Manual de Instalación y Despliegue de @firma v.6.4.0_001

Documento nº:	@firma-InstalacionyDespliegue-MAN
Revisión:	003
Fecha:	21-04-2022
Período de retención:	Permanente durante su período de vigencia + 3 años después de su anulación

CONTROL DE MODIFICACIONES

Documento nº:	@firma-InstalacionyDespliegue-MAN
Revisión:	003
Fecha:	21-04-2022

Rev.	001
Fecha	16-07-2021
Descripción	Documentación inicial.

Rev.	002
Fecha	03-11-2021
Descripción	Se actualiza el contenido definitivo del manual para la versión 6.4.0_000.

Rev.	003
Fecha	21-04-2022
Descripción	Se actualiza el contenido definitivo del manual para la versión 6.4.0_001.

CONTROL DE DISTRIBUCIÓN

Documento nº:	@firma-InstalacionyDespliegue-MAN
Revisión:	003
Fecha:	21-04-2022

Copias Electrónicas:

La distribución de este documento ha sido controlada a través del sistema de información.

Copias en Papel:

La vigencia de las copias impresas en papel está condicionada a la coincidencia de su estado de revisión con el que aparece en el sistema electrónico de distribución de documentos.

El control de distribución de copias en papel para su uso en proyectos u otras aplicaciones es responsabilidad de los usuarios del sistema electrónico de información.

Fecha de impresión 25 de abril de 2022

ÍNDICE

Control de Modificaciones **2**

Control de Distribución **3**

1 Objeto **5**

2 Alcance **5**

3 Siglas **5**

4 Introducción **5**

4.1 Cambios en la versión 6.4.0_000 con respecto a versiones anteriores 5

5 Guía de Instalación **11**

5.1 Consideraciones previas 11

5.2 Actualización desde @firma 6.4-beta 13

5.2.1 Despliegue del nodo de administración 13

5.2.2 Despliegue del resto de nodos (servicios y servicios asíncronos) 17

5.3 Actualización desde @firma 6.4.0_000 19

5.3.1 Despliegue del nodo de administración 19

5.3.2 Despliegue del resto de nodos (servicios y servicios asíncronos) 20

5.4 Instalación de los Esquemas de BD desde cero 21

5.4.1 Consideraciones previas 21

5.4.2 Oracle 21

5.4.3 HSQLDB: 26

5.5 Instalación de @Firma v6.4.0_001 desde cero 31

5.5.1 Consideraciones previas 32

5.5.2 Instalación en Red Hat 6.5 33

5.5.3 Arranque del JBOSS 53

5.5.4 Finalizar configuración 53

6 Anexo I: Mantenimiento de la base de datos HSQLDB de configuración **66**

6.1 Introducción 66

6.2 Estructura de la base de datos local de configuración del nodo de administración 66

6.3 Forzar sincronización de nodos de servicio o servicios asíncronos 67

6.4 Recuperación de una configuración estable o una copia de seguridad 67

7 Anexo II: Instalación de software base **69**

7.1 Variables Entorno Java 69

7.2 Consideraciones sobre instalación Cliente Oracle 69
7.3 Generación y Consumo de Entropía 70
7.3.1 RNG-Tools 71
7.3.2 Haveged 72

1 Objeto

Es objeto de este documento es describir el proceso de instalación y despliegue de la plataforma @firma versión 6.4.0_000.

2 Alcance

El objetivo global de este documento es describir el proceso completo de instalación y puesta en marcha de la plataforma.

3 Siglas

JDK	Java Development Kit
TSL	Trusted Service List
BD	Base de datos
HSQLDB	HyperSQL DataBase
CRL	Certificate Revocation List

4 Introducción

El presente documento recoge la instalación de la plataforma de Autenticación y Firma Digital @firma 6.4.0_000.

@firma ha sido desarrollada bajo la plataforma JBoss, un servidor de aplicaciones J2EE que ejecuta componentes software desarrollados en Java. Estos componentes pueden ser tanto componentes Web (directorios Web, páginas JSP, servicios Web) como componentes EJB's. Es por ello que todo lo referente a la aplicación (directorios, puesta en marcha, configuración) depende en gran medida y está condicionada por el servidor de aplicaciones JBoss y las herramientas que le dan soporte.

4.1 Cambios en la versión 6.4.0_000 con respecto a versiones anteriores

Respecto a la versión anterior, los cambios incluidos en la versión 6.4.0_000 de @firma son:

- Cambios en Arquitectura:

Se divide en tres componentes: Administración, Servicios Base (Validación y Firma) y Servicios Asíncronos, todo ello trabajando funcionalmente en "mono-nodo".

Adaptación del código para que sea compatible con JDK8 y WildFly 10.

Se extrae la lógica de validación mediante TSLs al Sistema VALet y se realiza la integración con este Sistema.

Se traslada el modelo de configuración y alarmas a bases de datos locales hsqldb.

Se desarrolla un mecanismo de sincronización de las bases de datos locales de configuración de los nodos de Servicios a partir de la registrada en el nodo de Administración.

Se elimina la posibilidad de configuración en clúster.

- Módulo de Administración:

Se añade la posibilidad de configurar varios certificados para representar un mismo PSC.

Se sacan del menú de servidores los apartados de "Destinatarios" y "Responsables Aplicaciones".

Se añade un botón para poder ver las aplicaciones asociadas a un responsable en el módulo de configuración.

Se añade la posibilidad de replicar en el resto de las políticas los cambios realizados en una de ellas

Se añade la posibilidad de hacer una selección múltiple de tipos de certificado en la gestión de políticas, así como que solo aparezcan aquellos tipos que no estén asociados a la política.

Nueva funcionalidad en módulo de gestión de aplicaciones: Se permite mover aplicaciones y unidades dentro del árbol manteniendo sus datos de configuración.

Módulo de Keystores. Se oculta del almacén de Certificados el Almacén de PSCs (no se muestra en árbol).

Se añade la opción de archivar certificado en la gestión de PSCs.

Se desactiva el botón de la creación de instancia de gestor documental mientras no hay seleccionado ningún gestor documental.

Se admite la configuración de múltiples métodos de validación para la descarga de varias CRLs.

Se deshabilita el botón de eliminar aplicación mientras no se haya seleccionado ninguna.

Se desvincula las aplicaciones de las listas TSL.

Se habilita un check en los parámetros de la aplicación para controlar si se permite la validación por TSL o no.

Se incluye una nueva interfaz que permite validar firma electrónica en el módulo de consultas.

Se paginan los almacenes de keystore.

Se realizan modificaciones para hacer que el navegador no autocomplete los campos usuario y password de los distintos módulos con el usuario de acceso.

Se añade la posibilidad de establecer cabeceras específicas HTTP/S en el módulo de los métodos de validación.

Se modifica el comportamiento al añadir pares de claves a los keystores para que se incluya la cadena de certificación asociada al alias si fuera necesario.

Nuevo desarrollo que posibilita a los usuarios la opción de autenticarse mediante un certificado asociado a su perfil dentro de la plataforma.

Se añade un nuevo botón para exportar la configuración de un método de validación CRLs a formato JSON para ser importados en @firmaCRL.

Se mejora el mensaje de eliminación de una Unidad Organizativa.

Se elimina la pestaña de consulta de log.

Se elimina la opción de consulta de elementos de cache ya que no es compatible con el modelo “mono-nodo”.

Se añade la opción de incorporación de configuraciones de PSCs generadas mediante la herramienta CADeskTool.

Se añade un nuevo check en los métodos de validación que, cuando está activado, permite lanzar alarmas si un determinado método falla.

Se elimina el grupo de propiedades 2 (contexto) ya que no contiene ninguna propiedad y deja de tener sentido (anteriormente contenía propiedades relativas al clúster, el cual ha dejado de existir).

Se modifican las condiciones que se deben reunir para mostrar el botón de guardado durante la edición y creación de una unidad organizativa. A partir de ahora, la plataforma permitirá la creación de unidades organizativas de tipo DIR3 sin necesidad de una sincronización previa cuando el tipo de integración con DIR3 es relajado.

Se añaden valores por defecto a los campos obligatorios durante la creación de unidades organizativas cuando dichos valores no han sido solicitados en la administración (y por tanto son nulos). Esto afecta únicamente a la creación de unidades organizativas de tipo DIR3, ya que los campos "nombre de unidad organizativa", "tipo de unidad organizativa", "comunidad autónoma" y "provincia" están deshabilitados en la administración.

Se añade un pool de conexiones para el acceso a las TSAs

- Módulo de Validación:

Se crea un nuevo método de bloqueo sobre métodos de validación para deshabilitarlos temporalmente si se detectan n intentos fallidos de acceso en un periodo de n segundos configurables en staticAfirmaConfig.properties.

Se evoluciona la caché L1 para la validación de certificados para solventar problemas de bloqueos y cambios en la configuración de esta.

Se añade la funcionalidad para la validación de certificados de CA mediante ValET. Para los tipos de certificado se habilita un nuevo atributo-bandera que determina la prioridad a la hora de realizar el proceso de validación, si priorizar ValET sobre la política de certificación o no. Como consecuencia, ha sido necesario adaptar el proceso de exportación/importación de políticas de certificación (por nuevos atributos en elementos como los tipos de certificado).

- Módulo de Alarmas:

Se realizan modificaciones para permitir el uso y la gestión de más de un servidor de correo para emitir alarmas.

- Módulo de Firma:

Se modifica la validación en firmas CAdES y XAdES (Baseline o no) para hacerla sobre todos los sellos de tiempo incluidos.

Se lleva a cabo la validación de todos los sellos de tiempo en firmas XAdES, CAdES y PAdES (Baseline y No Baseline) según la RFC 5816, de manera que se verifique correctamente el uso de los elementos ESSCertID y ESSCertIDv2 a partir de la fecha indicada en una propiedad configurable de la plataforma.

Se modifica el proceso que determina las tareas de validación que aplicar sobre firmas XAdES Baseline para que se apliquen siempre, al menos, las validaciones para el nivel B-Level.

Validación del algoritmo de firma y del tamaño de clave junto con el algoritmo de firma para firmantes, contra-firmantes, sellos de tiempo, y certificados firmantes.

Inclusión de un nuevo tipo de resultado asociado a validar un certificado firmante y que éste resulte revocado cuando, además, no se encuentra asegurado por ningún sello de tiempo.

Se añade una nueva comprobación durante la validación de los certificados firmantes en peticiones XAdES A. Ahora se comprueba (al igual que en el resto de las peticiones) si el algoritmo de firma del certificado firmante es seguro o no. NOTA: Para que esta comprobación se realice correctamente es necesario tener la propiedad "NOT_SECURE_SIGNATURE_ALGORITHM" con valor "W" en el fichero "critical_policy.properties", dentro de la tarea "SIGNER_CERTIFICATE_VAL".

Las firmas XAdES No Baseline generadas incluirán el elemento DataObjectFormat siempre.

Se añade una nueva característica a los procesos de actualización de firmas. Ahora, en las respuestas de actualización, la plataforma incluye la fecha de expiración del certificado firmante del sello de tiempo, el cual,a su vez, indica la fecha hasta la que será válida la firma. El campo añadido en la respuesta es "NextUpdate" y solo aparecerá cuando la firma tenga sellos de tiempo.

- Módulo de Servicios:

Se añade la posibilidad de forzar la validación de peticiones DSS mediante formatos no baseline, aunque la plataforma esté configurada para trabajar como baseline. Para poder realizar esta acción, las peticiones SOAP recibidas deben incluir el elemento " <afxp:ProcessAsNotBaseline xmlns:afxp="urn:afirma:dss:1.0:profile:XSS:schema"/> " dentro del OptionalInputs.

Se añade la posibilidad de devolver la fecha de próxima actualización en las respuestas de los servicios DSS de validación de firma (NextUpdate). Para que la plataforma devuelva dicha fecha, es necesario añadir el elemento ReturnNextUpdate en la petición de validación (dentro del OptionalInputs).

Se desarrolla funcionalidad para el control de peticiones repetidas por aplicación y de peticiones por aplicación y servicio.

Se añade nueva funcionalidad mediante la cual, cuando el resultado de la validación de un certificado mediante DSS o Nativo ofrece como resultado "certificado no soportado", se reenvía la petición a un servicio secundario de validación @firma que será configurado en una nueva sección en configuración.

Se incluye la posibilidad de mostrar URIs descriptivas del error producido en una validación solicitada por peticiones SOAP. Las URIs se deben colocar en el fichero de propiedades *_policy.properties junto a su criticidad, separado el caracter ';'.

Se incluye la posibilidad de mostrar URIs descriptivas del error producido en una validación solicitada por peticiones SOAP. Las URIs se deben colocar en el fichero de propiedades *_policy.properties junto a su criticidad, separado el caracter ';'.

- Otras mejoras:

Se añade en su completa funcionalidad el componente de provisión de información de monitorización SPIE.

Se modifica el proceso para hacer que las carpetas temporales se escriban en la carpeta temporal de jboss en lugar de en la carpeta bin. Se corrigen errores encontrados en el proceso de importaición cuando había PSCs con caracter / en el identificador.

Generación de un nuevo log en @firma que registre los tiempos de respuestas de los servicios WS y OCSP así como los tiempos empleados en las dependencias TS@, Métodos de Validación y VALet.

Se añade un nuevo módulo para la obtención de información de los tiempos de respuestas de los servicios, extrayendo dicha información de un fichero de log, que tras una serie de cálculos y clasificaciones se vuelca en base de datos. Con dicha información se generará un informe al mes.

Se evoluciona la versión de las siguientes librerías de IAIK:

iaik_cms-5.1.jar

iaik_jce-5.60.jar

iaik_tsp-2.32.jar

iaik_xades-2.13.jar

iaik_xsect-2.14.jar

Se añade una tarea para la realización periódica de backups de la base de datos HSQLDB de configuración.

- Bugs resueltos:

Se corrige la carga inicial del locale manager, de modo que si aún no se ha cargado la vista, obtiene el locale del navegador.

Se arregla el refresco de elementos de caché que no estaba funcionando correctamente.

Corrección del acceso autenticado a la plataforma mediante el cliente de firma.

Se corrigen mensajes de log en los que se indicaban que todas las operaciones que se realizaban en los keystores se hacían directamente en base de datos, cuando realmente se hacen todas con la información contenida en la caché (infinispan) de los almacenes de claves. Esto hacía confundir a Sistemas, pensando que se realizaban excesivas consultas a BBDD.

Se extiende la validación de seguridad sobre la vulnerabilidad por la que el servidor se saturaba al validar ciertos PDFs. Ahora se comprueba que el PDF y las revisiones acaben con %%EOF

Se solventa bug a la hora de modificar la configuración del proxy desde la administración.

4.2 Cambios en la versión 6.4.0_001 con respecto a la versión 6.4.0_000

- Bugs resueltos:

Se solventa problema al importar políticas. No se estaban asociando los certpsc con sus padres.

Se corrige un bug detectado en el proceso de validación de certificados de sello de tiempo.

Se corrige bug detectado en la validación de firma con certificados caducados sin sello de tiempo. El resultado de la validació no indicaba el estado EXPIRED del certificado.

Se añaden el caracter NULL, la tabulación y el FF al limpiado del final de un PDF de cara a ver si termina en %%EOF.

Se corrige un bug detectado a la hora de procesar los claimed roles en firmas CMS (o PAdES) y que de forma aleatoria se procesaba como inválido.

Se modifica la logica para reconocer certificados de sellos de tiempo en las TSL (un certificado de sello de tiempo es reconocido en la TSL si en las identidades digitales del TSP correspondiente se encuentra el certificado o el emisor del mismo).

Se solventa problema al hacer firmas servidor PADES cuando en la apliación está marcada la opción "con timestamp"

- Otras mejoras:

Se añade la opción de realizar la petición a TSA (WS o RFC3161) pasando el identificador de la aplicación de @firma, para que si la TSA a la que se le solicita está actualizada podrá incluir dicho identificador en el procesado de sus estadísticas.

5 Guía de Instalación

5.1 Consideraciones previas

Para realizar validaciones contra TSLs, el sistema necesita la convivencia con la plataforma valET. Si se necesitan este tipo de validaciones, es necesario instalarla. Los recursos, así como la documentación se encuentran en este mismo CD en el directorio “valET_1.0.0”.

La plataforma @firma Versión 6.4 está compuesta principalmente por dos grandes elementos a desplegar, descritos a continuación:

- **Esquemas de Base de Datos**: El sistema necesita dos tipos de bases de datos: Centralizadas (compartidos por todos los nodos) y locales (una instancia local para cada nodo). Por motivos de eficiencia y criticidad, el Sistema de Gestión de Base de Datos para los esquemas

centralizados debe ser ORACLE 11g release 2. El motor de Base de Datos debe colocarse en la red interna del organismo, sin visibilidad desde Internet. Los seis esquemas de Base de Datos necesarios para el correcto funcionamiento de la plataforma (centralizados y locales) son:

-Monitorización de Alarmas (local): donde se almacenan y monitorizan los eventos de alarmas que se puedan producir durante la ejecución de la plataforma. Además, gestiona el control de resúmenes de alarmas, almacenando la información necesaria para luego ser enviada a los destinatarios configurados en la fecha y hora adecuadas.

-Configuración (local): donde se almacena toda la configuración de la plataforma @firma (inclusive los almacenes de certificados). El nodo de administración tendrá la versión "Master" de la configuración que se irá distribuyendo de forma asíncrona al resto de nodos en sus versiones "Slave".

-Gestión de Procesos (compartida): donde se almacenan toda la información correspondiente a peticiones WS que deben ser procesadas de forma asíncrona.

-Quartz (compartida): donde la herramienta Quartz, utilizada para la gestión de tareas programadas, establece los datos necesarios para controlar la ejecución de diversas tareas en cluster (tareas de descarga de CRL, tareas de envíos de resúmenes de alarmas… etc).

-Transaccional (compartida): donde se almacenan todas las entidades asociadas a las transacciones de servicios realizadas en la plataforma.

-Caché de CRLs (compartida): donde se almacenan las CRLs descargadas de los Prestadores de servicios de Certificación configurados para ello. Este esquema se corresponde con el nivel 2 del sistema de caché de estados de certificados.

Adicionalmente, puede o no utilizarse el esquema de "Custodia" utilizado en la versión 5 de @firma si se necesita mantener la retrocompatibilidad con la custodia realizada en dicha versión.

