JUNTA DE ANDALUCIA

Consejería de Justicia y Administración Pública

Solicit@v4.0.0

Manual de Instalación y Configuración de Entrega

v04r00

Fecha: 28/01/2009

HOJA DE CONTROL

Título	**Manual de Instalación y Configuración de Entrega de solicit@v4.0.0**		
Entregable	Manual de Instalación y Configuración de Entrega		
Nombre del Fichero	SOL001E_MUS_Manual_Instalacion y Configuracion_Solicit@v4.0.0_v04r00.pdf		
Autor	**everis**		
Versión	v04r00	**Fecha Versión**	08/05/2009
		Nº Total Páginas	45

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio
V01r00	Adaptación Solicit@ 3.0.2	Juan María Reina Ortiz	Global	22-01-2008
V01r01	Revisión	Cristóbal Moral Medina	Global	14-02-2008
V03r02	Revisión	Juan Antonio López Marín	Global	18-04-2008
V03r03	Actualización conex@	Juan Antonio López Marín	Global	04-06-2008
V03r05	Revisión	Juan Antonio López Marín	Global	27-07-2008
V03r06	Revisión	Juan Antonio López Marín	Global	20-10-2008
V04r00	Revisión	everis	Global	28-01-2009
V04r01	Actualización solicit@ en alta disponibilidad	everis	Global	08-05-2009

ÍNDICE

1 INTRODUCCIÓN

En el presente documento se describen los distintos pasos que se deben seguir para realizar la instalación del software **Manual de Instalación y Configuración de Entrega de solicit@v4.0.0**.

Solicit@ se trata de una aplicación web (servidor de aplicaciones JBoss) que utiliza tecnología Java y accede a una base de datos Oracle. El propósito de este software es la generación y publicación de formularios para la adquisición de datos de forma telemática.

1.1 Datos de la entrega

- La versión que se entrega corresponde con **Manual de Instalación y Configuración de Entrega de solicit@v4.0.0**
- La instalación no precisa un hardware específico puesto que la aplicación utiliza tecnología Java. Además se utilizan: Jboss (servidor de aplicaciones) y Oracle (base de datos).

  La entrega se realiza en un CD con todo el software y manuales necesarios para la instalación, configuración y uso.

1.2 Registro de cambios

Ir a “Hoja de Control”.

1.3 Descripción del problema/cambio

N/A

1.4 Solución técnica

N/A

2 RECURSOS

2.1 Hardware

La aplicación utiliza tecnología Java, y, por tanto, no existen restricciones en el ámbito hardware.

Por optimización del rendimiento y de la administración se recomienda desplegar BD y servidor JBoss en máquinas diferenciadas, siendo recomendable un mínimo de 2Gb de Ram para la maquina de JBoss. Adicionalmente es recomendable el despliegue con certificado digital de servidor sobre servidor frontal Apache, para habilitar el acceso vía https desde el cliente web.

2.2 Software

El Software se presenta en un CD que contiene la siguiente información.

- Documentación:
  - Recoge todos los manuales e instrucciones necesarios para la instalación, configuración y uso de la aplicación.
  - Véase "Bibliografía y referencias"
- Software base:

  Aquí se encuentra todo el software requerido para poder instalar Solicit@ (servidor de aplicaciones, drivers, fuentes corporativas...)
  - Jboss 4.0.5: Servidor de aplicaciones y contenedor de EJB y servlets.
  - Jdk 1.5.0.11: Kit de desarrollo de Java. Contiene las librerías necesarias de Java.
- Aplicación **Manual de Instalación y Configuración de Entrega de solicit@v4.0.0**:
  - Aplicación: contiene los archivos .ear, .jar, .war, .ejb3 ya compilados listos para ser ejecutados.
- Scripts de base de datos:
  - Scripts de nueva instalación.
  - Scripts de actualización.

2.3 Humanos

Para la instalación de Solicit@ el equipo de administración de sistemas del organismo debería de ser capaz de instalar este software siguiendo las instrucciones de este manual

3 INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA

3.1 Datos paramétricos.

Para conseguir que la explicación del ciclo de instalación refleje el orden lógico de ejecución, la configuración de los datos paramétricos de la aplicación se explicará en el punto 3.4.4.3 *Configuración de parámetros de la aplicación*.

3.2 Planificación de procesos.

El alta de sistemas con los que solicit@ interacciona será explicado en el punto 3.4.1 *Integración con sistemas externos* con el fin de que el ciclo de instalación aquí explicado refleje el orden lógico de ejecución.

3.3 Ejecución de scripts.

La ejecución de los scripts de creación de la base de datos será explicado en el punto 3.4.2 *Instalación / actualización de la base de datos.*

3.4 Configuración del sistema

La instalación de la aplicación se dividirá en los siguientes apartados:

- Solicitud de acceso a aplicaciones externas:
- Creación/Actualización de la base de datos.
- Instalación del software base.
- Instalación y parametrización de la aplicación.

Tras la realización de la instalación y para comprobar que todo se ha instalado correctamente se deben seguir los pasos indicados en el manual *SOL001E_MUS_Manual_Usuario_Solicit@v4.0.0_CicloComprobInstalacción_v04r00.pdf.* Este manual mostrará a la persona que ha instalado la aplicación cómo se debe mover por ella con el fin de comprobar que todo lo configurado se ha realizado satisfactoriamente.

3.4.1 Integración con sistemas externos

3.4.1.1 Solicitud de acceso a sistemas externos.

Debido a la naturaleza del proyecto será necesario solicitar acceso a los siguientes sistemas externos.

- @firma, componente de firma y acceso al sistema que hace uso del certificado digital.
- @ries, registro telemático de la Junta de Andalucía.
- Notific@, plataforma de notificaciones fehacientes (opcional, sólo si desde un sistema de tramitación o backoffice externo a Solicit@ se van a realizar notificaciones vía plataforma de notificaciones telemáticas Notific@).

La aplicación Solicit@ necesita interactuar con distintos sistemas para completar su funcionalidad como puede ser con @firma para la autenticación y la firma digital de ficheros, @ries para realizar el registro telemático de la información que el usuario considera que han sido finalizados, y Notific@ para que el usuario pueda habilitar la posibilidad de que se le realicen notificaciones fehacientes.

3.4.1.1.1 Solicitud de acceso a @firma

Enviar a la cuenta de correo de la Oficina de Administración Electrónica soporte.admonelectronica@juntadeandalucia.es escribiendo en el asunto "SOLICITUD ALTA APLICACIONES @FIRMA v5", con la siguiente información:

- Datos del solicitante: Nombre, Apellidos, Teléfono, E-mail.
- Nombre de la empresa.
- Consejería para la que se desarrolla y responsable de la Consejería.
- Nombre de la Aplicación.
- Descripción de la aplicación.
- Responsable de la aplicación: Nombre, Apellidos, Teléfono, E-mail.
- Tipo de Servicio @firma solicitado:
    - Autenticación mediante web services.
    - Firma/Multifirma de Ficheros por usuario mediante web services.
    - Firma/Multifirma de Ficheros en bloque mediante web services.
- Parámetros configuración del servicio solicitado. (ver manual)
- Plataforma @firma en la que se solicita el alta: Desarrollo / Producción.
- Fecha prevista de puesta en producción.
- Volumen aproximado de accesos/día.

Para acceder a más información de este componente puede acceder a:

https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/afirma.jsp

3.4.1.1.2 Solicitud de acceso a @ries

Para la solicitud de alta en este componente es necesario rellenar la solicitud que se presenta en la siguiente imagen, y que se deberá descargar en la web de plutón,

https://ws024.juntadeandalucia.es/pluton/ofivirtual/guias/plataformasARIES.jsp

y enviarla a la dirección de correo electrónico proyecto.aries@juntadeandalucia.es

Solicitud de alta/baja/modificación de Aplicaciones con conexión al Registro Telemático

D/Dª_____ en calidad de _____solicita el _____ (alta/baja/modificación) de la aplicación indicada en el cuadro adjunto, para el acceso al registro telemático.

Datos Aplicación	Contenido	
Consejería		
Nombre de la Aplicación		
Nombre de Trámite		
Unidad Tramitadora		
Nombre del Asunto		
¿Asociada a trámite telemático? (ciudadanos/internet)	[X] SI	[ ] NO
Direcciones de red origen de las peticiones de registro	IP	Nombre Host
¿Anexado de ficheros?	[X] SI	[ ] NO
¿Uso de Notario Electrónico? (Timestamping)	[ ] SI	[X] NO
Responsable técnico del proyecto en la Junta Andalucía		
e-mail Responsable		
¿Desarrollada por empresa externa?	[X] SI	[ ] NO
Nombre de la Empresa Externa		
Responsable técnico del Proyecto en la Empresa Externa		
e-mail Responsable Empresa		

En _____, a _____ de _____ de _____

Firma y Sello

Para acceder a más información de este componente debe acceder a

https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/aries.jsp

3.4.1.1.3 Solicitud de acceso a Notific@

Para solicitar el alta de servicios en una entidad emisora es necesario rellenar el siguiente formulario:

JUNTA DE ANDALUCIA

Formulario de alta de Servicio en una Entidad de Notificaciones Telemáticas

Información Necesaria para crear el Servicio
Nombre de la Entidad a la que pertenece: _____
Nombre del Servicio: _____
Descripción del Servicio: _____
_____
_____
_____
Consultas
Para cualquier consulta sobre el presente documento, condiciones de servicio o incidencias en el funcionamiento del mismo, por favor enviar un correo electrónico a la dirección: soporte.admonelectronica@juntadeandalucia.es
En _____, a ____ de _____ de ____. Fdo. _____

Para acceder a más información de este componente debe acceder a

https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/notifica.jsp

3.4.1.2 Comprobación del acceso a sistemas externos.

Una vez recibimos la confirmación de que nuestra solicitud ha sido aceptada y nuestra aplicación dada de alta en el servicio solicitado, debemos comprobar que realmente tenemos acceso, antes de intentar parametrizar Solicit@ con los datos recibidos. A continuación se explica como verificar el acceso a cada uno de los servicios solicitados desde el servidor donde se alojará el servidor de aplicaciones (JBoss).

NOTA: Es muy importante que esta comprobación se realice desde el servidor indicado, ya que es la aplicación, instalada en dicho servidor, la que intentará acceder a los distintos servicios, y no el equipo cliente o cualquier otro equipo del Organismo.

3.4.1.2.1 Verificación del acceso a @Firma.

Para verificar el acceso @Firma comprobaremos que podemos acceder hasta los webservices publicados con el usuario y contraseña recibidos en respuesta a la solicitud de alta.

- ***Servidor Linux:***

- Comprobar que hay conexión por el puerto 443 con el servidor de firma mediante, por ejemplo, un telnet.

  telnet <SERVIDOR DE FIRMA> 443

  y, si la comunicación es correcta, deberíamos obtener:

  Trying <SERVIDOR DE FIRMA>...

