Sistema de Agenda Electrónica

Manual de Instalación y configuración

Creación:	10/10/2016	Autor:	SP
Revisado:		Aprobado:	
Versión:	1.16	Plantilla:	
Código:	AG-SAE-MA-ES-01	Página:	1 de 62

Table of Contents

1 Introducción

Este documento constituye el manual de instalación de la aplicación Sistema de Agenda Electrónica (SAE) para Agesic, la cual es a su vez una adaptación del sistema IMMSAE (Sistema de Agenda Electrónica de la Intendencia de Montevideo).

2 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 **PostgreSQL 9.4**. Es necesario instalar dichos productos antes de proceder a instalar la aplicación en sí misma, como se explica en este documento. Se sugiere instalar el servidor de aplicaciones y el servidor de bases de datos en diferentes equipos físicos, previendo que en el ambiente final se cuente con más de un servidor de aplicaciones (es decir, más de una instancia de la aplicación), todos ellos conectados al mismo servidor de bases de datos. En todos los equipos donde se instale un servidor de aplicaciones se debe contar con un entorno de ejecución **Java (JRE) versión 7** instalado (es válido también un entorno de desarrollo Java, o JDK). Está contemplada la utilización de un Apache HTTPd Server como balanceador de carga y control de acceso (WAF).

La utilización de HTTPS (HTTP+SSL) en las conexiones **no es opcional**. En el caso de no utilizar un Apache HTTPd Server frontal, la configuración debe hacerse directamente en el servidor JBoss AS, mientras que en el caso de sí utilizar un Apache HTTPd Server frontal la configuración debe realizarse únicamente en éste, aunque es necesario habilitar un conector AJP en el servidor JBoss AS. A continuación se muestra esquemáticamente ambas opciones:

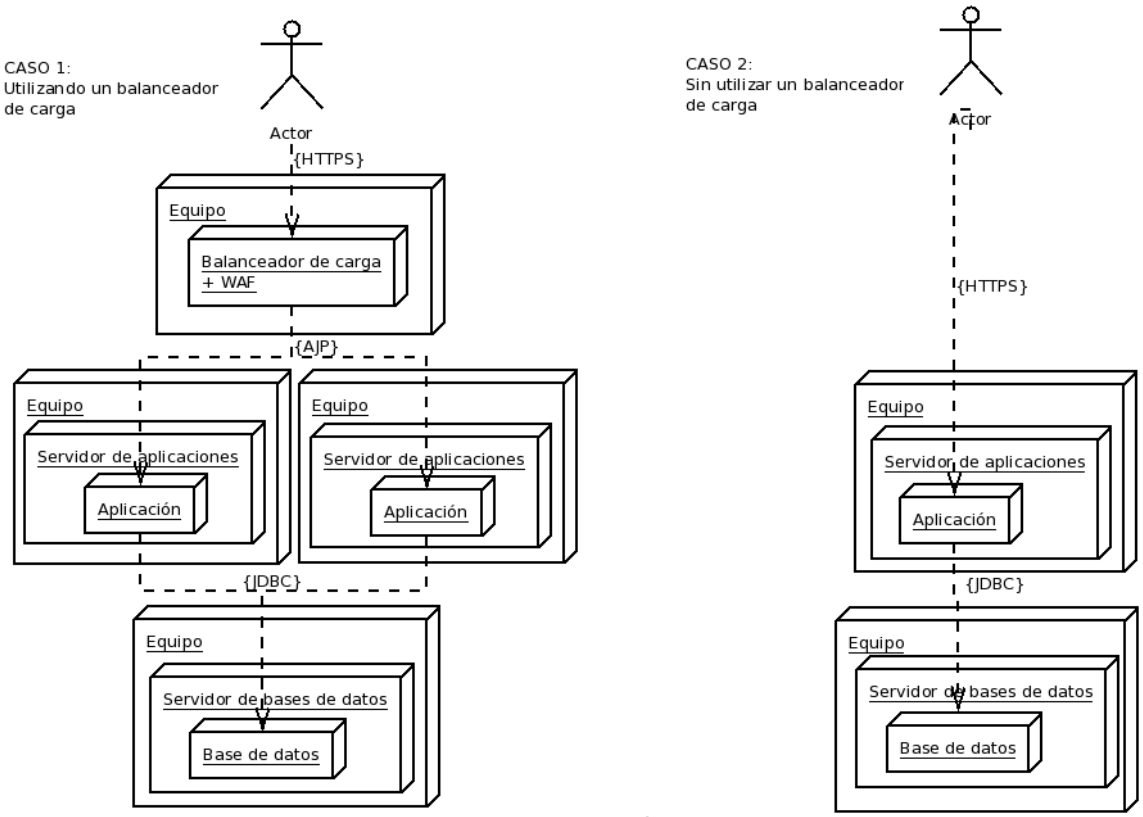

Figura 2.1: despliegue físico (casos)

Además, la aplicación puede funcionar integrada al Sistema de Control de Acceso de AGESIC (CDA) o no. En el caso de no utilizar CDA, el módulo público (“frontend”) no tendrá autenticación, mientras que el módulo privado (“backoffice”) tendrá un mecanismo de autenticación básico. En el caso de utilizar CDA, el módulo privado requerirá autenticación mediante CDA en todos los casos, mientras que el público podrá ser configurado para cada Agenda individual.

El software se distribuye en dos formatos, según sea determinado por AGESIC: para instalar desde cero, o empaquetado con un servidor de aplicaciones JBoss. El segundo de los formatos hace que la instalación sea mucho más sencilla por lo que de estar disponible se sugiere siempre solicitar dicha versión. El primero de los formatos, debido a que requiere mayor trabajo para la instalación, es recomendable para lograr un mayor entendimiento de la aplicación y tener más información en caso de tener que resolver problemas durante la operativa normal. Este documento describe el procedimiento de instalación según se cuente con los archivos en un formato o en el otro.

3 Procedimiento de instalación completo

Este documento provee los pasos necesarios para realizar la instalación de la aplicación AGESIC AGENDA partiendo de cero (no la versión empaquetada con el servidor de aplicaciones JBoss AS).

3.1 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. Alternativamente, en aquellos sistemas operativos que cuentan con un repositorio de software propio (típicamente, casi todas las distribuciones Linux) se puede utilizar el manejador de paquetes propio de la distribución.

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 "/usr/lib/jvm/jdk1.7.0_60/", entonces el valor de la variable de entorno JAVA_HOME debe ser también "/usr/lib/jvm/jdk1.7.0_60/".

Nota: si en el equipo donde se instala el servidor de aplicaciones JBoss AS 7 existe más de una versión de Java, y la variable JAVA_HOME ya existe y apunta a una versión diferente de Java (más antigua o más reciente) y no puede modificarse (porque afectaría a otras aplicaciones existentes previamente en el equipo) se puede aplicar la siguiente alternativa: editar el archivo **{JBOSS_HOME}/bin/standalone.conf** (en Linux) o **{JBOSS_HOME}/bin/standalone.conf.bat** (en Windows) y buscar la línea que comienza con el siguiente texto: "#JAVA_HOME="; a dicha línea quitarle el numeral (#) del comienzo y remplazar todo lo que está a continuación del símbolo de igual (=) por el mismo valor que se desea aplicar a la variable JAVA_HOME; este cambio solo aplica para esta instancia del servidor de aplicaciones JBoss AS 7.

3.2 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 7.1.1):

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.GZ según se haya elegido) en la ubicación deseada, la cual 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 en este momento 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. Se puede obtener más información observando el log del servidor de aplicaciones, el cual se encuentra en el directorio **{JBOSS_HOME}/standalone/logs**.

Luego de haber realizado la instalación del servidor de aplicaciones JBoss se debe copiar al directorio **{JBOSS_HOME}/standalone/configuration** el archivo llamado **jboss-ejb-client.properties** que se encuentra en la carpeta configuration del paquete de instalación. El contenido de este archivo no debe ser modificado.

3.3 Instalación de PostgreSQL

PostgreSQL también es un producto de distribución libre. Puede descargarse gratuitamente un instalador gráfico desde la siguiente URL (asegurarse que se trate de la versión 9.4):

http://www.enterprisedb.com/products-services-training/pgdownload

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 **postgres** (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. Alternativamente, en aquellos sistemas operativos que cuentan con un repositorio de software propio (típicamente, casi todas las distribuciones Linux) se puede utilizar el manejador de paquetes propio de la distribución.

3.4 Crear la base de datos para la aplicación

Antes de proceder a crear la base de datos es necesario una cuenta de usuario (rol de login) específica para la aplicación AGENDA, que será utilizada para la comunicación entre el servidor de aplicaciones (JBoss AS) y el servidor de bases de datos (PostgreSQL Server). Si bien puede utilizarse la cuenta **postgres** no se recomienda por razones de seguridad.

3.4.1 Conectarse a la base de datos

Para poder realizar las tareas que se describen a continuación es necesario iniciar sesión en el servidor de bases de datos usando el programa psql:

```
$> psql --host {host} --port {port} –user postgres
```

donde **{host}** y **{port}** deben ser sustituidos por el nombre o dirección IP del servidor de bases de datos y el puerto por el que atiende conexiones respectivamente.

3.4.2 Crear un usuario específico para la aplicación

Para crear una nueva cuenta de usuario debe ejecutarse la siguiente sentencia SQL:

```
psql> CREATE ROLE sae LOGIN PASSWORD '{contraseña}';
```

donde **{contraseña}** debe ser remplazado por la contraseña deseada.

3.4.3 Crear la base de datos para la aplicación

A continuación, para crear la base de datos para la aplicación se debe ejecutar la siguiente sentencia SQL:

```
psql> CREATE DATABASE sae WITH OWNER=sae ENCODING='UTF8' ;
```

3.4.4 Crear el esquema global

Luego de crear la base de datos es necesario crear el esquema global, donde se almacenarán todos los datos generales de la aplicación (no dependientes de tenant) y un esquema por cada tenant que se desee configurar. En el esquema global se almacena, entre otras cosas, los usuarios registrados en la aplicación y su relación con los roles. Para crear el esquema global se debe ejecutar el script esquema-global.sql, provisto como parte del paquete instalador, con la siguientes sentencias SQL:

```
psql> \c sae
psql> \i {ruta}
```

donde **{ruta}** debe ser sustituido por la ruta absoluta al archivo esquema-global.sql.

3.4.5 Registrar un usuario superadministrador para la aplicación

Para que la aplicación sea utilizable es necesario registrar al menos un usuario que pueda iniciar sesión en la aplicación luego de la instalación. Para hacerlo, se debe ejecutar la siguiente sentencia en la base de datos:

```
psql> INSERT INTO global.ae_usuarios (id, codigo, nombre, fecha_baja, password, correoe,
superadmin) VALUES (nextval('global.s_ae_usuario'), '<codigo>', '<nombre>', NULL,
'<contraseña>', '<correoelectronico>', true);
```

donde los valores entre símbolos “<” y “>” deben ser remplazados de la siguiente manera:

- **<codigo>**: ingresar el código de usuario deseado para el primer superadministrador. En el caso de utilizar CDA para el control de acceso, el valor ingresado debe corresponderse con un código de usuario registrado en CDA (normalmente la cédula de identidad). En el caso de no utilizar CDA se puede elegir el valor a gusto (por ejemplo, “admin”).
- **<nombre>**: ingresar el nombre real de la persona que será superadministrador. No se hace ninguna comprobación respecto del valor ingresado.
- **<contraseña>**: en el caso de utilizar CDA puede dejarse en blanco o ingresar cualquier valor ya que no es utilizado. En el caso de no utilizar CDA debe ingresarse la transformación MD5+Base64 de la contraseña elegida (por ejemplo, para utilizar la contraseña “admin” debe ingresarse el valor “ISMvKXpXpadDiUoOSoAfww==”).
- **<correoelectronico>**: ingresar la dirección de correo electrónico de la persona que será superadministrador. Puede dejarse en blanco pero se recomienda ingresar una dirección válida.

3.4.6 Crear un esquema para cada tenant

Cada “empresa” (organismo, unidad ejecutora, dirección, división, inciso, etc) que desee utilizar la aplicación y diferenciarse del resto deberá tener un “tenant” asociado. Para crear el esquema correspondiente a cada uno de los tenants que deben ser configurados en la aplicación se debe realizar el siguiente procedimiento:

1. Copiar el archivo esquema-template.sql cambiando 'template' por el nombre del tenant; por ejemplo, si el nombre del tenant es 'ejemplo1' se debe llamar al archivo esquema-ejemplo1.sql.
2. Editar el archivo copiado y remplazar todas las ocurrencias del texto “{esquema}” por el mismo nombre que el tenant; este nombre debe estar compuesto solo por letras minúsculas y dígitos y debe comenzar con una letra; en particular no se pueden usar espacios ni tildes.

3. Ejecutar el script modificado, utilizando la siguiente sentencia:

   ```
   psql> \i {ruta}
   ```

   donde **{ruta}** debe ser sustituido por la ruta absoluta al archivo copiado y modificado.

3.4.7 Crear la empresa (tenant) inicial

Para que el usuario administrador pueda acceder a la aplicación por primera vez es necesario crear la primer empresa o tenant. Para esto, se debe realizar el siguiente procedimiento:

1. Crear el esquema en la base de datos, según se explica en la sección 3.4.6.
2. Crear la empresa con los datos mínimos necesarios; para esto se debe ejecutar la siguiente sentencia SQL en la base de datos:

```
INSERT INTO global.ae_empresas (id, nombre, datasource, fecha_baja, org_id, org_codigo,
org_nombre, unej_id, unej_codigo, unej_nombre, logo, cc_finalidad, cc_responsable,
cc_direccion, logo_texto, timezone, formato_fecha, formato_hora, oid, pie_publico)
VALUES (nextval('global.s_ae_empresa'), 'Empresa 1', '<esquema>', NULL, 0, '0', 'No
configurado', 0, '0', 'No configurado', NULL, 'No configurado', 'No configurado', 'No
configurado', 'No configurado', 'GMT', 'dd/MM/yyyy', 'HH:mm', 'No configurado', '');
```

donde **<esquema>** debe ser sustituido por el nombre del esquema utilizado en el paso 1.

Notar que la sentencia SQL utiliza datos genéricos, que luego deberán ser modificados; alternativamente, si se conocen en este momento, puede modificarse la sentencia para incluir dichos datos.

3.4.8 Habilitar el acceso a la base de datos desde el equipo donde reside la aplicación

Para que la aplicación pueda conectarse a la base de datos es necesario indicarle al servidor PostgreSQL que debe aceptar conexiones desde el equipo donde reside la aplicación (el servidor JBoss AS). Para hacer esto se debe editar el archivo pg_hba.conf que se encuentra en el directorio data de la instalación de PostgreSQL y añadir una línea como la que sigue:

```
host    sae     sae     <ip>/<mascara>        md5
```

donde <ip> y <mascara> deben ser sustituidos por la dirección IP del equipo donde reside el JBoss, o su subred. Por ejemplo “192.168.1.0/24” indica que debe aceptar conexiones de cualquier equipo que se encuentre en la subred “192.168.1”, mientras que “10.1.5.201/32” indica que debe aceptar conexiones solo del equipo cuya dirección IP es “10.1.5.201”.

Si se opta por indicar una dirección IP específica y a la vez se debe configurar más de un servidor de aplicaciones JBoss AS, debe añadirse una línea por cada uno de ellos indicando su correspondiente dirección IP.

3.5 Crear y actualizar módulos en JBoss AS

Para que la aplicación funcione correctamente es necesario crear algunos módulos en JBoss y modificar otros existentes. Los módulos de JBoss AS pueden verse como extensiones al servidor de aplicaciones aplicadas mediante el añadido de librerías (archivos JAR).

Los módulos que deben ser creados son los siguientes:

- **Apache Digester**: crear el directorio **{JBOSS_HOME}/modules/org/apache/commons/digester/main** y copiar dentro de él los archivos contenidos por el paquete apache-digester.zip (commons-digester-1.8.jar y module.xml).

