





Plataforma de Tramitación w@ndA

Manual de Instalación y Configuración

Versión: v01r06

Fecha: 09/12/2010

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de la Junta de Andalucía.

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</small>	Consejería de Hacienda y Administración Pública D.G. de Tecnologías para Hacienda y la Administración Electrónica	Plataforma de Tramitación w@ndA Manual de Instalación y Configuración	
--	--	--	---

HOJA DE CONTROL

Título	Plataforma de Tramitación w@ndA		
Entregable	Manual de Instalación y Configuración		
Nombre del Fichero	PTW200_MIC_Manual_Instalación_PTw@ndA_v01r06.doc		
Autor	everis		
Versión/Edición	v01r06	Fecha Versión	09/12/2010
Aprobado por		Fecha Aprobación	
		Nº Total Páginas	37

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio
v01r00	Nuevo entregable del producto	everis	-	24/07/2009
v01r01	Actualización del documento	everis	-	05/11/2009
v01r02	Actualización del documento	everis	-	05/02/2010
v01r03	Actualización del documento	everis	-	07/04/2010
v01r04	Actualización del documento	everis	-	07/06/2010
v01r05	Actualización del documento	everis	-	04/11/2010
v01r06	Actualización del documento	everis	-	09/12/2010

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Área	Nº Copias
Manuel Perera Domínguez	Jefe de Servicio	Servicio de Coordinación de Administración Electrónica	1
Francisco González Guillén	Dirección del Proyecto	Servicio de Coordinación de Administración Electrónica	1
Francisco Mesa Villalba	Gte. Administración Electrónica	Servicio de Coordinación de Administración Electrónica	1
Rafael Cabello Ruiz-Barruecos	Gerente	everis	1
Daniel Martínez Sevilla	Jefe de Proyecto	everis	1

ÍNDICE

1.	REQUISITOS PREVIOS DE INSTALACIÓN	6
1.1	Tabla de compatibilidades	7
1.2	Notas de versión	7
2	ESTRUCTURA DE LA ENTREGA	9
3	PROCEDIMIENTO DE PRIMERA INSTALACIÓN	11
3.1	Instalación de componentes w@ndA complementarios	11
3.1.1	Instalación de Trew@ 1.5	11
3.1.2	Instalación de @Firma 5	11
3.1.3	Instalación de Port@firmas	11
3.1.4	Instalación de Avis@dor	11
3.1.5	Instalación de Solicit@ 5.0.0	12
3.2	Instalación de otros Componentes	12
3.2.1	Instalación y configuración de OpenOffice en modo servidor	12
3.3	Instalación de la Plataforma de Tramitación w@ndA	13
3.3.1	Configuración de los Componentes w@ndA	13
3.3.2	Configuración del editor de párrafos Trew@	13
3.3.3	Despliegue en el Servidor de aplicaciones Web	13
3.3.3.1	Instalación de la aplicación	13
3.3.3.2	Librerías comunes	14
3.3.3.3	Almacén de certificados	14
3.3.3.4	Ampliación de memoria de la Máquina Virtual Java en JBOSS	16
3.3.3.5	Configuración de la conexión a Internet del servidor	17
3.3.4	Configuración de la Plataforma de Tramitación w@ndA	17
3.3.4.1	Configuración de conexiones con Base de datos	17
3.3.4.2	Primer inicio de PTw@ndA	20
3.3.4.3	Ejecución de scripts de BBDD	21
3.3.4.4	Configuración de la Aplicación	21
3.3.4.5	Configuración del motor de indexación y búsqueda	26
3.3.4.5.1	Alta Disponibilidad	30
3.3.4.5.1.1	Introducción	30
3.3.4.5.1.2	Problemas a resolver	31
3.3.4.5.1.3	Solución persistencia de sesiones de usuario	31
3.3.4.5.1.4	Solución ejemplo basada en Apache	31

3.3.4.5.1.4.1	Definición de worker.properties.....	32
3.3.4.5.1.4.2	Inclusión de worker.properties en la configuración de APACHE.....	33
3.3.4.5.1.4.3	Configuración de los nodos JBoss.....	33
3.3.4.6	Configuración de trazas mediante Log4j	34
3.4	Instalación de Módulos Funcionales. Cambio de Versión.....	34
3.5	Definición de Menús.....	35
4	REFERENCIAS.....	36
5	ENLACES	37