  Connected to <SERVIDOR DE FIRMA>

  Escape character is '^]'

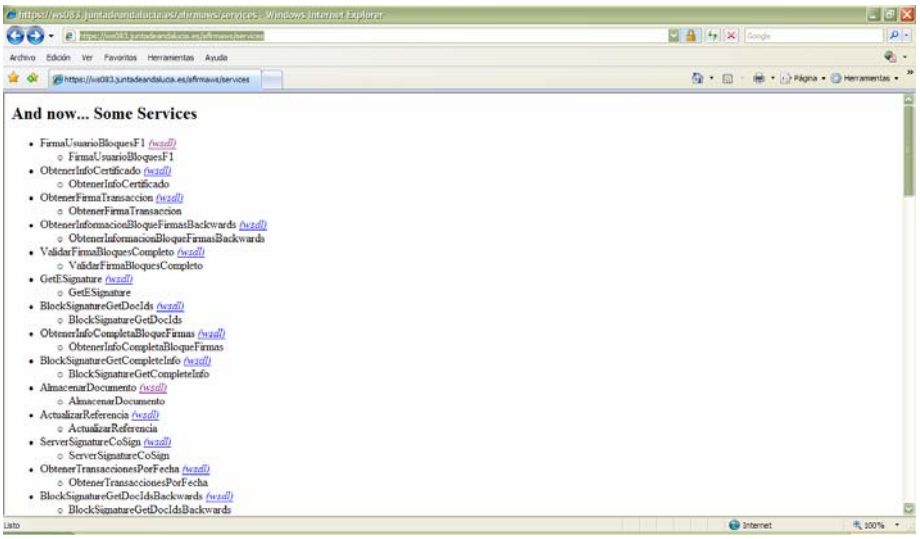
Si se dispone de interfaz gráfica y un navegador en el servidor, se puede hacer la misma prueba que en el caso de servidor Windows.

- ***Servidor Windows:***

- Desde el navegador, realizamos la llamada

https://<SERVIDOR DE FIRMA>/afirmaws/services

- Deberíamos obtener:



En ambos casos, si no se recibe el resultado esperado, se debe pensar en un error de comunicaciones entre el servidor y @Firma (puerto 443 cerrado, firewall, error de routing...)

3.4.1.2.2 Verificación del acceso a Notific@.

Para verificar el acceso Notifi@ comprobaremos que podemos acceder a la misma URL que utiliza Solicit@ para acceder.

- ***Servidor Linux:***

- Comprobar que hay conexión por el puerto 443 con el servidor de firma mediante, por ejemplo, un telnet.

telnet <SERVIDOR DE NOTIFIC@> 443

y, si la comunicación es correcta, deberíamos obtener:

Trying <SERVIDOR DE NOTIFIC@>...

Connected to <SERVIDOR DE NOTIFIC@>

Escape character is '^]'

Si se dispone de interfaz gráfica y un navegador en el servidor, se puede hacer la misma prueba que en el caso de servidor Windows.

- ***Servidor Windows:***

- Desde el navegador, realizamos la llamada

https://<SERVIDOR DE NOTIFIC@>/jboss-net/services/ServicioWEBSN

- A continuación pide un certificado digital de usuario y a continuación se debería obtener:

En ambos casos, si no se recibe el resultado esperado, se debe pensar en un error de comunicaciones entre el servidor y @Firma (puerto 443 cerrado, firewall, error de routing...)

3.4.1.2.3 Verificación del acceso a @ries.

El acceso a @ries, tanto de producción como de desarrollo, se permite únicamente desde la IP indicada en la solicitud.

- ***Servidor Linux y Windows:***

- Comprobar que hay conexión física (funciona un telnet a la IP y el puerto)

telnet ariesdes.cjap.junta-andalucia.es 8081

3.4.2 Creación / actualización de la base de datos.

3.4.2.1 Creación de una nueva base de datos.

Actualmente existen dos opciones a la hora de realizar una nueva instalación de la base de datos de Solicit@. Por un lado, se puede utilizar un script.bat que, una vez parametrizado, lanza automáticamente todos los scripts necesarios para crear la base de datos. Esta es la opción más rápida y sencilla para instalar la base de datos, y es la recomendada por **everis.**

Por otro lado, se entrega el juego completo de scripts preparados para lanzarlos de forma manual, previa parametrización de los mismos.

En la ruta "[CD_SOLICITA]/BBDD" se encuentra el fichero

"000-PASOS INSTALACION.txt"

que puede contener notas más actualizadas sobre la instalación que las que aquí se exponen.

3.4.2.1.1 Instalación Automatizada (recomendada)

Los scripts necesarios para este modo de instalación los podemos encontrar en:

"[CD_SOLICITA]/BBDD/Nueva Instalacion Automatizada"

Este modo de instalación permite, configurando un único fichero, instalar la base de datos. Tan sólo requiere una máquina con acceso al servidor de base de datos y tener instalado "SQLPlus", incluido en cualquier cliente de Oracle.

A. **Desde un cliente Windows**

- **Paso 1:** Copiar el contenido del directorio "[CD_SOLICITA]/BBDD/Nueva Instalacion Automatizada" a una ubicación donde poder editar los scripts.
- **Paso 2:** Configurar el script

  "100.-INSTALACION_AUTOMATIZADA.bat "

Este fichero tiene una primera sección donde se deben configurar los siguientes parámetros. Esta tarea es aconsejable que sea realizada previa consulta con el DBA de la base de datos.

- Nombre de la instancia de base de datos

  DATABASE_INSTANCE=

- Nombre de usuario system

  SYSTEM_USER= < system >

- Contraseña del usuario system

  SYSTEM_PWD=

- Nombre del usuario propietario

  OWNER= < PL_OWNER >

- Contraseña del usuario propietario

  OWNER_PWD= < SOLICITA > (Se recomienda cambiar las contraseñas por defecto)

- Nombre del usuario que utilizará la Solicit@ para conectarse a la base de datos.

  WEBUSR= <PL_WEBUSR >

- Contraseña del usuario que utilizará la Solicit@ para conectarse a la base de datos.

  WEBUSR_PWD= < SOLICITA2007 > (Se recomienda cambiar las contraseñas por defecto)

- ¿Desea crear sinónimos públicos? (0=no, 1=si)

  PUBLIC_SYN= < 0 >

- ¿Desea crear sinónimos privados? (0=no, 1=si)

  PRIV_SYN= < 1 >

- Nombre del tablespace de Datos

  TABLESPACE_D= < TS_SOLICITA_D >

- Datafile del tablespace de Datos (el directorio debe existir)

DATAFILE_D= < "C:\BBDD\TS_SOLICITA_D_000001.ora" >

- Tamaño inicial del tablespace de Datos (en MB)

DATAFILESIZE_D= < 1024 >

- Nombre tablespace de Índices

TABLESPACE_I= < TS_SOLICITA_I >

- Datafile del tablespace de Índices (el directorio debe existir)

DATAFILE_I= < "C:\BBDD\TS_SOLICITA_I_000001.ora" >

- Tamaño inicial del tablespace de Índices (en MB)

DATAFILESIZE_I= < 1024 >

- Nombre tablespace de Blobs

TABLESPACE_B= < TS_SOLICITA_LOB >

- Datafile del tablespace de Blobs (el directorio debe existir)

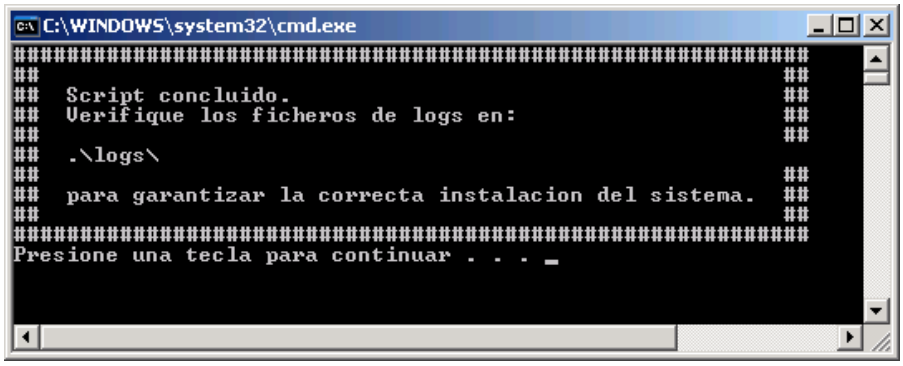
DATAFILE_B= < "C:\BBDD\TS_SOLICITA_B_000001.ora" >

- Tamaño inicial del tablespace de Blobs (en MB)

DATAFILESIZE_B= < 1024 >

- **Paso 3:** Ejecutar 100.-INSTALACION_AUTOMATIZADA.bat teniendo en cuenta que debe realizarse en el mismo lugar donde se encuentran los demás scripts.

Al finalizar el script debemos recibir por pantalla el siguiente mensaje:



- **Paso 4:** Verificar que no se ha producido ningún error al ejecutar los scripts.

Para ello acceder a la ruta indicada (normalmente ".\logs") y comprobar que no se han producido errores al ejecutar ninguno de los scripts. En caso de error consultar con el DBA encargado las posibles causas.

B. **Desde un cliente Linux**

Para realizar la instalación automatizada desde un cliente Linux, procederemos del mismo que con un cliente Windows, pero con las siguientes particularidades:

- **Paso 1:** Copiar el contenido del directorio "[CD_SOLICITA]/BBDD/Nueva Instalacion Automatizada" a una ubicación donde poder editar los scripts.

Paso 2: Otorgar permisos de ejecución al script "100.-INSTALACION_AUTOMATIZADA.sh "

chmod +x 100.-INSTALACION_AUTOMATIZADA.sh

- **Paso 3:** Para evitar posibles problemas con los retornos de carro en el script, pasar la herramienta "dos2unix" al script.

  dos2unix 100.-INSTALACION_AUTOMATIZADA.sh

- **Paso 4:** Editar y ejecutar el script del mismo modo que en Windows.

(NOTA: Si la base de datos se encuentra alojada en un servidor Windows, la ruta de los datafiles debe indicarse con doble barra invertida. Ej.: "C:\\BBDD\\TS_SOLICITA_D_000001.ora"**)**

3.4.2.1.2 Instalación Manual

Los scripts necesarios para este modo de instalación los podemos encontrar en:

"[CD_SOLICITA]/BBDD/Nueva Instalacion"

Para lanzarlos, podemos utilizar "SQLPlus" o alguna otra herramienta gráfica como "Toad" o "PLSQL Developer". Con este modo de creación, tendremos que editar la mayoría de los scripts para que la instalación se ajuste a las directrices propias del Organismo sobre bases de datos. Esta tarea es aconsejable que sea realizada previa consulta con el DBA del Organismo. A continuación se describen los pasos a seguir en una instalación por defecto:

(1) - Acceder con el usuario system y ejecutar el script (previa modificación de las localizaciones de los tablespaces.)