- **Apache Fileupload**: crear el directorio **{JBOSS_HOME}/modules/org/apache/commons/fileupload/main** y copiar dentro de él los archivos contenidos por el paquete apache-fileupload.zip (commons-fileupload-1.3.1.jar y module.xml).
- **JasperSoft JasperReports**: crear el directorio **{JBOSS_HOME}/modules/net/sourceforge/jasperreports/main** y copiar dentro de él los archivos contenidos por el paquete jaspersoft-jasperreports.zip (commons-javaflow-20060411.jar, itext-2.1.7.jar, jasper-compiler-jdt-5.5.15.jar, jasperreports-3.0.0.jar, jcommon-1.0.0.jar, jfreechart-1.0.0.jar y module.xml).
- **JBoss Remoting**: crear el directorio **{JBOSS_HOME}/modules/remote/main** y dentro de él copiar los archivos contenidos en el paquete jboss-remoting.zip (jboss-client.jar y module.xml). El archivo jboss-client.jar también puede ser obtenido de la propia instalación de JBoss, en el directorio **JBOSS_HOME}/bin/client**.
- **PostgreSQL JDBC Driver**: crear el directorio **{JBOSS_HOME}/modules/org/postgresql/main** y dentro de él copiar los archivos contenidos por el paquete postgresql-jdbcdriver.zip (postgresql-9.4-1204.jdbc4.jar y module.xml).
- **Sofis CDA Adapter**: crear el directorio **{JBOSS_HOME}/modules/cda/main** y dentro de él copiar los archivos contenidos por el paquete sofis-cdaadapter.zip (annotations-api-6.0.16.jar, bcprov-jdk15-1.45.jar, CDAServiceProvider.jar, commons-codec-1.3.jar, commons-collections-3.1.jar, commons-httpclient-3.0.jar, commons-lang-2.1.jar, jcip-annotations-1.0.jar, jcl-over-slf4j-1.6.1.jar, joda-time-1.6.2.jar, juli-6.0.16.jar, jul-to-slf4j-1.6.1.jar, junit-3.8.1.jar, log4j-over-slf4j-1.6.1.jar, not-yet-commons-ssl-0.3.9.jar, opensaml-2.4.1.jar, openws-1.4.1.jar, serializer-2.7.1.jar, slf4j-api-1.6.1.jar, tomcat-juli-5.5.23.jar, velocity-1.5.jar, xalan-2.7.1.jar, xml-apis-1.3.04.jar, xml-resolver-1.2.jar, xmlsec-1.4.4.jar, xmltooling-1.3.1.jar y module.xml).
- **Sofis PGE Adapter**: crear el directorio **{JBOSS_HOME}/modules/pge/main** y dentro de él copiar los archivos contenidos por el paquete sofis-pgeadapter.zip (bcprov-jdk15-1.43.jar, opensaml-2.3.1.jar, openws-1.3.0.jar, PGEClient-1.6.jar, xmltooling-1.2.1.jar y module.xml).
- **Sofis SAELogin**: crear el directorio **{JBOSS_HOME}/modules/saelogin/main** y dentro de él copiar los archivos contenidos por el paquete sofis-saelogin.zip (saelogin.jar, unboundid-ldapsdk-3.0.0.jar y module.xml).
- **Sofis SysLog**: crear el directorio **{JBOSS_HOME}/modules/syslog/main** y dentro de él copiar los archivos contenidos por el paquete syslog.zip (jboss-syslog-1.0.3.jar y module.xml).

Además, también deben ser actualizados los siguientes módulos:

- **JBoss AS Web**: editar el archivo **{JBOSS_HOME}/modules/org/jboss/as/web/main/module.xml** y comentar (o eliminar) la línea que hace referencia al archivo jasper-jdt-{version}.jar, donde {version} indica el número de versión del mencionado archivo (por ejemplo, jasper-jdt-7.0.3.Final.jar).
- **JBoss Javassist**: remplazar el contenido del directorio **{JBOSS_HOME}/modules/org/javassist/main** por los archivos contenidos por el paquete jboss-javassist.zip (javassist-3.18.1-GA.jar y module.xml).

- **Hibernate Envers**: remplazar el contenido del directorio **{JBOSS_HOME}/modules/org/hibernate/envers/main** por los archivos contenidos por el paquete hibernate-envers.zip (hibernate-envers-4.2.21.Final.jar y module.xml).
- **Hibernate JPA**: remplazar el contenido del directorio **{JBOSS_HOME}/modules/org/hibernate/main** por los archivos contenidos por el paquete hibernate-jpa.zip (hibernate-commons-annotations-4.0.2.Final.jar, hibernate-core-4.2.21.Final.jar, hibernate-entitymanager-4.2.21.Final.jar, hibernate-infinispan-4.2.21.Final.jar y module.xml).

Luego de crear y actualizar estos módulos, es necesario registrar los módulos JBoss Remoting y Sofis SAELogin como globales para todo el servidor de aplicaciones. Para esto, se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar el subsistema cuyo nombre es "**urn:jboss:domain:ee:1.0**"; normalmente este subsistema estará completamente vacío (<subsystem xmlns="urn:jboss:domain:ee:1.0"/>), debiendo ser remplazado con lo siguiente (el texto está disponible en el archivo **modules.txt** en la carpeta configuration del paquete instalador):

```
<subsystem xmlns="urn:jboss:domain:ee:1.0">
  <global-modules>
    <module name="remote" slot="main"/>
    <module name="saelogin" slot="main"/>
  </global-modules>
</subsystem>
```

En el caso de que este subsistema no estuviese vacío, deberá realizarse las modificaciones correspondientes de forma que la sección <global-modules> incluya a los dos módulos remote y saelogin además de otros módulos que pudieran existir previamente.

3.6 Configurar los orígenes de datos para la aplicación

Para que la aplicación pueda conectarse a la base de datos PostgreSQL a través del servidor de aplicaciones JBoss AS es necesario definir **dos** orígenes de datos en este último, uno para acceder al esquema **global** y otro para acceder a los esquemas correspondientes a cada uno de los **tenants** (dados que son, en principio, idénticos, contenido las mismas tablas y secuencias no hace falta definir un origen de datos diferente para cada tenant). Para hacerlo, se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** 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 (el texto está disponible en el archivo **datasources.txt** en la carpeta configuration del paquete instalador):

```
<datasource jndi-name="java:/postgres-sae-ds" pool-name="postgres-sae-ds" enabled="true"
use-java-context="true">
  <connection-url>jdbc:postgresql://{db-host}:{db-port}/sae</connection-url>
  <driver>postgres</driver>
  <security>
    <user-name>{db-user}</user-name>
    <password>{db-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>
    <prepared-statement-cache-size>100</prepared-statement-cache-size>
```

```
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
<datasource jndi-name="java:/postgres-sae-global-ds" pool-name="postgres-sae-global-ds"
enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql://{db-host}:{db-port}/sae</connection-url>
  <driver>postgres</driver>
  <security>
    <user-name>{db-user}</user-name>
    <password>{db-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>
    <prepared-statement-cache-size>100</prepared-statement-cache-size>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
```

Nota: remplazar "**{db-host}**" y "**{db-port}**" por la dirección IP y número de puerto donde atiende conexiones el servidor de bases de datos PostgreSQL (si el servidor de bases de datos PostgreSQL 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 "5432"). También remplazar "**{db-user}**" y "**{db-pass}**" por el nombre de usuario y la contraseña especificadas al crear la cuenta de usuario en el punto 3.4.

A continuación, dentro del mismo subsistema, ir hasta la sección **<drivers>** y copiar el siguiente texto (el texto está disponible en el archivo **jdbcdrivers.txt** en la carpeta configuration del paquete instalador):

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

3.7 Configurar el servidor de correo electrónico

Para que la aplicación pueda enviar correos electrónicos es necesario configurar un servidor SMTP en el servidor de aplicaciones JBoss AS. Para hacerlo, se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar la sección **<socket-binding-group>** (normalmente se encuentra al final del archivo) e incluir dentro de ella el siguiente texto (el texto está disponible en el archivo **socketbindings.txt** en la carpeta configuration del paquete instalador):

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

Nota: remplazar "**{smtp-host}**" y "**{smtp-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 (el texto está disponible en el archivo **mailsession.txt** en la carpeta configuration del paquete instalador):

```
<mail-session jndi-name="java:/sae/mail">
  <smtp-server ssl="true" outbound-socket-binding-ref="sae-mail-smtp">
```

```
    <login name="{smtp-user}" password="{smtp-pass}"/>
  </smtp-server>
</mail-session>
```

Nota: remplazar "{**smtp-user**}" y "{**smtp-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, sae@agesic.gub.uy).

3.8 Configurar el mecanismo de registro de actividades (log)

La aplicación registra casi todas las actividades que realiza en el archivo de log del servidor. Dado que la aplicación en realidad delega las actividades de registro al servidor de aplicaciones, es éste quien se encarga de efectivamente escribir los registros enviados por la aplicación en los medios correspondientes. A continuación se explica cómo realizar la configuración para dos medios de registro usuales: mediante **archivos locales** en el sistema de archivos y mediante el sistema **SysLog** (registro remoto a un servidor dedicado a esta actividad).

3.8.1 Registro en archivos locales en el sistema de archivos

El servidor de aplicaciones JBoss AS mantiene un registro de operaciones (log) en un directorio propio de propósito específico para ello; este directorio suele ser **{JBOSS_HOME}/standalone/log**. Los archivos generados en este directorio son rotados periódicamente. Si se desea que los archivos de registro sean almacenados en otro directorio (por ejemplo en el directorio de log general del sistema operativo, que en el caso de Linux suele ser el directorio /var/log) se debe hacer lo siguiente:

1. Si no existe aún, crear el directorio donde el servidor de aplicaciones debe escribir los archivos de log. En adelante este directorio será referenciado como "**{JBOSS_LOG_DIR}**". Por ejemplo, /var/log/jboss-as.
2. Asignarle permisos de lectura y escritura al usuario del sistema operativo con el cual se ejecutará el servidor de aplicaciones sobre el directorio **{JBOSS_LOG_DIR}**.
3. Editar el archivo **{JBOSS_HOME}/bin/standalone.conf** (en Linux) o **{JBOSS_HOME}/bin/standalone.conf.bat** (en Windows) y buscar la línea que contiene el siguiente texto: "*if ["x$JAVA_OPTS" = "x"]; then*". Debajo de esa línea deben verse varias líneas que comienzan con el texto "*JAVA_OPTS="$JAVA_OPTS ...*". Añadir una línea adicional (justo antes de la línea que comienza con el texto '*else*') con lo siguiente:

   ```
   JAVA_OPTS="$JAVA_OPTS -Djboss.server.log.dir={JBOSS_LOG_DIR}"
   ```

   sustituyendo el texto "**{JBOSS_LOG_DIR}**" por la ruta absoluta al directorio creado en el punto 1.

3.8.2 Configuración de SysLog

Para que además realice el registro en un servidor SysLog se debe realizar la siguiente configuración (se debe disponer de un servidor SysLog activo del cual se conozca al menos la dirección IP y el puerto por el que atiende conexiones UDP, y además debe existir conectividad entre el equipo donde se encuentra el servidor JBoss y el equipo donde se encuentra el servidor SysLog): editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar el subsistema

"**urn:jboss:domain:logging:1.1**" y copiar dentro de él el siguiente texto (el texto está disponible en el archivo **syslog.txt** en la carpeta configuration del paquete instalador):

```
<custom-handler name="SYSLOG" class="x1.jboss.syslog.SyslogHandler" module="syslog">
  <level name="INFO"/>
  <properties>
    <property name="loghost" value="{syslog-host}"/>
    <property name="port" value="{syslog-port}"/>
    <property name="protocol" value="udp"/>
    <property name="application" value="sae"/>
    <property name="facility" value="daemon"/>
  </properties>
</custom-handler>
```

Nota: remplazar "**{syslog-host}**" y "**{syslog-port}**" por la dirección IP (o nombre del equipo) y puerto por el que atiende conexiones el servidor SysLog (debe necesariamente ser UDP, no TCP).

Luego, buscar el elemento <root-logger> y añadir a la lista de "handlers" uno llamado "SYSLOG", debiendo quedar de la siguiente manera:

```
<root-logger>
  <level name="INFO"/>
  <handlers>
    <handler name="CONSOLE"/>
    <handler name="FILE"/>
    <handler name="SYSLOG"/>
  </handlers>
</root-logger>
```

3.9 Habilitar el acceso seguro (HTTPS)

La aplicación está diseñada para aceptar conexiones solo por HTTPS (HTTP + SSL/TLS) por lo que es obligatoria esta configuración. Sin embargo, la misma depende de si se utiliza un servidor Apache HTTPd Server frontal o no; en el caso de utilizar un servidor Apache HTTPd Server la configuración de SSL debe hacerse en éste, mientras que en el caso de no usarlo la configuración debe hacerse en el servidor JBoss AS.

De esta manera se tienen dos casos:

- Cuando no se utiliza Apache HTTPd Server:

  [browser] → {HTTPS} → [JBoss]

- Cuando se utiliza Apache HTTPd Server

  [browser] → {HTTPS} → [Apache] → {AJP} → [JBoss]

A continuación se describen ambos casos.

3.9.1 Configuración sin Apache HTTPd Server

De fábrica el servidor de aplicaciones JBoss AS no acepta conexiones SSL. Para habilitarlo se debe contar con un keystore (un repositorio de certificados digitales) que contendrá únicamente al certificado digital que entregará el servidor de aplicaciones a los navegadores web de los usuarios cada vez que inicien una conexión nueva.

Crear un keystore conteniendo el certificado digital

Se tienen dos casos: cuando se cuenta con un certificado digital emitido por una autoridad certificadora reconocida (AC), y cuando no se tiene dicho certificado. En ambos casos se creará

en el directorio **{JBOSS_HOME}/standalone/configuration/** un keystore nuevo, llamado **sae.jks**.

Caso 1: si se cuenta con un certificado digital de una autoridad certificadora reconocida éste debe ser importado al keystore con el siguiente comando:

```
$> keytool -v -importkeystore -srckeystore {archivo} -srcstoretype PKCS12 -destkeystore
          {JBOSS_HOME}/standalone/configuration/sae.jks -deststoretype JKS
```

remplazando el texto **{archivo}** por la ruta completa al archivo que contiene el certificado digital (usualmente de extensión .p12 o .pfx) y **{JBOSS_HOME}** por la ruta absoluta a donde se encuentra instalado el servidor de aplicaciones JBoss AS. 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 para el nuevo keystore, que debe ser la misma.

Caso 2: si no se cuenta con un certificado digital reconocido se puede crear uno "casero", aunque se debe tener en cuenta que todos los usuarios, al acceder a la aplicación, verán un mensaje de advertencia indicándoles que el servidor utiliza un certificado digital que no es de confianza. Para crear un keystore conteniendo el nuevo certificado digital se debe ejecutar el siguiente comando:

```
$>keytool -genkeypair -keystore {JBOSS_HOME}/standalone/configuration/sae.jks -alias sae
                        -keyalg RSA -keysize 1024 -validity 365
```

remplazando el texto **{JBOSS_HOME}** por la ruta absoluta a donde se encuentra instalado el servidor de aplicaciones JBoss AS. Al ejecutar este comando el programa keytool solicitará primero el ingreso de la contraseña para el nuevo 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", donde se debe ingresar el nombre de dominio que corresponde al servidor, por ejemplo "agenda.organismo.gub.uy") y el último ("país", donde se debe ingresar el texto "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: el comando keytool forma parte de cualquier entorno de ejecución de Java y se encuentra en el directorio {JAVA_HOME}/bin. 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 o modificar la variable de entorno PATH (tanto en MS Windows como en Linux) para incluir el directorio bin de la instalación del JRE.

Crear un conector SSL en el servidor JBoss AS

Una vez creado el keystore se debe indicarle al servidor de aplicaciones JBoss que permita las conexiones seguras, y que utilice dicho keystore, para lo cual se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y 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 (el texto está disponible en el archivo **httpssinapache.txt** en la carpeta configuration del paquete instalador):

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"
enabled="false"/>
<connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https"
secure="true">
```

```
  <ssl password="{cert-pass}" certificate-key-file="${jboss.server.config.dir}/sae.jks"
protocol="TLSv1"/>
</connector>
```

remplazando **{cert-pass}** por la contraseña del keystore.

3.9.2 Configuración con Apache HTTPd Server

Cuando se utiliza un servidor Apache HTTPd Server frontal la configuración de SSL/TLS debe realizarse en éste y no en el servidor JBoss; de todas maneras, en el JBoss AS sí es necesario habilitar un nuevo conector AJP para la comunicación entre el servidor Apache HTTPd Server y el servidor JBoss AS.

Habilitar un conector AJP en JBoss AS

Para habilitar un conector AJP en el servidor JBoss AS se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y 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 (el texto está disponible en el archivo **httpsconapache1.txt** en la carpeta configuration del paquete instalador):

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"
enabled="false"/>
<connector name="ajp13" protocol="AJP/1.3" scheme="https" socket-binding="ajp"/>
```

Obtener un certificado digital para Apache HTTPd Server

Se tienen dos casos: cuando se cuenta con un certificado digital emitido por una autoridad certificadora reconocida (AC), y cuando no se tiene dicho certificado.

Caso 1: si se cuenta con un certificado digital de una autoridad certificadora reconocida, la acción a seguir depende del formato del mismo:

- Si es PKCS 12 (o PFX), debe ser convertido a formato PEM, con el siguiente comando:

  $> openssl pkcs12 -in **{entrada}** -out **{salida}**

  donde **{entrada}** debe ser sustituido por el nombre del archivo que contiene el certificado y **{salida}** por el mismo valor anexando la extensión ".pem".

- Si ya está en formato PEM no es necesario realizar ninguna acción.

Caso 2: si no se cuenta con un certificado digital reconocido se puede crear uno "casero", aunque se debe tener en cuenta que todos los usuarios, al acceder a la aplicación, verán un mensaje de advertencia indicándoles que el servidor utiliza un certificado digital que no es de confianza. Para crear un certificado adecuado para ser utilizado con Apache HTTPd Server se debe realizar lo siguiente:

1. Crear un keystore conteniendo el nuevo certificado digital con el siguiente comando:

   ```
   $>keytool -genkeypair -keystore sae.jks -alias sae -keyalg RSA -keysize 1024
   -validity 365
   ```

   Al ejecutar este comando el programa keytool solicitará primero el ingreso de la contraseña para el nuevo 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", donde se debe ingresar el nombre de dominio que corresponde al servidor, por ejemplo "agenda.organismo.gub.uy") y el último ("país", donde se debe ingresar el texto "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.