1. REQUISITOS PREVIOS DE INSTALACIÓN

Para poder instalar y asegurar un buen funcionamiento de la Plataforma de Tramitación w@ndA, se deben cumplir los siguientes **requisitos**:



- Máquina virtual Java compatible con la especificación de Sun Microsystems JSR-176 (<http://jcp.org/en/jsr/detail?id=176>) para J2SE 6.0.
- URL de descarga:
http://www.java.com/es/download/windows_xpi.jsp?locale=es&host=www.java.com:80
- Servidor de Aplicaciones JBoss 5.1GA como contenedor de JSP y Servlet compatible con las especificaciones de Sun Microsystems JSR-154 (<http://www.icp.org/en/jsr/detail?id=154>) y JSR-142 (<http://www.icp.org/en/jsr/detail?id=152>).
URL de descarga (enlace [jboss-5.1.0.GA-jdk6.zip](#)):
<http://sourceforge.net/projects/jboss/files/JBoss/JBoss-5.1.0.GA>
- Acceso a base de datos con esquema compatible Trew@ 1.5
- Base de datos soportada por Hibernate 3.0, como, por ejemplo, Oracle (versiones 8i, 9i, 10g) y MySQL (versiones 3.23, 4.0, 4.1, 5.0), etc. para el modelo de datos propio de la Plataforma de Tramitación, el cual se dividirá en dos esquemas: uno para las tablas paramétricas y otro para las configurables (la lista completa de compatibilidades de Hibernate con diferentes SGDB puede consultarse en la página <http://www.hibernate.org/80.html>).
- El servidor donde se despliegue la Plataforma de Tramitación debe tener habilitada conexión a Internet –directa o mediante proxy-, ya que Struts requiere que el servidor pueda descargarse los documentos DTD de definición sintáctica de sus ficheros XML de configuración.

Requisitos opcionales en función de los componentes w@ndA que quieran integrarse en la Plataforma de Tramitación:

- Acceso a @visor.
- Acceso a Port@firmas
- Acceso a @firma 5, para autenticación de usuarios
- Acceso a Solicit@
- Acceso a Notific@
- Acceso a Ward@ o un servicio emulador equivalente (Alfresco 2.1)

Adicionalmente a los componentes w@ndA puede opcionalmente requerirse el acceso a Formul@ para integrarse con el motor de formularios para mostrar tareas de manipulación de datos en caso de un desarrollo vertical que haga uso de esta posibilidad.

En cuanto a los requisitos referentes a los clientes de la Plataforma de Tramitación, hay que cumplir los definidos por:

- La propia plataforma **@firma** en su versión v5.0 (para más información, se puede consultar el Manual de Usuario de @firma en la web Plutón de la Consejería de Justicia y Administración Pública de la Junta de Andalucía, <https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/afirma.jsp>).
- El generador de escritos **WebOffice 3.x** o superior (para más información, consúltese la documentación de este componente que se distribuye en los CDs oficiales de Trew@).



A partir de esta versión no es obligatorio el uso de WebOffice, pudiéndose utilizar en su lugar el editor de párrafos HTML proporcionado por Trew@.

1.1 Tabla de compatibilidades

A continuación se detalla la relación de productos con los que la Plataforma de Tramitación w@ndA ha sido probada y, por tanto, se garantiza su correcto funcionamiento.

Producto	Tipo	Versiones
Sun J2SE Runtime Environment	Máquina virtual Java	JSDK 6.0
JBoss	Servidor de Aplicaciones	5.1.x
Oracle	SGDBR	9i, 10G
MySQL	SGDBR	5.0

1.2 Notas de versión

En esta versión 2.0 de PTw@ndA se han realizado las siguientes modificaciones, que pueden afectar a la actualización que se realice proveniente de la versión anterior (1.0.3):

- Integración con el motor de tramitación Trew@ en su versión 1.5.0, asegurándose compatibilidad con versiones anteriores (1.4.X). Esto incluye la nueva versión de WebOffice3.
- Integración con el editor de documentos HTML Trew@ basado en párrafos.
- Integración con el motor de formularios Formul@ para la explotación de formularios de manipulación de datos.
- Adaptación a las directrices expuestas por MADEJA (Marco de Desarrollo de la Junta de Andalucía) para el desarrollo de Sistemas de Información. En este punto, se han modificado Servicios base de PTw@ndA, siendo

necesario adaptar aquellos desarrollos de módulos específicos realizados sobre PTw@ndA 1.0.3 por las entidades que han adoptado el producto:

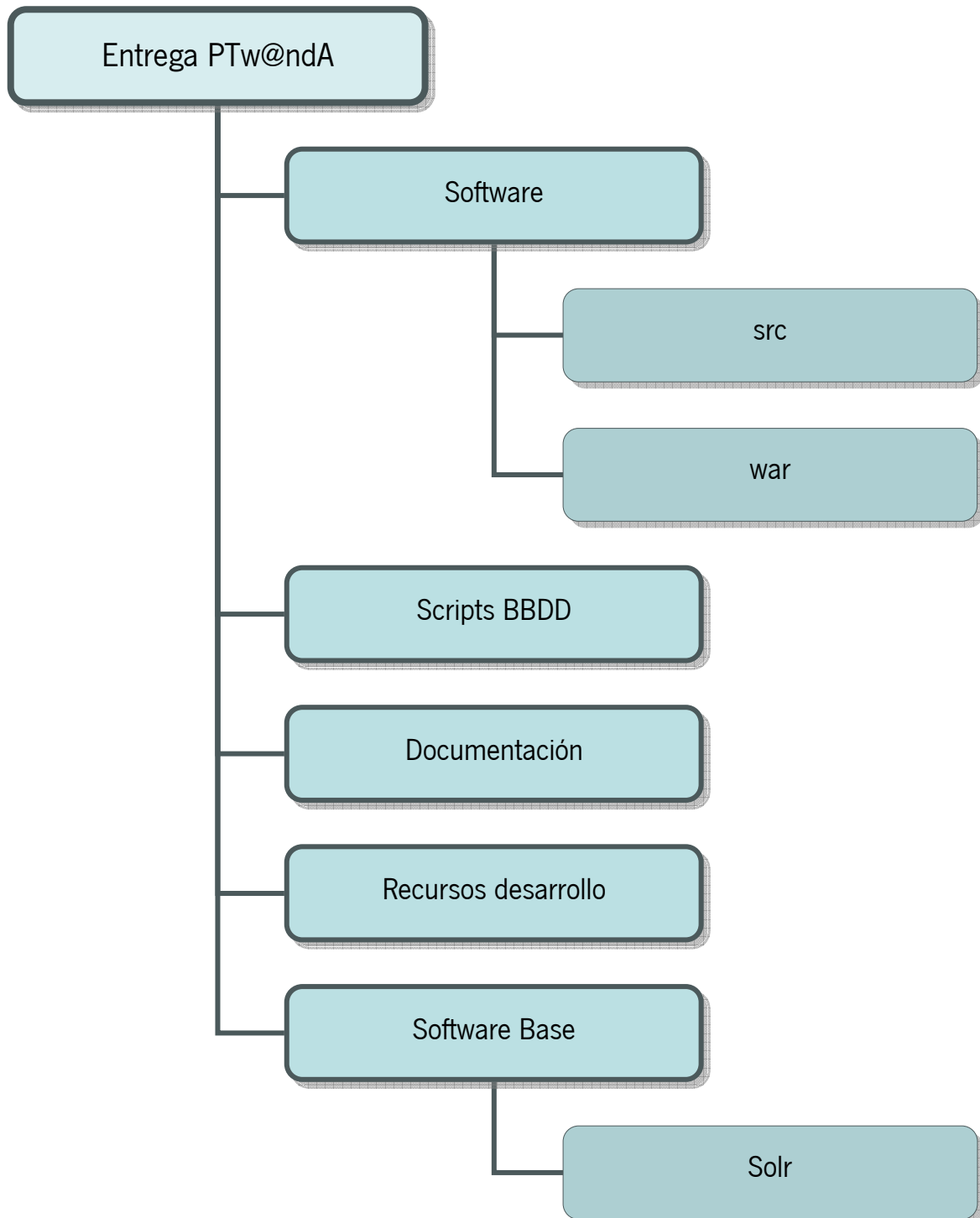
- Los servicios desarrollados en los módulos deben heredar del Servicio `PTWandaServiceImpl`, indicando expresamente en sus componentes de la capa de negocio o Servicios inyectados mediante Spring, la extensión del Servicio base citado.
- Adicionalmente, los desarrollos previos de módulos funcionales deben adaptarse a la última versión del componente `struts-core` (2.0.14).
- Inclusión de un nuevo esquema de Base de Datos asociado a los **datos** de la aplicación: Configuración, noticias, festivos, notas,... Esto implica que la aplicación dispondrá de dos esquemas diferenciados:
 - Plataforma: Información de datos paramétricos.
 - PlataformaDatos: Información referente a los datos gestionados por la aplicación.

