- **Servidores @firma:** Se dispone de tres tipos de servidores, administración, servicios y servicios asíncronos. Será necesario instalar al menos 2 de estos servidores, uno de administración y otro de servicios.

-Servidor de Administración: En él, se dispone de la interfaz de administración, así como de los elementos necesarios para la sincronización del resto de nodos y la realización de tareas en background.

-Servidor de servicios: En él, se dispone de todo lo necesario para responder a las peticiones de servicio.

El Servidor de @firma publica las interfaces XML-SOAP necesarias para la comunicación con el mismo (ver manual de programación de WebServices). El Servidor debe colocarse en la red interna del organismo, sin visibilidad desde Internet. Este servidor debe ser accesible únicamente por las aplicaciones que se integran con la plataforma @firma y por los puertos habilitados para ello (por defecto: 8080, 443 y 4433).

-Servidor de servicios asíncronos: En él, se dispone de todo lo necesario para responder a las peticiones asíncronas.

A continuación, en los siguientes apartados se describe la instalación de los componentes descritos: Esquemas de Base de datos y el Servidor de @Firma.

5.2 Actualización desde @firma 6.4-beta

A continuación, se describe el proceso de actualización desde la versión beta de @firma 6.4.

5.2.1 Despliegue del nodo de administración

El despliegue comenzará por el nodo de administración y continuará con el resto de nodos de servicio y servicios asíncronos que se deseen instalar.

Antes de nada, sería conveniente realizar una copia de seguridad de la instalación anterior por si surgieran problemas, de cara a volver a establecerla.

De cara a realizar la actualización de la base de datos HSQLDB de configuración, si durante la instalación de @firma 6.4_BETA se importaron los datos de la base de datos Oracle de la versión anterior (6.2.2_xxx), se recomienda volver a realizar el volcado de nuevo, ya que es posible que se hayan hecho cambios en la versión 6.2.2_xxx desde que se ejecutó la importación inicial, y estos cambios se deseen trasladar. En todo caso, si no se desea volver a realizar por algún motivo, existe un mecanismo para actualizar la base de datos HSQLDB de configuración desde la versión 6.4_BETA hasta la versión actual.

A continuación, se describen ambos procesos:

Importación de la base de datos de nuevo desde una versión anterior (6.2.2_xxx):

1) Parar la instancia de la base de datos HSQLDB de configuración (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopConfigDatabase.sh).

2) Eliminar por completo el contenido del directorio “JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/configuration/LocalHSQLDBDataBase/configuration”.

3) Arrancar de nuevo la instancia de la base de datos HSQLDB de configuración (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh).

4) Copiar el directorio “CDInstalacion/@firma_6.4.0_001/afirma6-standalone-importDDBB” y todos sus subdirectorios a la ruta deseada, desde ahora STANDALONE_IMPORT_HOME.

5) Configurar y ejecutar el componente afirma6-standalone-importDDBB de cara a volver a importar la base de datos desde Oracle. Para ello:

   - Editar el fichero “STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/configuration/staticImportDDBBConfig.properties” sustituyendo

RPLC.CONF_SCHM.HSQLDB.IP por la IP/interfaz en el que se iniciará la base de datos de configuración HSQLDB configurada en el fichero "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh".

RPLC.CONF_SCHM.HSQLDB.PORT por el puerto en el que se iniciará la base de datos de configuración HSQLDB configurado en el fichero "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh".

RPLC.CONF_SCHM.ORACLE.CNNTNURL por la URL de conexión hacia la base de datos que contiene el esquema de configuración origen (jdbc:oracle:thin:@IP:PORT:SID) que se desea migrar.

RPLC.CONF_SCHM.ORACLE.OWNER por el identificador del usuario propietario del esquema de configuración origen que se desea migrar.

RPLC.CONF_SCHM.ORACLE.PASS por el password del usuario propietario del esquema de configuración origen que se desea migrar.

Establecer valores para las propiedades

database.path con la ruta absoluta hacia la base de datos local de configuración "JBOSS_HOME/wildfly-10.1.0.Final-Afirma /standalone/configuration/LocalHSQLDBDataBase/configuration"

y

source.version con la versión origen desde la que se desea migrar o actualizar. Los valores posibles en este caso son 6_2_2_004, 6_2_2_009, 6_2_2_011 o 6_2_2_012.

- Editar el script "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/generateScript-importDDBB.sh" indicando en la primera línea (JDK_LIB=/RUTA_JDK/jdk1.8.0_131) la ruta hacia la jdk 1.8.0_131

- Ejecutar el script "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/generateScript-importDDBB.sh" para crear la base de datos comprobando en el log generado en "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/logs" que el proceso haya finalizado con éxito.

Actualización de la base de datos desde la versión 6.4_BETA:

1) Copiar el directorio "CDInstalacion/@firma_6.4.0_001/afirma6-standalone-importDDBB" y todos sus subdirectorios a la ruta deseada, desde ahora STANDALONE_IMPORT_HOME. Este componente actualizará desde versiones.

2) Configurar y ejecutar el componente afirma6-standalone-importDDBB de cara a actualizar la base de datos. Para ello:

- Editar el fichero "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/configuration/staticImportDDBBConfig.properties" sustituyendo

RPLC.CONF_SCHM.HSQLDB.IP por la IP/interfaz en el que se iniciará la base de datos de configuración HSQLDB configurada en el script "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh".

RPLC.CONF_SCHM.HSQLDB.PORT por el puerto en el que se iniciará la base de datos de configuración HSQLDB configurado en el script "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh".

Establecer valores para las propiedades

database.path con la ruta absoluta hacia la base de datos local de configuración "JBOSS_HOME/wildfly-10.1.0.Final-Afirma /standalone/configuration/LocalHSQLDBDataBase/configuration"

y

source.version con la versión origen desde la que se desea migrar o actualizar. En este caso el valor a establecer sería 6_4_BETA.

- Editar el script "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/generateScript-importDDBB.sh" indicando en la primera línea (JDK_LIB=/RUTA_JDK/jdk1.8.0_131) la ruta hacia la jdk 1.8.0_131

- Ejecutar el script "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/generateScript-importDDBB.sh" para crear la base de datos comprobando en el log generado en "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/logs" que el proceso haya finalizado con éxito.

Actualización del servidor

Una vez actualizada la base de datos, continuamos con el proceso de actualización del servidor wildfly. Para ello:

1) Parar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopWildFlyAfirma.sh).

2) Copiar el directorio “CDInstalacion/@firma_6.4.0_001/Actualizacion/from64Beta/wildfly-10.1.0.Final-Afirma” y todos sus subdirectorios a la ruta final de instalación (ruta donde está instalada la versión beta), machacando el contenido coincidente.

3) Copiar sobre la ruta final de instalación “JBOSS_HOME/wildfly-10.1.0.Final-Afirma”, machacando los recursos coincidentes, el contenido de:

   “CDInstalacion/@firma_6.4.0_001/**Actualizacion/from64Beta/**resources/admin/wildfly-10.1.0.Final-Afirma”.

   incluidos todos los subdirectorios.

4) Copiar sobre la ruta final de instalación “JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/deployments”, machacando el ear anterior, el contenido de:

   “CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/admin/wildfly-10.1.0.Final-Afirma/standalone/deployments” si no se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

   O

   “CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/admin-BC/wildfly-10.1.0.Final-Afirma/standalone/deployments” si se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

   incluidos todos los subdirectorios.

5) Editar el fichero de propiedades “JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/configuration/staticAfirmaVersion.properties” Estableciendo a true la propiedad “update.database.on.startup”.

6) Configurar el fichero de propiedades JBOSS_HOME/standalone/configuration/staticAfirmaConfig.properties tal y como se indica en el apartado 22) de la “Configuración del servidor de Aplicaciones JBOSS” dentro del apartado 5.5.2.

7) Dado que en la versión beta se incluyó un fichero “JBOSS_HOME/standalone/configuration/standalone.xml” donde se “obligaba” a configurar el datasource “AfirmaBackwardCustodyDS”, y dado que esto solo aplica en instalaciones donde se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5, habría que eliminar dicho datasource si en este caso NO se desea mantener dicha retrocompatibilidad. El contenido a eliminar es el siguiente:

```
<datasource jndi-name="java:jboss/datasources/AfirmaBackwardCustodyDS" enabled="true" use-java-context="true" pool-name="AfirmaBackwardCustodyDS" jta="true">
<connection-url>RPLC.CUST_SCHM.CNNTNURL</connection-url>
<driver>oracle-driver</driver>
<connection-property name="autoReconnect">true</connection-property>
<pool>
```

```
                <min-pool-size>10</min-pool-size>
                <max-pool-size>60</max-pool-size>
                <prefill>true</prefill>
</pool>
<security>
                <user-name>RPLC.CUST_SCHM.USER</user-name>
                <password>RPLC.CUST_SCHM.PASS</password>
</security>
<validation>
                <check-valid-connection-sql>SELECT 0 FROM DUAL</check-valid-connection-sql>
                <background-validation>true</background-validation>
                <background-validation-millis>600000</background-validation-millis>
</validation>
 </datasource>
```

8) Arrancar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startWildFlyAfirma.sh).

5.2.2 Despliegue del resto de nodos (servicios y servicios asíncronos)

1) Parar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopWildFlyAfirma.sh), y realizar una copia de seguridad de la instalación anterior.

2) Copiar el directorio "CDInstalacion/@firma_6.4.0_001/Actualizacion/from64Beta/wildfly-10.1.0.Final-Afirma" y todos sus subdirectorios a la ruta final de instalación (ruta donde está instalada la versión beta), machacando el contenido coincidente.

3) Copiar sobre la ruta final de instalación "JBOSS_HOME/wildfly-10.1.0.Final-Afirma", machacando los recursos coincidentes, el contenido de:

   "CDInstalacion/@firma_6.4.0_001/**Actualizacion/from64Beta/**resources/services/wildfly-10.1.0.Final-Afirma",

   O

   "CDInstalacion/@firma_6.4.0_001/**Actualizacion/from64Beta/**resources/async/wildfly-10.1.0.Final-Afirma" en función de si se está actualizando un nodo de servicio o de servicios asíncronos.

   Incluidos todos los subdirectorios.

4) Copiar sobre la ruta final de instalación "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/deployments", machacando el ear anterior, el contenido de:

   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/services/wildfly-10.1.0.Final-Afirma/standalone/deployments" o
   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/async/wildfly-10.1.0.Final-Afirma/standalone/deployments" (en función de si se trata de un nodo de servicio o de servicios

asíncronos) si no se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

O

"CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/services-BC/wildfly-10.1.0.Final-Afirma/standalone/deployments" o "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/async-BC/wildfly-10.1.0.Final-Afirma/standalone/deployments" (en función de si se trata de un nodo de servicio o de servicios asíncronos) si se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

Incluidos todos los subdirectorios.

5) Editar el fichero de propiedades "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/configuration/staticAfirmaVersion.properties" Estableciendo a true la propiedad "update.database.on.startup".

6) Configurar el fichero de propiedades JBOSS_HOME/standalone/configuration/staticAfirmaConfig.properties tal y como se indica en el apartado 22) de la "Configuración del servidor de Aplicaciones JBOSS" dentro del apartado 5.5.2.

7) Dado que en la versión beta se incluyó un fichero "JBOSS_HOME/standalone/configuration/standalone.xml" donde se "obligaba" a configurar el datasource "AfirmaBackwardCustodyDS", y dado que esto solo aplica en instalaciones donde se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5, habría que eliminar dicho datasource si en este caso NO se desea mantener dicha retrocompatibilidad. El contenido a eliminar es el siguiente:

```
<datasource jndi-name="java:jboss/datasources/AfirmaBackwardCustodyDS" enabled="true" use-java-context="true" pool-name="AfirmaBackwardCustodyDS" jta="true">
<connection-url>RPLC.CUST_SCHM.CNNTNURL</connection-url>
<driver>oracle-driver</driver>
<connection-property name="autoReconnect">true</connection-property>
<pool>
                <min-pool-size>10</min-pool-size>
                <max-pool-size>60</max-pool-size>
                <prefill>true</prefill>
</pool>
<security>
                <user-name>RPLC.CUST_SCHM.USER</user-name>
                <password>RPLC.CUST_SCHM.PASS</password>
</security>
<validation>
                <check-valid-connection-sql>SELECT 0 FROM DUAL</check-valid-connection-sql>
                <background-validation>true</background-validation>
                <background-validation-millis>600000</background-validation-millis>
</validation>
 </datasource>
```

8) Arrancar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startWildFlyAfirma.sh). Es importante que el nodo de administración se encuentre actualizado y arrancado antes de proceder al arranque del resto de nodos.

5.3 Actualización desde @firma 6.4.0_000

A continuación, se describe el proceso de actualización desde la versión de @firma 6.4.0_000.

5.3.1 Despliegue del nodo de administración

El despliegue comenzará por el nodo de administración y continuará con el resto de nodos de servicio y servicios asíncronos que se deseen instalar.

Antes de nada, sería conveniente realizar una copia de seguridad de la instalación anterior por si surgieran problemas, de cara a volver a establecerla.

Actualización del servidor

Para ello:

1) Parar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopWildFlyAfirma.sh).

2) Copiar el directorio "CDInstalacion/@firma_6.4.0_001/Actualizacion/from640_000/wildfly-10.1.0.Final-Afirma" y todos sus subdirectorios a la ruta final de instalación (ruta donde está instalada la versión 6.4.0_000), machacando el contenido coincidente.

3) Copiar sobre la ruta final de instalación "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/deployments", machacando el ear anterior, el contenido de:

   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/admin/wildfly-10.1.0.Final-Afirma/standalone/deployments" si no se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

   O

   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/admin-BC/wildfly-10.1.0.Final-Afirma/standalone/deployments" si se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

   incluidos todos los subdirectorios.

4) Añadir el siguiente fragmento al final del fichero JBOSS_HOME/standalone/configuration/staticAfirmaConfig.properties:

```
#-------------------------------------------------------------------------------
# PROPIEDADES TSA
#-------------------------------------------------------------------------------
# Bandera donde se indicará si se incluye en la petición SOAP a la TSA  el identificador de la aplicación de @firma (true) o no (false).
tsa.include.afirmaApp = false
```

5) Arrancar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startWildFlyAfirma.sh).

5.3.2 Despliegue del resto de nodos (servicios y servicios asíncronos)

1) Parar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopWildFlyAfirma.sh), y realizar una copia de seguridad de la instalación anterior.

2) Copiar el directorio "CDInstalacion/@firma_6.4.0_001/Actualizacion/from640_000/wildfly-10.1.0.Final-Afirma" y todos sus subdirectorios a la ruta final de instalación (ruta donde está instalada la versión 6.4.0_000), machacando el contenido coincidente.

3) Copiar sobre la ruta final de instalación "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/deployments", machacando el ear anterior, el contenido de:

   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/services/wildfly-10.1.0.Final-Afirma/standalone/deployments" o
   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/async/wildfly-10.1.0.Final-Afirma/standalone/deployments" (en función de si se trata de un nodo de servicio o de servicios asíncronos) si no se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

   O

   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/services-BC/wildfly-10.1.0.Final-Afirma/standalone/deployments" o
   "CDInstalacion/@firma_6.4.0_001/**Instalacion**/resources/async-BC/wildfly-10.1.0.Final-Afirma/standalone/deployments" (en función de si se trata de un nodo de servicio o de servicios asíncronos) si se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

   Incluidos todos los subdirectorios.

4) Añadir el siguiente fragmento al final del fichero JBOSS_HOME/standalone/configuration/staticAfirmaConfig.properties:

```
#--------------------------------------------------------------------------------
# PROPIEDADES TSA
#--------------------------------------------------------------------------------
# Bandera donde se indicará si se incluye en la petición SOAP a la TSA  el identificador de la aplicación de @firma (true) o no (false).
tsa.include.afirmaApp = false
```

5) Arrancar la instancia JBOSS del nodo (JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startWildFlyAfirma.sh). Es importante que el nodo de administración se encuentre actualizado y arrancado antes de proceder al arranque del resto de nodos.

5.4 Instalación de los Esquemas de BD desde cero

5.4.1 Consideraciones previas

El sistema necesita dos tipos de bases de datos: Centralizadas y locales.

Las centralizadas serán Oracle y deben ser visibles desde todos los nodos del sistema.

Las locales son bases de datos HSQLDB y estarán instanciadas en cada nodo sin visibilidad hacia el resto de nodos.

Recomendamos encarecidamente que la instalación y el mantenimiento de los esquemas de Base de Datos de la plataforma @firma lo realice personal técnico cualificado en Base de Datos (preferiblemente un DBA) que esté familiarizado con Oracle 11g release 2.

5.4.2 Oracle

Para llevar a cabo esta instalación se requiere de una máquina (independiente a la plataforma @firma) con un Servidor de Base de Datos Oracle 11g release 2 correctamente instalado y funcionando y con las herramientas administrativas básicas, entre ellas “sqlplus”.
Se parte de que se ha creado un SID BD nuevo o ya existe uno que albergará a los cuatro esquemas de BD necesarios.

Los cuatro esquemas centralizados de BD de la plataforma @firma se instalan automáticamente ejecutando una serie de scripts de creación para cada uno de ellos, que deben ser configurados previamente, adaptándolos a las necesidades de cada caso concreto.
Si la máquina desde la que se ejecuta el script no es el mismo servidor de BD, se debe instalar el sqlplus en esa máquina y configurar correctamente el fichero tnsnames.ora.

A continuación, se describen los parámetros a configurar en el script de creación en las diferentes plataformas y como realizar la ejecución del mismo.

Sistema de Gestión de Procesos

En este esquema se almacena la información necesaria para el procesado de peticiones WS asíncronas.
El Sistema de Gestión de Procesos está compuesto básicamente por los siguientes elementos:
- Un Tablespace: MPM_TABLESPACE.
- Dos Usuarios: MPMOWNER, MPMUSER.
- Varias Tablas de datos.
- Varios Sinónimos.
- Varias secuencias.