2. Exportar el par de claves generado a un certificado en formato PKCS 12 con el siguiente comando:

   $> keytool -importkeystore -srckeystore sae.jks -destkeystore sae.p12 -srcstoretype jks -deststoretype pkcs12

   El programa solicitará una contraseña para el archivo de destino (dos veces) y la contraseña del keystore. Se recomienda usar la misma.

3. Convertir el archivo en formato PKCS 12 a PEM, que es el formato que requiere Apache HTTPd Server con el siguiente comando:

   $> openssl pkcs12 -in sae.p12 -out sae.pem

   Esta vez el programa openssl solicitará una contraseña para el archivo de origen y de destino. Se recomienda usar la misma.

Si no se cuenta con OpenSSL instalado en el sistema es posible utilizar alguna herramienta externa, por ejemplo Portecle (http://portecle.sourceforge.net/):

1. Descargar Portecle de http://portecle.sourceforge.net.
2. Iniciar Portecle.
3. Crear un nuevo keystore de tipo “PKCS #12” (File → New keystore).
4. Generar un par de claves RSA (Tools → Generate key pair), especificando como nombre el nombre de dominio que corresponde al servidor, por ejemplo “agenda.organismo.gub.uy”) y país, donde se debe ingresar el texto “UY”. Cuando el programa solicite un “alias” ingresar el texto que se desee.
5. Exportar el par de claves como un certificado PEM (botón derecho sobre el alias especificado en el paso anterior, seleccionar Export y elegir “Private key and certificates” y “PEM Encoded”). Cuando el programa solicite una contraseña, ingresar un valor que se considere apropiado.
6. Elegir la ubicación donde se desea guardar el archivo.

Habilitar HTTPS en el servidor Apache HTTPd Server

Para permitir conexiones HTTPS en el servidor Apache se debe habilitar un conjunto de módulos, indicar que se desea aceptar conexiones por un nuevo puerto y crear un “host virtual” como se muestra a continuación (no es posible indicar en cuál archivo añadir este texto porque existen muchas variedades en cuanto a la instalación de Apache HTTPd Server y cada una de ellas utiliza una distribución de archivos de configuración diferentes; incluso es posible que deban modificarse más de un archivo a la vez; el texto está disponible en el archivo **httpsconapache2.txt** en la carpeta configuration del paquete instalador):

```
# Abrir un puerto exclusivo para HTTPS
listen: {apache-port}
…
# Habilitar los módulos requeridos
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule ssl_module modules/mod_ssl.so
…
# Configurar el host virtual
<VirtualHost *:{apache-port}>
```

```
  SSLEngine on
  SSLProxyEngine on
  SSLCertificateFile "{certificado}"
  <Proxy balancer://agenda2proxy>
    BalancerMember ajp://{host-jboss}:8009
  </Proxy>
  ProxyRequests Off
  ProxyPreserveHost On
  ProxyPass / balancer://agenda2proxy/ stickysession=JSESSIONID|jsessionid nofailover=On
  ProxyPassReverse / https://{apache-host}:{apache-port}/
</VirtualHost>
```

donde **{apache-port}** debe ser sustituido por el puerto por el cual se desea que Apache HTTPd Server atienda conexiones (usualmente es el 443), **{certificado}** debe ser sustituido por la ruta completa al archivo que contiene el certificado a utilizar para entregarle a los usuarios que se conecten (debe estar en formato PEM) y **{apache-host}** debe ser sustituido por el nombre o dirección IP del servidor Apache HTTPd Server.

3.10 Mejorar la seguridad del servidor

De fábrica tanto JBoss AS como Apache HTTPd Server ofrecen características que usualmente son consideradas inseguras. Por esta razón es necesario aplicar algunos cambios con el objetivo de hacer un poco más segura la instalación.

3.10.1 Mejoras en la seguridad de JBoss AS

En primer lugar, se debe deshabilitar la página de bienvenida del servidor, para lo cual se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y modificar el valor de la propiedad " enable-welcome-root" pasando de "true" a "false".

También es recomendable ocultar el nombre y versión del servidor JBoss AS en uso; para esto editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y añadir la sección "<system-properties>" justo debajo de "</extensions>" con el siguiente contenido (el texto está disponible en el archivo **jbosssystemproperties.txt** en la carpeta configuration del paquete instalador):

```
<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"/>
  <property name="org.apache.tomcat.util.http.Parameters.MAX_COUNT" value="10000"/>
  <property name="org.apache.coyote.http11.Http11Protocol.SERVER" value="Server
undisclosed"/>
</system-properties>
```

Nota: si la sección "<system-properties>" ya está presente en el archivo se debe añadir este contenido verificando que ninguna propiedad quede duplicada.

3.10.2 Mejoras en la seguridad de Apache HTTPd Server

Nota: la configuración que se explica a continuación debe hacerse dentro del host virtual definido para esta aplicación, según la sección 3.9.2.

El objetivo de esta configuración es modificar algunas cabeceras del servidor Apache HTTPd Server para ocultar el nombre y la versión del servidor en uso, impedir que la aplicación sea incluida dentro un i-frame de terceros, entre otras cosas (el texto está disponible en el archivo **apacheheaders.txt** en la carpeta configuration del paquete instalador):

```
#Habilitar el módulo headers
```

```
LoadModule headers_module modules/mod_headers.so
…
#Establecer las cabeceras deseadas
Header set Server "Undisclosed server"
Header set X-Powered-By "JSF"
Header set Strict-Transport-Security "max-age=31536000; includeSubDomains"
Header set X-Frame-Options "SAMEORIGIN"
Header set X-Content-Type-Options "nosniff"
```

3.11 Configurar el dominio de seguridad

La aplicación delega al servidor de aplicaciones JBoss AS 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 seguridad**.

Según se desee utilizar el Sistema de Control de Acceso de AGESIC (CDA) o no, el mecanismo de configuración es diferente. En esta sección se continuación se describen ambos mecanismos. Es importante tener en cuenta que si bien la configuración del dominio de seguridad se realiza en el servidor de aplicaciones JBoss AS y puede ser utilizada tanto desde la parte pública como de la parte privada de la aplicación, cada una de estas partes a su vez debe ser configurada en forma independiente como se explica en la secciones 5.1 y 5.2.

Nota: a los efectos de cumplir con las normas de seguridad requeridas por AGESIC es obligatorio el uso de CDA. En el caso de las instalaciones dentro de los ambientes de los organismos serán los responsables de seguridad de éstos la determinación si se exige el uso de CDA o no. Si no se está seguro si se requiere la configuración de CDA o no, se puede considerar lo siguiente: si alguna de las dos partes de la aplicación (pública o privada) requiere autenticación, se debe configurar CDA (como se describió, más adelante se explica cómo configurar cada una de las partes para utilizar CDA).

3.11.1 Configuración sin CDA

Cuando no se requiere el uso de CDA, la aplicación ofrece dos alternativas: realizar la autenticación de los usuarios contra la base de datos local propia de la aplicación, o realizarla contra un servidor LDAP corporativo.

Autenticación contra la base de datos local

En el caso de preferir que la autenticación de los usuarios sea realizada contra la base de datos local de la aplicación el proceso es el siguiente: editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml**, buscar el subsistema "**urn:jboss:domain:security:1.1**" y copiar dentro de él el siguiente texto (el texto está disponible en el archivo **securitydomain-sincda-bd.txt** en la carpeta configuration del paquete instalador):

```
<security-domain name="SDSAE" cache-type="default">
  <authentication>
    <login-module code="uy.gub.imm.sae.login.SAEPorEmpresaLoginModule" flag="sufficient"
module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
    </login-module>
    <login-module code="uy.gub.imm.sae.login.SAEAnonimoLoginModule" flag="sufficient"
module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
    </login-module>
  </authentication>
</security-domain>
```

Notar que no se requiere realizar ninguna modificación.

Autenticación contra un servidor LDAP externo

En el caso de contar con un servidor LDAP corporativo y desear que la autenticación de los usuarios sea realizada él el proceso es el siguiente: editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar el subsistema "**urn:jboss:domain:security:1.1**" y copiar dentro de él el siguiente texto (el texto está disponible en el archivo **securitydomain-sincda.txt** en la carpeta configuration del paquete instalador):

```
<security-domain name="SDSAE" cache-type="default">
  <authentication>
    <login-module code="uy.gub.imm.sae.login.SAEPorEmpresaLDAPLoginModule"
flag="sufficient" module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
      <module-option name="ldapHost" value="{host}"/>
      <module-option name="ldapPort" value="{port}"/>
      <module-option name="ldapUser" value="{user}"/>
      <module-option name="ldapPass" value="{pass}"/>
      <module-option name="ldapBase" value="{base}"/>
      <module-option name="ldapAttr" value="{attr}"/>
    </login-module>
    <login-module code="uy.gub.imm.sae.login.SAEAnonimoLoginModule" flag="sufficient"
module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
    </login-module>
  </authentication>
</security-domain>
```

donde {host}, {port} deben ser sustituidos por el nombre o la dirección IP del servidor LDAP, {user} y {pass} deben ser sustituidos por el usuario y la contraseña para establecer una conexión al mismo, {base} debe ser sustituido por el DN de la rama a partir del cual la aplicación debe realizar la búsqueda de usuarios y {attr} debe ser sustituido por el atributo que identifica a los usuarios (debe ser uno de "cn" o "uid" según esté configurado el servidor LDAP). Toda esta información debe ser provista por el administrador del servidor LDAP.

Nota importante: cuando no se utiliza CDA es obligatorio crear la empresa inicial según se explica en la sección 3.4.7. De no hacerlo la aplicación no permitirá realizar el inicio de sesión inicial.

3.11.2 Configuración con CDA

Cuando se requiere configurar CDA el proceso es algo más complejo, requiriendo incluso interacción con AGESIC para solicitar autorización para la conexión al Sistema. El procedimiento para configurar CDA es el siguiente:

1. Obtener un certificado digital para utilizar con el Sistema CDA.
2. Solicitar a AGESIC permiso para realizar la integración con el CDA.
3. Finalizar la configuración de CDA.
4. Configurar Apache HTTPd Server para permitir redirecciones (si aplica).

1 - Obtener el certificado digital

Obtener un certificado digital propio que pueda ser registrado ante AGESIC. Hay dos opciones para obtener este certificado: solicitar uno a una autoridad certificadora habilitada por AGESIC o crear uno "casero" (válido solo para ambientes de testing).

Caso 1: si se cuenta con un certificado digital emitido por una autoridad certificadora (AC) reconocida por AGESIC éste debe ser importado a un keystore llamado **cda-ks.jks** que debe crearse en el directorio **{JBOSS_HOME}/standalone/configuration/** con el siguiente comando:

$> keytool -v -importkeystore -srckeystore **{certificado}** -srcstoretype PKCS12 -destkeystore **{JBOSS_HOME}**/standalone/configuration/cda-ks.jks -deststoretype JKS

remplazando el texto **{certificado}** por la ruta completa al archivo que contiene el certificado digital (usualmente de extensión .p12 o .pfx) y **{JBOSS_HOME}** por la ruta absoluta a donde se encuentra instalado el servidor de aplicaciones JBoss AS. 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 para el nuevo keystore (ingresar la misma, habrá que ingresarla dos veces).

Caso 2: si no se cuenta con un certificado digital emitido por una autoridad certificadora (AC) reconocida por AGESIC y se está trabajando en el ambiente de testing se puede utilizar un certificado "casero", el cual puede generarse ejecutando el siguiente comando:

```
$>keytool -genkeypair -keystore {JBOSS_HOME}/standalone/configuration/cda-ks.jks -alias
saecda -keyalg RSA -keysize 1024 -validity 365
```

remplazando el texto **{JBOSS_HOME}** por la ruta absoluta a donde se encuentra instalado el servidor de aplicaciones JBoss AS. Al ejecutar este comando el programa keytool solicitará primero el ingreso de la contraseña para el nuevo keystore (habrá que ingresarla dos veces) y luego el ingreso de la información que será incluida en el certificado. El único dato obligatorio es el primero ("nombre", donde se sugiere utilizar un texto descriptivo del uso, por ejemplo "AGESIC-SAE-TEST"), el resto de los campos se puede dejar vacíos. 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.

En ambos casos como resultado se debe contar un keystore llamado **cda-ks.jks** ubicada en el directorio **{JBOSS_HOME}/standalone/configuration/**.

Nota: el comando **keytool** forma parte de cualquier entorno de ejecución de Java y se encuentra en el directorio {JAVA_HOME}/bin. 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 ejecutables. En este caso se puede ejecutar el mismo comando especificando la ruta completa al programa keytool o modificar la variable de entorno PATH (tanto en MS Windows como en Linux) para incluir el directorio bin de la instalación del JRE.

Nota: en ambos casos, si antes de ejecutar los siguientes comandos se detecta que ya existe el archivo **{JBOSS_HOME}/standalone/configuration/cda-ks.jks** se debe eliminarlo ya que es posible que entre en conflicto con esta configuración.

2 - Solicitar a AGESIC la integración con CDA

Para solicitar el permiso para la integración con CDA se debe completar un formulario y enviarlo a AGESIC junto con la clave pública del certificado obtenido en el paso anterior. De esta manera AGESIC registrará que dicha clave pública está autorizada para ser usada con CDA y quedará asociada a los datos incluidos en el formulario. A continuación se detalla el procedimiento.

En primer lugar se debe exportar la clave pública del certificado obtenido en el paso anterior para poder enviarla mediante correo electrónico a AGESIC. Esto puede hacerse con el siguiente comando

```
$>keytool -export -keystore {JBOSS_HOME}/standalone/configuration/cda-ks.jks -alias
{alias} -file saecda.cer
```

donde **{JBOSS_HOME}** debe ser sustituido por la ruta absoluta a donde se encuentra instalado el servidor de aplicaciones JBoss AS, y **{alias}** debe ser sustituido por el identificador del certificado que se encuentra dentro del archivo cda-ks.jks. En el caso del certificado casero se empleó el alias "saecda", mientras que en el caso del certificado emitido por una autoridad certificadora debe consultarse con el siguiente comando:

```
$>keytool -list -keystore {JBOSS_HOME}/standalone/configuration/cda-ks.jks
```

Una vez exportada la clave pública del certificado y almacenada en un archivo (saecda.cer) se debe solicitar a AGESIC permiso para realizar la integración con el CDA. Para determinar cómo hacer esto consultar a AGESIC mediante la siguiente dirección de correo electrónico: soporte@agesic.gub.uy indicando el interés por realizar esta integración. A modo de resumen, se deberá completar un formulario que será provisto por AGESIC de la siguiente manera (se debe completar dos formularios, uno para la parte pública de la aplicación y otro para la parte privada):

- Completar la primera parte ("1. Información de la Solicitud") con los datos de la persona y el organismo solicitante.
- Completar la segunda parte ("2. Información técnica del solicitante") de la siguiente manera:
  - **Entity ID**: ingresar el nombre con el cual se desea registrarse ante el CDA de AGESIC. Si bien puede ser cualquier texto, se recomienda que sea un URI para evitar colisiones con otros organismos. Debe utilizarse nombres diferentes para la parte pública y la privada, por ejemplo "**http://cda.agenda.sofis.com.uy/sp/sae**" para la parte pública y "**http://cda.agenda.sofis.com.uy/sp/saea-dmin**" para la parte privada. Ajustar el valor según el nombre del Organismo, teniendo en cuenta que **NO** es una URL real sino simplemente una URI.
  - **Assertion consumer service location**: especificar aquí la URL "de retorno" a la cual el CDA deberá redirigir a los usuarios debidamente autenticados. Esta URL debe conformarse de la siguiente manera: concatenar la **URL base** de la aplicación con el sufijo "**/cda**", resultando, "https://**{host}**:**{port}**/sae/cda" para el caso de la parte pública y "https://**{host}**:**{port}**/sae-admin/cda" (donde **{host}** y **{port}** deben ser sustituidos por el nombre o dirección IP del servidor y el puerto por el que atiende conexiones respectivamente; en el caso de utilizar Apache HTTPd Server corresponde a éste, en otro caso corresponde al JBoss AS).
  - **Assertion consumer service binding**: debe ser el literal "HTTP-POST".
  - **Single logout location**: copiar el texto ingresado en el campo Assertion consumer service location.
  - **Single logout response location**: copiar el texto ingresado en el campo Assertion consumer service location.
  - **Name identifier format**: debe ser el literal "unspecified".
  - **Certificado de firma**: debe ser el texto "Se adjunta certificado autofirmado".

- **Certificado de encriptación**: debe ser el texto “Es el mismo que el certificado de firma”.

Luego de completar ambos formularios enviarlos a AGESIC (a la casilla de correo electrónico **soporte@agesic.gub.uy**) adjuntando en el mensaje el archivo que contiene la clave pública del certificado a utilizar (el archivo **saecda.cer** generado en primer lugar). En el mismo mensaje se debe solicitar a AGESIC que envíe el certificado que contiene la clave pública del Sistema CDA.