Para actualizaciones de versión, será necesario migrar el contenido de la tabla `CONFIG` del esquema Plataforma a PlataformaDatos.

- Ampliación de la funcionalidad ya existente, desarrollándose evolutivos solicitados por distintas áreas usuarias: Gestión de Notas, Gestión de Noticias, Tramitación en Bloque, Gestión de Festivos, ...

2 ESTRUCTURA DE LA ENTREGA

A continuación se expone la estructura de la entrega de la versión de PTw@ndA distribuida en el CD oficial y el contenido de cada uno de los apartados:



Directorio	Subdirectorio	Contenido
 <Entrega>		-
	 01. Software	-
	 01. Software/src	Código fuente de la entrega de PTw@ndA
	 01. Software/war	Código ejecutable de la entrega de PTw@ndA, listo para desplegarse.
	 01. Software/lib	Librerías necesarias para la aplicación que debe ubicarse en el lib del servidor.
	 02. Scripts BBDD	Scripts de Base de Datos para crear los esquemas y datos paramétricos.
	 03. Documentación	 Documento Detallado de Requisitos  Manual de Instalación y Configuración  Manual de Administración  Manual de Usuario  Manual del Desarrollador  Guía de Estilos
	 04. Recursos desarrollo	Recursos de desarrollo de módulos facilitados.
	 05. Software Base	-
	 05. Software Base/Solr	-

3 PROCEDIMIENTO DE PRIMERA INSTALACIÓN

A continuación se procede a describir la instalación de la Plataforma de Tramitación en un entorno donde no haya sido instalada previamente.



Indicar que el procedimiento explicado a continuación es para la versión 2.0.X de la Plataforma de Tramitación. No existe una migración desde las versiones 1.X.

En el caso de haber estado utilizando una versión 1.X, se requiere una nueva instalación y la adaptación de los módulos verticales desarrollados. En el manual del desarrollador se encuentra toda la información necesaria para realizar estos cambios.

3.1 Instalación de componentes w@ndA complementarios

3.1.1 Instalación de Trew@ 1.5

Para proceder a la instalación de Trew@ 1.5 se remite al documento de instalación del mismo (ver [1] en apartado [5 Referencias](#)) distribuido con dicho componente de w@ndA. Se hace hincapié en que este es el único componente obligatorio para poner en marcha la Plataforma de Tramitación.

3.1.2 Instalación de @Firma 5

Para proceder a la instalación de @Firma 5 se remite al documento de instalación (ver [2] en apartado [5 Referencias](#)) distribuido con dicho componente de W@ndA. Hay que recordar que este componente es de uso opcional en la Plataforma de Tramitación.



3.1.3 Instalación de Port@firmas

Para proceder a la instalación de Port@firmas se remite al documento de instalación (ver [3] en apartado [5 Referencias](#)) distribuido con dicho componente de w@ndA. Hay que recordar que este componente es de uso opcional en la Plataforma de Tramitación.

La integración de la Plataforma con este componente se configura a través de la herramienta de Administración de Trew@, en la sección de administración de componentes. Para más información se puede consultar el manual de usuario de dicha herramienta.

3.1.4 Instalación de Avis@dor

Para proceder a la instalación de Avis@dor se remite al documento de instalación (ver [4] en apartado [5 Referencias](#)) distribuido con dicho componente de w@ndA. Hay que recordar que este componente es de uso opcional en la Plataforma de Tramitación.

	<p>Consejería de Hacienda y Administración Pública</p> <p>D.G. de Tecnologías para Hacienda y la Administración Electrónica</p>	<p>Plataforma de Tramitación w@ndA</p> <p>Manual de Instalación y Configuración</p>	
---	---	---	---

La integración de la Plataforma con este componente se configura a través de la herramienta de Administración de Trew@, en la sección de administración de componentes. Para más información se puede consultar el manual de usuario de dicha herramienta.

3.1.5 Instalación de Solicit@ 5.0.0

Para proceder a la instalación de Solicit@ 5 se remite al documento de instalación (ver [5] en apartado [5 Referencias](#)) distribuido con dicho componente de w@ndA. Hay que recordar que este componente es de uso opcional en la Plataforma de Tramitación.

