema creará el usuario pero no enviará la contraseña por correo.
- Ingresar a la base de datos de la aplicación, ya sea utilizando un administrador de bases de datos como MySQL Workbench o simplemente la línea de comando (en un sistema Linux) y ejecutar la siguiente sentencia (asegurarse que se está utilizando o se está parado sobre la base de siges):

```
update ss_usuario set usu_password = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX' where  
usu_correo_electronico = 'nuevousuario@organismo.gub.uy';
```

En la sentencia anterior sustituir nuevousuario@organismo.gub.uy por el correo del usuario ingresado en el sistema en el paso 2 y XXXXXXXXXXXXXXXXXXXXXXX por la contraseña del usuario. Tener en cuenta que el sistema utiliza MD5 para encriptar contraseñas por lo que no se podrá escribir directamente la contraseña deseada sino que la contraseña requerida deberá ser previamente encriptada con MD5 y se deberá escribir acá el resultado de esta transformación. Se podrá utilizar un servicio web para hacer esta encriptación como por ejemplo: <http://www.md5online.es/cifrar-md5.html>

- Una vez realizado esto, se podrá ingresar a Siges con el usuario creado para el organismo, utilizar el correo como usuario y la contraseña correspondiente.

- Ingresando con el usuario creado, ir al menú Administración – Configuración – General y hacer clic en “Buscar” (si no se ve este menú, es porque el usuario no fue creado con el rol “PMO Transversal”, en este caso se debe volver a ingresar como admin y modificar el usuario). El sistema mostrará la tabla con todos los parámetros para el organismo. Los mínimos a modificar son los siguientes:
 - CON_CORREO = true: habilita el uso del correo electrónico
 - ENVIO_NOTIFICACIONES = true: habilita las notificaciones
 - FOLDER_MEDIA = ‘path carpeta de fotos’: indicar acá el path completo de donde se almacenarán las fotos (sección 3.17)
 - MAIL_FROM = ‘correo@organismo.gub.uy’: poner acá la cuenta de correo que se utilizará en el organismo para que el sistema envía correos (debe ser la misma que se configuró en standalone.xml, sección 3.8)
 - MAIL_TLS = false o true: deberá estar en true si el servidor de correo electrónico del organismo requiere TLS
 - Es aconsejable verificar los valores de los parámetros: TAMANIO_MAX_ARCHIVO_DOCUMENTO, TAMANIO_MAX_ARCHIVO_MULTIMEDIA, TAMANIO_MAX_LOGO_ORGANISMO. Estos parámetros indican los tamaños máximos permitidos para los documentos subidos a los proyectos, las imágenes subidas a los proyectos y el logo del organismo respectivamente. Se recomienda reducir al menos el de imágenes. La cifra debe estar expresada en bytes.

3.19 Backups

Una vez instalado el sistema se recomienda realizar un backup de la aplicación (.ear) y todos los archivos de configuración nombrados anteriormente en este capítulo. Luego de cada upgrade es recomendable realizar un nuevo backup de este tipo.

En cuanto a los datos, más allá de la estrategia de backup elegida (diaria, semanal, incremental, etc), se recomienda respaldar dos cosas:

- La base de datos, realizando un dump de la misma
- La carpeta donde se almacenan las fotos de los proyectos que se detalla en la sección anterior.