Luego de enviar el correo electrónico con la solicitud (conteniendo los dos formularios y el archivo con la clave pública) cabe esperar la respuesta positiva por parte de AGESIC, la cual debe incluir el certificado conteniendo la clave pública del Sistema CDA para poder validar los mensajes.

Al recibir la respuesta de AGESIC, crear un nuevo keystore llamado **cda-ts.jks** en el directorio **{JBOSS_HOME}/standalone/configuration/** (es decir, el mismo lugar donde se creó el archivo **cda-ks.jks**) conteniendo solo el certificado que contiene la clave pública del CDA; esto puede hacerse con el siguiente comando:

```
$>keytool -import -keystore {JBOSS_HOME}/standalone/configuration/cda-ts.jks -file
{archivo}
```

donde **{archivo}** debe sustituirse por el nombre (y si es necesario la ruta completa) del archivo entregado por AGESIC (usualmente de extensión .cer o .crt). El programa keytool solicitará ingresar una nueva contraseña para el keystore creado (será necesaria ingresarla dos veces).

3 - Finalizar la configuración del dominio de seguridad

Para finalizar la configuración se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar el subsistema “**urn:jboss:domain:security:1.1**” y copiar dentro de él el siguiente texto (el texto está disponible en el archivo **securitydomain-concda.txt** en la carpeta configuration del paquete instalador):

```
<security-domain name="SDSAE" cache-type="default">
  <authentication>
    <login-module code="uy.gub.imm.sae.login.SAEPorEmpresaLoginModule" flag="sufficient"
module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
    </login-module>
    <login-module code="uy.gub.imm.sae.login.SAECDALoginModule" flag="sufficient"
module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
      <module-option name="keystorePath"
value="{JBOSS_HOME}/standalone/configuration/cda-ks.jks"/>
      <module-option name="certAlias" value="{cert-alias}"/>
      <module-option name="keystorePass" value="{cert-pass}"/>
    </login-module>
    <login-module code="uy.gub.imm.sae.login.SAEAnonimoLoginModule" flag="sufficient"
module="saelogin">
      <module-option name="dsJndiName" value="java:/postgres-sae-ds"/>
    </login-module>
  </authentication>
</security-domain>
```

donde **{JBOSS_HOME}** debe ser sustituido por la ruta absoluta a donde se encuentra instalado el servidor de aplicaciones JBoss AS, y **{cert-alias}** y **{cert-pass}** deben ser sustituidos por el mismo alias utilizado anteriormente y por la contraseña del certificado usado.

4 - Configuración de Apache HTTPd Server para permitir la redirección automática a CDA

En el caso de que se utilice un servidor Apache HTTPD delante del servidor JBoss AS es necesario garantizar que éste interprete las páginas XHTML como documento HTML y no como documentos XML. Para ello se debe buscar la configuración de los tipos MIME del servidor (que podría estar en el archivo httpd.conf, o en algún otro importado por éste, se reconoce porque es una sección que comienza con el texto "<IfModule mime_module>") y añadir (si no está presente) una línea con el texto "AddType text/html .xhtml", debiendo resultar algo como lo que sigue (solo la línea en negrita fue añadida para esto, las líneas restantes podrían o no estar, pero no deben alterarse excepto que exista otra configuración para ".xhtml"):

```
<IfModule mime_module>
  TypesConfig /etc/mime.types
  AddType application/x-compress .Z
  AddType application/x-gzip .gz .tgz
  AddType text/html .xhtml
  AddType text/html .shtml
  AddOutputFilter INCLUDES .shtml
</IfModule>
```

3.11.3 Limitar el acceso a los componentes de negocio

Ya sea que se utilice CDA o no, es necesario limitar el acceso a los componentes de negocio por fuera de la aplicación. Para hacerlo se debe editar el archivo **{JBOSS_HOME}/standalone/configuration/application-users.properties** y añadir al final del mismo el siguiente texto (el texto está disponible en el archivo **applicationuser.txt** en la carpeta configuration del paquete instalador):

```
anonimo=325621c25511c66dfe9580652de9291b
```

3.12 Configurar el huso horario del sistema

Dado que la aplicación debe soportar múltiples usos horarios en forma simultánea, es necesario que el manejo de fechas internamente sea realizado tomando como referencia el horario estándar (GMT). Para esto, se debe editar el archivo **{JBOSS_HOME}/bin/standalone.conf** (en Linux) o **{JBOSS_HOME}/bin/standalone.conf.bat** (en Windows) y buscar la línea que contiene el siguiente texto: "*if ["x$JAVA_OPTS" = "x"]; then*". Debajo de esa línea deben verse dos (tal vez alguna más o menos) líneas que comienzan con el texto "*JAVA_OPTS="$JAVA_OPTS ...*". Añadir una línea adicional (justo antes de la línea que comienza con el texto '*else*') con lo siguiente (el texto está disponible en el archivo **timezone.txt** en la carpeta configuration del paquete instalador):

```
JAVA_OPTS="$JAVA_OPTS -Duser.timezone=GMT"
```

3.13 Configurar integración con TrámitesUy

Para poder obtener información sobre los trámites que pueden realizarse en el estado la aplicación se integra con la Guía de Trámites del Estado Uruguayo (TrámitesUy) mediante el consumo de servicios web. Para esto, el servidor donde reside la aplicación debe tener acceso a Internet (la integración se realiza mediante servicios web que no están publicados en RedUy sino que están públicos en Internet). Estos servicios web requieren de la especificación de un código de acceso y contraseña. Para realizar la configuración de estos datos se debe modificar las siguientes propiedades en la tabla ae_configuracion en el esquema global de la base de datos:

- **WS_TRAMITE_USER** y **WS_TRAMITE_PASS**: código de acceso y contraseña respectivamente para invocar los servicios web de acceso a los trámites de AGESIC. Estos datos son proporcionados por Agesic, para lo cual se debe solicitarlos a través de la dirección de correo electrónico soporte@agesic.gub.uy.

De fábrica, el paquete instalador está configurado para acceder al ambiente de testing de TrámitesUy. Si se requiriera acceder a otro ambiente, por ejemplo producción, es necesario realizar algunos cambios para configurar el destino:

1. Abrir el archivo **sae-1-service.ear** con un manejador de archivos comprimidos.
2. Dentro de él, abrir el archivo **sae-ejb.jar** con el mismo manejador de archivos comprimidos.
3. Editar el archivo /**uy/gub/imm/sae/business/ws/guiatramites/GuiaTramites.wsdl** y dentro de él cambiar el valor del atributo **location** de la propiedad s**oap:address** para apuntar a la URL correcta del servicio según el ambiente. Esta información debe ser proporcionada por Agesic.
4. Editar el archivo **/uy/gub/imm/sae/business/ws/wstramite/WsTramite.wsdl** y dentro de él cambiar el valor del atributo **location** de la propiedad **soap:address** para apuntar a la URL correcta del servicio según el ambiente. Esta información debe ser proporcionada por Agesic.
5. Asegurarse de que los cambios realzados permanezcan.