A continuación, se describe cada uno de estos elementos:

1) Tablespace MPM_TABLESPACE: Espacio de Datos utilizado para almacenar todos los datos correspondientes a la gestión de peticiones asíncronas. Por defecto, el tamaño indicado en los scripts de creación para este TableSpace es de 128 MB.
2) Usuario MPMOWNER: Propietario de las tablas de datos del sistema de gestión de procesos y con todos los permisos sobre las mismas. **Este usuario es utilizado únicamente por el Administrador de Bases de Datos (DBA) y ninguna otra persona ha de tener acceso a él**. Las tablas son las siguientes:
    1) Tabla REQUESTS.
    2) Tabla NODES.
    3) Tabla PROCESSES.
    4) Tabla SEQUENCES_PROCESSES.
    6) Secuencia SQ_REQUESTS: para la clave de la tabla REQUESTS.
    7) Secuencia SQ_PROCESSES: para la clave de la tabla PROCESSES.

3) Usuario MPMUSER: Usuario utilizado para conectarse desde el Servidor de Firma al Sistema de Gestión de Procesos, tiene permisos sobre las tablas de datos de MPMOWNER y el permiso de conexión.

4) Sinónimos: proporcionan facilidad para el acceso a las tablas de datos de MPMOWNER, en concreto permiten referenciar una tabla sin precederla por el nombre usuario.

Sistema de Quartz

En este esquema se almacena toda la información que necesita Quartz para gestionar las distintas tareas programadas en cluster.
El Sistema de Quartz está compuesto básicamente por los siguientes elementos:
- Un Tablespace: QUARTZ_TABLESPACE.
- Un Usuario: QUARTZOWNER.

- Varias Tablas de datos.

A continuación, se describe cada uno de estos elementos:

1) Tablespace QUARTZ_TABLESPACE: Espacio de Datos utilizado para almacenar todos los datos necesarios por Quartz para la gestión de tareas en cluster. Por defecto, el tamaño indicado en los scripts de creación para este TableSpace es de 32 MB.

2) Usuario QUARTZOWNER: Propietario de las tablas de datos del sistema de Quartz y con todos los permisos sobre las mismas. Las tablas son las siguientes:
    1) Tabla QRTZ_JOB_DETAILS.
    2) Tabla QRTZ_TRIGGERS.
    3) Tabla QRTZ_SIMPLE_TRIGGERS.
    4) Tabla QRTZ_CRON_TRIGGERS.
    5) Tabla QRTZ_SIMPROP_TRIGGERS.
    6) Tabla QRTZ_BLOB_TRIGGERS.
    7) Tabla QRTZ_CALENDARS.
    8) Tabla QRTZ_PAUSED_TRIGGER_GRPS.
    9) Tabla QRTZ_FIRED_TRIGGERS.
    10) Tabla QRTZ_SCHEDULER_STATE.
    11) Tabla QRTZ_LOCKS.

Sistema de Validación

En este esquema se almacenan las CRLs descargadas de los Prestadores de servicios de Certificación configurados para ello. Este esquema se corresponde con el nivel 2 del sistema de caché de estados de certificados.
El Sistema de Validación está compuesto básicamente por los siguientes elementos:
- Un Tablespace: VALIDATION_TABLESPACE.
- Dos Usuarios: VALIDAOWNER, VALIDATION.
- Varias Tablas de datos.
- Varios Sinónimos.
- Varias secuencias.

A continuación, se describe cada uno de estos elementos:

1) Tablespace VALIDATION_TABLESPACE: Espacio de Datos utilizado para almacenar las CRLs obtenidas de distintos prestadores y así actuar como caché de segundo nivel. Por defecto, el tamaño indicado en los scripts de creación para este TableSpace es de 512 MB.
2) Usuario VALIDAOWNER: Propietario de las tablas de datos del sistema de validación y con todos los permisos sobre las mismas. **Este usuario es utilizado únicamente por el Administrador de Bases de Datos (DBA) y ninguna otra persona ha de tener acceso a él**. Las tablas son las siguientes:
    1) Tabla CRLS.

2) Tabla DELTACRLS.
3) Tabla CRLS_SEGMENTED_LDAP.
4) Secuencia SQ_CRL: para la clave de la tabla CRLS.
5) Secuencia SQ_DELTACRL: para la clave de la tabla DELTACRLS.

3) Usuario VALIDATION: Usuario utilizado para conectarse desde el Servidor de @firma al Sistema de Validación, tiene permisos sobre las tablas de datos de VALIDAOWNER y el permiso de conexión.
4) Sinónimos: proporcionan facilidad para el acceso a las tablas de datos de VALIDAOWNER, en concreto permiten referenciar una tabla sin precederla por el nombre usuario.

Sistema Transaccional

En este esquema de BD se almacenan las transacciones de firma realizadas y se custodian los documentos firmados, en el caso de que la política de "almacenamiento de documentos" de la aplicación en cuestión así se encuentre configurada. En el caso de que no se almacenen los documentos, se almacenará únicamente la transacción de firma electrónica. También se almacenan según la configuración aplicada, las peticiones recibidas y respuestas generadas de los distintos servicios.
El Sistema Transaccional está compuesto básicamente por los siguientes elementos:
- Tres Tablespaces: TRANSACTIONAL_TS, TRANSACTIONAL_BLOB_TS y TRANSACTIONAL_INDEX_TS DATAFILE.
- Dos Usuarios: TRANSOWNER y AFIRMATRANS.
- Varias Tablas de datos.
- Varios Sinónimos.
- Varias secuencias.

A continuación, se describe cada uno de estos elementos:

1) Tablespace TRANSACTIONAL_TS: Espacio de Datos utilizado para almacenar la estructura transaccional de las operaciones realizadas en la plataforma. Por defecto, el tamaño indicado en los scripts de creación para este TableSpace es de 3 Gb.
2) Tablespace TRANSACTIONAL_BLOB_TS: Espacio de Datos utilizado para almacenar todos los blobs asociados a las firmas y documentos almacenados. Por defecto, el tamaño indicado en los scripts de creación para este TableSpace es de 6 Gb.
3) Tablespace TRANSACTIONAL_INDEX_TS DATAFILE: Espacio de Datos utilizado para almacenar los índices necesarios de los dos tablespaces anteriores. Por defecto, el tamaño indicado en los scripts de creación para este TableSpace es de 1 Gb.
4) Usuario TRANSOWNER: Propietario de las tablas de datos del sistema transaccional y con todos los permisos sobre las mismas. **Este usuario es utilizado únicamente por el Administrador de Bases de Datos (DBA) y ninguna otra persona ha de tener acceso a él**. Las tablas y secuencias son las siguientes:
    1) Tabla ALARMS.
    2) Tabla DOCCONTENTS.
    3) Tabla DOCHASHS.
    4) Tabla FEDSTATISTICS.

5) Tabla FEDSTATISTICSAPP.
6) Tabla FEDSTATISTICSPENDING.
7) Tabla FEDSTATISTICSPLATFORM.
8) Tabla FEDSTATISTICSPSC.
9) Tabla FEDSTATISTICSSERVICE.
10) Tabla FEDSTATISTICSTIME.
11) Tabla FEDSTATISTICSTYPECERT.
12) Tabla FIELDTYPES.
13) Tabla HASHALGS.
14) Tabla MESSAGES.
15) Tabla MSGCONTENTS.
16) Tabla MSGHASHS.
17) Tabla MSGTYPES.
18) Tabla SERVICECONF.
19) Tabla SERVICETYPES.
20) Tabla SIGCONTENTS.
21) Tabla SIGDOCUMENTS.
22) Tabla SIGFORMATS.
23) Tabla SIGHASHS.
24) Tabla SIGNATURES.
25) Tabla STATSBUFFER.
26) Tabla SV_TRANSACTIONS.
27) Tabla TRACECONF.
28) Tabla TRACETYPES.
29) Tabla TRANSACTIONDATAS.
30) Tabla TR_FIELDS.
31) Tabla TR_TRACES.
32) Secuencia ALARMPK_SEQ: para la clave de la tabla ALARMS.
33) Secuencia DOCCONTENTPK_SEQ: para la clave de la tabla DOCCONTENTS.
34) Secuencia DOCHASHPK_SEQ: para la clave de la tabla DOCHASHS.
35) Secuencia FEDSTATISTICSAPPPK_SEQ: para la clave de la tabla FEDSTATISTICSAPPPK.
36) Secuencia FEDSTATISTICSPENDINGPK_SEQ: para la clave de la tabla FEDSTATICSPENDINGPK.
37) Secuencia FEDSTATISTICSPK_SEQ: para la clave de la tabla FEDSTATISTICSPK.
38) Secuencia FEDSTATISTICSPLATFORMPK_SEQ: para la clave de la tabla FEDSTATISTICSPLATFORMPK.
39) Secuencia FEDSTATISTICSPSCPK_SEQ: para la clave de la tabla FEDSTATISTICSPSCPK.
40) Secuencia FEDSTATISTICSSERVICEPK_SEQ: para la clave de la tabla FEDSTATISTICSSERVICEPK.
41) Secuencia FEDSTATISTICSTIMEPK_SEQ: para la clave de la tabla FEDSTATISTICSTIMEPK.

42) Secuencia FEDSTATISTICSTYPECERTPK_SEQ: para la clave de la tabla FEDSTATISTICSTYPECERTPK.
43) Secuencia MESSAGEPK_SEQ: para la clave de la tabla MESSAGES.
44) Secuencia MSGCONTENTPK_SEQ: para la clave de la tabla MSGCONTENTS.
45) Secuencia MSGHASHPK_SEQ: para la clave de la tabla MSGHASHS.
46) Secuencia SIGCONTENTPK_SEQ: para la clave de la tabla SIGCONTENTS.
47) Secuencia SIGDOCUMENTPK_SEQ: para la clave de la tabla SIGDOCUMENTS.
48) Secuencia SIGHASHPK_SEQ: para la clave de la tabla SIGHASHS.
49) Secuencia STATSBUFFERPK_SEQ: para la clave de la tabla STATSBUFFERPK.
50) Secuencia TRANSACTIONDATAPK_SEQ: para la clave de la tabla TRANSACTIONDATAS.
51) Secuencia TR_FIELDPK_SEQ: para la clave de la tabla TR_FIELDS.
52) Secuencia TR_TRACEPK_SEQ: para la clave de la tabla TR_TRACES.

5) Usuario AFIRMATRANS: Usuario utilizado para conectarse desde el Servidor de Firma al Sistema Transaccional, tiene permisos sobre las tablas de datos de TRANSOWNER y el permiso de conexión.

6) Sinónimos: proporcionan facilidad para el acceso a las tablas de datos de TRANSOWNER, en concreto permiten referenciar una tabla sin precederla por el nombre usuario.

Ejecución de Scripts

Para la creación de los esquemas de BD descritos anteriormente se entregan una serie de scripts sql y un archivo “orden.txt”, para cada esquema concreto.
El proceso consiste únicamente en ejecutar los scripts, en el Sistema de Gestión de Base de Datos, en el orden establecido en el archivo “orden.txt” y con el usuario indicado en dicho archivo. Previo a la ejecución de dichos scripts, hay que editarlos para establecer la contraseña de los usuarios a crear y la ruta para los tablespaces.
Los scripts se encuentran dentro del CD de Instalación en la ruta “Instalacion/ScriptsSQL/”.

Se recomienda que la ejecución de estos scripts la realice personal técnico cualificado en Base de Datos, preferiblemente un DBA, que esté familiarizado con Oracle 11g Release 2.

5.4.3 HSQLDB:

Se dispondrá de dos bases de datos locales HSQLDB en cada uno de los nodos. Una para el esquema de Alarmas y otra para el esquema de configuración.

La base de datos de configuración será administrada desde el nodo de administración y dispone de un mecanismo de sincronización a través de servicios REST para garantizar que la información que se modifique en el nodo de administración sea trasladada al resto de nodos.

La base de datos de alarmas se suministra ya creada.

Para crear la base de datos de configuración se suministra un componente standalone que permite dos opciones: Crearla "limpia", o realizar una migración desde versiones anteriores ORACLE o HSQLDB. El mecanismo se describirá más adelante.

Base de datos de Alamas

En este esquema se almacenan la información necesaria para la monitorización y gestión en tiempo real de las alarmas (para gestionar de forma automática el bloqueo de las alarmas), además de los resúmenes de estas (se almacena hasta su envío los mensajes de eventos que se van produciendo relacionando alarma y resumen).

El Sistema de Monitorización (de alarmas) está compuesto básicamente por los siguientes elementos:

1) Tabla SUMMARIES_EVENTS.
2) Secuencia SQ_SUMMARIES_EVENTS: para la clave de la tabla SUMMARIES_EVENTS.

Base de datos de Configuración

Esquema fundamental para la plataforma @firma 6.4. En este esquema se almacenará toda la configuración relacionada con prestadores, políticas de validación, políticas de firma, aplicaciones, usuarios, almacenes de certificados… etc.

El Sistema de Configuración está compuesto básicamente por los siguientes elementos:

1) Tabla C_TSA_AUTHENTICATION_TYPE.
2) Tabla C_TSA_HASH_ALGORITHM.
3) Tabla C_RFC3161_CONNECTION_TYPE.
4) Tabla C_TSA_COMMUNICATION_TYPE.
5) Tabla TSA_WEB_SERVICE.
6) Tabla TSA_RFC3161_SERVICE.
7) Tabla TSA.
8) Tabla X_APP_TSA.
9) Tabla C_OU_IDENTIFICATOR_TYPE.
10) Tabla C_DIR3_SERVICE.
11) Tabla DIR3_UPDATE.
12) Tabla C_AUTONOMOUS_COMMUNITY.
13) Tabla C_PROVINCE.
14) Tabla CACHE.
15) Tabla C_AFIRMA_VERSION.
16) Tabla MAIL_SERVER.
17) Tabla CRL_DOWNLOAD_TASK.
18) Tabla SIGNATURE_POLICY_PARSER.
19) Tabla SIGNATURE_POLICY_VERSION.

20) Tabla ASN1_POLICY.
21) Tabla C_DOWNLOAD_TYPE.
22) Tabla SIGNATURE_POLICY.
23) Tabla XML_POLICY.
24) Tabla USER_AFIRMA.
25) Tabla MANAGER_TYPE.
26) Tabla CONTENTS_MANAGER.
27) Tabla ADDITIONAL_PARAMETER.
28) Tabla X_CM_ADDITIONAL_PARAMETER.
29) Tabla X_DELEGATED_MANAGEMENT_USER.
30) Tabla X_AP_CONTENTS_MANAGER.
31) Tabla C_OCSP_RESTRICTION_MODE.
32) Tabla C_OCSP_RQ_SIGNATURE_MODE.
33) Tabla C_SIG_VER_POLICY_RESULT.
34) Tabla C_SECURITY_TOKEN_IM.
35) Tabla X_SIGNATURE_CERTIFICATE.
36) Tabla ORGANIZATIONAL_UNIT.
37) Tabla C_RESPONSE_SIGNATURE_FORMAT.
38) Tabla APPLICATION.
39) Tabla C_STATUS_CERTIFICATES.
40) Tabla SYSTEM_CERTIFICATE.
41) Tabla X_CERT_POLICY_RESTRICTION.
42) Tabla C_DATA_TYPE.
43) Tabla X_OCSP_CERTIFICATE.
44) Tabla C_SIGNATURE_ALGORITHM.
45) Tabla OCSP_SERVER.
46) Tabla C_ASSOCIATION_TYPE.
47) Tabla MAPPING.
48) Tabla LOGICAL_FIELD.
49) Tabla C_ACCESS_TYPE.
50) Tabla C_OCSP_RQ_REQUESTORNAME_TYPE.
51) Tabla VALIDATION_METHOD.
52) Tabla C_HASH_ALGORITHM.
53) Tabla C_AUTHENTICATION_TYPE.
54) Tabla POLICY.
55) Tabla C_VALIDATION_METHOD_TYPE.
56) Tabla C_REPOSITORY_TYPE.
57) Tabla X_CERT_TYPE_POLICY.
58) Tabla X_VALIDATION_METHOD_POLICY.
59) Tabla X_VALIDATION_METHOD_HEADER.
60) Tabla C_PSC_STATUS.
61) Tabla PSC.
62) Tabla C_EUROPEAN_BRIDGE_STATUS.
63) Tabla C_OPERATION_TYPE.

64) Tabla CERTIFICATE_TYPE.
65) Tabla DISCRIMINATOR.
66) Tabla C_PATTERN_TYPE.
67) Tabla X_ALARM_ADDRESSEE.
68) Tabla X_APPLICATION_RESPONSIBLE.
69) Tabla ALARM.
70) Tabla TASK.
71) Tabla X_TASK_PLANNER.
72) Tabla X_TASK_KEYSTORE.
73) Tabla PLANNER.
74) Tabla C_PLANNER_TYPE.
75) Tabla X_ALARM_SUMMARY.
76) Tabla ADDRESSEES.
77) Tabla C_ADDRESSEE_TYPES.
78) Tabla X_SUMMARY_ADDRESSEE.
79) Tabla SUMMARY.
80) Tabla C_NOMENCLATURE_TYPE.
81) Tabla APPLICATION_USER.
82) Tabla KEYSTORE.
83) Tabla X_WS_CERTIFICATE.
84) Tabla C_SERVICE.
85) Tabla X_APPLICATION_SERVICE.
86) Tabla PARTITION.
87) Tabla X_VAL_ME_CRL_DOWN_TASK.
88) Tabla PROPERTY.
89) Tabla C_ORG_UNIT_TYPE.
90) Tabla SCHEDULER.
91) Tabla C_PROPERTY_GROUP.
92) Tabla C_DIR3_CHANGE_CAUSE.
93) Tabla DIR3_OU_CHANGE.
94) Tabla C_DIR3_OU_STATUS.
95) Tabla SMPP_SERVERS.
96) Tabla FEDPLATFORM.
97) Tabla RESPONSIBLES.
98) Tabla X_DISC_CERT_TYPE.
99) Tabla C_BASIC_OPERATOR.
100) Tabla CERT_PSC.
101) Tabla SECONDARY_VALSER.
102) Tabla VALET_SERVER.
103) Secuencia SQ_ADDITIONAL_PARAMETER: Secuencia necesaria para la clave de la tabla ADDITIONAL_PARAMETER.
104) Secuencia SQ_ADDRESSEES: Secuencia necesaria para la clave de la tabla ADDRESSEES.