110-CREA_TABLESPACES_ESQUEMA.sql

(2) - Acceder con el usuario PL_OWNER/SOLICITA y ejecutar los scripts siguientes en el orden dado:

(2.1) Scripts de creación de tablas:

210-CREA_TABLAS_ADMINISTRACION.sql

211-CREA_TABLAS_GENERADOR_FORMULARIOS.sql

212-CREA_TABLAS_GENERADOR_ESCRITOS.sql

213-CREA_TABLAS_OFICINA_VIRTUAL.sql

214-CREA_TABLAS_REPLICA.sql

(2.2) Se cargan de datos las tablas:

220-INSERTA_PARAMETRICAS_ADMINISTRACION.sql

221-INSERTA_PARAMETRICAS_GENERADOR_FORMULARIOS.sql

223-INSERTA_DATOS_OFICINA_VIRTUAL.sql

224-INSERTA_DATOS_TABLAS_REPLICA.sql

(2.3) Se crean las constraints necesarias:

230-CREA_CONSTRAINTS_ADMINISTRACION.sql

231-CREA_CONSTRAINTS_GENERADOR_FORMULARIOS.sql

232-CREA_CONSTRAINTS_GENERADOR_ESCRITOS.sql

233-CREA_CONSTRAINTS_OFICINA_VIRTUAL.sql

234-CREA_CONSTRAINTS_REPLICA.sql

(2.4) Se crean las secuencias necesarias:

240-CREA_SECUENCIAS_ADMINISTRACION.sql

241-CREA_SECUENCIAS_GENERADOR_FORMULARIOS.sql

242-CREA_SECUENCIAS_GENERADOR_ESCRITOS.sql

243-CREA_SECUENCIAS_OFICINA_VIRTUAL.sql

(2.5) Se crean los triggers necesarios:

251-CREA_TRIGGERS_GENERADOR_FORMULARIOS.sql

252-CREA_TRIGGERS_GENERADOR_ESCRITOS.sql

(2.6) Se crean los roles necesarios:

260-CREA_ROL.sql

(2.7) Se crean los índices necesarios:

270-CREA_INDICES_ADMINISTRACION.sql

271-CREA_INDICES_GENERADOR_FORMULARIOS.sql

272-CREA_INDICES_GENERADOR_ESCRITOS.sql

273-CREA_INDICES_OFICINA_VIRTUAL.sql

(3) - Acceder con el usuario system/sys y ejecutar el siguiente script que crea el usuario para la aplicación:

310-CREA_USUARIO.sql

(4) - A continuación creamos los sinónimos. Debemos elegir entre públicos o privados:

(4.1) En caso de usar sinónimos públicos usar usuario propietario del esquema PL_OWNER/SOLICITA

410-CREA_SINONIMOS_PUBLICOS.sql

(4.2) En caso de querer sinónimos privados:

- Acceder con el usuario PL_OWNER/SOLICITA y ejecutar el siguiente script.

420-PERMISOS_SINONIMOS.sql

- Acceder con el usuario PL_WEBUSR/SOLICITA2007 y ejecutar el siguiente script.

421-CREA_SINONIMOS.sql

Nota: Es importante que la persona responsable de lanzar estos scripts en la base de datos verifique que la nomenclatura de los usuarios que se crean (ficheros 110-CREA_TABLESPACES_ESQUEMA.sql y 310-CREA_USUARIO.sql) sigue los criterios de nomenclatura corporativos. En el caso en que no se usen los nombres distribuidos será necesario modificar el fichero "solicita_orcl_pool-ds.xml" que se debe encontrar en JBOSS tal y como se explica en el punto 3.4.4.2 y en el orden de ejecución mostrado anteriormente se deberá acceder a la base de datos con los usuarios que se hayan creado.

3.4.2.2 Actualización de la base de datos.

En caso de tener ya instalada una versión previa de Solicit@ se utilizarán los scripts que permiten actualizar la base de datos para que pueda funcionar con la nueva versión de la aplicación. Estos se encuentran en la ruta:

"[CD_SOLICITA]/BBDD/Upgrades"

Dentro del directorio Upgrades se pueden encontrar clasificados estos scripts. El nombre de cada directorio indica la versión de partida y destino de la actualización. La actualización debe realizarse de **forma gradual, pasando los scripts necesarios consecutivamente hasta llegar a la versión actual.**

Dentro de este mismo directorio está el fichero

000-PASOS ACTUALIZACION.txt

que puede contener notas más actualizadas sobre la actualización que las que aquí se exponen.

Será necesario editar los scripts ajustándolos a la instalación existente (usuario propietario del esquema, tablespaces...) Esta tarea es aconsejable que sea realizada previa consulta con el DBA del Organismo.

Estos scripts deben lanzarse utilizando "SQLPlus" o alguna otra herramienta gráfica como "Toad" o "PLSQL Developer".

Para realizar cada actualización, basta ejecutar los scripts según se indica a continuación:

(9) - Con el usuario PL_OWNER/SOLICITA ejecutar el siguiente script que realiza la actualización a la siguiente versión.

900-UPGRADEvxxxTOvxxx.sql

(9.1) En caso de usar sinónimos públicos usar usuario propietario del esquema PL_OWNER/SOLICITA

910-CREA_SINONIMOS_PUBLICOS.sql

(9.2) En caso de querer sinónimos privados:

- Acceder con el usuario PL_OWNER/SOLICITA y ejecutar el siguiente script.

920-PERMISOS_SINONIMOS.sql

- Acceder con el usuario PL_WEBUSR/SOLICITA2007 y ejecutar el siguiente script.

921-CREA_SINONIMOS.sql

Todos los scripts están configurados para colocar un fichero de log en "C:\". Se debe comprobar, revisando dichos ficheros, que la actualización se ha realizado con éxito y sin errores.

Después de haber realizado las distintas actualizaciones, es muy posible que sea necesario revisar la parametrización de la aplicación en busca de campos añadidos para las nuevas funcionalidades.

3.4.3 Instalación del software base

El software que es necesario instalar se encuentra en el CD distribuido en la ruta "CD Solicit@\Software" base.

3.4.3.1 Instalación de JDK:

En la ruta "CD Solicit@\Software Base\Jdk 1.5.0_11" se encuentran los instaladores de JDK_1.5.0_11 para los distintos sistemas operativos. Debemos elegir la carpeta que corresponda con el S.O. y ejecutar el archivo correspondiente para iniciar la instalación. Para que el funcionamiento sea el correcto, es necesario declarar una variable de entorno en el sistema operativo (*JAVA_HOME*) con la ruta en la que se haya instalado.

3.4.3.2 Instalación del servidor de aplicaciones

Dado que Solicit@ hace uso de EJB3, es necesario aplicar un parche al JBoss. En está distribución de Solicit@ y con el fin de simplificar el proceso de instalación, se ofrecen dos posibilidades. Por un lado se distribuye una versión de JBoss ya parcheada (punto A, recomendado por **everis).** Y por otro, se distribuye un JBoss sin modificar, así como las herramientas e instrucciones necesarias para aplicar dicho parche, para aquellos administradores que prefieran tener mayor control sobre el proceso (Punto B).

A.- JBoss con parche previamente aplicado:

En este caso, para instalar JBoss basta con descomprimir el archivo "CD Solicit@\Software Base\JBoss+Parche\jboss-4.0.5.GA+PARCHE.zip" en la ruta deseada y definir la variable de sistema (JBOSS_HOME) apuntando a dicha ruta.

B.- JBoss sin modificar:

En este caso se hace uso de los archivos que podemos encontrar en la ruta "CD Solicit@\Software Base\JBoss_Original\"

a. **Instalación de JBOSS**: Debemos descomprimir el archivo "**jboss-4.0.5.GA.zip"** en la ruta en la que queramos instalar este componente. Para que el funcionamiento sea el correcto, es necesario declarar una variable de entorno del sistema operativo (*JBOSS_HOME*) con la ruta en la que se haya instalado.

b. **Instalación de ANT**: Debemos descomprimir el archivo "**apache-ant-1.7.0-bin.zip**" en la ruta deseada. Se ha detectado un bug en ANT, para solventarlo es suficiente con descomprimir el archivo en una carpeta cuyo nombre

no tenga caracteres extraños, se recomienda poner el nombre "ant". Con objeto de evitar conflictos, de manera opcional, puede declararse una variable de entorno del sistema operativo (*ANT_HOME*) con la ruta como destino la subcarpeta "bin" de la carpeta de instalación del ANT.

c. **Instalación del parche para el uso de EJB 3**: Debemos descomprimir el archivo "**jboss-EJB-3.0_RC9_Patch_1.zip**" en la ruta deseada. Para llevar a cabo la instalación deberemos ejecutar el "**install.xml**" mediante el comando:

ant -f install.xml -Djboss.server.config=[configuración de despliegue (por defecto all)].

Éste comando debe ser ejecutado desde consola en el directorio del parche descomprimido.

3.4.3.3 Instalación de la librería de acceso a la BBDD:

Sea cual sea el camino elegido para instalar JBoss, es necesario, para su correcto funcionamiento, copiar la librería "ojdbd14.jar" en el directorio {JBOSS_HOME}/server/all/lib. Estas librerías se pueden encontrar en la ruta "CD Solicit@\Software Base\OracleDrivers\". Es importante comprobar, una vez que la instalación ha concluido y Solicit@ está operativo, que los documentos se están almacenando correctamente en la base de datos. Para ello se puede ir a la tabla PL_OWNER.OFIVDOCSOLICI y comprobar que los .pdf almacenados en las columnas LB_DOCFIRMADO, LB_DOCPRESENTADO y LB_SOLICITUD no están corruptos y son correctos (contienen su identificador de firma o presentación, según la columna). Es un fallo típico, debido a un driver incorrecto, el encontrar un documento corrupto de 86 bytes en lugar del .pdf esperado.

En principio se recomienda utilizar el driver más nuevo disponible (aunque sea de una versión de Oracle posterior a la que se está usando), aunque se han visto casos en los que esta no es necesariamente la mejor opción. También podemos descargar drivers más actualizados de la web de Oracle.

3.4.3.4 Configuración / optimización del servidor de aplicaciones.

Pese a que es posible utilizar el servidor de aplicaciones (JBoss) tal y como viene por defecto, es posible, y recomendable, hacerle algunas modificaciones en su configuración para mejorar su funcionamiento o adaptarlo a las necesidades del entorno de sistemas. A continuación se detallan algunas sugerencias.

3.4.3.4.1 Modificaciones en el tomcat

JBoss incluye un Tomcat para ejercer de contenedor de servlets. Es por lo tanto parte fundamental de la funcionalidad del servidor de aplicaciones y merece revisar su configuración. Este Tomcat lo tenemos en {JBOSS_HOME}/server/all/deploy/jbossweb-tomcat55.sar y el fichero de configuración del mismo es el server.xml, dentro de este directorio.