3.14 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 editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y 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>
```

3.15 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 3.2.

4 Instalación del paquete con JBoss AS

Esta instalación es más sencilla y rápida que la descrita en la sección anterior, pero sin embargo no proporciona mucha información sobre los componentes que forman parte de la solución y la interacción entre ellos, por lo que en el caso de tener que resolver problemas se contará con menos información.

Para poder realizar este procedimiento se debe contar con los siguientes elementos que deben haber sido provistos por AGESIC:

- Una carpeta llamada "**jboss**", que contiene un servidor JBoss AS preconfigurado (excepto por los detalles propios de la instalación).
- Una carpeta llamada "**app**", que contiene los archivos que conforman la aplicación propiamente dicha.
- Una carpeta llamada "**sql**", que contiene los archivos necesarios para crear la base de datos.
- Una carpeta llamada "**doc**", que contiene la documentación necesaria para realizar la instalación (este documento).

Nota: si no se cuenta con la carpeta "app" seguramente esté trabajando con un paquete anterior a la versión 1.2 de la aplicación; se recomienda solicitar a AGESIC un nueva paquete más actualizado.

Los pasos para realizar la instalación son los siguientes:

4.1 Instalación de Java

Realizar la instalación de Java como se explica en la sección 3.1, "Instalación de Java".

4.2 Instalación de JBoss

Tomar la carpeta jboss que forma parte del paquete instalador y copiarla a la ubicación deseada (esta 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}**".

4.3 Instalación de PostgreSQL

Realizar la instalación de Java como se explica en la sección 3.3, "Instalación de PostgreSQL".

4.4 Crear la base de datos para la aplicación

Realizar el procedimiento descrito en la sección 3.4, "Crear la base de datos para la aplicación" para crear la base de datos, los esquemas iniciales, el primer usuario administrador y la primera empresa.

4.5 Configurar los orígenes de datos para la aplicación

Configurar el servidor de correo electrónico según se explica en la sección 3.6, "Configurar los orígenes de datos para la aplicación", considerando que el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** ya tiene la configuración necesaria, siendo solo necesario remplazar los asteriscos (***) por los valores correspondientes.

4.6 Configurar el servidor de correo electrónico

Configurar el servidor de correo electrónico según se explica en la sección 3.7, “Configurar el servidor de correo electrónico“, considerando que el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** ya tiene la configuración necesaria, siendo solo necesario remplazar los asteriscos (***) por los valores correspondientes.

4.7 Habilitar el acceso seguro (HTTPS)

Habilitar el acceso seguro (HTTPS) según se explica en la sección 3.9, “Habilitar el acceso seguro (HTTPS)“, considerando que el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** ya tiene la configuración necesaria, siendo solo necesario remplazar los asteriscos (***) por los valores correspondientes.

4.8 Mejorar la seguridad del servidor

Mejorar la seguridad del servidor según se explica en la sección 3.10, “Mejorar la seguridad del servidor“. Esto solo aplica si utiliza un servidor Apache HTTPd Server delante ya que la configuración necesaria para el servidor JBoss AS ya está incluida en el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y no requiere modificaciones.

4.9 Configurar el dominio de seguridad

Configurar el dominio de seguridad según se explica en la sección 3.11, “Configurar el dominio de seguridad“, considerando que el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** ya tiene la configuración necesaria, siendo solo necesario remplazar los asteriscos (***) por los valores correspondientes.

4.10 Configurar el huso horario del sistema

Configurar el huso horario del sistema según se explica en la sección 3.12, “Configurar el huso horario del sistema“.

4.11 Configurar integración con TrámitesUy

Configurar integración con TrámitesUy según se explica en la sección 3.12, “Configurar el huso horario del sistema“.

4.12 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.

5 Configuración e instalación de la aplicación

Antes de proceder a instalar la aplicación en el servidor JBoss AS es necesario realizar algunas configuraciones adicionales, tanto a la parte pública como privada, que dependen de si utiliza CDA o no, y si se utiliza un servidor Apache HTTPd Server o no.

5.1 Configuración de la parte privada de la aplicación

5.1.1 Configuración del backend según se usa o no CDA

Configuración cuando no se usa CDA

Cuando no se utiliza CDA es necesario indicarle a la aplicación que se desea utilizar el módulo de autenticación local. Para esto se debe realizar lo siguiente:

- **Deshabilitar la válvula de control de acceso en la aplicación**. Para esto editar el archivo jboss-web.xml que se encuentra en la carpeta WEB-INF dentro del archivo sae-backoffice.war que a su vez se encuentra dentro del archivo sae-2-backoffice.ear y comentar (o quitar) toda la válvula cuya clase es "com.sofis.agesic.sae.cda.CDAServiceProviderValve".
- **Habilitar la solicitud de credenciales mediante una página propia de la aplicación**. Para esto editar el archivo web.xml que se encuentra en la carpeta WEB-INF dentro del archivo sae-backoffice.war que a su vez se encuentra dentro del archivo sae-2-backoffice.ear y buscar la línea con el texto "<auth-method>"; en esta línea asegurarse que el valor de la propiedad sea "FORM".

Configuración cuando se usa CDA

Cuando se utiliza CDA es necesario indicarle a la aplicación que no se desea utilizar el módulo de autenticación local y además configurar los parámetros de acceso al Sistema CDA. Para esto se debe realizar lo siguiente:

- **Habilitar la válvula de control de acceso en la aplicación**. Para esto editar el archivo jboss-web.xml que se encuentra dentro del archivo sae-backoffice.war que a su vez se encuentra en la carpeta WEB-INF dentro del archivo sae-2-backoffice.ear y descomentar toda la válvula cuya clase es "com.sofis.agesic.sae.cda.CDAServiceProviderValve". Luego configurar los parámetros de acceso al CDA en la aplicación de la siguiente manera (los valores no comentados dejarlos en su valor original):
  - **IdpUrlLogin**: indicar la URL a la cual se debe redirigir al usuario cuando se necesita que se autentique. Esta información debe ser provista por AGESIC.
  - **ProviderId**: indicar exactamente el mismo valor que se ingresó en el campo "Entity ID" al completar el formulario en la sección 3.11.2.
  - **SpReturnUrl**: indicar exactamente el mismo valor que se ingresó en el campo "Assertion consumer service location" al completar el formulario en la sección 3.11.2.
  - **ReturnPath**: dejarlo siempre en "/cda".
  - **KeystorePath**: ingresar la ruta absoluta al archivo cda-ks.jks creado en la sección 3.11.2.
  - **keystorePass**: ingresar la cortraseña archivo cda-ks.jks creado en la sección 3.11.2.
  - **CertAlias**: ingresar el alias (nombre) que identifica al certificado a utilizar dentro del keystore creado en la sección 3.11.2.
  - **TruststorePath**: ingresar la ruta absoluta al archivo cda-ts.jks creado en la sección 3.11.2.
  - **keystorePass**: ingresar la cortraseña archivo cda-ts.jks creado en la sección 3.11.2.

- **IdpUrlLogout**: indicar el mismo valor que fue especificado en el atributo IdpUrlLogin.

- **Deshabilitar la solicitud de credenciales mediante una página propia de la aplicación**. Para esto editar el archivo web.xml que se encuentra dentro del archivo sae-backoffice.war que a su vez se encuentra dentro del archivo sae-2-backoffice.ear y buscar la línea con el texto "<auth-method>"; en esta línea asegurarse que el valor de la propiedad sea "NONE".

Nota: recordar que para que un usuario registrado en CDA pueda trabajar con la aplicación debe estar también registrado en la base de datos de la propia aplicación. Para ello, su número de documento (que lo identifica en CDA) debe figurar en el campo "codigo" de la tabla global.ae_usuarios.

5.1.2 Configuración según se utiliza Apache HTTPd Server o no

Configuración cuando no se usa Apache HTTPd Server

- **Habilitar las cabeceras especiales de seguridad**. Para esto editar el archivo jboss-web.xml que se encuentra dentro del archivo sae-backoffice.war que a su vez se encuentra en la carpeta WEB-INF dentro del archivo sae-2-backoffice.ear y descomentar toda la válvula cuya clase es "uy.gub.imm.sae.web.security.SecurityHeadersValve".

Configuración cuando se usa Apache HTTPd Server

- **Deshabilitar las cabeceras especiales de seguridad**. Para esto editar el archivo jboss-web.xml que se encuentra dentro del archivo sae-backoffice.war que a su vez se encuentra en la carpeta WEB-INF dentro del archivo sae-2-backoffice.ear y comentar toda la válvula cuya clase es "uy.gub.imm.sae.web.security.SecurityHeadersValve".

5.2 Configuración de la parte pública de la aplicación

5.2.1 Configuración del frontend según se usa o no CDA

Configuración cuando no se usa CDA

Cuando no se utiliza CDA es necesario indicarle a la aplicación que se desea utilizar el módulo de autenticación local. Para esto se debe realizar lo siguiente:

- **Deshabilitar la válvula de control de acceso en la aplicación**. Para esto editar el archivo jboss-web.xml que se encuentra en la carpeta WEB-INF dentro del archivo sae-frontend.war que a su vez se encuentra dentro del archivo sae-2-frontend.ear y comentar (o quitar) toda la válvula cuya clase es "com.sofis.agesic.sae.cda.CDAServiceProviderValve".

Configuración cuando se usa CDA

Cuando se utiliza CDA es necesario indicarle a la aplicación que no se desea utilizar el módulo de autenticación local y además configurar los parámetros de acceso al Sistema CDA. Para esto se debe realizar lo siguiente:

- **Habilitar la válvula de control de acceso en la aplicación**. Para esto editar el archivo jboss-web.xml que se encuentra dentro del archivo sae-frontend.war que a su vez se encuentra en la carpeta WEB-INF dentro del archivo sae-2-frontend.ear y descomentar toda la válvula cuya clase es "com.sofis.agesic.sae.cda.CDAServiceProviderValve". Luego configurar los parámetros de acceso al CDA en la aplicación de la siguiente manera (los valores no comentados dejarlos en su valor original):

- **IdpUrlLogin**: indicar la URL a la cual se debe redirigir al usuario cuando se necesita que se autentique. Esta información debe ser provista por AGESIC.
- **ProviderId**: indicar exactamente el mismo valor que se ingresó en el campo “Entity ID” al completar el formulario en la sección 3.11.2.
- **SpReturnUrl**: indicar exactamente el mismo valor que se ingresó en el campo “Assertion consumer service location” al completar el formulario en la sección 3.11.2.
- **ReturnPath**: dejarlo siempre en “/cda”.
- **KeystorePath**: ingresar la ruta absoluta al archivo cda-ks.jks creado en la sección 3.11.2.
- **keystorePass**: ingresar la cortraseña archivo cda-ks.jks creado en la sección 3.11.2.
- **CertAlias**: ingresar el alias (nombre) que identifica al certificado a utilizar dentro del keystore creado en la sección 3.11.2.
- **TruststorePath**: ingresar la ruta absoluta al archivo cda-ts.jks creado en la sección 3.11.2.
- **keystorePass**: ingresar la cortraseña archivo cda-ts.jks creado en la sección 3.11.2.
- **IdpUrlLogout**: indicar el mismo valor que fue especificado en el atributo IdpUrlLogin.

- **Deshabilitar la solicitud de credenciales mediante una página propia de la aplicación**. Para esto editar el archivo web.xml que se encuentra dentro del archivo sae-frontend.war que a su vez se encuentra dentro del archivo sae-2-frontend.ear y buscar la línea con el texto “<auth-method>”; en esta línea asegurarse que el valor de la propiedad sea “NONE”.

Nota: recordar que para que un usuario registrado en CDA pueda trabajar con la aplicación debe estar también registrado en la base de datos de la propia aplicación. Para ello, su número de documento (que lo identifica en CDA) debe figurar en el campo “codigo” de la tabla global.ae_usuarios.

5.2.2 Configuración según se utiliza Apache HTTPd Server o no

Configuración cuando no se usa Apache HTTPd Server

- **Habilitar las cabeceras especiales de seguridad**. Para esto editar el archivo jboss-web.xml que se encuentra dentro del archivo sae-frontend.war que a su vez se encuentra en la carpeta WEB-INF dentro del archivo sae-2-frontend.ear y descomentar toda la válvula cuya clase es “uy.gub.imm.sae.web.security.SecurityHeadersValve”.

Configuración cuando se usa Apache HTTPd Server

- **Deshabilitar las cabeceras especiales de seguridad**. Para esto editar el archivo jboss-web.xml que se encuentra dentro del archivo sae-frontend.war que a su vez se encuentra en la carpeta WEB-INF dentro del archivo sae-2-frontend.ear y comentar toda la válvula cuya clase es “uy.gub.imm.sae.web.security.SecurityHeadersValve”.

5.3 Instalar a aplicación en el servidor JBoss AS

Para instalar (“deployar”) la aplicación en el servidor de aplicaciones JBoss AS, solo se debe copiar todos los archivos contenidos por el paquete sae.zip a la carpeta **{JBOSS_HOME}/standalone/deployments** en el siguiente orden:

1. **sae-1-recursos.ear**
2. **sae-1-service.ear**
3. **sae-2-backoffice.ear**
4. **sae-2-frontend.ear**

Tras unos pocos segundos luego de la copia de cada archivo en dicha carpeta aparecerá un nuevo archivo con el mismo nombre que el copiado y extensión ".deployed" indicando que la aplicación se ha instalado correctamente; si no es así, o en su lugar aparece un archivo con extensión ".failed" significa que ocurrió un error y debe examinarse el archivo de registro (**{JBOSS_HOME}/standalone/log/server.log**) para intentar determinar la causa del mismo.

6 Acceso a la aplicación

La aplicación ofrece dos interfaces web diferentes: una para los usuarios administradores (quienes configuran las agendas, los recursos y las disponibilidades) y otra para los usuarios comunes (quienes usan la agenda para realizar reservas).

Para acceder a la interfaz de administración se debe apuntar un navegador web (Internet Explorer, Mozilla Firefox, Google Chrome) a la siguiente URLs:

https://**{host}:{port}**/sae-admin

donde se debe remplazar {host} para dirección IP o el nombre del equipo donde reside el servidor de aplicaciones JBoss o el servidor Apache HTTPd Server (si se utiliza), y {port} por el número de puerto por el cual atiende conexione. Si todo funciona correctamente debería verse la pantalla inicial de la interfaz de administración de la aplicación Agenda, solicitando el nombre de usuario y la contraseña; si no es así, se debe ver el archivo de registro del servidor para intentar detectar posibles errores.

Para acceder a la interfaz de reserva se debe apuntar el navegador web a la siguiente URLs:

https://**{host}:{port}**/sae/agendarReserva/Paso1.xhtml?e={e}&a={a}&r={r}

donde se debe remplazar {host} para dirección IP o el nombre del equipo donde reside el servidor de aplicaciones JBoss AS o el servidor Apache HTTPd Server (si se utiliza), {port} por el número de puerto por el cual atiende conexiones y los parámetros {e}, {a} y {r} por los identificadores de una empresa, una agenda y un recurso respectivamente (para esto será necesario haber definido por lo menos una empresa, dentro de ella una agenda y dentro de ella un recurso). Si todo funciona correctamente debería verse la pantalla inicial de la interfaz de reserva de la aplicación Agenda, lista para comenzar el proceso de reserva.

7 Configuraciones opcionales

7.1 Cambio de puertos del JBoss AS

Por defecto, los puertos usados por JBoss AS para atender conexiones HTTPS es el 8443 (8009 para AJP). Si se desea cambiar esto, se debe hacer lo siguiente:

- Para cambiar el puerto HTTPS: editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar la línea que comienza con "**<socket-binding name="https"**" , remplazando en ella 8443 por el puerto deseado.

- Para cambiar el puerto AJP: editar el archivo **{JBOSS_HOME}/standalone/configuration/standalone.xml** y buscar la línea que comienza con "**<socket-binding name="ajp"**", remplazando en ella 8009 por el puerto deseado.

8 Interacción con el Sistema de Trazabilidad

Cuando ocurren ciertas acciones en la aplicación, las mismas deben ser registradas en el sistema de trazabilidad de Agesic. Esto se realiza mediante la invocación a un par de servicios web expuestos en la Plataforma de Interoperabilidad (PDI) la Plataforma de Gobierno Electrónico (PGE) de Agesic. Dados los requerimientos de seguridad de dicha plataforma debe realizarse el siguiente proceso de configuración:

1. Tramitar los certificados digitales necesarios para invocar servicios web publicados en la PDI. Para esto debe solicitarse instrucciones a Agesic a través de la dirección de correo electrónico soporte@agesic.gub.uy. Como resultado de esta tramitación deberían obtenerse tres juegos de keystores:
   - Un keystore, llamado truststore de SSL conteniendo los certificados correspondientes a la PDI. Este keystore debe contener el certificado que entrega la PDI y opcionalmente uno o más certificados de autoridades certificadoras de confianza.
   - Un keystore, llamado keystore de SSL, conteniendo un certificado utilizado para establecer conexiones SSL con la PDI. Este keystore debe contener el certificado propio que se debe entregar a la PDI cada vez que se establece una conexión. Este certificado es emitido por Agesic tanto en el ambiente de testing como de producción.