105) Secuencia SQ_FEDPLATFORMPK :Secuencia necesaria para la clave de la tabla FEDPLATFORM.
106) Secuencia SQ_APPLICATION: Secuencia necesaria para la clave de la tabla APPLICATION.
107) Secuencia SQ_APPLICATION_USER: Secuencia necesaria para la clave de la tabla APPLICATION_USER.
108) Secuencia SQ_ASN1_POLICY: Secuencia necesaria para la clave de la tabla ASN1_POLICY.
109) Secuencia SQ_CERTIFICATE_TYPE: Secuencia necesaria para la clave de la tabla 110) CERTIFICATE_TYPE.
110) Secuencia SQ_CONTENTS_MANAGER: Secuencia necesaria para la clave de la tabla CONTENTS_MANAGER.
111) Secuencia SQ_CRL_DOWNLOAD_TASK: Secuencia necesaria para la clave de la tabla CRL_DOWNLOAD_TASK.
112) Secuencia SQ_DIR3_OU_CHANGE: Secuencia necesaria para la clave de la tabla DIR3_OU_CHANGE.
113) Secuencia SQ_DIR3_UPDATE: Secuencia necesaria para la clave de la tabla DIR3_UPDATE.
114) Secuencia SQ_DISCRIMINATOR: Secuencia necesaria para la clave de la tabla DISCRIMINATOR.
115) Secuencia SQ_LOGICAL_FIELD: Secuencia necesaria para la clave de la tabla LOGICAL_FIELD.
116) Secuencia SQ_MANAGER_TYPE: Secuencia necesaria para la clave de la tabla MANAGER_TYPE.
117) Secuencia SQ_MAPPING: Secuencia necesaria para la clave de la tabla MAPPING.
118) Secuencia SQ_ORGANIZATIONAL_UNIT: Secuencia necesaria para la clave de la tabla ORGANIZATIONAL_UNIT.
119) Secuencia SQ_PARTITION: Secuencia necesaria para la clave de la tabla PARTITION.
120) Secuencia SQ_PLANNER: Secuencia necesaria para la clave de la tabla PLANNER.
121) Secuencia SQ_POLICY: Secuencia necesaria para la clave de la tabla POLICY.
122) Secuencia SQ_PSC: Secuencia necesaria para la clave de la tabla PSC.
123) Secuencia SQ_SIGNATURE_POLICY Secuencia necesaria para la clave de la tabla SIGNATURE_POLICY.
124) Secuencia SQ_SUMMARY: Secuencia necesaria para la clave de la tabla SUMMARY.
125) Secuencia SQ_SYSTEM_CERTIFICATE: Secuencia necesaria para la clave de la tabla SYSTEM_CERTIFICATE.
126) Secuencia SQ_TASK: Secuencia necesaria para la clave de la tabla TASK.
127) Secuencia SQ_TSA: Secuencia necesaria para la clave de la tabla TSA.
128) Secuencia SQ_TSA_WEB_SERVICE: Secuencia necesaria para la clave de la tabla TSA_WEB_SERVICE.
129) Secuencia SQ_TSA_RFC3161_SERVICE: Secuencia necesaria para la clave de la tabla TSA_RFC3161_SERVICE.
130) Secuencia SQ_USER_AFIRMA: Secuencia necesaria para la clave de la tabla USER_AFIRMA.

131) Secuencia SQ_VALIDATION_METHOD: Secuencia necesaria para la clave de la tabla VALIDATION_METHOD.
132) Secuencia SQ_XML_POLICY: Secuencia necesaria para la clave de la tabla XML_POLICY.
133) Secuencia SQ_SIGNATURE_POLICY_PARSER: Secuencia necesaria para la clave de la tabla SIGNATURE_POLICY_PARSER.
134) Secuencia SQ_SIGNATURE_POLICY_VERSION: Secuencia necesaria para la clave de la tabla SIGNATURE_POLICY_VERSION.
135) Secuencia SQ_RESPONSIBLES: Secuencia necesaria para la clave de la tabla RESPONSIBLES.
136) Secuencia SQ_CERT_PSC: Secuencia necesaria para la clave de la tabla CERT_PSC.
137) Secuencia SQ_MAIL_SERVER: Secuencia necesaria para la clave de la tabla MAIL_SERVER.

Instalación/Migración de la base de datos de configuración

El proceso de instalación desde cero o migración desde un esquema Oracle será descrito durante el procedimiento de instalación de los nodos en el apartado 5.5.2.

5.5 Instalación de @Firma v6.4.0_001 desde cero

Antes de comenzar con la instalación del servidor es necesario realizar la instalación de los esquemas Oracle definidos en el punto 5.4.2.

En este apartado se detallará el **proceso de instalación de la plataforma @firma v6.4.0_001 desde cero**.

Debido al cambio tecnológico realizado en esta versión de la plataforma @firma, no se definen procedimientos de actualización desde versiones, se debe realizar una instalación “limpia”.

La instalación de @firma necesita un servidor de aplicaciones JBOSS (versión wildfly-10.1.0.Final) para cada uno de los nodos que se decidan desplegar (mínimo uno para administración y otro para servicios).

El servidor viene incluido en el kit de instalación proporcionado.

También se proporciona un componente para la instalación o la migración de la base de datos HSQLDB local de configuración desde cero o bien desde una versión anterior 6.2.2_004 o superior.

En los siguientes apartados se detallará la instalación y configuración de cada uno de estos componentes.

5.5.1 Consideraciones previas

La plataforma @firma ha sido diseñada y certificada para su ejecución en servidores Red Hat 6.5. Al cierre de este documento, no se garantiza el funcionamiento de la plataforma con otros SO, si bien es cierto que al haber sido desarrollada 100% con tecnología Java, en teoría el coste de la implantación en otro tipo de sistemas sería mínimo.

Previo a la instalación del Servidor de Firma es necesario realizar los siguientes pasos:

1) Instalar la JDK 1.8.0_131 junto con su extensión criptográfica JCE (no se proporcionan en el CD).

2) Instalar el Cliente Oracle (versión 11g) en la máquina Servidor de Firma. Este elemento es necesario para que el Servidor de Firma pueda establecer conexión con el sistema de Custodia.

3) **IMPORTANTE. La plataforma necesita para su funcionamiento las librerías del proveedor criptográfico IAIK. Estas librerías ya se encuentran incluidas. Los distintos componentes son:**

    - **JCE v5.6**
    - **CMS v5.1**
    - **TSP v2.32**
    - **XSEC v2.14**
    - **XADES v2.13**
    - **eccelerate v6.0.1**

Todas estas librerías vienen incluidas dentro del desplegable correspondiente del CD de Instalación.

Advertencias:

- La plataforma utiliza para su funcionamiento un conjunto de carpetas, ficheros y estructuras de datos con un formato, contenido y nomenclatura determinados. A menos que se indique lo contrario en este manual, cualquier modificación - ya sea parcial o total de estos - puede provocar un mal funcionamiento de la plataforma. Por tanto, se recomienda no realizar modificaciones, tanto en el contenido del CD de Instalación antes del despliegue, como en el contenido generado por el proceso de instalación.

- No se garantiza que, en versiones futuras de la plataforma, los ficheros y estructuras que utiliza para su funcionamiento, vayan a existir con los mismos formatos y/o contenidos. Por esta razón, se recomienda no utilizar estos ficheros y estructuras de datos como fuente de datos para aplicaciones o procesos externos a la plataforma @firma.

4) Comprobar el estado de generación de entropía del sistema, y si fuera necesario, aumentar su disponibilidad mediante el uso de una herramienta-servicio adicional. Para más información al respecto, se remite al lector al anexo 1.

5.5.2 Instalación en Red Hat 6.5

A continuación, se detallará el proceso de instalación de la plataforma en un Red Hat 6.5.

Los ficheros necesarios para realizar la instalación de la plataforma @firma v6.4.0_001 se encuentran ubicados en la carpeta "Instalacion" del CD. El proceso de instalación se divide en dos fases, el despliegue, donde se copiarán los componentes necesarios a su ubicación final y se generarán los ficheros necesarios para su funcionamiento, y la configuración de la plataforma donde se modificarán los ficheros de configuración de la misma para su primer arranque.

El sistema dispone de tres tipos de nodos (administración, servicios y servicios asíncronos). Debe realizarse una instalación del servidor JBOSS por cada uno de estos nodos. Como mínimo, son necesarios dos, uno de administración y otro de servicios.

A continuación, se detallan los pasos a realizar para llevar a cabo la instalación.

Despliegue del nodo de administración

El despliegue comenzará por el nodo de administración y continuará con el resto de nodos de servicio y servicios asíncronos que se deseen instalar.

1) Asegurarse de haber definido la variable *JAVA_HOME* al instalar la JDK 1.8.0_131.

2) Incluir la librería JDBC del cliente Oracle instalado en el sistema en la ruta "CDInstalacion/@firma_6.4.0_001/Instalación/wildfly-10.1.0.Final-Afirma/modules/system/layers/base/Oracle/driver/oci/main/ojdbc6.jar" y en "CDInstalacion/@firma_6.4.0_001/afirma6-standalone-importDDBB/lib/ojdbc6.jar"

3) Copiar el directorio "CDInstalacion/@firma_6.4.0_001/Instalación/wildfly-10.1.0.Final-Afirma" y todos sus subdirectorios a la ruta final de instalación, desde ahora, JBOSS_HOME .

4) Copiar sobre la ruta final de instalación el contenido de:

   "CDInstalacion/@firma_6.4.0_001/Instalación/resources/admin/wildfly-10.1.0.Final-Afirma" si no se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

O

"CDInstalacion/@firma_6.4.0_001/Instalación/resources/admin-BC/wildfly-10.1.0.Final-Afirma" si se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5.

incluidos todos los subdirectorios.

De esta forma, se añadirán los ficheros específicos del nodo de administración a la instalación de wildfly estándar. En ningún caso debe eliminarse el contenido previo.

5) Copiar el directorio "CDInstalacion/@firma_6.4.0_001/afirma6-standalone-importDDBB" y todos sus subdirectorios a la ruta deseada, desde ahora STANDALONE_IMPORT_HOME. Este componente migrará desde versiones anteriores o creará la base de datos de configuración vacía HSQLDB.

6) Editar el fichero "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh" hay que realizar los siguientes cambios:

   - Indicar en la primera línea la ruta de instalación de java 1.8.0.131
   - Sustituir en todo el fichero las apariciones del token "RPLC.CONF_SCHM.PORT" por el puerto en el que se iniciará la base de datos de configuración HSQLDB.
   - Sustituir en todo el fichero las apariciones del token "RPLC.CONF_SCHM.IP" por la IP/interfaz en el que se iniciará la base de datos de configuración HSQLDB. Usualmente es la misma IP de binding que la del servidor de aplicaciones.

7) Editar el fichero "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopConfigDatabase.sh" hay que realizar los siguientes cambios:

   - Sustituir en todo el fichero las apariciones del token "RPLC.CONF_SCHM.PORT" por el puerto en el que se iniciará la base de datos de configuración HSQLDB.

8) Ejecutar el script "JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/startConfigDatabaseMaster.sh" para arrancar la base de datos de configuración. Este primer arranque está destinado a la creación o migración de la base de datos.

9) Configurar y ejecutar el componente afirma6-standalone-importDDBB.para ello:

   -Editar el fichero "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/configuration/staticImportDDBBConfig.properties" sustituyendo

   RPLC.CONF_SCHM.HSQLDB.IP por la IP/interfaz en el que se iniciará la base de datos de configuración HSQLDB configurada en el paso 6).

   RPLC.CONF_SCHM.HSQLDB.PORT por el puerto en el que se iniciará la base de datos de configuración HSQLDB configurado en el paso 6).

Las siguientes propiedades relativas a la configuración Oracle serán necesarias solo y exclusivamente si se desea migrar desde una versión anterior (6.2.2_XXX). No será necesario establecerlas si se desea realizar una instalación en limpio desde cero. Las propiedades son las siguientes.

RPLC.CONF_SCHM.ORACLE.CNNTNURL por la URL de conexión hacia la base de datos que contiene el esquema de configuración origen (jdbc:oracle:thin:@IP:PORT:SID) que se desea migrar.

RPLC.CONF_SCHM.ORACLE.OWNER por el identificador del usuario propietario del esquema de configuración origen que se desea migrar.

RPLC.CONF_SCHM.ORACLE.PASS por el password del usuario propietario del esquema de configuración origen que se desea migrar.

Establecer valores para las propiedades

database.path con la ruta absoluta hacia la base de datos local de configuración "JBOSS_HOME/wildfly-10.1.0.Final-Afirma /standalone/configuration/LocalHSQLDBDataBase/configuration"

y

source.version con la versión origen desde la que se desea migrar o actualizar. Los valores posibles son:

- Dejarlo vacío: La instalación se realizará en limpio desde cero sin migrar información desde ninguna base de datos Oracle.
- 6_2_2_004, 6_2_2_009, 6_2_2_011 o 6_2_2_012: La instalación migrará la información de la base de datos Oracle de la versión 6.2.2 anterior y aplicará las actualizaciones necesarias para la versión 6.4.0_000.
- 6_4_X: No procede. Este valor se establecerá si se está utilizando el componente para realizar una actualización desde una versión anterior de la 6.4.

-Editar el script "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/generateScript-importDDBB.sh" indicando en la primera línea (JDK_LIB=/RUTA_JDK/jdk1.8.0_131) la ruta hacia la jdk 1.8.0_131

-Ejecutar el script "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/generateScript-importDDBB.sh" para crear la base de datos comprobando en el log generado en "STANDALONE_IMPORT_HOME/afirma6-standalone-importDDBB/logs" que el proceso haya finalizado con éxito.

10) Ejecutar el script “JBOSS_HOME/wildfly-10.1.0.Final-Afirma/bin/stopConfigDatabase.sh” para parar la base de datos de configuración.

11) Realizar una copia de seguridad de la base de datos generada (directorio “JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/configuration/LocalHSQLDBDataBase/configuration”) para utilizarla en la instalación del resto de nodos.

Despliegue del resto de nodos (servicios y servicios asíncronos)

1) Asegurarse de haber definido la variable *JAVA_HOME* al instalar la JDK 1.8.0_131.

2) Incluir la librería JDBC del cliente Oracle instalado en el sistema en la ruta “CDInstalacion/@firma_6.4.0_001/Instalación/wildfly-10.1.0.Final-Afirma/modules/system/layers/base/Oracle/driver/oci/main/ojdbc6.jar”.

3) Copiar el directorio “CDInstalacion/@firma_6.4.0_001/Instalación/wildfly-10.1.0.Final-Afirma” y todos sus subdirectorios a la ruta final de instalación, desde ahora, JBOSS_HOME .

4) Copiar sobre la ruta final de instalación el contenido de:

   “CDInstalacion/@firma_6.4.0_001/Instalación/resources/services/wildfly-10.1.0.Final-Afirma” (para nodo de servicios) o “CDInstalacion/@firma_6.4.0_001/Instalación/resources/async/wildfly-10.1.0.Final-Afirma” (para nodo de servicios asíncronos) si no se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5,

   O

   “CDInstalacion/@firma_6.4.0_001/Instalación/resources/services-BC/wildfly-10.1.0.Final-Afirma” (para nodo de servicios) o “CDInstalacion/@firma_6.4.0_001/Instalación/resources/async-BC/wildfly-10.1.0.Final-Afirma” (para nodo de servicios asíncronos) si se desea mantener la retrocompatibilidad con el sistema de custodia de @firma 5,

   incluidos todos los subdirectorios.

   De esta forma, se copiarán los ficheros específicos del tipo nodo deseado a la instalación de wildfly estándar. En ningún caso debe eliminarse el contenido previo.

5) Copiar la base de datos de configuración generada en durante el despliegue del nodo de administración al directorio “JBOSS_HOME/wildfly-10.1.0.Final-Afirma/standalone/configuration/LocalHSQLDBDataBase/configuration”.

Configuración del Servidor de Aplicaciones JBOSS

El siguiente paso será configurar debidamente el servidor de aplicaciones JBOSS en cada uno de los nodos instalados. A continuación, se listan los ficheros a modificar junto con las modificaciones a realizar en cada uno.

1) JBOSS_HOME/bin/*.sh

Descripción. Conjunto de scripts ejecutables proporcionados con el servidor de aplicaciones.

Modificación. Hay que darle permisos de ejecución a los scripts.

2) JBOSS_HOME/bin/startConfigDatabase*.sh (master o salve en cada caso en función de si se trata de un nodo de servicio, servicio asíncrono o administración)

Hay que realizar los siguientes cambios:

- Indicar en la primera línea la ruta de instalación de java 1.8.0.131
- Sustituir en todo el fichero las apariciones del token “RPLC.CONF_SCHM.PORT” por el puerto en el que se iniciará la base de datos de configuración HSQLDB.
- Sustituir en todo el fichero las apariciones del token “RPLC.CONF_SCHM.IP” por la IP/interfaz en el que se iniciará la base de datos de configuración HSQLDB. Usualmente es la misma IP de binding que la del servidor de aplicaciones.

3) JBOSS_HOME/bin/stopConfigDatabase.sh

Hay que realizar los siguientes cambios:

- Sustituir en todo el fichero las apariciones del token “RPLC.CONF_SCHM.PORT” por el puerto en el que se iniciará la base de datos de configuración HSQLDB.

4) JBOSS_HOME/bin/startAlarmsDatabase.sh

Hay que realizar los siguientes cambios:

- Indicar en la primera línea la ruta de instalación de java 1.8.0.131
- Sustituir en todo el fichero las apariciones del token “RPLC.ALARMS_SCHM.PORT” por el puerto en el que se iniciará la base de datos de configuración HSQLDB.
- Sustituir en todo el fichero las apariciones del token “RPLC. ALARMS_SCHM.IP” por la IP/interfaz en el que se iniciará la base de datos de configuración HSQLDB. Usualmente es la misma IP de binding que la del servidor de aplicaciones.

5) JBOSS_HOME/bin/stopAlarmsDatabase.sh

Hay que realizar los siguientes cambios:

- Sustituir en todo el fichero las apariciones del token “RPLC. ALARMS_SCHM.PORT” por el puerto en el que se iniciará la base de datos de configuración HSQLDB.

6) JBOSS_HOME/bin/checkStatus.sh

Hay que realizar los siguientes cambios:

- Definir el valor adecuado a la variable JBOSS_INSTANCE_NAME_DIR, en la que hay que indicar el nombre de la referencia al directorio concreta del WidFly de @firma. Por defecto “afirma_wildfly101”.

- Sustituir en todo el fichero las apariciones del token “RPLC.CONF_SCHM.PORT” por el puerto en el que se iniciará la base de datos de configuración HSQLDB.