Modificaciones posibles son:

- Cambiar el puerto de escucha http.
- Eliminar la "escucha" por http si no fuera necesaria (este es el caso de tener configurado un Apache con modjk). Para ello borramos o comentamos del fichero el siguiente párrafo:

```
<Connector port="8080" address="${jboss.bind.address}"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true"/>
```

Asimismo, se debe examinar el fichero {JBOSS_HOME}/server/all/deploy/jbossweb-tomcat55.sar/conf/web.xml, el cual contiene las particularidades de este Tomcat instalado dentro de un JBoss. Modificaciones posibles son:

- Eliminar el listado de directorios en los que no se halle el fichero definido como *welcome file*, es decir, aquel que por omisión se sirve al acceder a dicho directorio. Esto lo hacemos modificando el siguiente párrafo:

```
<init-param>
    <param-name>listings</param-name>
    <param-value>true</param-value>
</init-param>
```

Debemos poner false en lugar de true.

- También es recomendable inhabilitar el modo "development". Esto se hace añadiendo justo detrás del párrafo:

```
<servlet>
    <servlet-name>jsp</servlet-name>
    <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
```

Lo siguiente:

```
<init-param>
    <param-name>development</param-name>
    <param-value>false</param-value>
</init-param>
```

3.4.3.4.2 Configurar el log4j

El logging de jboss tiene un importante impacto en el rendimiento. Por defecto, deja log tanto en la consola como el fichero de log con el nivel mínimo de INFO. Recomendamos eliminar el logging hacia la consola y poner un nivel mínimo de log "ERROR" cuando pasemos la aplicación a producción. Para esto, editamos el fichero {JBOSS_HOME}/server/all/conf/log4j.xml y modificamos el siguiente fragmento:

```
<root>
<appender-ref ref="CONSOLE"/>
<appender-ref ref="FILE"/>
</root>
```

Para que quede así:

```
<root>
<priority value="ERROR" />
<appender-ref ref="FILE"/>
</root>
```

3.4.3.4.3 Configuración de la máquina virtual java (JVM).

Debemos configurar la máquina virtual para que arranque con las siguientes asignaciones de memoria:

-Xms1024m -Xmx1024m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:PermSize=1024m -XX:MaxPermSize=1024m

Si por alguna razón no se pudiera asignar estos tamaños a la máquina virtual, se haría manteniendo siempre la relación existente. Es decir

Xms = Xmx = PermSize = MaxPermSize

y

NewSize = MaxNewSize = ¼ Xms

Esto se configura de manera diferente según el sistema operativo utilizado:

- Si se trata de **Linux** se especifican estos parámetros en el fichero run.conf, dentro de directorio bin. Modificamos la línea que contiene las opciones de arranque de la máquina virtual, e insertamos estos parámetros entre los que ya trae:

  JAVA_OPTS="-Xms1024m -Xmx1024m -XX:NewSize=256m -XX:MaxNewSize=256m

  -XX:PermSize=1024m -XX:MaxPermSize=1024m (resto de parámetros)

- Si estamos en un sistema **Windows**, el fichero a modificar será el mismo run.bat. Buscamos lo siguiente:

  rem JVM memory allocation pool parameters. Modify as appropriate.

  set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m

  y lo dejamos así:

  rem JVM memory allocation pool parameters. Modify as appropriate.

  set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx1024m -XX:NewSize=256

  -XX:MaxNewSize=256m -XX:PermSize=1024m -XX:MaxPermSize=1024m

3.4.3.5 Configuración de Solicit@ en alta disponibilidad

3.4.3.5.1 Introducción.

La configuración de un sistema de aplicaciones web en alta disponibilidad requiere siempre la redundancia en la instalación de los servidores de aplicaciones. Existen dos modos de hacer uso de esta redundancia:

- Se dispone de un nodo activo y uno o más nodos en espera pasiva. Habrá de habilitarse un sistema que permita detectar problemas de funcionamiento en el nodo activo y en este caso entrará en funcionamiento alguno de los nodos pasivos.
- Se dispone de dos o más nodos activos que se reparten la carga del trabajo. Un sistema externo a estos nodos detectará los posibles problemas que puedan presentar y en este caso el nodo afectado dejará de recibir trabajo hasta su recuperación.

Es obvio que de las dos posibilidades planteadas es preferible la segunda ya que además de lograr la alta disponibilidad se logra un aprovechamiento completo de los recursos destinados a lograrla con las correspondientes mejoras de rendimiento y su escalabilidad con la incorporación de nodos activos al sistema. Es por ello que en este documento se explicará cómo conseguir una infraestructura de este tipo para el despliegue del sistema Solicit@.

3.4.3.5.2 Problemas a resolver.

- Acceso local al sistema de archivos

El despliegue de diversos servidores de aplicaciones en máquinas distintas puede suponer problemas si estas aplicaciones escriben datos en el sistema de archivos local de los equipos. Es obvio que esta escritura de datos no estará coordinada y originará problemas. En el caso que nos ocupa de Solicit@ el sistema sólo requiere escribir en el sistema local de archivos los ficheros JSP correspondientes a los modelos que se van confeccionando desde el editor de formularios. Si no se presta atención a este problema ocurrirá que el despliegue del formulario sólo se realizará en el nodo asignado por el sistema de balanceo de carga quedando el resto de los nodos con información obsoleta.

- Persistencia de sesiones de usuario

Las aplicaciones web almacenan la información de sesión de la memoria volátil de los servidores. Si no se presta atención a este problema se darán situaciones de pérdida de sesión de los usuarios finales, ya sea por un mal reparto en la carga del trabajo o por la caída no controlada de nodos del sistema.

3.4.3.5.3 Solución acceso local al sistema de archivos.

Para este problema existen tres soluciones que se detallan a continuación:

- El acceso local de los servidores se realiza a una partición compartida del sistema de archivos. Existen diversos modos de lograr este recurso compartido como por ejemplo NFS o GFS. No entra en el alcance de este documento detallar el uso de estos sistemas o cualquier otra alternativa.
- La solución alternativa consiste en garantizar que toda escritura local en el sistema de archivos la haga un único nodo y habilitar adicionalmente un procedimiento de réplica (rsync o similares) que se ejecutará periódicamente. En el caso que nos ocupa sería por tanto necesario garantizar que toda la edición de formularios se realice en un único nodo sin balancear.
- La última de las soluciones, específica para Solicit@, consistiría en el acceso individual a todos los nodos del sistema para desplegar el JSP de un formulario editado.

3.4.3.5.4 Solución persistencia de sesiones de usuario.

Para este problema existen dos soluciones que se detallan a continuación:

- El balanceador de reparto de carga de trabajo entre nodos debe identificar la sesión a la que pertenece cada una de las peticiones que ha de gestionar. Todas las peticiones de una única sesión serán redirigidas siempre al mismo nodo. Del mismo modo se deberá detectar la creación de nuevas sesiones para asignar nodos a los nuevos clientes. Generalmente las aplicaciones web basadas en JAVA identifican la sesión mediante el envío al navegador cliente de una cookie de sesión única con el nombre JSESSIONID vinculada a la ruta y al servidor de la aplicación. Es necesario destacar en este punto que la caída de un nodo del sistema provocaría en una reasignación a otro nodo de los usuarios conectados provocándoles una pérdida de sesión.
- Existe la alternativa de que todos los nodos intercambien mensajes de los datos de sesión que manejan. De este modo no es necesario gestionar ninguna asignación de nodo a los clientes ni se ven afectados por la caída de alguno de ellos. Por el contrario esta solución implica una mayor carga de trabajo y tráfico de red para los servidores de aplicaciones.

3.4.3.5.5 Solución ejemplo basada en Apache.

Se plantea a continuación una solución al problema presentado basado en el servidor web APACHE como elemento balanceador de carga entre nodos, en este caso los servidores de aplicaciones JBOSS con Solicit@ instalado.

Aunque no se detallará en este documento es también posible la configuración de varios servidores web APACHE tal como se detallará y realizar sobre ellos un balanceo con un sistema externo de alta disponibilidad hardware o software.

La vinculación entre APACHE y JBOSS se realizará con el módulo “mod-jk” de APACHE que permite vincularlo a cualquier servidor de aplicaciones que implemente el protocolo Ajpv13.

3.4.3.5.5.1 Definición de worker.properties

En primer lugar será necesario identificar todos los nodos donde ha sido desplegado JBOSS mediante un fichero de texto plano que denominaremos “worker.properties”. En él se definirán nuestros “workers” donde un “worker” es un servidor o conjunto de servidores. Se supone que los servidores JBOSS escuchan en el puerto por defecto apj13 8009.

```
# Listado de workers públicos definidos en este fichero
worker.list=solicita

# Se definirá un worker de tipo host para cada instalación de JBOSS
# Definimos un primer nodo JBOSS
worker.solicita1.host=nodo1.midominio.es
worker.solicita1.port=8009
worker.solicita1.type=ajp13
worker.solicita1.lbfactor=1
worker.solicita1.socket_timeout=5
worker.solicita1.recycle_timeout=10

# Definimos un segundo nodo JBOSS
worker.solicita2.host=nodo2.midominio.es
worker.solicita2.port=8009
worker.solicita2.type=ajp13
worker.solicita2.lbfactor=1
worker.solicita2.socket_timeout=5
worker.solicita2.recycle_timeout=10

# Se podrían definir más nodos y asignarles pesos de reparto de carga
# haciendo uso de worker.solicitaN.lbfactor

# Definimos el worker global solicita de tipo lb load balancing
worker.solicita.type=lb
worker.solicita.balance_workers=solicita1,solicita2
worker.solicita.sticky_session=false
worker.solicita.session_cookie=JSESSIONID
```

El valor del atributo "`worker.solicita.sticky_session`" determinará si se realizará la asignación de nodos a los usuarios (true) o no (false). No será necesario si los servidores JBOSS intercambian información de sesión.

El valor del atributo "`worker.solicita.session_cookie`" sólo es relevante si establecimos a "true" el parámetro anterior. Indica el identificador de la cookie de sesión enviada por el servidor de aplicaciones JBOSS a los navegadores clientes.