   - Un keystore, llamado keystore de organismo, contenido un certificado utilizado para demostrar la identidad del invocante. Este certificado suele ser emitido por Agesic para el ambiente de testing (puede incluso ser el mismo que el utilizado para establecer conexiones SSL), y es emitido por El Correo en el ambiente de producción.
2. Tramitar la solicitud de consumo de los dos servicios web necesarios: Trazabilidad_Cabezal y Trazabilidad_Linea. Para esto debe solicitarse instrucciones a Agesic a través de la dirección de correo electrónico soporte@agesic.gub.uy. Como resultado de esta tramitación debería obtenerse la autorización para consumir ambos servicios web, y el siguiente juego de datos (más abajo se explica la utilidad de cada uno, pero es necesario asegurarse que cuando se reciba la autorización de consumo de os servicios también se reciba esta información:
   - URL del sistema de emisión de tokens SAML.
   - Rol a utilizar.
   - URLs lógicas de cada uno de los servicios web (dos).
   - Tipo de token a solicitar.
3. Modificar los valores de la tabla **ae_configuracion** en el esquema global de la base de datos como se explica a continuación:
   - WS_TRAZABILIDAD_HABILITADO: poner a esta variable el valor "true" si se desea habilitar el sistema de trazabilidad, o "false" en caso contrario.

- WS_TRAZABILIDAD_TIMEOUT: indicar el tiempo máximo (en milisegundos) que puede esperar la aplicación para obtener una respuesta de alguno de los servicios web del Sistema de Trazabilidad. Se recomienda no especificar un tiempo muy largo debido a que el usuario de la aplicación tendría que esperar, en el peor de los casos, este tiempo antes de obtener una respuesta del servidor.
- WS_TRAZABILIDAD_MAXINTENTOS: cantidad máxima de veces que se intentará enviar cada traza si se obtiene un error.
- WS_TRAZABILIDAD_VERSION: indicar el número de versión del servicio web que se está invocando. Este valor debió haber sido informado por Agesic, pero por el momento es seguro poner el valor “101”.
- WS_TRAZABILIDAD_URLSTS: indicar la URL del servicio de emisión de Tokens SAML de la PDI. Este dato debe ser provisto por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_ROL: indicar el rol a usar. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_WSAACTION_CABEZAL: indicar la URL lógica (WSA-TO) correspondiente al servicio web Cabezal. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_WSAACTION_LINEA: indicar la URL lógica (WSA-TO) correspondiente al servicio web Línea. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_WSATO_CABEZAL: indicar la operación (WSA-ACTION) correspondiente al servicio web Cabezal. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_WSATO_LINEA: indicar la operación (WSA-ACTION) correspondiente al servicio web Línea. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_POLICY: indicar el tipo de token a utilizar. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo de los servicios web.
- WS_TRAZABILIDAD_ORG_KS_PATH: indicar la URL absoluta al keystore de organismo mencionado en el punto 1.
- WS_TRAZABILIDAD_ORG_KS_PASS: indicar la contraseña del keystore de organismo.
- WS_TRAZABILIDAD_ORG_KS_ALIAS: indicar el alias correspondiente al certificado a utilizar dentro del keystore de organismo.
- WS_TRAZABILIDAD_SSL_KS_PATH: indicar la URL absoluta al keystore de SSL mencionado en el punto 1.
- WS_TRAZABILIDAD_SSL_KS_PASS: indicar la contraseña del keystore de SSL.
- WS_TRAZABILIDAD_SSL_KS_ALIAS: indicar el alias correspondiente al certificado a utilizar dentro del keystore de SSL.
- WS_TRAZABILIDAD_SSL_TS_PATH: indicar la URL absoluta al truststore de SSL mencionado en el punto 1.
- WS_TRAZABILIDAD_SSL_TS_PASS: indicar la contraseña del truststore de SSL.

De fábrica, el paquete instalador está configurado para acceder al ambiente de testing de Trazabilidad. Si se requiriera acceder a otro ambiente, por ejemplo producción, es necesario realizar algunos cambios para configurar el destino:

1. Abrir el archivo **sae-1-service.ear** con un manejador de archivos comprimidos.
2. Dentro de él, abrir el archivo **sae-ejb.jar** con el mismo manejador de archivos comprimidos.
3. Editar el archivo **/uy/gub/agesic/itramites/bruto/web/ws/cabezal/CabezalService.wsdl** y dentro de él cambiar el valor del atributo **location** de la propiedad s**oap:address** para apuntar a la URL correcta del servicio según el ambiente. Esta información debe ser proporcionada por Agesic.
4. Editar el archivo **/uy/gub/agesic/itramites/bruto/web/ws/linea/LineaService.wsdl** y dentro de él cambiar el valor del atributo **location** de la propiedad **soap:address** para apuntar a la URL correcta del servicio según el ambiente. Esta información debe ser proporcionada por Agesic.
5. Asegurarse de que los cambios realzados permanezcan.
6. Reinstalar la aplicación (en realidad, solo es necesario volver a instalar el módulo **sae-1-service.ear**.
7. Reiniciar el servidor de aplicaciones.

8.1 Habilitar y deshabilitar la integración con el Sistema de Trazabilidad

La habilitación o deshabilitación de la integración con el Sistema de Trazabilidad se puede realizar a nivel global (para todo el sistema) modificando la propiedad WS_TRAZABILIDAD_HABILITADO, aunque requiere el reinicio del servidor de aplicaciones debido a que la tabla configuración solo se accede durante el arranque.

Nota: también puede habilitarse o deshabilitarse la integración con trazabilidad a nivel de Agenda (trámite) aunque esto corresponde a una configuración operativa. Es importante tener en cuenta que aunque se habilite la integración con el Sistema de Trazabilidad a nivel de Agenda si está deshabilitado a nivel global NO se realizará el registro de trazas.

9 Interacción con el Sistema de Notificación de Novedades (Publish&Subscribe)

Al igual que la aplicación permite la integración con el Sistema de Trazabilidad también permite la integración con el sistema de Notificación de Novedades (también llamado Publish and Suscribe) de la Plataforma de Gobierno Electrónico. Con esto, cada vez que ocurre algún evento relevante (cuando un ciudadano confirma una reserva, cuando cancela una reserva, o cuando un funcionario marca que un ciudadano fue atendido o no se presentó al ser llamado) se registra el evento en el mencionado sistema. Esto se realiza mediante la invocación de un servicio web expuesto en la Plataforma de Interoperabilidad (PDI). Dados los requerimientos de seguridad de dicha plataforma debe realizarse el siguiente proceso de configuración (**nota importante**: los pasos 1 y 2 pueden no hacerse si ya se hicieron para la integración con el Sistema de Trazabilidad ya que pueden usarse los mismos certificados digitales) :

1. Tramitar los certificados digitales necesarios para invocar servicios web publicados en la PDI. Para esto debe solicitarse instrucciones a Agesic a través de la dirección de correo electrónico soporte@agesic.gub.uy. Como resultado de esta tramitación deberían obtenerse tres juegos de keystores:

- Un keystore, llamado truststore de SSL conteniendo los certificados correspondientes a la PDI. Este keystore debe contener el certificado que entrega la PDI y opcionalmente uno o más certificados de autoridades certificadoras de confianza.
- Un keystore, llamado keystore de SSL, conteniendo un certificado utilizado para establecer conexiones SSL con la PDI. Este keystore debe contener el certificado propio que se debe entregar a la PDI cada vez que se establece una conexión. Este certificado es emitido por Agesic tanto en el ambiente de testing como de producción.
- Un keystore, llamado keystore de organismo, contenido un certificado utilizado para demostrar la identidad del invocante. Este certificado suele ser emitido por Agesic para el ambiente de testing (puede incluso ser el mismo que el utilizado para establecer conexiones SSL), y es emitido por El Correo en el ambiente de producción.

2. Tramitar la solicitud de consumo del servicio de Notificaciones indicando que se trata del tópico “SAENovedades”. Para esto debe solicitarse instrucciones a Agesic a través de la dirección de correo electrónico soporte@agesic.gub.uy, indicando expresamente que se trata del sistema de Notificaciones y no de un servicio web común. Como resultado de esta tramitación debería obtenerse la autorización para consumir ambos servicios web, y el siguiente juego de datos (más abajo se explica la utilidad de cada uno, pero es necesario asegurarse que cuando se reciba la autorización de consumo de os servicios también se reciba esta información:
   - URL del sistema de emisión de tokens SAML.
   - Rol a utilizar.
   - URLs lógicas de cada uno de los servicios web (dos).
   - Tipo de token a solicitar.
   - Nombre del productor de mensajes.
3. Modificar los valores de la tabla ae_configuracion en el esquema global de la base de datos como se explica a continuación:
   - WS_NOVEDADES_HABILITADO: poner a esta variable el valor “true” si se desea habilitar el sistema de notificaciones, o “false” en caso contrario.
   - WS_NOVEDADES_TIMEOUT: indicar el tiempo máximo (en milisegundos) que puede esperar la aplicación para obtener una respuesta del servicio web.
   - WS_NOVEDADES_MAXINTENTOS: cantidad máxima de veces que se intentará enviar la notificación si se obtiene un error.
   - WS_NOVEDADES_URLSTS: indicar la URL del servicio de emisión de Tokens SAML de la PDI. Este dato debe ser provisto por Agesic en el momento de autorizar el consumo del servicio web.
   - WS_NOVEDADES_ROL: indicar el rol a usar. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo del servicio web.
   - WS_NOVEDADES_TOPICO: indicar el nombre “SAENovedades” excepto que AGESIC indique otra cosa al autorizar el consumo del servicio web.

- WS_NOVEDADES_PRODUCTOR: indicar el nombre del productor de novedades a usar. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo del servicio web.
- WS_NOVEDADES_WSAACTION: indicar la URL lógica (WSA-TO) correspondiente al servicio web. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo del servicio web.
- WS_NOVEDADES_WSATO: indicar la operación (WSA-ACTION) correspondiente al servicio web. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo del servicio web.
- WS_NOVEDADES_POLICY: indicar el tipo de token a utilizar. Este dato debió haber sido informado por Agesic en el momento de autorizar el consumo del servicio web.
- WS_NOVEDADES_ORG_KS_PATH: indicar la URL absoluta al keystore de organismo mencionado en el punto 1.
- WS_NOVEDADES_ORG_KS_PASS: indicar la contraseña del keystore de organismo.
- WS_NOVEDADES_ORG_KS_ALIAS: indicar el alias correspondiente al certificado a utilizar dentro del keystore de organismo.
- WS_NOVEDADES_SSL_KS_PATH: indicar la URL absoluta al keystore de SSL mencionado en el punto 1.
- WS_NOVEDADES_SSL_KS_PASS: indicar la contraseña del keystore de SSL.
- WS_NOVEDADES_SSL_KS_ALIAS: indicar el alias correspondiente al certificado a utilizar dentro del keystore de SSL.
- WS_NOVEDADES_SSL_TS_PATH: indicar la URL absoluta al truststore de SSL mencionado en el punto 1.
- WS_NOVEDADES_SSL_TS_PASS: indicar la contraseña del truststore de SSL.

9.1 Habilitar y deshabilitar la integración con el Sistema de Trazabilidad

La habilitación o deshabilitación de la integración con el Sistema de Notificaciones se puede realizar a nivel global (para todo el sistema) modificando la propiedad WS_NOVEDADES_HABILITADO, aunque requiere el reinicio del servidor de aplicaciones debido a que la tabla configuración solo se accede durante el arranque.

Nota: también puede habilitarse o deshabilitarse la integración con notificaciones a nivel de Agenda (trámite) aunque esto corresponde a una configuración operativa. Es importante tener en cuenta que aunque se habilite la integración con el Sistema de Notificaciones a nivel de Agenda si está deshabilitado a nivel global NO se realizará el registro de novedades.

10 Configuración de las agendas para utilizar control de acceso

En el caso de que se utilice CDA para la parte pública, cada agenda puede ser configurada en forma independiente para requerir control de acceso o no; por omisión se define que todas las agendas **no** requieren control de acceso. Para requerir control de acceso para una agenda particular se debe hacer lo siguiente:

1. Acceder a la interfaz de administración de la aplicación. Esto debe hacerse con un usuario superadministrador o con un usuario con el rol ADMINISTRADOR.

2. Seleccionar, en la lista de la parte superior derecha, la empresa a la cual pertenece la agenda que se quiere limitar el acceso.
3. En el menú de la derecha desplegar la rama “Agendas” y seleccionar “Consultar agendas”.
4. Seleccionar en el listado la empresa a la cual se desea limitar el acceso y hacer clic en el ícono de edición de la misma.
5. Marcar la casilla “Requiere Control de acceso”.
6. Hacer clic en el botón “Guardar”.

11 Adaptación y traducción de textos

Todos los textos que forman parte de la aplicación puede ser modificados y/o traducidos a otros idiomas sin necesidad de modificar la propia aplicación. A continuación se explica cómo hacer esto.

11.1 Adaptación de los textos

Todos los textos que forman parte de la aplicación (es decir, los que no son especificados por los usuarios, sino los “fijos”, como las etiquetas de campos) se encuentran almacenados en la tabla **ae_textos** del esquema **global** en la base de datos. Esta tabla contiene dos campos: **codigo** y **texto**. Cualquiera de los textos que se encuentran en dicha tabla puede ser modificado para adaptarlo a las necesidades; para ello se debe modificar solo los datos de la columna texto, sin modificar los datos de la columna codigo.

Se debe tener en cuenta que al modificar los textos almacenados en la tabla **ae_textos** del esquema **global** se afecta a toda la instalación de la aplicación, es decir a todas las empresas y a todas las agendas y recursos de ellas. Si solo se desea cambiar los textos para una empresa particular, sin afectar al resto, se debe realizar una traducción de los textos, como se explica en la siguiente sección, indicando que el idioma a traducir es también español.

11.2 Traducción de los textos a otros idiomas

La aplicación está preparada para trabajar en múltiples idiomas. Esto significa que, de estar disponibles las traducciones, los usuarios del módulo público y del privado pueden seleccionar el idioma en el cual desean visualizar la interfaz. La configuración debe hacerse para cada empresa en forma individual; esto permite que dos empresas admitan diferentes idiomas, e incluso que dos empresas tengan diferentes traducciones para el mismo texto en el mismo idioma. Para poder aprovechar esta característica de la aplicación se debe realizar tres cosas:

1. Traducir todos los textos al idioma deseado.
2. Habilitar el nuevo idioma para ser utilizado en la aplicación.
3. Configurar cada una de las agendas para permitir que los usuarios utilicen el idioma.

A continuación se explica cada uno de estas actividades.

11.2.1 Traducir todos los textos

De fábrica, la aplicación incluye los textos solo en el idioma español. Para traducir estos textos a otro idioma (o para modificar los textos en español) para una empresa particular hay que hacer lo siguiente:

1. Tomar como base todos los textos que se encuentran en la tabla **ae_textos** del esquema **global** en la base de datos.

2. Copiar los valores tomados en el punto 1 a la tabla **ae_textos** del esquema correspondiente a la empresa. Esta tabla tiene los mismos dos campos que la anterior (**codigo** y **texto**) más un campo **idioma** en el cual se debe especificar el código ISO de dos letras al cual corresponde la traducción (por ejemplo, “es” para español, “en” para inglés, “fr” para francés y “pt” para portugués). Notar que se puede especificar que el idioma es español, lo que permite modificar los textos en español para una empresa específica, sin afectar a otras empresas.
3. Modificar los valores de la columna **texto** con la traducción correspondiente al valor original. No modificar los valores de la columna **codigo**. Por ejemplo, si se quiere traducir al inglés el texto “Acepto los términos” debe hacerse lo siguiente:

   3.1. Buscar en la tabla ae_textos del esquema global, en la columna texto, el valor “Acepto los términos”; se debería encontrar lo siguiente:

   - codigo: “acepto_los_terminos”
   - texto: “Acepto los términos”

   1.2. Copiar los valores anteriores en la tabla ae_textos del esquema correspondiente a la empresa, especificando además en la columna idioma el valor “en”; se debería lograr lo siguiente:

   - codigo: “acepto_los_terminos”
   - texto: “I agree with the terms”
   - idioma: “en”

   Notar que no se modifica el campo codigo.

Si el usuario del a aplicación (por ejemplo, en la parte pública para realizar la reserva) selecciona un cierto idioma, por ejemplo inglés, y algunos de los textos de la aplicación no han sido traducidos entonces dichos textos serán desplegados en el idioma español.

11.2.2 Habilitar el nuevo idioma

El hecho de traducir los textos de la aplicación a otro idioma no implica que dicho idioma quede automáticamente disponible para ser usado. Para hacer disponible un nuevo idioma se debe añadir el código ISO de dos letras de dicho idioma a la lista de valores de la propiedad “IDIOMAS_SOPORTADOS” de la tabla ae_configuracion en el esquema global de la base de datos, separando con una coma cada valor posible. Por ejemplo, para permitir los idiomas español, inglés y portugués en la aplicación (para todas las empresas) se debería ejecutar la siguiente consulta SQL en la base de datos:

```
UPDATE global.ae_configuracion SET valor = 'es,en,pt' WHERE clave =
'IDIOMAS_SOPORTADOS';
```

11.2.3 Configurar los idiomas soportados por una agenda

Los idiomas soportados por una agenda particular es una configuración propia de dicha agenda. Los usuarios administradores de dicha agenda (y los superadministradores) pueden seleccionar los idiomas que la agenda soporta de entre los idiomas habilitados en la aplicación (según se explicó en la sección 11.2.2). Para hacer esto, deben acceder a la página de configuración de la agenda y en el campo Idiomas soportados marcas las casillas correspondientes a los idiomas deseados.

Nota: los idiomas disponibles son los que el administrador de la instalación de la aplicación configuró como disponibles.

11.3 Gestión de preguntas de captcha

La reserva pública requiere que como último paso el ciudadano que realiza la reserva responda una pregunta de captcha, a los efectos de prevenir la generación automatizada de reservas. Estas preguntas son configurables y propias de cada empresa (diferentes empresas pueden contar con diferentes preguntas).

La gestión de las preguntas de captcha se debe hacer directamente en la tabla ae_preguntas_captcha del esquema correspondiente a la empresa para la cual se desea gestionar las preguntas. En esta tabla debe insertarse una fila por cada pregunta, indicando los siguientes campos:

- **Clave**: un valor cualquiera que identifica a la pregunta y que no puede repetirse en la misma tabla.
- **Pregunta**: texto de la pregunta, tal cual será desplegado al ciudadano que realiza la reserva.
- **Idioma**: código ISO de dos letras correspondiente al idioma en el cual está expresada la pregunta; para un ciudadano solo se considerarán las preguntas que están expresadas en el idioma en el cual está trabajando dicho ciudadano.
- **Respuesta**: respuesta que debe dar el ciudadano; el sistema verificará que lo que ingrese el ciudadano sea exactamente el valor especificado en este campo (sin considerar mayúsculas o minúsculas), por lo que se sugiere evitar ambigüedades tales como el uso de números (que podrían ser escritos como texto o dígitos), espacios (la respuesta debería ser una sola palabra), caracteres especiales (tildes, signos de puntuación), etc.

12 Procedimientos de respaldo y recuperación

En esta sección se describe el procedimiento de respaldo y recuperación de la aplicación.

12.1 Respaldo

La aplicación almacena toda la información que gestiona en la base de datos (en múltiples esquemas), no dejando nada a nivel del sistema de archivos. Sin embargo, para funcionar sí requiere de otros archivos que son generados por el usuario que realiza la instalación y configuración inicial de la aplicación que también deben respaldarse.

Existen dos formas de hacer un respaldo: total o parcial. A continuación se describen ambas.

12.1.1 Respaldo total

El respaldo total es el recomendado ya que es más seguro y hace la recuperación mucho más sencilla, aunque puede tomar más tiempo hacerlo y más espacio almacenarla. Consiste en lo siguiente;

- Hacer un respaldo de la base de datos completa (debe incluir todos los esquemas existentes) para lo cual debe usarse la herramienta que se considere más apropiada para hacer respaldos de PostgreSQL.
- Hacer un respaldo del servidor de aplicaciones JBoss AS completo, para lo cual debe hacerse un archivo comprimido (.zip, .tar.gz, .bzip o el formato que se prefiera) de la carpeta {JBOSS_HOME}. Se recomienda eliminar previamente el contenido de las carpetas data, log y tmp ya que no serán útiles en caso de realizar una restauración. Nota importante: antes de hacer el respaldo del servidor de aplicaciones es

recomendable detenerlo por completo con el fin de asegurarse de que ningún archivo quede almacenado en forma parcial.

12.1.2 Respaldo parcial

El respaldo parcial no se recomienda si es viable hacer un respaldo total, aunque puede ser útil en los casos en los cuales se tiene más de una instalación similar con pequeñas variantes. Las cosas que deben respaldarse en este caso son las siguientes:

- La base de datos. Debe incluir todos los esquemas existentes en la base de datos para lo cual debe usarse la herramienta que se considere más apropiada para hacer respaldos de PostgreSQL.
- Keystores con certificados digitales. Esto corresponde a los archivos creados en los puntos 1 y 2 de las secciones 5 y 6 de este documento, y también a los siguientes archivos que fueron creados en el directorio {JBOSS_HOME}/standalone/configuration: sae.jks, cda-ks.jks y cda-ts.jks.
- Se recomienda también guardar una copia de los archivos que se mencionan a continuación; estos archivos no serán usados por sí mismos, pero pueden ser utilizados para abreviar el tiempo de recuperación en caso de un incidente (a los efectos de copiar los valores originales):
  - {JBOSS_HOME}/standalone/configuration/standalone.xml.
  - {JBOSS_HOME}/bin/standalone.conf (en Linux) o {JBOSS_HOME}/bin/standalone.conf.bat (en Windows).

12.2 Recuperación

La recuperación de la aplicación en caso de un incidente depende del mecanismo utilizado para hacer el respaldo (total o parcial), aunque en ambos casos debe comenzarse por restaurar el respaldo de la base de datos utilizano la herramienta que se considere más apropiada. A continuación se describe el procedimiento en cada caso.

12.2.1 Recuperación de un respaldo total

En el caso de que se desee recuperar un respaldo total debe realizarse lo siguiente:

1. Restaurar la base de datos utilizando el respaldo hecho anteriormente, para lo cual debe utilizarse la herramienta que se considere más apropiada.
2. Restaurar el servidor de aplicaciones respaldado anteriormente, para lo cual basta con descromprimir el zip hecho anteriormente en su ubicación original (en caso de utilizar otra ubicación diferente tal vez sea necesario modificar las rutas que se describen a lo largo del documento para reflejar este cambio).

12.2.2 Recuperación de un respaldo parcial

En el caso de que se desee recuperar un respaldo parcial debe realizarse lo siguiente:

1. Restaurar la base de datos utilizando el respaldo hecho anteriormente, para lo cual debe utilizarse la herramienta que se considere más apropiada.
2. Realizar el procedimiento de instalación, tal si fuera una instalación nueva, solo que donde se indique crear archivos nuevos (como certificados) o configuraciones deben utilizarse los archivos respaldados anteriormente.

13 Apéndice 1: Incorporación de acciones y validaciones personalizadas

En la versión 1.4 de la aplicación se reincorporaron dos funcionalidades que estaban originalmente en la aplicación SAE tomada como base pero que no se habían migrado a la nueva tecnología; estas funcionalidades son las que permiten incorporar a la aplicación validaciones y acciones personalizadas.

13.1 Validaciones personalizadas

13.1.1 Sobre las validaciones

La aplicación SAE permite definir los datos que se le solicitarán a los ciudadanos al momento de realizar una reserva. Para cada dato el usuario que realiza la configuración debe indicar el nombre y el tipo de dato (texto, número, fecha o sí/no) y también debe indicar si el dato es requerido (debe ser completado obligatoriamente por el ciudadano) o no. Cuando se crea un recurso automáticamente se crean tres datos a solicitar que no pueden ser eliminados: el tipo de documento, el número de documento y el correo electrónico.

Cuando el ciudadano realiza una reserva la aplicación SAE hace validaciones básicas sobre los datos ingresados:

- Verifica que todos los datos requeridos tengan un valor asignado.
- Verifica que el valor ingresado para cada campo coincida con el tipo de datos definido.
- En el caso de que el tipo de documento sea “CI” (cédula de identidad) verifica que el número de documento sea un número de cédula válido (debe validar el dígito verificador).
- En el caso del correo electrónico verifica que tenga un formato correcto.

A partir de la versión 1.4 la aplicación SAE permite incorporar validaciones personalizadas. En el caso de que alguna de las validaciones personalizadas falle (según el criterio establecido por la validación) la aplicación no permitirá continuar con el proceso de reserva.

Para crear una validación personalizada se debe programar un EJB (Enterprise Java Bean) que implemente una interfaz definida, empaquetarlo en un archivo EAR (Enterprise Application) e instalarlo en el servidor de aplicaciones JavaEE (este procedimiento es explicado a continuación). Luego un usuario administrador en la aplicación deberá registrar la validación y asociarla a los recursos en los cuales desee aplicarla (esto se explica en el manual de usuario).

13.1.2 Creación de una validación personalizada

Nota importante: el procedimiento descrito en esta sección requiere de conocimientos de programación en el lenguaje Java, y particularmente en la tecnología JavaEE, relativamente avanzados.

A continuación se describe el procedimiento para crear una validación personalizada utilizando el entorno de desarrollo Eclipse (con otros entornos de desarrollo, como Netbeans o IntelliJ, no debería ser muy diferente el procedimiento):

1. Crear un proyecto de tipo EJB. No crear un proyecto EAR aún.
2. Añadir al class-path del proyecto el archivo sae-validaciones-ejbclient.jar (este archivo puede obtenerse haciendo un build del proyecto SAE-Validaciones-EJBClient).

3. Programar en el proyecto un Stateless Session Bean que implemente además la interfaz uy.gub.imm.sae.validaciones.business.ejb.ValidadorReservaRemote. Esto exigirá que se implemente el método public ResultadoValidacion validarDatosReserva(String nombreValidacion, Map<String, Object> params), donde el mapa params contiene todos los parámetros que sean configurados por el usuario administrador que registre la validación en la aplicación (ver el manual de usuario).
4. Crear un proyecto de tipo EAR conteniendo únicamente al proyecto EJB creado en el punto 1.
5. Crear una carpeta con el nombre “lib” dentro de la carpeta EarContent del proyecto EAR y colocar en ella a los archivos sae-ejb-client.jar (este archivo puede obtenerse haciendo un build del proyecto SAE-EJBClient) y sae-validaciones-ejbclient.jar (el mismo usado en el punto 2).
6. Generar el archivo EAR. Debería obtenerse un archivo con extensión .ear conteniendo un archivo con extensión .jar (el proyecto EJB creado en el punto 1) y una carpeta llamada lib (con los dos archivos mencionados en el punto 5).
7. Deployar el EAR en el servidor de aplicaciones, copiando el archivo con extensión .ear a la carpeta standalone/deployments del servidor de aplicaciones JBoss AS. Observar el log hasta ver el mensaje «*2) JBAS018559: Deployed* "*<nombre>*"» donde <nombre> corresponde al nombre del archivo EAR generado en el punto 6. Una vez que se observa este mensaje la validación está lista para ser incorporada a la aplicación, según se explica en el manual de usuario.
8. Comunicar al administrador de la aplicación el nombre JNDI asignado por el servidor de aplicaciones JBoss al EJB que implementa la validación; este nombre se determina de la siguiente manera: “java:global/<proyecto-ear>/<proyecto-ejb>/<nombre-ejb>”, donde “<proyecto-ear>” es el nombre asignado al proyecto EAR creado en el punto 4, “<proyecto-ejb>” es el nombre asignado al proyecto EJB creado en el punto 1 y “<nombre-ejb>” es el nombre asignado al Stateless Session Bean creado en el punto 2. Este nombre JNDI es el que deberá especificar el administrador de la aplicación en el campo “Servicio” al momento de registrar la validación.

Nota: pueden crearse múltiples validaciones dentro del mismo proyecto EJB, siempre que tengan nombres diferentes.

13.2 Acciones personalizadas

13.2.1 Sobre las acciones

La aplicación SAE permite integrar a las agendas con el Sistema de Trazabilidad y con el Sistema de Notificaciones. Si una agenda está integrada con el Sistema de Trazabilidad, cuando un ciudadano confirma o cancela una reserva la aplicación notifica al Sistema de Trazabilidad del hecho. Algo análogo ocurre si la agenda está integrada con el Sistema de Notificaciones.

A partir de la versión 1.4 la aplicación SAE también permite incorporar acciones personalizadas que pueden ser ejecutadas cuando el ciudadano confirma o cancela una reserva. En el caso de que una acción personalizada no pueda ejecutarse correctamente la aplicación no permitirá continuar con el proceso de reserva.

Para crear una acción personalizada se debe programar un EJB (Enterprise Java Bean) que implemente una interfaz definida, empaquetarlo en un archivo EAR (Enterprise Application) e instalarlo en el servidor de aplicaciones JavaEE (este procedimiento es explicado a continuación). Luego un usuario administrador en la aplicación deberá registrar la acción y asociarla a los recursos en los cuales desee aplicarla (esto se explica en el manual de usuario).

13.2.2 Creación de una acción personalizada

Nota importante: el procedimiento descrito en esta sección requiere de conocimientos de programación en el lenguaje Java, y particularmente en la tecnología JavaEE, relativamente avanzados.

A continuación se describe el procedimiento para crear una acción personalizada utilizando el entorno de desarrollo Eclipse (con otros entornos de desarrollo, como Netbeans o IntelliJ, no debería ser muy diferente el procedimiento):

1. Crear un proyecto de tipo EJB. No crear un proyecto EAR aún.
2. Añadir al class-path del proyecto el archivo sae-acciones-ejbclient.jar (este archivo puede obtenerse haciendo un build del proyecto SAE-Acciones-EJBClient).
3. Programar en el proyecto un Stateless Session Bean que implemente además la interfaz uy.gub.sae.acciones.business.ejb.EjecutorAccionRemote. Esto exigirá que se implemente el método public ResultadoAccion ejecutar(String nombreAccion, Map<String, Object> params), donde el mapa params contiene todos los parámetros que sean configurados por el usuario administrador que registre la acción en la aplicación (ver el manual de usuario).
4. Crear un proyecto de tipo EAR conteniendo únicamente al proyecto creado en el punto 1.
5. Crear una carpeta con el nombre “lib” dentro de la carpeta EarContent del proyecto EAR y colocar en ella a los archivos sae-ejb-client.jar (este archivo puede obtenerse haciendo un build del proyecto SAE-EJBClient) y sae-acciones-ejbclient.jar (el mismo usado en el punto 2).
6. Generar el archivo EAR. Debería obtenerse un archivo con extensión .ear conteniendo un archivo con extensión .jar (el proyecto EJB creado en el punto 1) y una carpeta llamada lib (con los dos archivos mencionados en el punto 5).
7. Deployar el EAR en el servidor de aplicaciones, copiando el archivo con extensión .ear a la carpeta standalone/deployments del servidor de aplicaciones JBoss AS. Observar el log hasta ver el mensaje «*2) JBAS018559: Deployed* "*<nombre>*"» donde <nombre> corresponde al nombre del archivo EAR generado en el punto 6. Una vez que se observa este mensaje la acción está lista para ser incorporada a la aplicación, según se explica en el manual de usuario.
8. Comunicar al administrador de la aplicación el nombre JNDI asignado por el servidor de aplicaciones JBoss al EJB que implementa la acción; este nombre se determina de la siguiente manera: “java:global/<proyecto-ear>/<proyecto-ejb>/<nombre-ejb>”, donde “<proyecto-ear>” es el nombre asignado al proyecto EAR creado en el punto 4, “<proyecto-ejb>” es el nombre asignado al proyecto EJB creado en el punto 1 y “<nombre-ejb>” es el nombre asignado al Stateless Session Bean creado en el punto 2. Este nombre JNDI es el que deberá especificar el administrador de la aplicación en el campo “Servicio” al momento de registrar la acción.

Nota: pueden crearse múltiples acciones dentro del mismo proyecto EJB, siempre que tengan nombres diferentes.

14 Actualización de versiones

La aplicación SAE es un software en constante desarrollo. Cada vez que se libera una nueva versión, se lo hace de dos dos maneras: como paquetes para realizar una instalación de cero, y como patches para migrar de la versión anterior. En esta sección se describe el procedimiento para migrar de una versión a la siguiente.

14.1 Archivos necesarios para realizar la migración

Para realizar la migración de una versión a la siguiente se debe contar con los siguientes archivos:

- Archivo “migracion_de_vX_a_vY.txt”, donde X e Y corresponden al número de versión que se debe tener actualmente instalado y la versión con la que se acabará luego de completar el procedimiento de migración. Este archivo contiene instrucciones puntuales para migrar de la versión X a la versión Y.
- Archivo “migracion_de_vX_a_vY-global.sql”, donde X e Y corresponden al número de versión que se debe tener actualmente instalado y la versión con la que se acabará luego de completar el procedimiento de migración. Este archivo contiene cambios que deben hacerse al esquema global de la base de datos.
- Archivo “migracion_de_vX_a_vY-esquema.4.sql”, donde X e Y corresponden al número de versión que se debe tener actualmente instalado y la versión con la que se acabará luego de completar el procedimiento de migración. Este archivo contiene cambios que deben hacerse a cada uno de los esquemas de la base de datos correspondientes a empresas.
- Archivos sae-1-recursos.ear, sae-1-service.ear, sae-2-backoffice.ear y sae-2-frontend.ear. Estos archivos corresponden a la versión nueva de la aplicación.

Nota importante: el archivo “migracion_de_vX_a_vY.txt” podría hacer referencia a otros archivos necesarios para realizar la migración en casos particulares.

14.2 Procedimiento de migración de una versión a la siguiente

El procedimiento que se describe en esta sección es sumamente genérico, y cada migración particular podría requerir cosas específicas de la versión. Siempre debe consultarse el archivo “migracion_de_vX_a_vY.txt” por instrucciones precisas relativas a la versión a la que se está migrando.

Para migrar de una versión de la aplicación a la siguiente debe realizarse el siguiente procedimiento:

1. Detener el servidor de aplicaciones JBoss. Si hay más de un servidor de aplicaciones utilizando la misma base de datos se debe detener todos ellos.
2. Aplicar en el esquema global de la base de datos los cambios indicados en el archivo “migracion_de_vX_a_vY-global.sql”.
3. Aplicar en cada uno de los esquemas de la base de datos correspondientes a empresas los cambios indicados en el archivo “migracion_de_vX_a_vY-esquema.sql”, teniendo en cuenta que es necesario remplazar, en dicho archivo, el texto “{esquema}” por el nombre del esquema correspondiente; este procedimiento debe repetirse en todos los esquemas que son usados por empresas.

4. Hacer una copia de seguridad de los cuatro archivos EAR que están actualmente deployados en el servidor de aplicaciones JBoss AS y que corresponden a la versión desde la cual se está migrando.
5. Extraer del paquete sae-2-backoffice.ear actualmente instalado en el servidor de aplicaciones JBoss el archivo jboss-web.xml que se encuentra dentro del archivo sae-backoffice.war, en la carpeta WEB-INF y copiarlo en la misma ubicación del paquete correspondiente a la versión a la cual se está migrando.
6. Extraer del paquete sae-2-frontend.ear actualmente instalado en el servidor de aplicaciones JBoss el archivo jboss-web.xml que se encuentra dentro del archivo sae-frontend.war, en la carpeta WEB-INF y copiarlo en la misma ubicación del paquete correspondiente a la versión a la cual se está migrando.
7. Copiar los cuatro archivos EAR correspondiente a la versión a la cual se está migrando a la carpeta standalone/deployments del servidor JBoss AS, teniendo especial cuidado de que los archivos sae-2-backoffice.ear y sae-2-frontend.ear sean los modificados en los puntos 5 y 6.
8. Iniciar el servidor de aplicaciones JBoss AS.

14.3 Migración de una versión a otra no consecutiva

En el caso de tener que migrar de una versión a otra no consecutiva, el procedimiento recomendado es el siguiente:

1. Detener el servidor de aplicaciones JBoss. Si hay más de un servidor de aplicaciones utilizando la misma base de datos se debe detener todos ellos.
2. Aplicar en el esquema global de la base de datos los cambios indicados en los archivos “migracion_de_vX_a_vY-global.sql” correspodientes a todas las migraciones intermedias en orden. Por ejemplo, para migrar de la versión 1.1 a la 1.4 es necesario aplicar los archivos “migracion_de_v1.1_a_v1.2-global.sql”, “migracion_de_v1.2_a_v1.3-global.sql” y “migracion_de_v1.3_a_v1.4-global.sql”
3. Aplicar en cada uno de los esquemas de la base de datos correspondientes a empresas los cambios indicados en los archivos “migracion_de_vX_a_vY-esquema.sql” correspodientes a todas las migraciones intermedias en orden, teniendo en cuenta que es necesario remplazar, en dicho archivo, el texto “{esquema}” por el nombre del esquema correspondiente; este procedimiento debe repetirse en todos los esquemas que son usados por empresas. Por ejemplo, para migrar de la versión 1.1 a la 1.4 es necesario aplicar los archivos “migracion_de_v1.1_a_v1.2-esquema.sql”, “migracion_de_v1.2_a_v1.3-esquema .sql” y “migracion_de_v1.3_a_v1.4-esquema .sql”
4. Hacer una copia de seguridad de los cuatro archivos EAR que están actualmente deployados en el servidor de aplicaciones JBoss AS y que corresponden a la versión desde la cual se está migrando.
5. Extraer del paquete sae-2-backoffice.ear actualmente instalado en el servidor de aplicaciones JBoss el archivo jboss-web.xml que se encuentra dentro del archivo sae-backoffice.war, en la carpeta WEB-INF y copiarlo en la misma ubicación del paquete correspondiente a la versión a la cual se está migrando.
6. Extraer del paquete sae-2-frontend.ear actualmente instalado en el servidor de aplicaciones JBoss el archivo jboss-web.xml que se encuentra dentro del archivo sae-frontend.war, en la carpeta WEB-INF y copiarlo en la misma ubicación del paquete correspondiente a la versión a la cual se está migrando.

7. Copiar los cuatro archivos EAR correspondiente a la versión a la cual se está migrando a la carpeta standalone/deployments del servidor JBoss AS, teniendo especial cuidado de que los archivos sae-2-backoffice.ear y sae-2-frontend.ear sean los modificados en los puntos 5 y 6. No es necesario aplicar los archivos EAR de las versiones intermedias.
8. Iniciar el servidor de aplicaciones JBoss AS.

15 Apéndice 1: Hardening de CentOS 7

Nota: este apéndice fue proporcionado por AGESIC y se transcribe a continuación ajustando solo la tipografía y formato de títulos para acompasarlo con el estilo general del documento.

15.1 Hardening de CentOS 7

15.1.1 Alcance

El presente documento propone una lista de puntos a tener en cuenta al momento de endurecer la postura de seguridad de un sistema operativo Linux CentOS 7 básico.

15.1.1.1 Supuestos

Los lineamientos que se proponen en esta guía suponen el endurecimiento del sistema operativo base, no de las aplicaciones o servicios que se disponibilicen sobre el mismo.

15.1.1.2 Material de Referencia

La presente guía se elaboró en base a las mejores prácticas documentadas, entre otros, en los siguientes enlaces:

- HighOn.Coffee (2016) *Security Harden CentOS 7*
- The University of Texas at Austin (2016) *Red Hat Enterprise Linux 7 Hardening Checklist*

15.1.2 Guía de Trabajo

15.1.2.1 Premisas iniciales y precondiciones

- **Medios de Instalación:** La instalación se realizará utilizando medios obtenidos desde los canales oficiales y verificada su integridad mediante los *hashes* que a tal efecto se disponen en dichos sitios.
- **Información del Servidor:** Se deberá documentar la información referida a el o los roles que deberá cumplir el servidor así como toda información relevante para el inventario y/o auditoría.\\Se deberá incluir al menos:
  - Para cada tarjeta de red:
    - Dirección física (*MAC Address*)
    - Dirección IP
  - Nombre del servidor
  - Rol o roles a cumplir en la infraestructura
  - Listado de software autorizado (no detallado, sino a alto nivel)
  - Fecha de última actualización de software
  - Fecha de instalación
  - Administrador responsable
  - Categorización de Seguridad (si existiere)

15.1.2.1.1 Preparación y seguridad física

- Si el servidor es una nueva instalación se deberá aislar de cualquier tráfico de red hasta que se hayan completado las configuraciones de endurecimiento detalladas en esta guía.
- Se debera configurar una contraseña de *BIOS/firmware*.
- Configurar en *BIOS* el orden de arranque para restringir el inicio del sistema haciendo uso de medios alternativos.
- Configurar usuario y contraseña de *bootloader* para prevenir manipulación de parámetros de arranque. Se recomienda que dicho usuario no tenga un nombre estándar (*admin, root, superuser, etc.*) y en particular que la contraseña no sea la misma que la utilizada en otros usuarios del sistema (*contraseña única*).
- Configurar autenticación para acceder en el modo monousuario.
- Deshabilitar el reinicio mediante el uso de Ctrl-Alt-Delete.
- Instalar *scren* para permitir el bloqueo de consola.

15.1.2.2 Parámetros de Kernel

- Activar ASLR
- Habilitar XD (*Ejecución Deshabilitada*) o NX (*No Ejecutar*) si estuviere disponible.
- Habilitar protección contra *buffer overflow* mediante el escudo de ejecución (*exec-shield*)

15.1.2.3 Sistema de Archivos

Para garantizar la posibilidad de aplicar políticas de montaje granulares en el sistema de archivos se requiere que el mismo sea creado previendo esa posibilidad. Se sugiere a continuación un esquema de particionado de acuerdo a esa premisa así como los parámetros recomendados para cada partición. Los tamaños de cada una dependerán de cada caso en particular. En caso de duda se podrá hacer uso de volúmenes lógicos (*LVM*) que permiten ampliación posterior. Caso especial será el punto de montaje /boot que no deberá ser un volúmen lógico y, de preferencia, ser apuntado mediante su UUID. Todas las particiones excepto la de *swap* deberán configurar un valor de

Punto de Montaje	Parámetros	Comentarios
/	defaults	
/boot	defaults,nosuid,noexec,nodev	
/home	defaults,nodev	
/tmp	defaults,nosuid,noexec,nodev	
/var	defaults,nosuid	
/var/tmp	defaults,nosuid,noexec,nodev	montado en /tmp

/var/log	defaults,nosuid,noexec,nodev	
/var/log/audit	defaults,nosuid,noexec,nodev	
/var/www	defaults,nosuid,noexec,nodev	
swap	defaults	
/dev/shm	nodev,nosuid,noexec	*opcional*

Ejemplo de instalación automática en Apéndice, Kickstart
Ejemplo de archivo /etc/fstab en Apéndice, /etc/fstab

15.1.2.4 Sincronización de Reloj

Se deberá configurar un cliente NTP para que el servidor mantenga el reloj sincronizado con el resto de la infraestructura. Este elemento es importante para facilitar la correlación de eventos provenientes de múltiples servidores.

15.1.2.5 Políticas de Auditoría

- Habilitar el *daemon auditd* para auditoría.
- Habilitar la auditoría de los procesos que se inician **antes** del arranque del *daemon auditd*

- Configurar los parámetros de auditoría tal como se indica en Apéndice, auditd

15.1.2.6 Detección de Cambios

- Instalar, configurar y monitorizar **AIDE** para la prevención de modificaciones no autorizadas a los archivos protegidos.

15.1.2.7 Manejo de Registros y Trazas

- Habilitar *rsyslogd*, *syslog-ng* o similar para la colección y tratamiento inicial de registros.
- Mantener copia de los registros en un servidor centralizado, en particular los concernientes a eventos de auditoría y acceso privilegiado.

15.1.2.8 Manejo de Medios Extraíbles

- Deshabilitar el uso de medios extraíbles.
- Prevenir que los usuarios monten discos USB.

15.1.2.9 Gestión de Usuarios y Permisos

- Todo usuario del sistema que sea utilizado por una persona deberá ser nominado y su uso personal e intransferible.
- Utilizar SHA512 para el almacenamiento de contraseñas.

- Configurar complejidad de contraseña con al menos 3 de cuatro clases obligatorias y verificaciones de repetición e inclusión.
- En la medida de lo posible activar verificación de contraseña contra diccionario.
- Las contraseñas deberán:
  - Tener un largo mínimo 12 caracteres.
  - Cambiarse con una periodicidad máxima de 90 días.
  - No ser reutilizadas como mínimo en los últimos 8 cambios.
  - No volver a cambiarse en menos de 3 días.
- Habilitar notificación de último acceso.
- Limitar el número máximo de intentos de login por sesión.
- Bloquear nuevos intentos de login frente a fallos reiterados.

15.1.2.9.1 Usuarios Privilegiados

- El número de usuarios con privilegios administrativos deberá limitarse al mínimo necesario.
- El uso de *sudo* como alternativa de gestión de acceso administrativo deberá ser cuidadosamente configurado para evitar *efectos secundarios* (accesos no previstos) no deseados.

15.1.2.10 Software Instalado y Gestión de Parches

- **Software Instalado:** Se deberá limitar el software instalado al estrictamente necesario para cumplir con las tareas asociadas al rol del servidor y su gestión. Todo otro software deberá ser completamente desinstalado incluyendo archivos de configuración y bibliotecas en las que dependa (cuando sea posible por dependencias cruzadas).
- **Orígenes de Software:** En la medida de lo posible se deberá instalar software solamente de repositorios oficiales y mediante el sistema de empaquetado provisto por la distribución. Para los casos en los que se requiera una excepción se priorizarán aquellos repositorios mantenidos por los propios desarrolladores del software en cuestión. Se deberán mantener actualizados los almacenes de claves de publicación de paquetes para garantizar la correcta verificación de firma de los mismos.
- **Compiladores:** Se deberá eliminar todo software compilador con el objeto de impedir que se pueda generar código ejecutable por fuera del instalado por los gestores de paquetes propios de la distribución. En caso de ser necesario compilar algún software en particular se dará preferencia a la compilación fuera de línea en otro equipo similar. Si ello no fuera posible, se deberá mantener la instalación del compilador el menor tiempo posible eliminando toda traza del mismo una vez finalizada la tarea.
- **Intérpretes de lenguajes:** Todo intérprete de lenguaje que no sea estrictamente necesario para la ejecución de las funciones atribuídas al servidor o su gestión deberá desinstalarse.
- **Bibliotecas:** Las bibliotecas instaladas deberán restringirse a las estrictamente necesarias para el correcto funcionamiento del software instalado, eliminando aquellas que hayan dejado de usarse y en particular las versiones anteriores o desactualizadas.
- **Gestión de Parches:** Se deberá mantener el software actualizado y en concordancia con la Política de Gestión de Actualizaciones si la hubiere. La instalación de actualizaciones **no**

deberá hacerse de forma automatizada y se recomienda probar dichas actualizaciones en sistemas que no estén en producción **antes** de instalarlos en éstos. No obstante lo anterior, la verificación de actualizaciones deberá ser automatizada por los mecanismos previstos en la distribución.

15.1.2.11 Configuraciones Automáticas

- Deshabilitar toda configuración automatizada incluyendo, pero no limitándose a, *avahi* y *zeroconf*.

15.1.2.12 Redes y Conectividad

- Deshabilitar IPv6 si no se lo está usando. En caso contrario configurarlo no dejando que el protocolo resuelva por autoconfiguración.
- Deshabilitar el cliente DHCP.
- Deshabilitar todo acceso a través de *TCP Wrappers*.
- Habilitar el firewall permitiendo únicamente tráfico entrante destinado a los servicios autorizados.
- Denegar en el firewall todo tráfico a servicios no autorizados registrando dichos intentos de acceso.

Se indícan varios parámetros de configuración de red recomendados en Apéndice, sysctl

15.1.2.13 Control de Acceso Mandatorio

- Habilitar y configurar SELinux en modo *enforcing* verificando que ningún proceso *daemon* queda por fuera de los controles.

15.1.2.14 Acceso Administrativo

- El acceso administrativo deberá ser nominado y deberán estar configurados todos los controles que permitan registro y auditoría de los mismos.

15.1.2.14.1 SSH

- Habilitar únicamente SSHv2
- Restringir el acceso únicamente a segmentos de red de gestión u orígenes autorizados.
- Deshabilitar el acceso por SSH al usuario *root*.
- En la medida de lo posible deshabilitar el acceso mediante usuario y contraseña configurando el mismo únicamente mediante par de claves.
- Habilitar el acceso solamente a los usuarios administrativos.
- De ser posible cambiar el puerto por omisión para el servicio y no usar los puertos alternativos tradicionales.
- Implementar protección contra ataques por fuerza bruta bloqueando los orígenes implicados.
- Configurar temporizadores de inactividad que cierren la sesión.

- Deshabilitar soporte a *rhost*.
- Deshabilitar autenticación basada en *host*.
- Deshabilitar el acceso mediante passwords nulas.
- Deshabilitar la posibilidad de uso de opciones de ambiente.
- Configurar un mensaje de atención a ser desplegado previo al ingreso que indique los principales puntos de la política de acceso privilegiado.
- Limitar los cifrados autorizados a aquellos más robustos. Mantener dicha lista actualizada.
- Configurar *UserRoaming no* para prevenir CVE-2016-0777

15.1.2.14.2 SNMP

SNMP debería ser deshabilitado a menos que sea un requerimiento de gestión del servidor. Las siguientes recomendaciones son para los casos en los que no se puede prescindir de dicho protocolo.

- Habilitar únicamente el protocolo SNMPv3.
- Configurar autenticación y cifrado.
- Restringir el acceso únicamente a los segmentos de red o direcciones IP orígen autorizados.

15.1.2.15 Auditorías Periódicas de Postura y Cumplimiento

- Configurar un sistema de verificación de postura y cumplimiento que realice revisiones periódicas en lapsos no mayores a 15 días, como por ejemplo OpenSCAP.

15.1.3 Apéndices

15.1.3.1 Kickstart