- Sustituir en todo el fichero las apariciones del token “RPLC.ALARM_SCHM.PORT” por el puerto en el que se iniciará la base de datos de alarmas HSQLDB.

7) JBOSS_HOME/bin/standalone.sh

Hay que realizar los siguientes cambios:

- Definir el valor adecuado a la variable JBOSS_INSTANCE_NAME, en la que hay que indicar el nombre de la referencia al directorio concreta del WidFly de @firma. Por defecto, “afirma_wildfly101”.

8) JBOSS_HOME/bin/startWildFlyAfirma.sh

Hay que realizar los siguientes cambios:

- Definir el valor adecuado a la variable JBOSS_INSTANCE_NAME_DIR, en la que hay que indicar el nombre de la referencia al directorio concreta del WidFly de @firma. Por defecto “afirma_wildfly101”.

9) JBOSS_HOME/bin/stopWildFlyAfirma.sh

Hay que realizar los siguientes cambios:

- Definir el valor adecuado a la variable JBOSS_INSTANCE_NAME_DIR, en la que hay que indicar el nombre de la referencia al directorio concreta del WidFly de @firma. Por defecto “afirma_wildfly101”.

- Sustituir en todo el fichero las apariciones del token “RPLC.JBOSS.IP_BINDING” por la IP/nombre de binding con la que se inicia el servidor de aplicaciones.

- Sustituir en todo el fichero las apariciones del token “RPLC.JBOSS.PORTMANAGEMENTNATIVE” por el puerto asignado para el “management-native” del servidor de aplicaciones.

10) JBOSS_HOME/bin/standalone.conf

Descripción. Es un fichero de configuración que permite la inclusión de parámetros específicos para el arranque del JBOSS.

Modificación. Se deben modificar los parámetros referentes a la memoria asignada a la JVM donde se va a ejecutar el servidor de aplicaciones. Para ello, se deberá modificar la línea:

```
JAVA_OPTS="-Xms1024m -Xmx2048m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -
XX:MinMetaspaceFreeRatio=20 -XX:MaxMetaspaceFreeRatio=50 -XX:+UseG1GC -XX:MaxNewSize=896m -
XX:NewSize=896m -XX:SurvivorRatio=6 -Djava.net.preferIPv4Stack=true"
```

Los parámetros que se deben establecer son los siguientes:

-Xms*:*

Indica el tamaño inicial, en bytes, que tendrá el pool de memoria. Este valor debe ser múltiplo de 1024 y mayor a 1MB. Añada la letra k o K para indicar kilobytes, m o M para indicar megabytes. El valor por defecto es de 256 MB. Ejemplos:

-Xms6291456
-Xms6144k
-Xms6m

Los valores establecidos en los parámetros “-XX:MaxNewSize” y “-XX:NewSize” se corresponden con 7/8 de lo establecido en “-Xms“.

-Xmx*:*

Especifica el tamaño máximo, en bytes, del pool de memoria. Este valor debe ser múltiplo de 1024 y mayor a 1MB. Añada la letra k o K para indicar kilobytes, m o M para indicar megabytes. El valor por defecto es de 2048 MB. Ejemplos:

-Xmx83886080
-Xmx81920k
-Xmx80m

-XX:MetaspaceSize:

Especifica el tamaño inicial asignado al almacenamiento de clases, métodos, etc… (el área de memoria donde la JVM almacena sus metadatos).

-XX:MaxMetaspaceSize:

Indica el tamaño máximo asignado al almacenamiento de clases, métodos, etc… (el área de memoria donde la JVM almacena sus metadatos). Si una aplicación carga en memoria una gran cantidad de clases, será necesario ampliar el espacio dedicado al almacenamiento de metadatos. Esto se lleva a cabo por medio del parámetro XX:MaxMetaspaceSize=n, donde n especifica el tamaño. Ejemplos:

- XX:MaxPermSize=83886080
- XX:MaxPermSize=81920k
- XX:MaxPermSize=80m

-XX:MaxNewSize y -XX:NewSize

Estas propiedades deben tener siempre asignado como valor 7/8 del valor mínimo de memoria establecido (propiedad Xms).

Dependiendo de la memoria física de la máquina y de la utilización de la misma el administrador del sistema debe de asignarle valores a dichos parámetros.

11) JBOSS_HOME/modules/system/layers/base/es/mpr/pltm/ws/main/conf/ws-pm-config.properties

Descripción. Fichero de propiedades de configuración usado por la plataforma de mensajería del Ministerio en el que se determinan las URL de los servicios.

Modificación. Solo en el caso de que se vaya a hacer uso de la plataforma de mensajería del Ministerio, se deben establecer las rutas hacia los distintos servicios que permiten su uso:

```
# Propiedades necesarias para establecer las conexiones con los distintos servicios
# ofrecidos por la Plataforma de Mensajeria del Ministerio.
servicioEnvioMensajes=<URL_MSG_PLTM>/PlataformaMensajeriaWS/EnvioMensajes
servicioSeguimientoMensajes=<URL_MSG_PLTM>/PlataformaMensajeriaWS/SeguimientoMensajes
servicioOperacionesMensajes=<URL_MSG_PLTM>/PlataformaMensajeriaWS/OperacionesMensajes
```

12) JBOSS_HOME/standalone/configuration/KeystoreSSL-Tomcat.jks

Descripción. Es el keystore que contiene el certificado de servidor utilizado para las conexiones SSL contra la plataforma.

Modificación. Se deben importar al keystore el certificado y su clave privada necesarios para la autenticación SSL de servidor. La plataforma incluye un certificado SSL de test que será recomendable sustituir. Además, inicialmente, el password del keystore es 12345, el cual es recomendable cambiar por una contraseña segura.

13) JBOSS_HOME/standalone/configuration/TrustStoreSSL-Tomcat.jks

Descripción. Es el keystore que contiene los certificados de confianza utilizado para las conexiones SSL con autenticación mútua contra la plataforma.

Modificación. Se deben importar al keystore aquellos certificados de confianza necesarios para la autenticación SSL de autenticación mútua. Por defecto, la plataforma incluye un certificado SSL de test que será recomendable sustituir. Además, inicialmente, el password del keystore es 12345, el cual es recomendable cambiar por una contraseña segura.

14) JBOSS_HOME/configuration/shared-quartz.properties

Hay que realizar los siguientes cambios:

- Establecer en la propiedad “org.quartz.dataSource.AfirmaQuartz.URL” la URL de conexión hacia la base de datos donde se encuentra el esquema de Quartz.

- Establecer en la propiedad “org.quartz.dataSource.AfirmaQuartz.user” el identificador correspondiente al usuario propietario para el esquema de Quartz.

- Establecer la propiedad “org.quartz.dataSource.AfirmaQuartz.password” el password de acceso para el usuario propietario del esquema de Quartz.

15) JBOSS_HOME/configuration/server-config.wsdd

Hay que realizar los siguientes cambios: (**NO ES NECESARIO EN EL NODO DE ADMINISTRACIÓN**)

- Sustituir en todo el fichero las apariciones de:

RPLC.WSDD.URL.ENDPOINT/afirmaws/services/

Por la URL absoluta donde se publican los servicios web para el entorno.

16) JBOSS_HOME/standalone/configuration/application-roles.properties

Descripción. Fichero de propiedades que determina los roles de los que dispondrán los usuarios aquí declarados para acceder a la consola web de JBoss.

Modificación. Inicialmente se incluye el usuario *afirmaApplicationUser* con los roles *PowerUser* y *BillingAdmin*. Este usuario puede ser eliminado, modificado o añadir otros según el administrador del sistema estime oportuno.

```
#
# Properties declaration of users roles for the realm 'ApplicationRealm'.
#
# This includes the following protocols: remote ejb, remote jndi, web, remote
# jms.
#
# Users can be added to this properties file at any time, updates after the
# server has started will be automatically detected.
#
# The format of this file is as follows: -
# username=role1,role2,role3
#
# A utility script is provided which can be executed from the bin folder to add
# the users: -
# - Linux
#  bin/add-user.sh
#
# - Windows
#  bin\add-user.bat
#
# The following illustrates how an admin user could be defined.
#
#admin=PowerUser,BillingAdmin,
#guest=guest

afirmaApplicationUser=PowerUser,BillingAdmin
```

17) JBOSS_HOME/standalone/configuration/application-users.properties

Descripción. Fichero de propiedades que determina el password asociado a los usuarios aquí declarados para acceder a la consola web de JBoss.

Modificación. Inicialmente se incluye el usuario *afirmaApplicationUser* con el password *12345*. Este usuario puede ser eliminado, modificado o añadir otros según el administrador del sistema estime oportuno.

```
#
# Properties declaration of users for the realm 'ApplicationRealm' which is the
# default realm.
#
# This includes the following protocols: remote ejb, remote jndi, web, remote
# jms.
#
```

```
# Users can be added to this properties file at any time, updates after the
# server has started will be automatically detected.
#
# The format of this realm is as follows: -
# username=HEX( MD5( username ':' realm ':' password))
#
# A utility script is provided which can be executed from the bin folder to add
# the users: -
# - Linux
#  bin/add-user.sh
#
# - Windows
#  bin\add-user.bat
#
# The following illustrates how an admin user could be defined.
#
#admin=2a0923285184943425d1f53ddd58ec7a

afirmaApplicationUser=00a596d374527fae7c9aa303951816c4
```

Tal y como se puede observar en el comentario incluído en el propio fichero, el password no se representa en claro, sino que es la representación hexadecimal del cálculo MD5 de concatenar “usuario:realm:password”. En concreto y de forma generica la definición sería la siguiente:

username=HEX(MD5(username ':' realm ':' password))

18) JBOSS_HOME/standalone/configuration/mgmt-gropus.properties

Descripción. Fichero de propiedades que determina los roles de los que dispondrán los usuarios aquí declarados para acceder a la consola mantenimiento de JBoss.

Modificación. Inicialmente se incluye el usuario *afirmaApplicationUser* con los roles *PowerUser* y *BillingAdmin*. Este usuario puede ser eliminado, modificado o añadir otros según el administrador del sistema estime oportuno.

```
#

# Properties declaration of users groups for the realm 'ManagementRealm'.

#

# This is used for domain management, users groups membership information is used to assign the user

# specific management roles.

#
```

```
# Users can be added to this properties file at any time, updates after the server has started
# will be automatically detected.
#
# The format of this file is as follows: -
# username=role1,role2,role3
#
# A utility script is provided which can be executed from the bin folder to add the users: -
# - Linux
#  bin/add-user.sh
#
# - Windows
#  bin\add-user.bat
#
# The following illustrates how an admin user could be defined.
#
#admin=PowerUser,BillingAdmin,
afirmaManagementUser=PowerUser,BillingAdmin
```

19) JBOSS_HOME/standalone/configuration/mgmt-users.properties

Descripción. Fichero de propiedades que determina el password asociado a los usuarios aquí declarados para acceder mediante la consola java a la información de consumo de memoria y servicios desplegados en JBoss.

Modificación. Inicialmente se incluye el usuario *afirmaManagementUser* con el password *12345*. Este usuario puede ser eliminado, modificado o añadir otros según el administrador del sistema estime oportuno.

```
#
# Properties declaration of users for the realm 'ManagementRealm' which is the
# default realm.
# Further authentication mechanism can be configured
```

```
# as part of the <management /> in standalone-full-ha.xml.
#
# Users can be added to this properties file at any time, updates after the
# server has started will be automatically detected.
#
# By default the properties realm expects the entries to be in the format: -
# username=HEX( MD5( username ':' realm ':' password))
#
# A utility script is provided which can be executed from the bin folder to add
# the users: -
# - Linux
#  bin/add-user.sh
#
# - Windows
#  bin\add-user.bat
#
# The following illustrates how an admin user could be defined.
#
#admin=2a0923285184943425d1f53ddd58ec7a
afirmaManagementUser=37ebfc4f9190f0a0c7523a6267f62003
```

Tal y como se puede observar en el comentario incluído en el propio fichero, el password no se representa en claro, sino que es la representación hexadecimal del cálculo MD5 de concatenar “usuario:realm:password”. En concreto y de forma generica la definición sería la siguiente:

username=HEX(MD5(username ':' realm ':' password))

20) JBOSS_HOME/standalone/configuration/afirma-log4j.xml

Se debe revisar el nivel de logging de cada uno de los appenders y categoría, y adaptarlo según las necesidades.

21) JBOSS_HOME/standalone/configuration/standalone.xml

Fichero XML que describe por módulos la configuración completa del servidor de aplicaciones.

En este fichero se configuran los datasources para el acceso a base de datos será necesario sustituir en la sección datasources:

- El token “RPLC.ALARMS_SCHM.IP” por la IP de binding con la que se inicia la base de datos de alarmas hsqldb, y el token “RPLC. ALARMS_SCHM.PORT” por el puerto con el que se inicia esta misma base de datos.

- El token “RPLC. ALARMS_SCHM.USER” por el identificador del usuario de acceso a la base de datos de alarmas hsqldb (por defecto es “sa”).

- El token “RPLC. ALARMS_SCHM.PASS” por el password del usuario de acceso a la base de datos de alarmas hsqldb (por defecto es la cadena vacía).

- El token “RPLC.CONF_SCHM.IP” por la IP de binding con la que se inicia la base de datos de configuración hsqldb, y el token “RPLC. CONF_SCHM.PORT” por el puerto con el que se inicia esta misma base de datos.

- El token “RPLC. CONF_SCHM.USER” por el identificador del usuario de acceso a la base de datos de configuración hsqldb (por defecto es “sa”).

- El token “RPLC. CONF_SCHM.PASS” por el password del usuario de acceso a la base de datos de configuración hsqldb (por defecto es la cadena vacía).

- Solo si se ha optado por la opción de mantener la retrocompatibilidad con el sistema de custodia de @firma 5, sustituir el token “RPLC.CUST_SCHM.CNNTNURL” por la URL de conexión hacia la base de datos que contiene el esquema de custodia de @firma 5 (jdbc:oracle:oci:@IP:PORT:SID.

- Solo si se ha optado por la opción de mantener la retrocompatibilidad con el sistema de custodia de @firma 5, sustituir el token “RPLC. CUST _SCHM.USER” por el identificador del usuario NO propietario del esquema de custodia.

- Solo si se ha optado por la opción de mantener la retrocompatibilidad con el sistema de custodia de @firma 5, sustituir el token “RPLC. CUST _SCHM.PASS” por el password del usuario NO propietario del esquema de custodia.

- El token “RPLC.MPM_SCHM.CNNTNURL” por la URL de conexión hacia la base de datos que contiene el esquema de procesos asíncronos (jdbc:oracle:oci:@IP:PORT:SID).

- El token “RPLC. MPM_SCHM.USER” por el identificador del usuario NO propietario del esquema de procesos asíncronos.

- El token “RPLC. MPM_SCHM.PASS” por el password del usuario NO propietario del esquema de procesos asíncronos.

- El token “RPLC.TRANS_SCHM.CNNTNURL” por la URL de conexión hacia la base de datos que contiene el esquema transaccional (jdbc:oracle:oci:@IP:PORT:SID).

- El token “RPLC.TRANS_SCHM.USER” por el identificador del usuario NO propietario del esquema transaccional.

- El token “RPLC.TRANS_SCHM.PASS” por el password del usuario NO propietario del esquema transaccional.

- El token “RPLC.VAL_SCHM.CNNTNURL” por la URL de conexión hacia la base de datos que contiene el esquema de validación (jdbc:oracle:oci:@IP:PORT:SID).

- El token “RPLC.VAL_SCHM.USER” por el identificador del usuario NO propietario del esquema de validación.

- El token “RPLC.VAL_SCHM.PASS” por el password del usuario NO propietario del esquema de validación.

Inicialmente, como ya se indicó en apartados anteriores, el password de los keystores para las conexiones SSL es “12345”, el cual puede ser modificado en el siguiente módulo (siempre según el que se haya establecido en el keystore):

```
…
    <security-realm name="AfirmaConnectorsRealm">
        <server-identities>
        <ssl protocol="TLS">
                        <engine enabled-protocols="TLSv1 TLSv1.1 TLSv1.2" enabled-cipher-suites="TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_DHE_DSS_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA TLS_DHE_DSS_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_DH_anon_WITH_AES_256_GCM_SHA384 TLS_DH_anon_WITH_AES_128_GCM_SHA256 TLS_DH_anon_WITH_AES_256_CBC_SHA256
TLS_DH_anon_WITH_AES_256_CBC_SHA TLS_DH_anon_WITH_AES_128_CBC_SHA256 TLS_DH_anon_WITH_AES_128_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_DES_CBC_SHA SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA SSL_DH_anon_WITH_DES_CBC_SHA SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_WITH_NULL_SHA256 SSL_RSA_WITH_NULL_SHA SSL_RSA_WITH_NULL_MD5 TLS_KRB5_WITH_3DES_EDE_CBC_SHA
TLS_KRB5_WITH_3DES_EDE_CBC_MD5 TLS_KRB5_WITH_DES_CBC_SHA TLS_KRB5_WITH_DES_CBC_MD5
TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5"/>
                        <!-- <engine enabled-protocols="TLSv1 TLSv1.1 TLSv1.2" enabled-cipher-
suites="TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA TLS_DHE_DSS_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_DH_anon_WITH_AES_256_GCM_SHA384 TLS_DH_anon_WITH_AES_128_GCM_SHA256 TLS_DH_anon_WITH_AES_256_CBC_SHA256
TLS_ECDH_anon_WITH_AES_256_CBC_SHA TLS_DH_anon_WITH_AES_256_CBC_SHA TLS_DH_anon_WITH_AES_128_CBC_SHA256
TLS_ECDH_anon_WITH_AES_128_CBC_SHA TLS_DH_anon_WITH_AES_128_CBC_SHA TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_DES_CBC_SHA SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA SSL_DH_anon_WITH_DES_CBC_SHA SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_WITH_NULL_SHA256 TLS_ECDHE_ECDSA_WITH_NULL_SHA TLS_ECDHE_RSA_WITH_NULL_SHA SSL_RSA_WITH_NULL_SHA
TLS_ECDH_ECDSA_WITH_NULL_SHA TLS_ECDH_RSA_WITH_NULL_SHA TLS_ECDH_anon_WITH_NULL_SHA SSL_RSA_WITH_NULL_MD5
TLS_KRB5_WITH_3DES_EDE_CBC_SHA TLS_KRB5_WITH_3DES_EDE_CBC_MD5 TLS_KRB5_WITH_DES_CBC_SHA TLS_KRB5_WITH_DES_CBC_MD5
TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5"/> -->
```

```
                <keystore provider="JKS" path="KeystoreSSL-Tomcat.jks" relative-to="jboss.server.config.dir" keystore-password="12345"
alias="sslserver" key-password="12345"/>
            </ssl>
            </server-identities>
            <authentication>
                        <truststore provider="JKS" path="TrustStoreSSL-Tomcat.jks" relative-to="jboss.server.config.dir" keystore-
password="12345"/>
              <local default-user="$local" allowed-users="*" skip-group-loading="true"/>
            </authentication> …
```

Se definen los conectores “http”, “https1way” (SSL) y “https2way” (SSL con autenticación mútua). La asignación de los puertos a cada uno de estos conectores se realiza al final del fichero:

```
…
      <socket-binding name="http" port="8080"/>
      <socket-binding name="https1way" port="443"/>
      <socket-binding name="https2way" port="4433"/>
…
```

El administrador de la plataforma deberá indicar los puertos a utilizar si no son los de por defecto, o incluso eliminar o añadir los conectores que crea necesarios.