En el siguiente enlace es posible encontrar documentación adicional sobre configuración avanzada del fichero worker.properties:

Enlace: http://tomcat.apache.org/connectors-doc/reference/workers.html

3.4.3.5.5.2 Inclusión de worker.properties en la configuración de APACHE.

En la configuración de APACHE será necesario incluir y cargar el módulo mod_jk. También se establecerá el vínculo con el fichero "worker.properties" que habremos ubicado previamente en el directorio ${APACHE_HOME}/conf.

```
Include conf/mod-jk.conf
LoadModule jk_module modules/mod_jk.so
JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogLevel info
```

Finalmente se establecerán los puntos de montaje de los contextos de Solicita. Esta inclusión se realizará en el lugar apropiado del fichero de configuración, podrá estar en un Host Virtual o a nivel global en función de la particularidad de cada instalación.

```
JkMount /oficinaVirtual* solicita
JkMount /administracion* solicita
JkMount /solicita20* solicita
```

3.4.3.5.5.3 Configuración de los nodos JBoss.

Todos los servidores de aplicación JBoss con Solicit@ instalado deberán estar agrupados en una partición JBoss. Estos nodos deberán encontrarse ubicados en una misma VLAN que permita el intercambio de mensajes UDP entre ellos. Es posible la construcción de una partición JBoss con servidores instalados en diferentes VLAN pero en este caso el intercambio de mensajes UDP no es posible y ha de efectuarse mediante TCP/IP lo que requiere una configuración avanzada que no será cubierta en este documento.

Para cada uno de los servidores JBoss habrá que configurar el identificador de partición a la que pertenecen. El modo más sencillo para realizarlo es mediante la edición del fichero ${JBOSS_HOME}/bin/run.conf donde añadiremos un nuevo parámetro a la variable JAVA_OPTS. La nueva variable a añadir será la siguiente:

```
-Djboss.partition.name=SolicitaDesa
```

El nombre asignado a la partición debe ser el mismo para todos sus nodos. También pueden convivir en la misma VLAN instalaciones de JBoss con Solicit@ pertenecientes a diversos entornos siempre que se les asignen distintos nombres de partición: SolicitaDesa, SolicitaPruebas,.....

Finalmente para cada nodo JBoss es necesario indicarle su correspondencia con los declarados en el fichero worker.properties. Para el nodo "solicitaX" habrá que editar el siguiente fichero:

```
${JBOSS_HOME}/server/all/deploy/jbossweb-tomcat50.sar/server.xml
```

Será necesario localizar el elemento "Engine" dentro del documento XML y establecer un atributo jvmRoute con el nombre de nodo asignado.

```
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="solicitaX">
... ...
</Engine>
```

También será necesario editar el siguiente fichero:

```
${JBOSS_HOME}/server/all/deploy/jbossweb-tomcat50.sar/META-INF/jboss-service.xml
```

Donde habrá que establecer a "true" el atributo UseJK:

```
<attribute name="UseJK">true</attribute>
```

Documentación adicional en el siguiente enlace:

Enlace: http://docs.jboss.org/jbossas/guides/clusteringguide/r2/en/html/

3.4.3.5.6 Eliminar el cluster de JBoss.

Para el caso en el que no se desea configurar Solicit@ en alta disponibilidad es recomendable la eliminación del cluster de JBoss.

Por defecto, la configuración "all", utilizada para el correcto despliegue de la aplicación, inicializa el funcionamiento en cluster de JBoss. Esto puede dar lugar a constantes intentos de búsqueda de un segundo servidor, trayendo como consecuencia una sobrecarga de tráfico UDP en la red.

Sugerimos, por este motivo y en aras a un servidor más "ligero", inhabilitar esta funcionalidad en el jboss, siguiendo los siguientes pasos:

- Eliminar {JBOSS_HOME}/server/all/farm
- Eliminar {JBOSS_HOME}/server/all/deploy-hasingleton
- Eliminar {JBOSS_HOME}/server/all/deploy/cluster-service.xml
- Eliminar {JBOSS_HOME}/server/all/deploy/tc5-cluster.sar
- Eliminar {JBOSS_HOME}/server/all/deploy/deploy.last/farm-service.xml
- Eliminar {JBOSS_HOME}/server/all/deploy/deploy-hasingleton-service.xml
- Dentro del directorio {JBOSS_HOME}/server/all/deploy/jms, eliminar su contenido, y sustituirlo por el de {JBOSS_HOME}/server/default/deploy/jms folder.
- Editar el fichero {JBOSS_HOME}/server/all/deploy/jbossweb-tomcat55.sar/META-INF/jboss-service.xml, y eliminar este fragmento:

```
<!--
  Needed if using HTTP Session Clustering or if the
  ClusteredSingleSignOn valve is enabled in the tomcat server.xml file
-->
<depends>jboss.cache:service=TomcatClusteringCache</depends>
```

3.4.3.6 Configuración del juego de caracteres

Debemos configurar correctamente el juego caracteres que va a utilizar la aplicación: iso-8859-15. Esto es un parámetro adicional que debemos pasar a la Java Virtual Machina, por lo que las JAVA_OPTIONS quedarán de la siguiente forma:

- Si se trata de **Linux** se especifican estos parámetros en el fichero run.conf, dentro de directorio bin. Modificamos la línea que contiene las opciones de arranque de la máquina virtual, y lo añadimos a los anteriormente descritos en el epígrafe anterior:

  JAVA_OPTS="-Xms1024m -Xmx1024m -XX:NewSize=256m -XX:MaxNewSize=256m

  -XX:PermSize=1024m -XX:MaxPermSize=1024m ***–Dfile.encoding=iso-8859-15*** *(resto de parámetros)*

- Si estamos en un sistema **Windows**, el fichero a modificar será el mismo run.bat. Añadimos este nuevo parâmetro a lãs opciones de Java:

  rem JVM memory allocation pool parameters. Modify as appropriate.

  set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx1024m -XX:NewSize=256

  -XX:MaxNewSize=256m -XX:PermSize=1024m -XX:MaxPermSize=1024m ***–Dfile.encoding=iso-8859-15***

3.4.3.7 Renderizado de PDF

En sistemas Red Hat se debe habilitar el modo "headless", para indicar que si bien requerimos recursos gráficos para el renderizado de los PDF, el propio servidor no tiene configurado ningún dispositivo de visualización. Así evitamos que falle esta funcionalidad al no encontrar un servidor X, algo que no es necesario en ningún caso.

Esto se hace añadiendo a las opciones de la máquina virtual (JAVA_OPTS) lo siguiente:

JAVA_OPTS="-Xms1024m -Xmx1024m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:PermSize=1024m -XX:MaxPermSize=1024m –Dfile.encoding=iso-8859-15 ***-Djava.awt.headless=true*** *(resto de parámetros)*

3.4.4 Instalación y parametrización de la aplicación.

Como se ha comentado en la introducción Solicit@ se trata de una aplicación web que es servida por peticiones a un servidor JBOSS. A continuación se describen los pasos para instalar el software y parametrizarlo para cada entorno. Los pasos a seguir serán:

- Carga de la aplicación en el servidor de aplicaciones (JBoss)
- Configuración del acceso a la base de datos.
- Parametrización de la aplicación

3.4.4.1 Carga de la aplicación en el servidor de aplicaciones.

Una forma típica de instalar el software sería copiar los ficheros que se encuentran en la ruta del CD-ROM *"\Aplicacion\solicita\"* en la ruta del servidor "{JBOSS_HOME}\ *server\all\deploy\".*

3.4.4.2 Configuración del acceso a la base de datos.

El fichero **"solicita_orcl_pool-ds.xml"** ("CD Solicit@\Software Base") contiene toda la información sobre la conexión a la base de datos. Especifica los parámetros de conexión y es necesario editarlo y modificarlo según la base de datos que se use.

Este fichero debe ubicarse en la carpeta: "{JBOSS_HOME}\server\all\deploy" y su contenido es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
```