```
#version=RHEL7

install
# System authorization information
auth --enableshadow --passalgo=sha512

# Use CDROM installation media
cdrom
# Accept EULA
eula --agreed

services --enabled=NetworkManager,sshd
reboot

# Run the Setup Agent on first boot
#firstboot --enable
ignoredisk --only-use=sda
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
```

```
# System language
lang en_US.UTF-8
# SELinux
selinux --enforcing
# Network information
network  --bootproto=dhcp --device=eno16777736 --onboot=on --ipv6=off
network  --hostname=default-vm
# Root password
rootpw --iscrypted HASHGOESHERE
# System timezone
timezone Europe/London --isUtc --ntpservers=prime.transformers
# System bootloader configuration
bootloader --location=mbr --boot-drive=sda
# Partition clearing information
clearpart --all --drives=sda
ignoredisk --only-use=sda
# LVM

# Disk partitioning information
part pv.18 --fstype="lvmpv" --ondisk=sda --size=8004
part pv.11 --fstype="lvmpv" --ondisk=sda --size=8004
part /boot --fstype="ext4" --ondisk=sda --size=1000
volgroup lg_data --pesize=4096 pv.18
volgroup lg_os --pesize=4096 pv.11
logvol /  --fstype="xfs" --size=4000 --name=lv_root --vgname=lg_os
logvol /home  --fstype="xfs" --size=2000 --name=lv_home --vgname=lg_data
logvol /tmp  --fstype="xfs" --size=1000 --name=lv_tmp --vgname=lg_os
logvol /var  --fstype="xfs" --size=2000 --name=lv_var --vgname=lg_os
logvol /var/tmp  --fstype="xfs" --size=1000 --name=lv_var_tmp --vgname=lg_os
logvol /var/www  --fstype="xfs" --size=5000 --name=lv_var_www --vgname=lg_data
logvol /var  --fstype="xfs" --size=1000 --name=lv_var --vgname=lg_os
logvol /var/log  --fstype="xfs" --size=1500 --name=lv_var_log --vgname=lg_os
logvol /var/log/audit  --fstype="xfs" --size=500 --name=lv_var_log_audit
--vgname=lg_os
logvol /var/tmp  --fstype="ext4" --size=500 --name=lv_var_tmp --vgname=lg_os
logvol swap  --fstype="swap" --size=1000 --name=lv_swap --vgname=lg_data

%packages
@core
 %end

%post
%end
```

15.1.3.2 auditd

Configuración de auditoría en archivo */etc/audit/audit.rules*

```
# audit_time_rules - Record attempts to alter time through adjtime
-a always,exit -F arch=b64 -S adjtimex -k audit_time_rules

# audit_time_rules - Record attempts to alter time through settimeofday
-a always,exit -F arch=b64 -S settimeofday -k audit_time_rules

# audit_time_rules - Record Attempts to Alter Time Through stime
```

```
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -S clock_settime
-k audit_time_rules

# audit_time_rules - Record Attempts to Alter Time Through clock_settime
-a always,exit -F arch=b64 -S clock_settime -k audit_time_rules

# Record Attempts to Alter the localtime File
-w /etc/localtime -p wa -k audit_time_rules

# Record Events that Modify User/Group Information
# audit_account_changes
-w /etc/group -p wa -k audit_account_changes
-w /etc/passwd -p wa -k audit_account_changes
-w /etc/gshadow -p wa -k audit_account_changes
-w /etc/shadow -p wa -k audit_account_changes
-w /etc/security/opasswd -p wa -k audit_account_changes

# Record Events that Modify the System's Network Environment
# audit_network_modifications
-a always,exit -F arch=ARCH -S sethostname -S setdomainname -k
audit_network_modifications
-w /etc/issue -p wa -k audit_network_modifications
-w /etc/issue.net -p wa -k audit_network_modifications
-w /etc/hosts -p wa -k audit_network_modifications
-w /etc/sysconfig/network -p wa -k audit_network_modifications

#Record Events that Modify the System's Mandatory Access Controls
-w /etc/selinux/ -p wa -k MAC-policy

#Record Events that Modify the System's Discretionary Access Controls - chmod
-a always,exit -F arch=b32 -S chmod -F auid>=500 -F auid!=4294967295 -k perm_mod
-a always,exit -F arch=b64 -S chmod  -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls - chown
-a always,exit -F arch=b32 -S chown -F auid>=500 -F auid!=4294967295 -k perm_mod
-a always,exit -F arch=b64 -S chown -F auid>=500 -F auid!=4294967295 -k perm_mod

#Record Events that Modify the System's Discretionary Access Controls - fchmod
-a always,exit -F arch=b32 -S fchmod -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fchmod -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fchmodat
-a always,exit -F arch=b32 -S fchmodat -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fchmodat -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls - fchown
-a always,exit -F arch=b32 -S fchown -F auid>=500 -F auid!=4294967295 -k
```

```
perm_mod
-a always,exit -F arch=b64 -S fchown -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fchownat
-a always,exit -F arch=b32 -S fchownat -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fchownat -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fremovexattr
-a always,exit -F arch=b32 -S fremovexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fremovexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fsetxattr
-a always,exit -F arch=b32 -S fsetxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fsetxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls - lchown
-a always,exit -F arch=b32 -S lchown -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S lchown -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
lremovexattr
-a always,exit -F arch=b32 -S lremovexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S lremovexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
lsetxattr
-a always,exit -F arch=b32 -S lsetxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S lsetxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
removexattr
-a always,exit -F arch=b32 -S removexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S removexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod-a always,exit -F arch=b32 -S fchmodat -F auid>=500 -F auid!=4294967295
-k perm_mod
-a always,exit -F arch=b64 -S fchmodat -F auid>=500 -F auid!=4294967295 -k
```

```
perm_mod

#Record Events that Modify the System's Discretionary Access Controls - fchown
-a always,exit -F arch=b32 -S fchown -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fchown -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fchownat
-a always,exit -F arch=b32 -S fchownat -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fchownat -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fremovexattr
-a always,exit -F arch=b32 -S fremovexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S fremovexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
fsetxattr
-a always,exit -F arch=b32 -S lsetxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S lsetxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
removexattr
-a always,exit -F arch=b32 -S removexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S removexattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Events that Modify the System's Discretionary Access Controls -
setxattr
-a always,exit -F arch=b32 -S setxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod
-a always,exit -F arch=b64 -S setxattr -F auid>=500 -F auid!=4294967295 -k
perm_mod

#Record Attempts to Alter Logon and Logout Events
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins

#Record Attempts to Alter Process and Session Initiation Information
-w /var/run/utmp -p wa -k session
-w /var/log/btmp -p wa -k session
-w /var/log/wtmp -p wa -k session

#Ensure auditd Collects Unauthorized Access Attempts to Files (unsuccessful)
```

```
-a always,exit -F arch=b32 -S creat -S open -S openat -S open_by_handle_at -S
truncate -S ftruncate -F exit=-EACCES -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S open_by_handle_at -S
truncate -S ftruncate -F exit=-EPERM -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b64 -S creat -S open -S openat -S open_by_handle_at -S
truncate -S ftruncate -F exit=-EACCES -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b64 -S creat -S open -S openat -S open_by_handle_at -S
truncate -S ftruncate -F exit=-EPERM -F auid>=500 -F auid!=4294967295 -k access

#Ensure auditd Collects Information on the Use of Privileged Commands
#
#  Find setuid / setgid programs then modify and uncomment the line below.
#
##  sudo find / -xdev -type f -perm -4000 -o -perm -2000 2>/dev/null
#
# -a always,exit -F path=SETUID_PROG_PATH -F perm=x -F auid>=500 -F auid!
=4294967295 -k privileged

#Ensure auditd Collects Information on Exporting to Media (successful)
-a always,exit -F arch=ARCH -S mount -F auid>=500 -F auid!=4294967295 -k export

#Ensure auditd Collects File Deletion Events by User
-a always,exit -F arch=ARCH -S rmdir -S unlink -S unlinkat -S rename -S renameat
-F auid>=500 -F auid!=4294967295 -k delete

#Ensure auditd Collects System Administrator Actions
-w /etc/sudoers -p wa -k actions

#Ensure auditd Collects Information on Kernel Module Loading and Unloading
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules
-a always,exit -F arch=b64 -S init_module -S delete_module -k modules

#Make the auditd Configuration Immutable
-e 2
```

15.1.3.3 sysctl

Parámetros de configuración de red en el archivo */etc/sysctl.conf*

```
net.ipv4.ip_forward = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.tcp_max_syn_backlog = 1280
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

```
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_timestamps = 0
```

15.1.3.4 fstab

Ejemplo de configuración de particionado y puntos de montaje en el archivo /etc/fstab.

```
#
# /etc/fstab
# Created by anaconda on Sat Oct 11 14:28:47 2014
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/lg_os-lv_root /                       xfs     defaults        1 1
UUID=d73c5d22-75ed-416e-aad2-8c1bb1dfc713 /boot                   ext4
defaults,nosuid,noexec,nodev        1 2
/dev/mapper/lg_data-lv_home /home                   xfs     defaults        1 2
/dev/mapper/lg_os-lv_tmp /tmp                     xfs
defaults,nosuid,noexec,nodev        1 2
/dev/mapper/lg_os-lv_var /var                     xfs     defaults,nosuid
1 2
/dev/mapper/lg_os-lv_var_tmp /var/tmp               xfs
defaults,nosuid,noexec,nodev        1 2
/dev/mapper/lg_os-lv_var_tmp /var/log               xfs
defaults,nosuid,noexec,nodev        1 2
/dev/mapper/lg_os-lv_var_tmp /var/log/audit               xfs
defaults,nosuid,noexec,nodev        1 2
/dev/mapper/lg_data-lv_var_www /var/www               xfs
defaults,nosuid,noexec,nodev        1 2
/dev/mapper/lg_data-lv_swap swap                    swap    defaults        0 0
```

16 Versiones y cambios

Versión	Vers. Apl.	Fecha	Autor	Comentarios
1.0	0.0	14/12/2015	SP	Versión inicial.
1.1	0.1	15/12/2015	SP	Actualización.
1.2	0.2	17/12/2015	SP	Errores corregidos.
1.3	0.3	11/01/2016	SP	Segunda versión.
1.4	0.4	12/01/2016	SP	Errores corregidos.
1.5	0.5	20/01/2016	SP	Añadida la sección “Configurar propiedades”.
1.6	0.5	10/02/2016	SP/SA	Añadida la sección de SysLog y otras configuraciones de seguridad.
1.7	0.7	22/02/2016	SP	Incorporación de CDA y reorganización del documento.
1.8	0.7	08/04/2016	SP	Incorporación de CDA en la parte privada y correcciones en los módulos.
1.9	0.8	18/05/2016	SP	Reescritura casi total por cambios en los mecanismos de seguridad y control de acceso.
1.10	0.9	08/06/2016	SP	Agregado de cambios necesarios para configurar el ambiente en TrámitesUy y Trazabilidad.
1.11	1.1	13/06/2016	SP	Añadida información sobre multi-idioma. Se incluye detalles para la creación de la empresa inicial.
1.12	1.1	23/06/2016	SP	Corrección en configuración del frontend sin CDA.
1.13	1.2	03/08/2016	SP	Añadida las secciones de configuración a partir de la distribución empaquetada con JBoss, la

Versión	Vers. Apl.	Fecha	Autor	Comentarios
				autenticación contra un servidor LDAP y la gestión de frases de captcha.
1.14	1.3	12/09/2016	MG/SP	Ajustes por nueva versión del software.
1.15	1.4	05/10/2016	SP	Ajustes por nueva versión del software.
1.16	1.4	10/10/2016	SP	Correcciones menores.