Adicionalmente, hay que:

- Revisar las propiedades de todos los datasources aquí editados, y adaptar el número de conectores mínimos y máximo según las necesidades del entorno.
- Sustituir el token “JBOSS_INSTANCE_NAME” por el nombre de la instancia de wildfly establecido en fichero standalone.sh en el punto 7)
- Sustituir todas las apariciones del token “RPLC.JBOSS.IP_BINDING” por la IP/nombre de binding a usar para el servidor de aplicaciones.
- Sustituir el token “RPLC.JBOSS.PORTMANAGEMENTNATIVE” por el puerto a asignar para la gestión nativa del servidor.
- Sustituir el token “RPLC.JBOSS.PORTMANAGEMENTHTTP” por el puerto a asignar para la gestión por HTTP del servidor.
- Sustituir el token “RPLC.JBOSS.PORTMANAGEMENTHTTPS” por el puerto a asignar para la gestión por HTTPS del servidor.
- Sustituir el token “SERVER.MAIL.IP” por la IP/nombre del servidor de correo si se dispone de el, y el token “SERVER.MAIL.PORT” por el puerto .

22) JBOSS_HOME/standalone/configuration/staticAfirmaConfig.properties

Este fichero contiene todas aquellas propiedades propias de la plataforma @firma que por determinadas razones de comportamiento funcional no ha sido posible extraer y llevar a la configuración en base de datos.

Entre estas propiedades se encuentran:

- Configuración de la base de datos local de configuración: Ubicación, URL, usuario, contraseña, ubicación de la carpeta de backups, bandera para indicar si se hacen backups, número máximo de intentos de conexión con el esquema de configuración antes de enviar alarma, y el tiempo, en milisegundos, para reiniciar a cero el contador de intentos de conexión con el esquema de configuración antes de enviar alarma.

- Clase y mensaje para la excepción de conexión del proveedor de persistencia utilizado.

- Ruta JNDI para localizar el datasource. Se necesita para el acceso al esquema de custodia de la versión anterior de @firma.

- Configuración general del HSM.

- Configuración de procesamiento de mensajes de auditoría, pudiendo escoger entre modo acoplado (se procesan directamente los mensajes en base de datos) o desacoplado (mediante log y procesado externo). El modo desacoplado es el establecido por defecto.

- Configuración de los distintos proveedores de custodia disponibles.

- Ruta absoluta de almacenamiento de peticiones y respuestas (en caso de configurarse su almacenamiento), que por defecto es JBOSS/standalone/configuration/ReqResp.

- Configuración de almacenamiento en ficheros de las CRL descargadas, así como de sus copias de seguridad. En estas propiedades se especifica la ruta absoluta hacia el directorio donde se almacenarán las CRL, que por defecto es JBOSS/standalone/configuration/CRL, pero si el administrador así lo considera, podría modificarla para cada nodo.

- Bandera que indica si esta instancia de @firma ejecutará tareas compartidas o no. Solo el nodo de administración debe ejecutarlas.

- Configuración asociada a la caché de configuración, permitiendo escoger la implementación a usar, la propia de JBoss, Infinispan, o una custom propia de Java, JavaConcurrentMap. No se recomienda modificar esta configuración salvo problemas en recarga de la caché.

- Configuración específica para el comportamiento de la plataforma respecto a las firmas XAdES.

- Configuración específica para el comportamiento de la plataforma respecto a como tratar los formatos de firma en las respuestas.

- Listado de caracteres especiales permitidos (codificados en UTF-8) para los alias de los certificados.
- Bandera que indica si este nodo se encarga de ejecutar procesos asíncronos (procesos de validaciones en lote, por ejemplo). Solo estará activo en nodos asíncronos.
- Rutas hacia los directorios donde se almacenan los logs del servidor.
- Aplicación que aparecerá en los logs de control de repetición de peticiones si se hace uso del servicio OCSP sin especificar aplicación.
- GrayLog: Bandera que indica si debe usarse (por defecto deshabilitado), y listado de propiedades que definen su configuración.
- Propiedades relativas al servicio de provisión de monitorización de estado.
- Descriptores de auditoría. Estas propiedades no deben ser modificadas.
- Tipos de validadores de parámetros para los distintos servicios.
- Parámetros configurables de los validadores:
    - Tamaño máximo de cadena permitido para el validador tipo String.
    - Tamaño máximo de cadena permitido para el validador tipo String para DSS.
    - Tamaño máximo permitido (en bytes) para la petición SOAP.
- Propiedades para las tareas del sistema no configurables desde el módulo de administración.
- Nombre de cada uno de los properties de los informes a generar.
- Configuración de los algoritmos de canonicalización.
- Ficheros de política de verificación.
- Definición de algoritmos no seguros.
- OIDs no registrados en iaik. No deben ser modificados.
- Configuración para la evaluación de papeos y discriminadores de certificados.
- Configuración para la evaluación del bloqueo temporal de métodos de validación.
- Propiedades del servicio REST TASKFINISHER para finalizar tareas de procesos asíncronos. Solo en el nodo de servicios asíncronos.

- Propiedades del servicio REST DDBBVERSION para gestionar el versionado de la base de datos. Solo en nodos de servicio y servicios asíncronos.

- Propiedades del servicio REST asíncrono para finalizar tareas de procesos asíncronos. Solo en el nodo de servicios.

- Path donde se encuentran localizadas las TSLs obtenidas desde valET.

- Bandera donde se indicará si se incluye en la petición SOAP a la TSA el identificador de la aplicación de @firma.

Cada una de estas propiedades se encuentra descrita adecuadamente mediante los comentarios correspondientes.

Deben configurarse las siguientes propiedades:

- database.configuration.path: Sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS.

- afirma.config.databaseUrl: Sustituir el token "RPLC.CONF_SCHM.HSQLDB.IP" por la IP de binding con la que se inicia la base de datos de configuración, y el token "RPLC.CONF_SCHM.HSQLDB.PORT" por el puerto con el que se inicia esta misma base de datos.

- afirma.config.backup.path: Con la ruta absoluta hacia el directorio donde se almacenarán copias de seguridad de la base de datos local de configuración. Se propone ABS_PATH_TO_JBOSS/standalone/configuration/LocalHSQLDBDataBase/configurationBackups, sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS. **(SOLO ES NECESARIO EN EL NODO DE ADMINISTRACIÓN. SE EXCLUYEN LOS NODOS DE SERVICIO ASÍCRONO Y DE SERVICIO).**

- afirma.config.dobackup: Bandera que indica al sistema si debe realizar copias de seguridad diarias de la base de datos local de configuración en la ruta configurada en afirma.config.backup.path. Establecer "true" si se desean realizar copias de seguridad, y "false" en caso contrario. **(SOLO ES NECESARIO EN EL NODO DE ADMINISTRACIÓN. SE EXCLUYEN LOS NODOS DE SERVICIO ASÍCRONO Y DE SERVICIO).**

- wsCustodyPolicy.fileSystemPath: Con la ruta absoluta hacia el directorio donde se almacenarán las peticiones y respuestas SOAP si así se configura en la plataforma. Se propone ABS_PATH_TO_JBOSS/standalone/configuration/ReqResp, sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS.

- crlRepository.absPath: Con la ruta absoluta hacia el directorio donde se almacenarán las CRL que actúan como caché de segundo nivel de validación (solo en el caso de que se configure en la plataforma para almacenamiento en fichero). Se propone

ABS_PATH_TO_JBOSS/standalone/configuration/CRL, sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS.

- crlRepository.absPath.backup: Con la ruta absoluta hacia el directorio donde se almacenarán los backup de las CRL que actúan como caché de segundo nivel de validación (solo en el caso de que se configure en la plataforma para almacenamiento en fichero). Se propone ABS_PATH_TO_JBOSS/standalone/configuration/CRL/Backups, sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS.

- logs.path.1: Con la ruta absoluta hacia el directorio donde se almacenan los logs generados por la instancia. Se propone ABS_PATH_TO_JBOSS/standalone/logs, sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS.

- taskfinisher.urlServer: Sustituir el token "RPCL.TASKFINISHER.IP" con la ip del punto de entrada de los nodos de servicio y el token "RPCL.TASKFINISHER.PORT" con el puerto **(SOLO ES NECESARIO EN LOS NODOS DE SERVICIO ASÍCRONO SE EXCLUYEN LOS NODOS DE ADMINISTRACIÓN Y DE SERVICIO).**

- common.operations.urlServer: Sustituir el token "RPCL.ADMIN_NODE.IP" con la ip del nodo de administración y el token "RPCL.ADMIN_NODE_PORT" con el puerto **(SOLO ES NECESARIO EN LOS NODOS DE SERVICIO Y SERVICIO ASÍNCRONO. SE EXCLUYE EL NODO DE ADMINISTRACIÓN).**

- async.urlServer: Sustituir el token "RPCL.ASYNC.IP" con la ip del punto de entrada de los nodos de servicios asíncronos y el token "RPCL.ASYNC.PORT" con el puerto **(SOLO ES NECESARIO EN LOS NODOS DE SERVICIO. SE EXCLUYEN LOS NODOS DE SERVICIO ASÍNCRONO Y EL NODO DE ADMINISTRACIÓN).**

- integra.tsl.path: Con la ruta absoluta hacia el directorio donde se almacenarán las TSLs descargadas a modo de caché. Se propone ABS_PATH_TO_JBOSS/standalone/configuration/TSLs, sustituyendo el token ABS_PATH_TO_JBOSS por la ruta absoluta hacia el directorio de JBOSS.

- tsa.include.afirmaApp: Dejar seteado con valor false.

23) JBOSS_HOME/standalone/configuration/messages/Language.properties

Descripción. Fichero donde se especifica el lenguaje deseado para los mensajes de la plataforma.

Modificación. Se debe establecer:

- LANGUAGE: Propiedad que determina el idioma a utilizar para los mensajes de log de la plataforma. Por defecto está configurado para español.

```
# Language setting for the platform. Allowed Values:
```

```
# es_ES (for spanish).
# en_US (for english).

LANGUAGE = es_ES
```

5.5.3 Arranque del JBOSS

Una vez modificados los ficheros de configuración y los keystores necesarios, se procederá a arrancar la plataforma, con el fin de completar la configuración desde el módulo de administración de la misma.

Es importante recordar que cada instancia de la plataforma dispone de una base de datos de configuración y otra de alarmas HSQLDB.

Nota: para arrancar el servidor de aplicaciones es necesario haberse logado con un usuario con permisos de administrador para que el servidor JBOSS pueda abrir los puertos 8080, 443 y 4433 que vienen configurados por defecto.

- JBOSS_HOME/bin/startup.sh

Inicia las bases de datos de configuración, alarmas (si no lo están ya), y si no está iniciada la instancia del servidor de aplicaciones, elimina los ficheros temporales y procede a iniciarla. Todo esto en segundo plano y desviando la salida de consola a *null*.

- JBOSS_HOME/bin/shutdown.sh

Detiene la instancia del servidor de aplicaciones (primero lo intenta mediante la propia consola del servidor, si no lo consigue, manda una señal para que el proceso de detenga correctamente, y finalmente, si aún no lo ha hecho, mata el proceso) y posteriormente detiene las bases de datos de configuración y alarmas.

- JBOSS_HOME/bin/checkStatus.sh

Muestra información en la consola indicando si la instancia del servidor de aplicaciones está ejecutándose o detenida, y del mismo modo, información relativa a las bases de datos locales.

5.5.4 Finalizar configuración

Este apartado es solo aplicable si se ha hecho una instalación en limpio, es decir, sin migrar el contenido de la base de datos de la versión anterior.

El proceso descrito con anterioridad permite el arranque mínimo de la plataforma, lo que significa que el resto de configuración deberá realizarse a través del módulo de administración.

Para acceder al módulo de administración, introduzca en un navegador la siguiente URL:

Modo Seguro	https://IP_MAQUINA:PUERTO_SEGURO/Administracion
Modo no Seguro	http://IP_MAQUINA:PUERTO/Administracion

Donde IP_MAQUINA es la dirección IP pública de la máquina donde se ha desplegado la plataforma, y PUERTO es el puerto de escucha por el cual ha sido configurado el/los servidores de aplicaciones que alojan la plataforma (por defecto 8080, 443 y 4433).

Una vez insertada la URL en el navegador debe aparecer la siguiente pantalla:

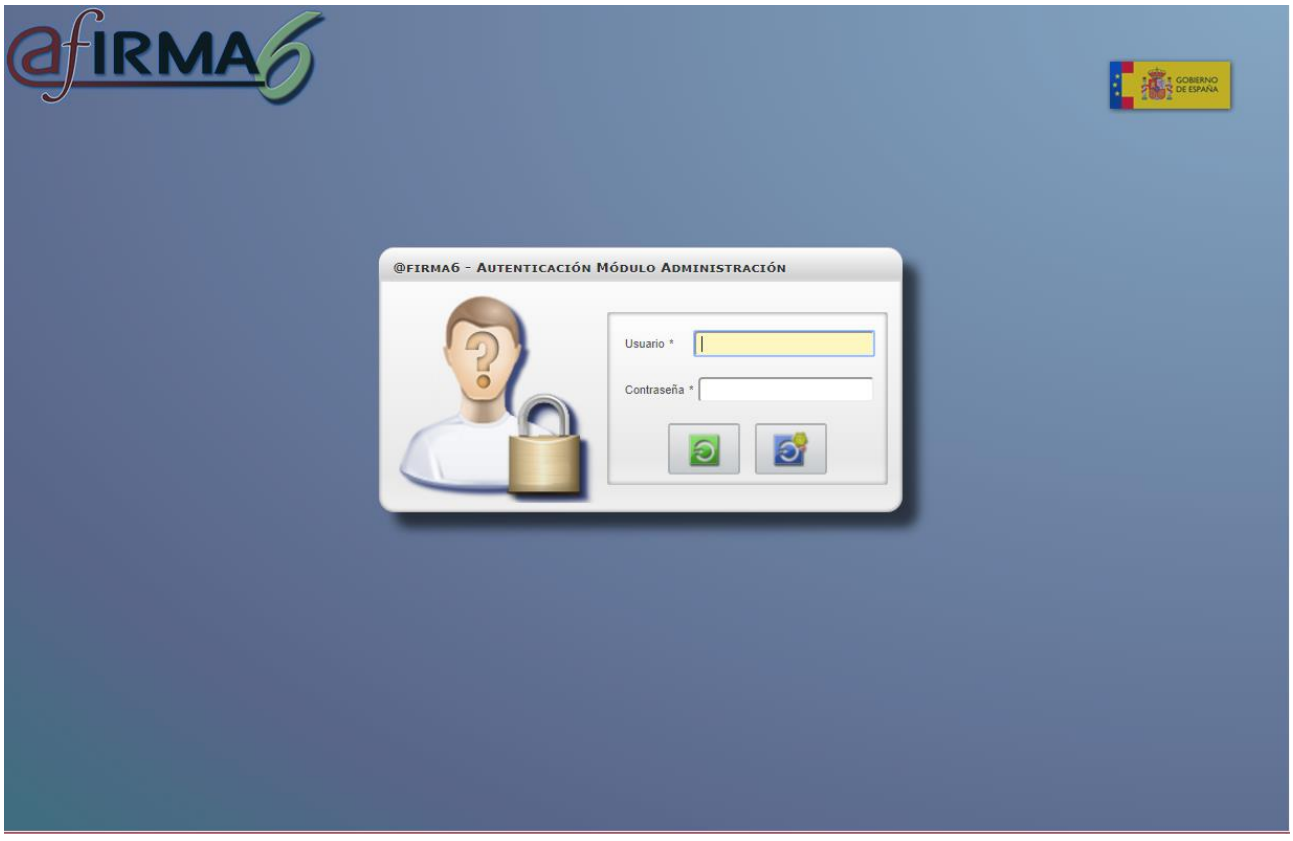


Figura 1 – Interfaz de acceso al módulo de Administración de @firma

El usuario configurado por defecto, si se ha generado la base de datos sin importar desde la versión anterior, es *admin* y su contraseña *12345.* Estas credenciales podrán ser modificadas por medio del módulo de gestión de usuarios.

Configuración de la plataforma desde la Administración

En este apartado se describe como configurar la plataforma @firma 6 desde su administración. Al igual que el apartado anterior, esta configuración tan solo debe realizarse una vez, ya que la configuración es compartida por todos los nodos.