```
<local-tx-datasource>
        <jndi-name>PlataformaDS</jndi-name>
    <connection-url>jdbc:oracle:thin:@[HOSTNAME]:[PUERTO_ORACLE]:[INSTANCIA]</connection-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <user-name>[USUARIO]</user-name>
    <password>[CONTRASEÑA]</password>
    <min-pool-size>10</min-pool-size>
   <max-pool-size>50</max-pool-size>
    <idle-timeout-minutes>20</idle-timeout-minutes>
 </local-tx-datasource>
</datasources>
```

Nota: Si la instalación de base de datos ha sido la que se especifica por defecto únicamente será necesario modificar en este fichero las siguientes anotaciones: [P Servidor], [puerto escucha], [instanciaBBDD] (estos datos se deben especificar sin las marcas "[ ]").

3.4.4.3 Configuración de parámetros de la aplicación.

Salvo parte de la configuración de los parámetros para la integración con Notific@ y @ries, la mayoría de los parámetros de la aplicación se debe realizar directamente sobre las tablas PLATP_PARAMETROS y PLATP_PARAMCONFIG. Estos parámetros se leen únicamente al arrancar el JBoss, por lo que, en caso de modificar alguno de ellos, será necesario reiniciar el JBoss.

- **TABLA PLATP_PARAMETROS**

NOMBRE	DESCRIPCION	VALOR
NUM_LINEAS_PAGINACION	Número de líneas de paginación. Indica el número de registros que se mostrarán por listado de modelos.	15
PREFIJO_SOLICITUD	Prefijo para el código de la solicitud	just-
SUFIJO_SOLICITUD	Sufijo para el código de la solicitud	-tra

Nota: El parámetro PREFIJO_SOLICITUD y SUFIJO_SOLICITUD componen en la oficina virtual el código de operación que identifica de forma unívoca una solicitud. Este código de operación engloba todos los documentos que conlleve el trámite que se esté realizando. El formato del código de operación será el siguiente:

PREFIJO_SOLICITUD<año><secuencial 6 cifras>SUFIJO_SOLICITUD

SCHEMA	Esquema propietario. Este parámetro se utiliza para hallar los dominios de la aplicación.	PL_OWNER

- **PLATP_PARAMCONFIG**

NOMBRE	DESCRIPCION	VALOR
[CONEXIONBD]	Nombre de la conexión a la base de datos en el fichero **"solicita_orcl_pool-ds.xml"**	java:/PlataformaDS
[PREFIJOSOLICITA]	Prefijo para construir el nombre de los campos de un modelo.	plat
NOMBRE_POOL_CONEXIONES	TREWA, NOMBRE_POOL_CONEXIONES	Default

USAR_ENCRIPTACION	¿Encriptar codigo de firma para su visualización en Oficina Virtual?		1
[TEXTO_FIRMA]	Texto que acompaña a la firma digital	Permite la verificación de una copia de este documento electrónico en la dirección: https//ws050.juntadeandalucia.es/verificarFirma/; este documento incorpora firma electrónica reconocida de acuerdo a la ley 59/2003, de 19 de diciembre, de firma electrónica.	
[SSL_ENABLED]	¿Utiliza la aplicación SSL?		0

Nota: Este parámetro debe configurarse a 1 cuando el acceso a la aplicación se vaya a realizar utilizando https (colocando un Apache delante o similar).

(Parámetros de integración con @ries)

USAR_REGISTRO	¿Utilizará Solicit@ una Aplicación externa para el registro de documentos?	0
[REGISTRO_LOCATION]	Url para registrar en la plataforma Aries.	http://ariesweb.cjap.junta-andalucia.es:8080/WSAriesRTL/services/RegistroTelematicoSOAP
IMPLEMENTA_REGISTRO	Ruta de la clase que implementa el registro telemático.	es.juntadeandalucia.registroAries.Aries

(Parámetros de integración con @Firma)

AUTENTICAR	Indica si se utiliza los servicios de atenticacion (@Firma – LDAP) o no. Con valor 0 la aplicación funciona en "modo demo".	1

Nota: Con valor 0 se puede utilizar la aplicación en "modo demo" o desarrollo. De esta manera se puede probar Solicit@ sin necesidad de acceso a la plataforma @firma. Para poder acceder a la Administración en este modo es necesario dar de alta un usuario de prueba (ver 5.3) con los siguientes datos:

Nombre: *Usuario*

Apellido1: *de Prueba*

Apellido2: *de Desarrollo*

DNI: *12345678X*

[SERVIDORFIRMA]	Nombre o IP del servidor de @firma	https://ws083.juntadeandalucia.es/afirmaws/services
[IDAPLICACIONFIRMA]	Identificador de la aplicación en @firma	solicita
[USUARIOFIRMA]	Usuario de acceso a @firma	membermull
[PASSWORDFIRMA]	Password de acceso a @firma	a4210
[TRUSTEDSTORE]	Path al keystore con certificado de servidor de @firma	c:\\trustkeystore
[TRUSTEDSTOREPASSWORD]	Password del keystore de @firma	changeit
FIRMAFICHEROS	URL donde redireccionar para instalar los componentes necesarios para autenticación.	https://ws022.juntadeandalucia.es/firmadigital/servicio/paginas/DescargasFicheros.jsp

Nota: El parámetro [TRUSTEDSTORE] indica la ruta del almacén de claves donde se almacena la clave pública del servidor de firma. Con esto, el JBoss reconoce el servidor de firma como un servidor confiable y permite la comunicación mediante HTTPS con @firma. Normalmente este fichero nos lo proporciona el servicio de soporte y administración de @Firma, aunque si fuera necesario, se puede crear de la manera según epígrafe 3.4.4.4. **Es necesario, para la integración con @Firma y el uso de la autenticación y firma eletrónica que este fichero exista.**

(Parámetros de integración con LDAP)

Solicit@ permite autenticar el acceso a Administración y Generador de Formularios mediante usuario y contraseña almacenados en un servidor LDAP, en lugar de obtener estos datos del certificado digital. Para ello se debe dar de alta un usuario (ver punto 5.3) y configurar los siguientes parámetros.

AUTENTICACION_LDAP	¿El acceso a la administración y generador de formularios se valida contra LDAP?	0
LDAP_INITIAL_CONTEXT_FACTORY	Contexto de conexión con el LDAP (No modificar)	com.sun.jndi.ldap.LdapCtxFactory
LDAP_URL	Url del servidor LDAP contra el que se validan los usuarios en Administración o Generador de Formularios.	ldap://ldap1.cap.junta-andalucia.es:389
LDAP_DN	DN de los usuarios de LDAP	o=empleados,o=juntadeandalucia,c=es
LDAP_ATRIBUTO	Atributo del esquema LDAP del objeto usuario a comparar con el nombre y apellidos del usuario en la base datos de Solicit@.	janombreapellidos
LDAP_ID	Atributo del esquema LDAP del objeto usuario que se utilizar como atributo de login.	uid=

(Parámetros de integración con servidor de correo electronico)

MAIL_PRESENTA	¿Se envia un correo a la dirección predeterminada siempre que se presente una solicitud-documento?	1
SERVIDOR_CORREO	Servidor de correo para informar de documentos presentados.	192.168.1.195
PUERTO_CORREO	Puerto del servidor de correo	25
CORREO_CORPORATIVO	Direccion remitente de los correos	solicita@dominio.es
MAIL_AUTHENTICATION_USER	Usuario para la autenticación del envio de correo electrónico.	usuario
MAIL_AUTHENTICATION_PASSWORD	Password para la autenticación del envio de correo electrónico.	password
TIENEPROXY	¿Se utiliza proxy para acceder al servidor de correo?	0
PROXY	Nombre o IP de Proxy para acceso servidor de correo.	NULL
PUERTOPROXY	Puerto del Proxy para acceso servidor de correo.	NULL

(Parámetros de integración con la plataforma Notific@)

USAR_NOTIFICA	¿Utilizará Solicit@ la aplicación externa Notific@?	1
MCSN	Ruta del fichero mscn.properties de configuración de Notific@.	C:\\mcsn.properties
NOTIFICA_KEYSTORE	Ruta al almacen de claves que contiene el certificado del servidor de notifica para el caso que notifica este corriendo como servidor seguro HTTPS	C:\\notificakeystore
NOTIFICA_KEYSTOREPASS	Password de acceso al almacen de claves de notifica	changeit

Nota: El servicio de Notific@ utiliza el parámetro mcsn.properties que se ofrece en el CD de solicita. En el se configura el acceso a este servicio. Un ejemplo comentado de este fichero sería:

```
# Características de la conexión con el servidor del notario.
protocolo=https
direccion_ip=ws031.juntadeandalucia.es
puerto=443
#puerto=80
path_acceso=jboss-net/services/ServicioWEBSN

# ------------------
# Conexión con proxy
```

```
# -----------------
# Con conexionproxy a true indica que el acceso es via proxy.
conexionproxy = false

# Nombre de servidor proxy o dirección IP:
proxyhost = 10.111.0.11

# Puerto del servidor proxy:
proxyport = 8080

# login conexión al proxy (vacio=>sin autenticación):
proxylogin =

# password conexión con proxy:
proxypassword =

# ------------------------------
# Configuración del Log de trazas
# ------------------------------
# Fichero de configuración log para la salida con el logger de Log4j
#xml_log=C:/mcneLog4jConfig.xml


# ----------------------------------------
# PKCS#12 para firma de las solicitudes SOAP
# ----------------------------------------
# Fichero PKCS#12 que contiene la pareja de claves
#pkcs12.archivo=c:/cacerts
#pkcs12.archivo=c:/notifica.p12
##Esta clave se obtiene cuando la entidad se da de alta en la aplicación ##Notific@, puede
incluirse en el keystore del Sistema.

pkcs12.archivo=c:/cjap.p12

# Contraseña del PKCS#12 que protege la clave privada
pkcs12.pass = 12345678
#pkcs12.pass = changeit
```

(Parámetros de integración con un tramitador)

USA_TRAMITADOR	Parámetro que indica si se integra solicit@ con un tramitador o no. (1=SI, 0=NO)	0
URL_TRAMITADOR	URL de conexion con un tramitador	http://localhost:8080/conexa/WebServices/ConexaWS
IMPLEMENTA_TRAMITADOR	Clase que implementa la interfaz de integracion con un tramitador	es.everis.solicita.tramitador.ConexionConexa

Si se desea integrar solicit@ con una pasarela para volcar solicitudes desde solicit@ a un tramitador es imprescindible que USA_TRAMITADOR este configurado a 1 y que la URL_TRAMITADOR apunte correctamente a los webservices del tramitador. Además, IMPLEMENTA_TRAMITADOR debe ser el paqueta a la clase que implementa la interfaz de tramitación, por defecto esta implementación es la que permite volcar solicitudes desde solicit@ a la plataforma de tramitación wand@.

(Nombre del .ear y version de solicit@)

NOMBRE_EMPAQUETADO	Nombre del fichero .ear	solicita/
SOLICITA_VERSION	Parámetro que indica la versión de solicit@	4.0.0

Nota: El nombre del empaquetado es necesario para versiones de solicit@ que incluyan un .ear, en otro caso este parámetro contendrá el valor **null**. Si el .ear se llama solicita400.ear, este parámetro tiene que establecerse a solicita400/

(URL de más información)

URL_ADOBE_READER	URL de descarga de Acrobat Reader para visualizacion de documentos PDF.	http://get.adobe.com/es/reader/
URL_CERES	URL de CERES para obtención y ayuda del certificado	http://www.cert.fnmt.es/index.ph

	digital	p?cha=cit&sec=obtain_cert

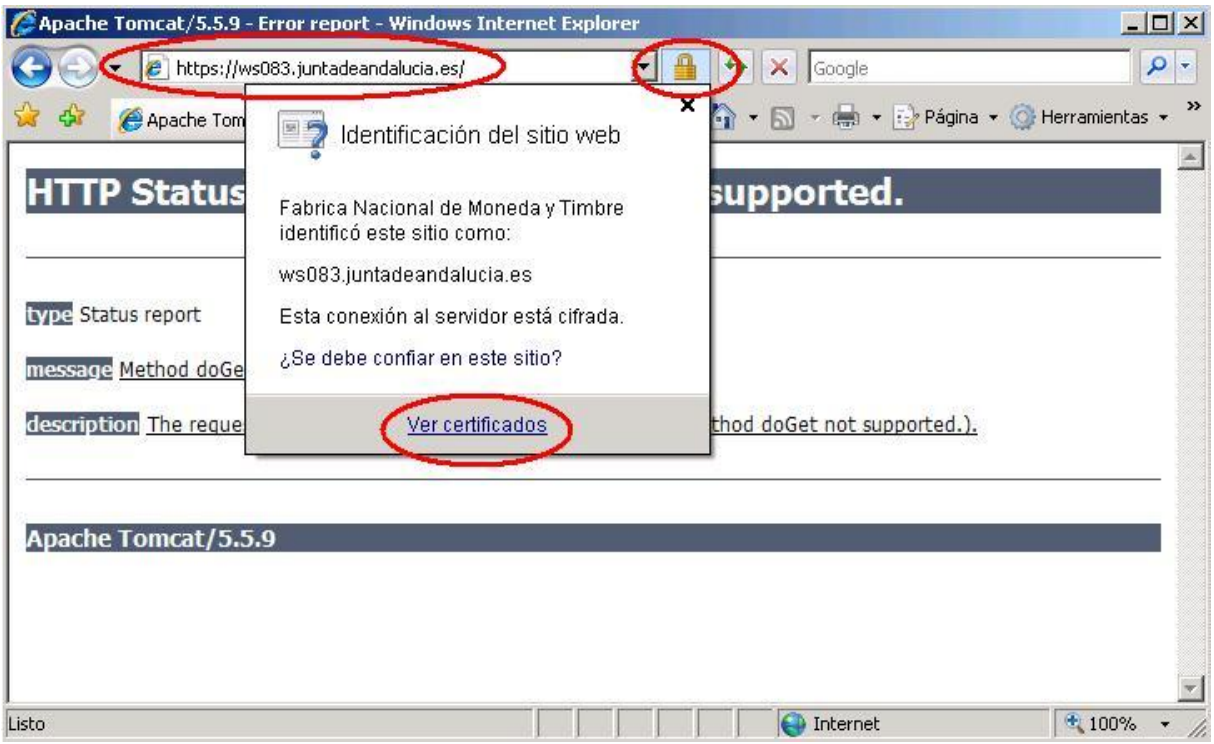
(Parámetros de properties)

Nombre parámetro	Descripción parámetro	Valor parámetro
OFIV_TITULO_ORGANIZACION	Titulo de la organización (cabecera de las paginas)	Consejería de Justicia y Administración Pública
OFIV_TEXTO_CONSEJERIA_INDEX	Nombre de la organización	de la Consejería de Justicia y Administración Pública
TAMANO_MAXIMO_FICHEROS_SUBIDOS_MB	Tamaño máximo permitido a los ficheros subidos al servidor.	2

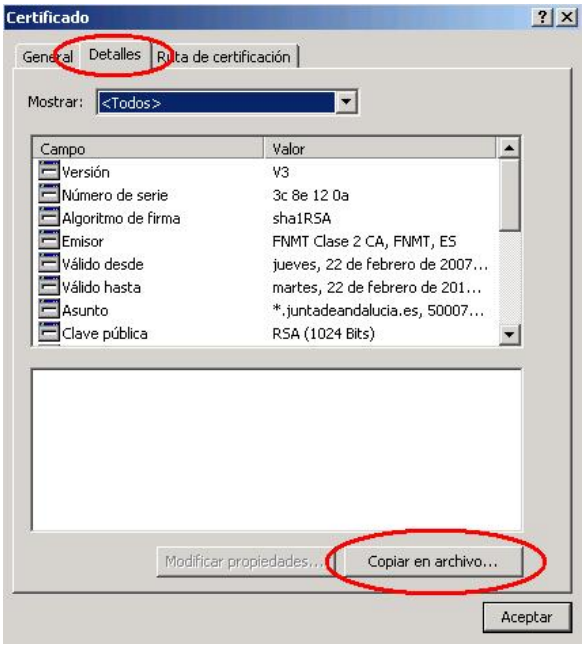
Nota: En esta tabla se almacenan los textos configurables que aparecen en la oficinaVirtual. Modificar solo los campos "Valor parámetro"

3.4.4.4 Configuración del Almacén de Certificados.

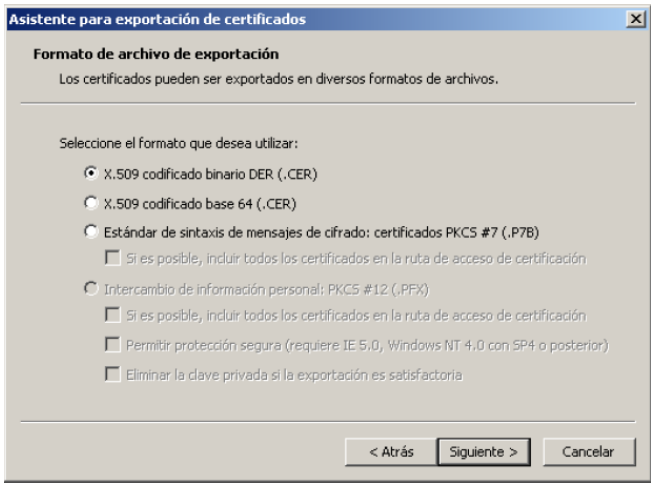
Para poder establecer la comunicación SSL entre nuestro servidor de aplicaciones y el servidor de custodia de @firma debemos cargar la clave publica del mismo dentro del almacén de certificados de nuestro servidor de aplicaciones. Para ello, la forma más sencilla es conectarnos con un navegador al servidor de @firma donde se publican los webservices ejemplo:
Se pulsa sobre el icono del candado, y a continuación en "Ver certificado".



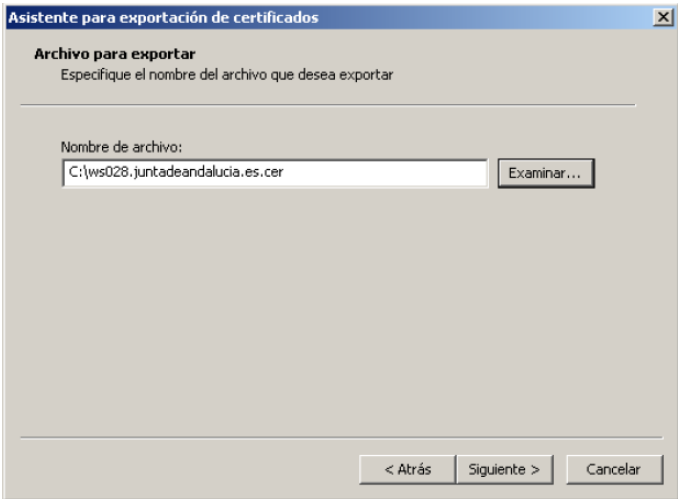
Se pulsa en la pestaña de "Detalles" y a continuación en "Copiar en archivo..."



Se mostrará un asistente mediante el cual podemos exportar a un archivo el certificado del servidor. Después de una pantalla de bienvenida, nos preguntará por el formato de exportación.



Escogemos el primero de los formatos, y nos pedirá a continuación también la ruta donde queremos guardarlo.



Tras esto ya tenemos el certificado del servidor, que podremos importar mediante la herramienta "keytool" de java sobre el almacén por defecto, "cacerts" (la clave de este almacén es "changeit") del runtime de java, que por lo general se encuentra en el directorio:
%JAVA_HOME%/jre/lib/security
Para incluirlo deberemos importarlo mediante el siguiente comando, que siguiendo con el ejemplo sería:

```
Keytool –import –keystore %JAVA_HOME%/jre/lib/security/cacerts –file C:\ws028.juntadeandalucia.es.cer –alias custodia
```

3.4.4.5 Configuración del contexto de la aplicación.

Solicit@ esta compuesta de tres modulos:

- oficinaVirtual
- administración
- solicita20

Por defecto tienen configurado el contexto para que se desplieguen como:

protocolo://servidor:puerto/oficinaVirtual

protocolo://servidor:puerto/administracion

protocolo://servidor:puerto/solicita20

Estos contextos son configurables, para ello hay que editar el fichero application.xml que se encuentra en solicita.ear/ META-INF y modificar las lineas que se resaltan en negrita:

```
<module>
 <web>
  <web-uri>oficinaVirtual.war</web-uri>
  <context-root>/oficinaVirtual</context-root>
 </web>
</module>
<module>
 <web>
  <web-uri>administracion.war</web-uri>
  <context-root>/administracion</context-root>
 </web>
</module>
<module>
 <web>
  <web-uri>solicita20.war</web-uri>
  <context-root>/solicita20</context-root>
 </web>
</module>
```

Planificación de tareas.

N/A

4 MARCHA ATRÁS DE LA INSTALACIÓN Y CONFIGURACIÓN

4.1 Marcha atrás de la entrega

N/A

4.2 Marcha atrás del software base

Para desinstalar o dar marcha atrás la configuración de ***Solicit@***.

- Eliminar todo el software instalado.
- Eliminar las posibles variables de entorno creadas.
- Eliminar la instancia de Base de Datos generada.

5 ANEXOS

5.1 Anexo 1. Instalación Apache

La instalación de Apache difiere según el Sistema Operativo en cuestión. No se encuentra dentro del objetivo de este manual explicar la instalación básica de este software. Los cambios a implementar en la configuración de Apache una vez instalado con respecto a la aplicación SOLICITA serían:

5.1.1 Configuración Apache con soporte SSL

Editar el fichero de configuración de apache httpd.conf

Descomentar las siguientes líneas:

LoadModule ssl_module modules/mod_ssl.so

Include conf/ssl.conf

Debemos comprobar que los archivos mod_ssl.so y ssl.conf existen y se encuentran ubicados en el lugar descrito anteriormente. Un ejemplo del fichero ssl.conf puede verse al final de este anexo.

Editar el fichero de configuración del apache ssl.conf. Especificar el lugar donde ubicaremos certificado del servidor:

SSLCertificateFile C:/Archivos de programa/Apache/Apache2.2/conf/server.crt

Especificar el lugar donde ubicaremos la clave privada del servidor:

SSLCertificateKeyFile C:/Archivos de programa/Apache/Apache2.2/conf/server.key

El procedimiento para obtener estos 2 archivos es el siguiente:

Abrimos una consola de comandos y nos ubicamos en el directorio donde se encuentra el binario de openssl:

cd C:\Archivos de programa\Apache\Apache2.2\conf

Generamos la clave privada (Server.key):

openssl genrsa -out server.key 1024

Generamos una solicitud de certificado autofirmado (server.csr)

openssl req -new -key server.key -out server.csr -config ./openssl.cnf

Country Name (2 letter code) [ES]:ES

State or Province Name (full name) [Sevilla]:Sevilla

Locality Name (eg, city) [Newbury]:Sevilla

Organization Name (eg, company) [My Company Ltd]:Agencia Andaluza del Agua

Organizational Unit Name (eg, section) [ ]:Information Technology

Common Name (eg, your name or your server's hostname) [ ]:servidor.juntadeandalucia.es

Email Address [ ]:usuario@juntadeandalucia.es

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password [ ]:

An optional company name [ ]:

Este archivo debe enviarse a una entidad certificadora (como la FNMT) para que la firme y nos devuelva nuestro certificado. Si no se tiene en el momento de la instalación este certificado devuelto por la entidad certificadora podemos crear un certificado autofirmado para utilizarlo de manera temporal, para ello:

openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

Copiar el server.crt y server.key en el directorio según el fichero de configuración de apache ssl.conf.

Reiniciar el apache y comprobar funcione con soporte SSL accediendo a https://hostname.domainame

5.1.2 Configuración Apache con el módulo mod_jk para su conexión con JBOSS

5.1.2.1 Para instalaciones Windows

Copiar el fichero mod_jk-apache-2.2.4.so como mod_jk.so en el directorio modules de apache y agregar en el fichero de configuración de apache httpd.conf la línea:

LoadModule jk_module modules/mod_jk.so

5.1.2.2 Para instalaciones Linux

Copiar el fichero tomcat-connectors-1.2.23-src.tar.gz en un directorio de trabajo.

Descomprimirlo mediante el comando:

tar –xvzf tomcat-connectors-1.2.23-src.tar.gz

Entrar al directorio tomcat-connectors-1.2.23-src:

cd tomcat-connectors-1.2.23-src/native

Compilar:

./configure --with-apxs=/usr/sbin/apxs (o donde se encuentre apxs/apxs2)

make

make install

Verificar se haya agregado en el fichero de configuración de apache la línea:

LoadModule jk_module modules/mod_jk.so

5.1.2.3 En ambos sistemas operativos.

Crear el fichero de configuración de apache conf/workers.properties con las líneas dentro (sólo en caso de no configurar Solicit@ en alta disponibilidad):

ps=/

worker.list=nothost

worker.nothost.host=127.0.0.1

worker.nothost.port=8009

worker.nothost.type=ajp13

worker.nothost.lbfactor=100

Agregar en el fichero de configuración de apache httpd.conf las líneas cambiando las IPs desde las cuales se desea permitir el acceso:

```
Include conf/mod-jk.conf
JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogLevel info

JkMount /manager/html* nothost
JkMount /solicita20* nothost
JkMount /oficinaVirtual* nothost
JkMount /administracion* nothost

<Location "/administracion*/*">
        order deny,allow
        deny from all
        allow from 10.244.2.0/24
                allow from 10.244.44.191/32
</Location>

<Location "/solicita20/jsp/solicita20/formularios/*">
        allow from all
</Location>

<Location "/solicita20/jsp/solicita20/*">
        order deny,allow
        deny from all
        allow from 10.244.2.0/24
                allow from 10.244.44.191/32
</Location>

<Location "/solicita20/servlets/generarpdf*">
        allow from all
</Location>
```

Reiniciar el apache

5.1.3 Ejemplo del Fichero ssl.conf

```
#
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailing information about these
# directives see <URL:http://httpd.apache.org/docs-2.0/mod/mod_ssl.html>
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
```

```
#
# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the SSL library.
# The seed data should be of good random quality.
# WARNING! On some platforms /dev/random blocks if not enough entropy
# is available. This means you then cannot use the /dev/random device
# because it would lead to very long connection times (as long as
# it requires to make more entropy available). But usually those
# platforms additionally provide a /dev/urandom device which doesn't
# block. So, if available, use this one instead. Read the mod_ssl User
# Manual for more details.
#
# Note: This must come before the <IfDefine SSL> container to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512

#<IfDefine SSL>

#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
# Note: Configurations that use IPv6 but not IPv4-mapped addresses need two
#       Listen directives: "Listen [::]:443" and "Listen 0.0.0.0:443"
#
Listen 443

##
##  SSL Global Context
##
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts.
##

#
#   Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

#   Pass Phrase Dialog:
#   Configure the pass phrase gathering process.
#   The filtering dialog program (`builtin' is a internal
#   terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog  builtin

#   Inter-Process Session Cache:
#   Configure the SSL Session Cache: First the mechanism
#   to use and second the expiring timeout (in seconds).
#SSLSessionCache         none
#SSLSessionCache         shmht:/opt/apache2ssl/logs/ssl_scache(512000)
#SSLSessionCache         shmcb:/opt/apache2ssl/logs/ssl_scache(512000)
SSLSessionCache         dbm:/opt/apache2ssl/logs/ssl_scache
SSLSessionCacheTimeout  300

#   Semaphore:
#   Configure the path to the mutual exclusion semaphore the
#   SSL engine uses internally for inter-process synchronization.
SSLMutex  file:/opt/apache2ssl/logs/ssl_mutex

##
## SSL Virtual Host Context
##
```

```
<VirtualHost _default_:443>

#   General setup for the virtual host
DocumentRoot "/opt/apache2ssl/htdocs"
ServerName www.example.com:443
ServerAdmin you@example.com
ErrorLog /opt/apache2ssl/logs/error_log
TransferLog /opt/apache2ssl/logs/access_log

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
SSLCertificateFile /opt/apache2ssl/conf/ssl.crt/wildcardJuntadeandaluciaes.cer
#SSLCertificateFile /opt/apache2ssl/conf/ssl.crt/ws074.cer
#SSLCertificateFile /opt/apache2ssl/conf/ssl.crt/server.crt
#SSLCertificateFile /opt/apache2ssl/conf/ssl.crt/server-dsa.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /opt/apache2ssl/conf/ssl.key/wildcardJuntadeandaluciaesApache.key
#SSLCertificateKeyFile /opt/apache2ssl/conf/ssl.key/ws074.key
#SSLCertificateKeyFile /opt/apache2ssl/conf/ssl.key/server.key
#SSLCertificateKeyFile /opt/apache2ssl/conf/ssl.key/server-dsa.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /opt/apache2ssl/conf/ssl.crt/ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /opt/apache2ssl/conf/ssl.crt
#SSLCACertificateFile /opt/apache2ssl/conf/ssl.crt/ca-bundle.crt

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /opt/apache2ssl/conf/ssl.crl
#SSLCARevocationFile /opt/apache2ssl/conf/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
```

```
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#           and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#           and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#           and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#           and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#          or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o CompatEnvVars:
#     This exports obsolete environment variables for backward compatibility
#     to Apache-SSL 1.x, mod_ssl 2.0.x, Sioux 1.0 and Stronghold 2.x. Use this
#     to provide compatibility to existing CGI scripts.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +CompatEnvVars +StrictRequire
<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>
<Directory "/opt/apache2ssl/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
```

```
#    SSL close notify alert is send and mod_ssl waits for the close notify
#    alert of the client. This is 100% SSL/TLS standard compliant, but in
#    practice often causes hanging connections with brain-dead browsers. Use
#    this only for browsers where you know that their SSL implementation
#    works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.
SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

#   Per-Server Logging:
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
CustomLog /opt/apache2ssl/logs/ssl_request_log \
         "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

# Modulo rewrite
<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteLog /opt/apache2ssl/logs/rewrite.log
   RewriteLogLevel 3

   RewriteRule ^/$      https://ws079.juntadeandalucia.es/oficinaVirtual/ [L]
</IfModule>


</VirtualHost>

#</IfDefine>
```

5.2 Anexo 2. Compatibilidad IE con ssl.

Se ha detectado un problema de desconexiones aleatorias cuando el cliente es IE (cualquier versión, incluida la 7). Éste problema estaba causado por la mala implementación de SSL en el IE, y se arregla añadiendo en el httpd.conf del apache, la siguiente condición:

```
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-response-1.0
```

5.3 Anexo 3. Creación de usuarios y asignación de perfiles.

La creación de un usuario para acceder al módulo de administración únicamente se tendrá que realizar para la creación del primer usuario. El resto de usuario que vayan a acceder a la aplicación de administración podrán ser dados de alta desde la opción de creación de usuarios existente para un usuario con perfil “Administrador de la configuración”. Para dar de alta a un nuevo usuario, tenemos que seguir una serie de pasos:

1. Se tendrá que insertar un registro en la tabla “PLATD_USUARIOSADMTREW” de la base de datos y añadir en los campos correspondientes la información correspondiente:

ID_SAWA	Identificador que representa al usuario en la administración.
ID_TREWA	Cadena que representa al usuario, similar a login.
DS_NOMBRE	Nombre del usuario tal y como aparece en el certificado digital.
TX_APELLIDO1	Primer apellido del usuario tal y como aparece en el certificado digital.

TX_APELLIDO2	Segundo apellido del usuario tal y como aparece en el certificado digital.
TX_NIF	Número y letra del DNI tal y como aparece en el certificado digital.

Para realizar esta inserción se podría ejecutar el siguiente script:

```
INSERT INTO PLATD_USUARIOSADMTREW ( ID_SAWA, ID_TREWA, DS_NOMBRE, TX_APELLIDO1, TX_APELLIDO2,
TX_NIF, ID_REL_SAWA, CREADOR, F_CREACION, MODIFICADOR, F_MODIFICACION )
VALUES
( 1, 'xxxxx', 'Usuario', 'De Prueba', 'De Desarrollo', '12345678X', 1, NULL, NULL, NULL, NULL);
COMMIT;
```

2. Para relacionar el usuario con el perfil correspondiente, se debe insertar en la "PLATR_USUARIOPERFIL":

```
INSERT INTO PLATR_USUARIOPERFIL ( ID_PERFIL, ID_USUARIO ) VALUES ( 1, 1);
COMMIT;
```

Realizando estos dos pasos el usuario que se haya dado de alta podrá acceder con administración a la administración. Una en la administración el usuario podrá dar de alta a cualquier otro usuario.

5.4 Anexo 4. Modificación de puertos del jboss.

En algunas ocasiones, el JBoss que sirve Solicit@ debe coexistir en el mismo servidor con otro JBoss o tomcat que ya estará utilizando los puertos que se requieren para funcionar. Es necesario, por lo tanto cambiarlos y tener en cuenta cuales son los nuevos.

En la siguiente tabla especificamos cuales son los puertos a modificar y en que fichero de configuración podemos hacer esto.

- %JBOSS_HOME%\server\all\deploy\jbossweb-tomcat50.sar**server.xml**

  Change HTTP/1.1 Connector port from 8080 to 8888

  Change AJP 1.3 Connector port from 8009 to 8099

  Change SSL/TLS Connector port from 8443 to 8493

- %JBOSS_HOME%\server\all\conf**jboss-service.xml**

  Change WebService port from 8083 to 8899

  Change NamingService Port from 1099 to 9999

  Change RMIport from 1098 to 9998

  Change RMIObjectPort from 4444 to 9444

  Change PooledInvoker ServerBindPort from 4445 to 9445

- %JBOSS_HOME%\server\all\deploy**cluster-service.xml**

  Change ha.jndi.HANamingService port from 1100 to 1190 and correspondingly in...

- %JBOSS_HOME%\server\all\deploy\jms**hajndi-jms-ds.xml**

  Change java.naming.provider.url from 1100 to 1190

- %JBOSS_HOME%\server\all\conf**jacorb.properties**

  Change OAPort from 3528 to 9528

  Change OASSLPort from 3529 to 9529

Una vez hecho esto, debemos modificar la aplicación, ya que ésta utiliza recursos del servidor de aplicaciones como el jndi. Dentro de los siguientes EJBs:

- GFAPI.jar
- solicitaBusiness.ejb3
- solicitaData.jar

debemos editar el fichero **jndi.properties**, para que el puerto del proveedor de nombres sea el que hemos configurado en el fichero $JBOSS_HOME/server/all/conf/ jboss-service.xml:

```
java.naming.provider.url=localhost:1099
```

6 GLOSARIO

Término	Descripción
BBDD	Base de Datos
JBoss	http://es.wikipedia.org/wiki/JBoss

7 BIBLIOGRAFÍA Y REFERENCIAS

Título	Descripción
SOL001E_MUS_Manual_Usuario_Solicit@v4.0.0._ManualDeIntegrador_v04r00.pdf	Manual dirigido a desarrolladores para integrar sus aplicaciones con solicit@.
SOL001E_MUS_Manual_Usuario_Solicit@v4.0.0._Administracion_v04r00.pdf	Descripción de la administración de la aplicación.
SOL001E_MUS_Manual_Usuario_Solicit@v4.0.0. GeneradorFormularios v04r00.pdf	Descripción del uso de la herramienta web de diseño de formularios.
SOL001E_MUS_Manual_Usuario_Solicit@v4.0.0._OficinaVirtual_v04r00.pdf	Descripción de la oficina virtual de solicit@.
SOL001E_MUS_Manual_Usuario_Solicit@v4.0.0._Ciclo_Comprobación_Instalacción_v04r00.pdf	Manual que muestra como realizar un ciclo completo de verificación del correcto funcionamiento de la aplicación tras su instalación.