Independientemente a la configuración, es importante revisar los datos de acceso a la administración de la plataforma, para ello hay que acceder a la ventana de usuario:

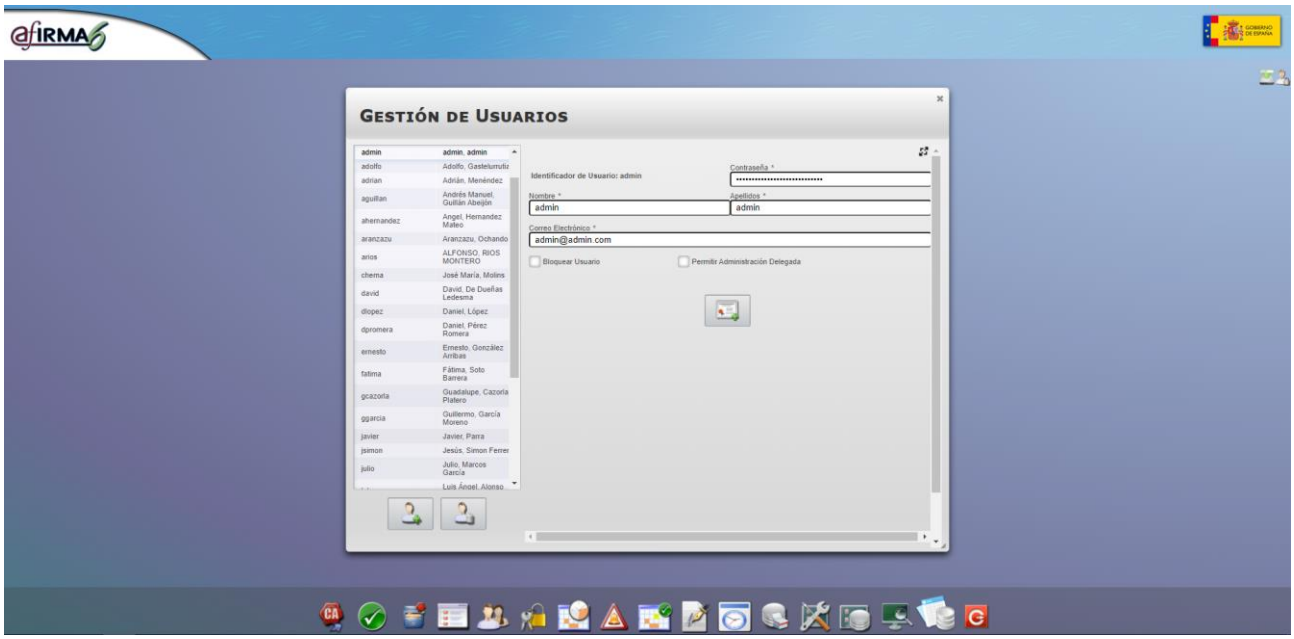


Figura 2 – Administración – Gestión de Usuarios

Para una explicación completa de la interfaz, se remite al lector al manual de administración.

Para iniciar el proceso de configuración de la plataforma, tras acceder a la administración, hay que entrar en la pantalla de “Gestión de Keystores”:

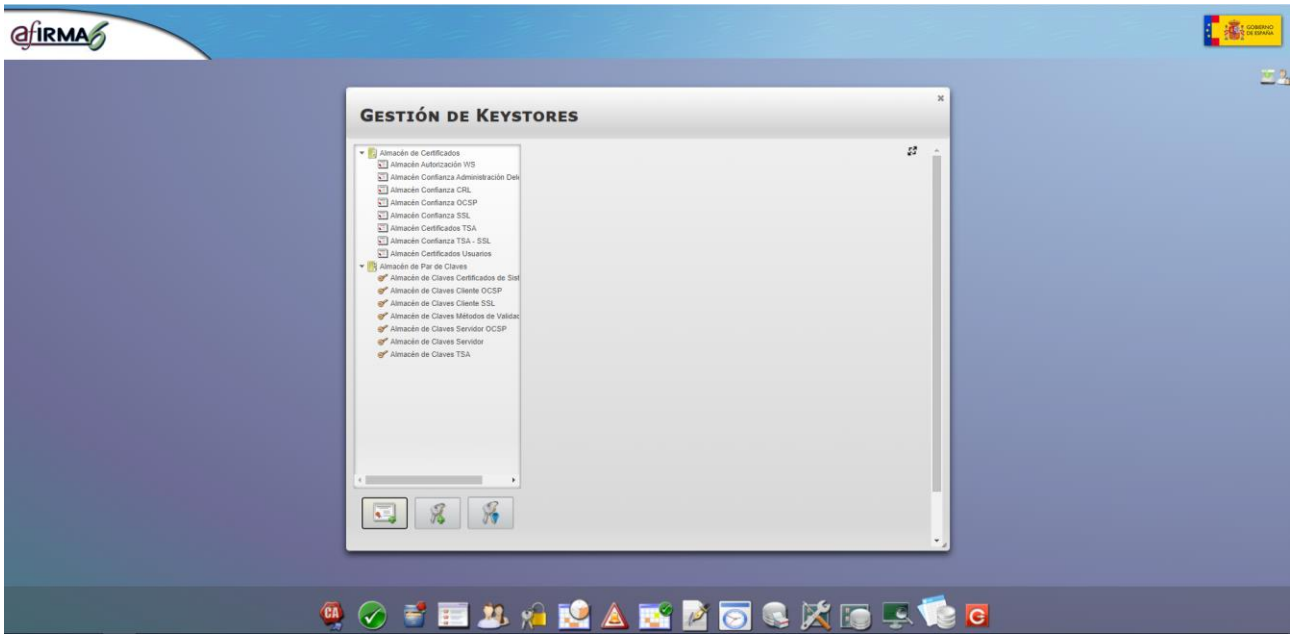


Figura 3 – Administración – Gestión de Keystores

A través de esta pantalla, se permite al administrador añadir/eliminar/modificar certificados en los distintos almacenes de la plataforma. Para una mayor información de manejo de la interfaz, se remite al lector al manual de administración.

Para continuar con el proceso de configuración de la plataforma, hay que entrar en la pantalla de "Configuración":

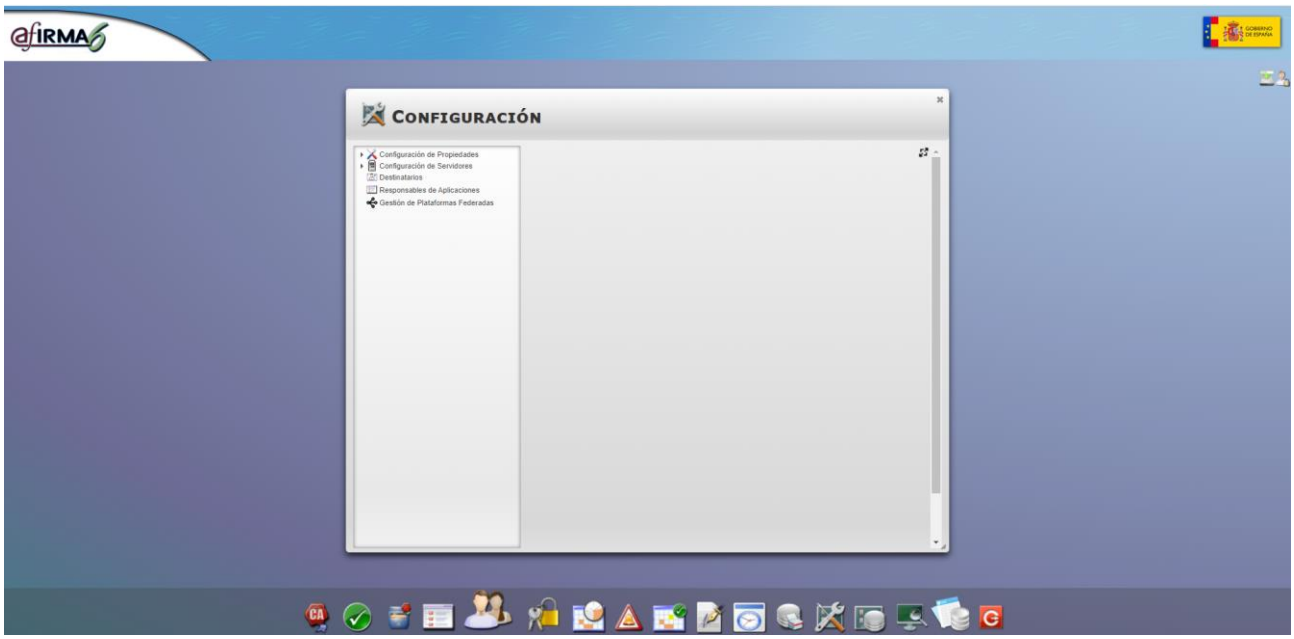


Figura 4 – Administración – Configuración de la Plataforma

En esta pantalla dispondremos de un árbol en el lado izquierdo que nos mostrará los distintos apartados para realizar las configuraciones, y una parte derecha en la que se mostrarán las propiedades a configurar según el elemento seleccionado en el árbol.

El primer apartado son las propiedades:

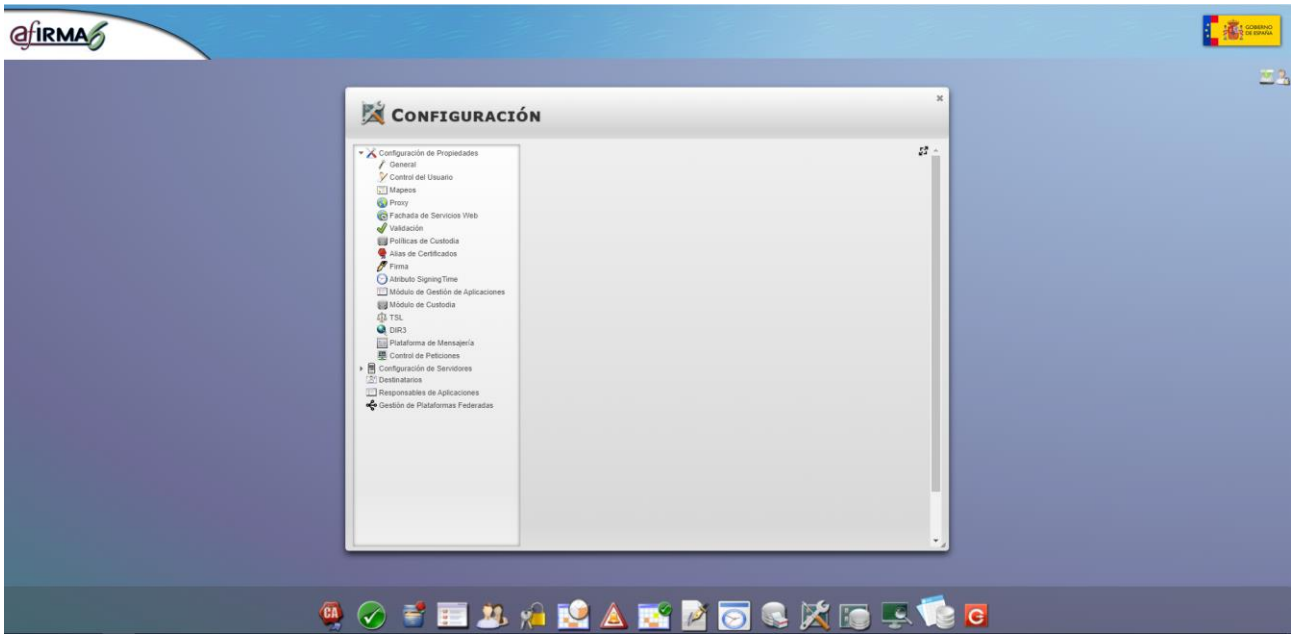


Figura 5 – Administración – Configuración - Propiedades

Dentro de este apartado, se podrá establecer la configuración de las siguientes propiedades (se adjunta una breve descripción de cada una de ellas):

- General:
    - Tamaño máximo (en bytes) permitido para los ficheros a importar desde la administración.
    - Número máximo de transacciones de auditoría a mostrar en una administración delegada.
    - Nombre del campo lógico que contendrá el nombre del usuario a logarse.
    - Nombre del campo lógico que contendrá el apellido del usuario a logarse.
    - Listado de protocolos restringidos para el cifrado SSL de conexiones.
    - Listado de algoritmos restringidos para el cifrado SSL de conexiones.
    - Tiempo de espera antes de detener la anterior caché local tras realizar una recarga de esta.
    - Activación de la restricción a la definición del CONTENT-LENGTH en la cabecera HTTP (error 411 de HTTP 1.1).
    - Clave de cifrado AES para las contraseñas del sistema.
    - Password de acceso al HSM.
- Control del Usuario:
    - Expresión regular que controla las contraseñas de usuario que van a ser creadas. Por defecto: "^.*(?=.{10,})(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&+=]).*$"
    - Número de intentos para insertar la contraseña antes de bloquear el usuario.
- Mapeos:
    - Listado de partículas aplicables a los mapeos tanto del primer como del segundo apellido. Por defecto: "DEL, LOS, LAS, SOC, DE, LA, AL, SA, SL, SC, CB, EL, Y, I, DO, DOS, E, DA, DAS, SAN".
- Proxy:
    - Modo de operación a utilizar en el proxy: A escoger entre "NONE", "NONE_AUTHENTICATION", "BASIC_AUTHENTICATION" y "NTLM_AUTHENTICATION".
    - Modo de operación a utilizar en el proxy por el canal seguro (por ejemplo, HTTPS): A escoger entre "NONE", "NONE_AUTHENTICATION", "BASIC_AUTHENTICATION" y "NTLM_AUTHENTICATION".

- Dirección IP del proxy.
- Dirección IP del proxy para el canal seguro.
- Puerto asignado al proxy.
- Puerto asignado al proxy para el canal seguro.
- Nombre de usuario en caso de necesitar autenticación.
- Nombre de usuario en caso de necesitar autenticación para el canal seguro.
- Contraseña asignada al usuario en caso de necesitar autenticación.
- Contraseña asignada al usuario en caso de necesitar autenticación para el canal seguro.
- Dominio del usuario en caso de autenticación NTLM.
- Dominio del usuario en caso de autenticación NTLM para el canal seguro.
- Listado de direcciones para las que no se utilizará el proxy.
- Indicado para utilizar el analizador de direcciones, de modo que, si se trata de una dirección local, no sea utilizado el proxy para realizar la conexión.

- Fachada de servicios Web:
  - URL donde se publicarán los SCHEMAS asociados a las peticiones de entrada y salida a la plataforma.
  - En el resto de propiedades se definen para los servicios web ofrecidos en la plataforma, los paquetes (a nivel de código) de los distintos traductores (estas propiedades no deben ser modificadas inicialmente), los espacios de nombres asignados y los esquemas asociados.

- Validación:
  - Indicador para validar los certificados a dar de alta de manera estricta, tal y como se especifica en la RFC 3280.
  - Indicador para utilizar la librería de Jakarta para las conexiones HTTP, o la proporcionada por Sun en la JRE.
  - Indicador para utilizar la librería de Novell para las conexiones LDAP, o la proporcionada por Sun en la JRE.

- Tamaño máximo permitido (en bytes) de descarga en las conexiones que se realicen. Por defecto 5242880 bytes.
- Indicador para comprobar el estado de revocación del certificado incluido en el token de seguridad BinarySecurityToken del mensaje SOAP.
- Fecha en formato dd/MM/yyyy a partir de la cual todos los sellos de tiempo serán validados respecto a la RFC 5816.

- Políticas de Custodia:
  - Indicador para almacenar las peticiones SOAP recibidas.
  - Indicador para almacenar las respuestas SOAP generadas por la plataforma.
  - Formato del hash que se almacena de la petición/respuesta SOAP. Se debe escoger entre “SHA1”, “SHA256” y “SHA512”.
  - Modo de almacenamiento de las peticiones/respuestas SOAP o de sus respectivos hashes. Si el valor escogido es 0, las peticiones/respuestas se almacenarán en base de datos. Si el valor escogido es 1, las peticiones/respuesta se almacenarán en el sistema de ficheros.
- Alias de Certificados:
  - Alias del certificado de servidor utilizado para la firma de las respuestas SOAP. Por defecto “defaultFirma”.
  - Alias del certificado de servidor utilizado para la firma de los ficheros de registro de eventos. Por defecto “defaultFirma”.
- Firma:
  - Indicador para registrar la firma origen.
  - Indicador para registrar la firma actualizada.
  - Indicador para conservar las firmas originales registradas que hayan sido actualizadas.
- Atributo SigningTime:
  - Indicador para validar la propiedad firmada SigningTime.
  - Ventana permitida de desviación temporal, en milisegundos, entre el momento de validación y el tiempo indicado por la propiedad firmada SigningTime.

- SigningTime para PAdES: Ventana permitida de desviación temporal, en milisegundos, entre el momento de validación y el tiempo indicado por la propiedad firmada SigningTime.

- Módulo de gestión de aplicaciones:
  - Custodia de Peticiones
  - Custodia de Respuestas
  - Formato de firma de la respuesta
  - Inclusión del Token de seguridad
  - Política de custodia
  - Política de Timestamp
  - Política de resultado de verificación de firma.
  - Política de validación
  - Excluir del control de peticiones
  - Indicar si por defecto, al crear una aplicación, se le habilitará (o no) el uso de TSL.
- Módulo de Custodia:
  - Modo de custodia de las firmas electrónicas: Custodia completa de la firma (valor 0) o custodia del hash de la firma (valor 1).
  - Algoritmo de resumen utilizado para la custodia de firma. Se debe escoger entre “SHA1”, “SHA256”, “SHA384” y “SHA512”.
- TSL:
  - Fecha inicial (dd/MM/yyyy) a partir de la cual se tiene en consideración el uso de TSL para el reconocimiento y verificación de certificados.
  - Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certClassification' por NATURAL_PERSON: Números enteros separados por comas.
  - Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certClassification' por LEGAL_PERSON: Números enteros separados por comas.

- Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certClassification' por ESIG: Números enteros separados por comas.
- Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certClassification' por ESEAL: Números enteros separados por comas.
- Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certClassification' por WSA: Números enteros separados por comas.
- Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certQualified' por YES.
- Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certQualified' por NO.
- Lista de valores correspondientes al campo lógico 'clasificacion' que se mapean en el campo lógico 'certClassification' por TSA.

- Integración DIR3:
  - Modo de integración con DIR3: WITHOUT INTEGRATION, RELAXED, STRICT
  - DIR3: Nombre de Usuario.
  - DIR3: Contraseña de Usuario.
  - DIR3: URL servicios.
  - DIR3: Timeout de conexión en milisegundos: por defecto viene el valor 10000.
- Plataforma de mensajería:
  - Indicador para usar la Plataforma de Mensajería.
  - Usuario de acceso a la Plataforma de Mensajería.
  - Contraseña de acceso a la Plataforma de Mensajería.
  - Nombre del lote a asignar para los correos electrónicos enviados.
  - Nombre del servicio asignado para el envío directo de correos electrónicos.
  - Nombre del lote a asignar para los mensajes de texto enviados.
  - Nombre del servicio asignado para el envío directo de mensajes de texto.
- Control de peticiones:

- Ventana de tiempo a observar al controlar las peticiones iguales permitidas (en milisegundos). Se enviará alarma si se detectan N peticiones iguales por parte de una aplicación durante el tiempo configurado aquí.
- Ventana de tiempo a observar al controlar las peticiones por aplicación y servicio permitidas (en milisegundos). Se enviará alarma si se detectan N peticiones de la misma aplicación a un servicio durante el tiempo configurado aquí.
- Número N de peticiones iguales permitidas durante el tiempo configurado.
- Número N de peticiones de una aplicación a un servicio permitidas durante el tiempo configurado.
- Tiempo (en milisegundos) que se esperará antes de seguir comprobando si una petición se sigue repitiendo tras el envío de una alarma para informar que se ha superado el número N de peticiones iguales permitidas durante el tiempo configurado.
- Tiempo (en milisegundos) que se esperará antes de seguir comprobando si una aplicación sigue realizando peticiones a un servicio tras el envío de una alarma por superar el número N de peticiones permitidas durante el tiempo configurado.
- Activar el control de peticiones.

El siguiente apartado es relativo a la configuración de servidores:

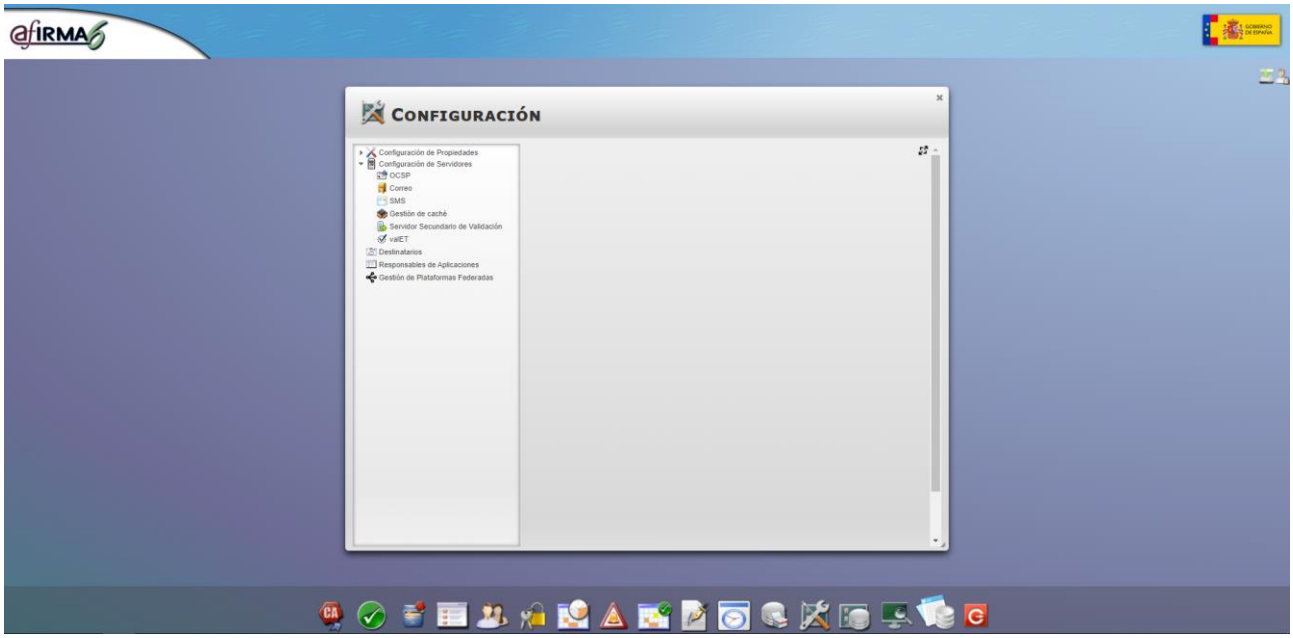


Figura 6 – Administración – Configuración - Servidores

Dentro de este apartado, se podrá establecer la configuración de las siguientes propiedades (se adjunta una breve descripción de cada una de ellas):

- OCSP:
    - Algoritmo de resumen utilizado en la respuesta del servicio OCSP. Se debe escoger entre “SHA1”, “SHA256” y “SHA384”.
    - Algoritmo de firma para la respuesta del servicio OCSP. Se debe escoger entre “sha1WithRSAEncryption”, “sha256WithRSAEncryption”, “sha384WithRSAEncryption” y “sha512WithRSAEncryption”.
    - Selección del certificado a utilizar para la firma de las respuestas del servicio OCSP.
    - Indicador de si es obligatorio el atributo RequestorName en las peticiones recibidas.
- Correo:
    - Identificador del emisor de los correos emitidos por la plataforma.
    - Host SMTP del servidor de correos.
    - Puerto SMTP del servidor de correos.
    - Usuario y contraseña de acceso para el uso del servidor de correos.
- SMS:
    - Marcador que indica si este servidor se encuentra habilitado o no.
    - Dirección IP donde se encuentra desplegado el servidor SMPP.
    - Número del puerto de conexión con el servidor SMPP.
    - Número de teléfono que aparecerá como emisor de los mensajes de texto enviados por la plataforma.
    - Usuario y contraseña de acceso.
- Gestión de Caché de Nivel 1:
    - Se define el tiempo de vida, intervalo de refresco y número de entradas para las respuestas de validación obtenidas mediante un método de validación tipo CRL almacenadas en la caché de nivel 1.

  - Se define el tiempo de vida, intervalo de refresco y número de entradas para las respuestas de validación obtenidas mediante un método de validación tipo OCSP almacenadas en la caché de nivel 1.
  - Se define el tiempo de vida, intervalo de refresco y número de entradas para las respuestas de validación obtenidas mediante un método de validación tipo WS-Longevo almacenadas en la caché de nivel 1.

- Servidor secundario de validación:
  - Habilitado
  - Tipo de acceso
  - URL:Puerto Servidor
  - Timeout de conexión (en ms)
  - Identificador de aplicación
  - Tipo de Autenticación
  - Usuario. En función del tipo de autenticación
  - Contraseña. En función del tipo de autenticación
  - Certificados disponibles. En función del tipo de autenticación
- valET:
  - URL del servidor de valET
  - Timeout de conexión (en ms)
  - Identificador de aplicación

El siguiente apartado es relativo a la gestión de destinatarios. Pueden añadirse eliminarse y editarse datos de destinatarios a los que la plataforma pueda notificar.

El siguiente apartado permite añadir, eliminar y editar responsables de las aplicaciones.

El último apartado es relativo a la gestión de plataformas federadas, y como es lógico, tan solo debe usarse por la plataforma central del Ministerio, por lo que es omitido para la configuración.

6 Anexo I: Mantenimiento de la base de datos HSQLDB de configuración

6.1 Introducción

Es necesario conocer algunos aspectos sobre la base de datos local de configuración para un correcto mantenimiento del sistema.

Cada uno de los nodos que forman el sistema dispone de su propia base de datos de configuración independiente.

Estas bases de datos están sincronizadas mediante un mecanismo de comunicación entre los nodos de servicio o servicios asíncronos y el nodo de administración. Cuando se realizan modificaciones en la administración y se genera una nueva versión de la base de datos (ver manual de administración), esta nueva versión incremental queda disponible para ser "utilizada" por el resto de nodos para actualizar sus propias bases de datos locales.

Si no existe comunicación entre los nodos, esta sincronización no podrá hacerse correctamente, y los datos incluidos en las bases de datos de los nodos de servicio y servicios asíncronos estarán incompletos y no se corresponderán con los contenidos en el nodo de administración.

6.2 Estructura de la base de datos local de configuración del nodo de administración

Bajo la ruta JBOSS_HOME/standalone/configuration/LocalHSQLDBDataBase/configuration se dispondrá de una serie de directorios cuyo contenido pasamos a describir.

- Directorio database:

  Aquí se encuentran los ficheros propios de la base de datos. Entre ellos, encontramos el fichero incremental.log que contendrá los cambios en la base de datos desde la última versión generada.

- Directorio blobs:

  En este directorio se encuentran los blobs de la base de datos (keystores, políticas, etc).

- Directorio version:

  Aquí se almacenan las distintas versiones que se han ido generando desde la administración. Están numeradas de menor a mayor, y el contenido de estas será lo que se envíe al resto de nodos cuando hagan una petición de actualización desde una versión anterior.

- Directorio lastFullVersion:

Contiene la versión completa de la base de datos correspondiente a la última versión generada. Este contenido se enviará a los nodos cuando estos pidan una generación completa de la base de datos mediante un procedimiento que describiremos a continuación.

- Directorio stableConf:

Contiene las distintas versiones estables generadas desde la administración por petición del administrador. Mas adelante se explicará el procedimiento para reestablecer una configuración estable.

Por otro lado, hay que indicar que existe un mecanismo de backup diario (o con el periodo deseado) configurable en el fichero staticAfirmaConfig.properties, que irá generando copias de seguridad en el directorio especificado en la configuración. Estas copias de seguridad pueden ser también establecidas mediante el procedimiento que se explicará más adelante.

6.3 Forzar sincronización de nodos de servicio o servicios asíncronos

Si en algún momento se sospecha que alguno de los nodos puede no tener correctamente sincronizada su base de datos de configuración con la del nodo de administración, existe un mecanismo para forzar la sincronización.

Bastaría con editar el fichero JBOSS_HOME/standalone/configuration/staticAfirmaVersion.properties del nodo en cuestión y establecer los siguientes valores en las propiedades:

Dejar vacía la propiedad last.database.version.date (last.database.version.date=)

Establecer a true el valor de la propiedad update.database.on.startup (update.database.on.startup=true).

Tras guardar los cambios realizados en el fichero de propiedades bastará con reiniciar el nodo en cuestión y este se sincronizará con la última versión completa disponible de la base de datos del nodo de administración.

6.4 Recuperación de una configuración estable o una copia de seguridad.

Desde la herramienta de administración, pueden generarse configuraciones estables en la base de datos del servidor (ver manual de administraicón). Mediante el proceso descrito en el manual de administración, se generarán directorios que contendrán una copia de la base de datos del momento en el que se realiza dicha operación.

Por otro lado, el sistema dispone de una tarea en background habilitada para realizar copias de seguridad periódicas de la base de datos. La generación de estas copias de seguridad puede configurarse en el fichero staticAfirmaConfig.properties.

Para recuperar en un momento dato una de estas copias, se pasa a describir el procedimiento a realizar.

- En el nodo de Administración:
  - Parada del servidor de aplicaciones WildFly del nodo de Administración.
  - Parada de la HSQLDB correspondiente (Configuración). Es extremadamente importante asegurarse de que la base de datos se encuentra detenida antes de realizar la manipulación en los directorios que se indica a continuación.
  - Eliminar por completo los directorios "blobs" y "database" de la BBDD (contenida en la ruta "JBOSS_HOME/standalone/configuration/LocalHSQLDBDataBase/configuration").
  - Copiar los directorios "blobs" y "database" contenidos en la configuración estable (ruta JBOSS_HOME/standalone/configuration/LocalHSQLDBDataBase/configuration/stableConf/xxxxxxxxx/) o el backup correspondiente en la ruta "JBOSS_HOME/standalone/configuration/LocalHSQLDBDataBase/configuration") previamente eliminados .
  - En el fichero "staticAfirmaVersion.properties", contenido en la ruta "JBOSS_HOME/standalone/configuration", establecer el valor indicado a los siguientes parámetros:
    - Eliminar el valor de fecha y hora del parámetro "last.database.version.date".
    - Establecer el valor «true» al parámetro "update.database.on.startup".
  - Arrancar la HSQLDB de Configuración.
  - Arrancar el servidor de aplicaciones WildFly.

- En el resto de nodos. Importante que haya finalizado el arranque del nodo de Administración.
  - Parada del servidor de aplicaciones WildFly del nodo en cuestión.
  - En el fichero "staticAfirmaVersion.properties", contenido en la ruta "JBOSS_HOME/standalone/configuration", establecer el valor indicado a los siguientes parámetros:
    - Eliminar el valor de fecha y hora del parámetro "last.database.version.date".
    - Establecer el valor «true» al parámetro "update.database.on.startup".

- Arrancar el servidor de aplicaciones WildFly.

7 Anexo II: Instalación de software base.

7.1 Variables Entorno Java

Resulta necesario e indispensable la declaración de las siguientes variables de entorno respecto a la JDK (1.8.0_131) instalada en el sistema:

```
JAVA_HOME=<Ruta instalacion JVM>

PATH=${PATH}:$JAVA_HOME/bin

CLASSPATH=${CLASSPATH}:${JAVA_HOME}/bin
```

7.2 Consideraciones sobre instalación Cliente Oracle

El Servidor de Firma necesita para su funcionamiento tener instalado el Cliente Oracle para disponer del driver OCI. Solamente se permite este driver por la necesidad de transferir ficheros de mayor tamaño (del orden de Mbytes) en el módulo de Firma de Documentos.

Se debe de instalar la versión 11g para un correcto funcionamiento de la plataforma. Para instalar el Cliente Oracle utilizar el manual y software apropiados de Oracle. No obstante, a continuación, se comentan unas consideraciones adicionales para poder utilizar dicho driver desde JAVA.

Una vez instalado el Cliente Oracle Linux.

1) Configurar el fichero "*tnsnames.ora*".

2) Definir las siguientes variables de entorno:

```
ORA_CLIENT=<directorio_instalacion_oracle>/instantclient_11_2

TNS_ADMIN=<directorio_instalacion_oracle>/network/admin

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORA_CLIENT

CLASSPATH=$CLASSPATH:$ORA_CLIENT/ojdbc6.jar

PATH=$PATH:$ORA_CLIENT
```

NOTA: Esta información PUEDE que sea diferente en función de la versión de Oracle, drivers, etc… Así pues, solo debe tomarse como una ayuda complementaria en caso de encontrar problemas en la instalación del Cliente del manual de Oracle.

Así mismo, se puede dar por válida la instalación de un cliente "lite", aunque lo preferible es que la instalación sea de un cliente completo.

7.3 Generación y Consumo de Entropía

Debido a la naturaleza del funcionamiento de la plataforma, concretamente y focalizado en los procesos de generación de firma, resulta necesario disponer de una serie de datos que se hayan generado de forma aleatoria. Estos datos se obtienen de la entropía generada por el propio sistema. En este sentido, y de forma general, puede considerarse el concepto "entropía" como un buffer gestionado por el propio sistema en el que se almacenan una serie de datos aleatorios (que dependiendo del origen pueden haber sido calculados de distintas formas), que a medida que es consumido, se va rellenando de nuevo. Para un sistema *Red Hat 6.5*:

Para consultar la capacidad disponible de entropía en un determinado momento:

```
cat /proc/sys/kernel/random/entropy_avail
```

Al ejecutar este comando, el sistema devuelve la cantidad de datos aleatorios disponibles. Por defecto, el valor debe estar alrededor de 300.

Esta cantidad, para una plataforma @firma que disponga de una elevada carga de generaciones de firma, puede ser pobre, llegando a consumir más entropía de la que el sistema puede generar, produciéndose en consecuencia bloqueos por espera de recarga del buffer. De hecho, casi para cualquier aplicativo criptográfico, un valor inferior a 1000 no es suficiente.

Para comprobar el rendimiento de consumo y generación de datos aleatorios:

```
dd if=/dev/random of=/dev/null bs=1024 count=1 iflag=fullblock
```

Al ejecutar este comando, el sistema consume bloques de datos aleatorios y va recargando el buffer de entropía, mostrando en pantalla la velocidad de regeneración, la cual, por debajo de 50kb/s se considera pobre para el uso con @firma.

Para solucionar este problema existen múltiples herramientas que se encargan de aumentar el tamaño del buffer de entropía, y además de realizar una recarga aún más rápida de este (prácticamente a medida que se consume). **Se aconseja la instalación y uso de alguna de estas herramientas**. En los siguientes subapartados se comentan brevemente dos de ellas, quedando como responsabilidad del administrador del sistema si instalar alguna de estas o cualquier otra debido a restricciones o necesidades del entorno.

7.3.1 RNG-Tools

Conjunto de utilidades que se instalan como servicio del sistema y que, entre otras cosas, aumenta el tamaño del buffer y rendimiento de generación de números aleatorios disponibles para el sistema.

El detalle de esta herramienta es que hace uso de una característica propia del microprocesador para la generación de números aleatorios (extendida en la mayoría de los microprocesadores existentes actualmente). En consecuencia, según el modelo del procesador (habría que comprobar si es compatible) y/o del entorno virtual si es el caso, se podrá hacer uso de esta utilidad o no.

Para una información más concreta y detallada, se remite al lector a:

- https://wiki.archlinux.org/index.php/Rng-tools
- https://github.com/cernekee/rng-tools

A modo resumen, a partir de la documentación de estas páginas, se puede concluir una serie de pautas/recomendaciones:

- Si rng-tools no detecta automáticamente TRNG físico del sistema:

Deben añadirse en la variable de entorno RNGD_OPTS los siguientes valores:

```
RNGD_OPTS="-o /dev/random -r /dev/my_hw_random_device"
```

- Si el sistema no tiene un "TMP Module”:

Deben añadirse en la variable de entorno RNGD_OPTS los siguientes valores:

```
RNGD_OPTS="--no-tpm=1"
```

- Por defecto el buffer toma el tamaño 2048, para aumentarlo (por ejemplo, a 4096):

Deben añadirse en la variable de entorno RNGD_OPTS los siguientes valores:

```
RNGD_OPTS="--fill-watermark=4096"
```

La herramienta RNG-Tools se instala como cualquier otro servicio del sistema, por lo que con los comandos start/stop se puede inicializar y detener. Otra prueba para comparar el rendimiento (una vez instalada) es probar a ejecutar el siguiente comando antes y después de iniciar el servicio:

```
rngtest -c 1000 </dev/random
```

Si como resultado se obtienen menos de 10 errores, se considera que el rendimiento es adecuado.

7.3.2 Haveged

Para aquellos casos en los que no se pueda hacer uso de RNG-Tools debido a que el procesador no dispone de las características necesarias, o como consecuencia de tratarse de un entorno virtualizado, recomendamos el uso de esta otra herramienta, que básicamente funciona exactamente igual que la anterior, pero la generación de entropía se ejecuta a nivel software mediante distintos algoritmos.

Para una información más concreta y detallada, se remite al lector a:

- https://wiki.archlinux.org/index.php/Haveged
- http://www.issihosts.com/haveged/