3.2 Instalación de otros Componentes

3.2.1 Instalación y configuración de OpenOffice en modo servidor

El servidor de Aplicaciones donde se encuentre el software de PTw@ndA necesita tener instalado el producto OpenOffice .org, ejecutado en modo servidor, para el tratamiento de algunos archivos a utilizar en la plataforma (típicamente, la conversión documentos a formato PDF). El presente apartado es una pequeña guía de instalación que resume los pasos necesarios para su instalación.

Para la instalación de Open Office en modo servidor no se necesita ningún requerimiento de hardware adicional al necesario para la ejecución del software de la Plataforma de Tramitación. Normalmente, será necesaria la intervención de un técnico de sistemas para la instalación de Open Office y los paquetes necesarios.

Este manual presupone que ya se tienen instalados los siguientes componentes:

- Listado de recursos software S.O Linux
 - Open Office (<http://es.openoffice.org/> para descarga y manual de Instalación).
- Listado de recursos software S.O Windows
 - Open Office (<http://es.openoffice.org/> para descarga y manual de Instalación).

El procedimiento de instalación sería el siguiente:

- Para servidores Linux
 - Agregar en el fichero `/etc/rc.local` la siguiente línea:

```

/usr/lib/openoffice/program/soffice -accept=
"socket,host=localhost,port=2002;urp;StarOffice.ServiceManager" -nologo -
headless

```
- Para servidores Windows
 - Crear el fichero `openoffice_modoservidor.bat` y dentro insertarle la línea:

```

soffice -accept=socket,host=localhost,port=2002;urp;StarOffice.ServiceManager -
nologo -headless

```
 - Luego configurar el servidor para que ejecute este script cuando reinicie el sistema.

El Open Office en modo servidor se configura para que responda sólo a solicitudes desde localhost (127.0.0.1). Para desinstalar, simplemente hay que dejar de ejecutar los scripts o servicios anteriormente descritos.

3.3 Instalación de la Plataforma de Tramitación w@ndA

3.3.1 Configuración de los Componentes w@ndA

Las configuraciones de los componentes w@ndA Port@firmas, Avis@dor y Notific@ se realizarán sobre la instalación que se utilice de Trew@, habilitando estos componentes en cada uno de los sistemas a utilizar y estableciendo los parámetros adecuados de conexión. Esta configuración podrá ser realizada haciendo uso de la herramienta de administración de Trew@ (*TrewaAdm*).

Para más información acerca de la configuración de componentes en Trew@, se debe consultar la documentación distribuida en la última versión liberada de este producto.

3.3.2 Configuración del editor de párrafos Trew@

Para la utilización del editor de párrafos habrá que definir la constante `URL_REPORT_JAVA` con el valor: `http[s]://servidor[:puerto]/ruta-despliegue-plataforma/editor/config`.

dentro de los parámetros de configuración de Trew@.

3.3.3 Despliegue en el Servidor de aplicaciones Web

3.3.3.1 Instalación de la aplicación

La plataforma de tramitación se distribuye en formato de fichero WAR (Web Application Archive). Se facilita a continuación un enlace con documentación relativa al proceso a seguir para desplegar una aplicación WAR en un servidor de aplicaciones JBOSS:



Enlace: <https://www.jboss.org/community/docs/DOC-12923> [Installation and Getting Started Guide]

<https://www.jboss.org/community/docs/DOC-12927> [Administration and Configuration Guide]



El despliegue de la aplicación PTw@ndA debe realizarse en un directorio que contenga el nombre de la aplicación con la extensión “.war”. Ej: pt-wanda.war

Para ello, el fichero WAR distribuido en el CD de la aplicación, debe descomprimirse en un directorio situado en la ruta `{JBoss}/server/all/deploy` y eliminarse de dicho directorio.

De esta manera, no existirá desplegado un fichero .war con la aplicación, sino únicamente el directorio descomprimido con la extensión .war

Es imprescindible realizar el despliegue de la Plataforma en el servidor **all** de JBoss, debido a que se requieren determinados servicios para la indexación que sólo se arrancan para la configuración **all** de jboss.

3.3.3.2 Librerías comunes

Es necesario incluir en el directorio `{JBOSS}/server/all/lib` las siguientes librerías (disponibles en la ruta del CD 01. `Software/lib`), requeridas para el correcto funcionamiento de los WebServices y conexión a Base de Datos:

- `jdkom-1.0.jar`
- `xbean-2.2.0.jar`
- `ojdbc14.jar` (o versión posterior del driver de Oracle)

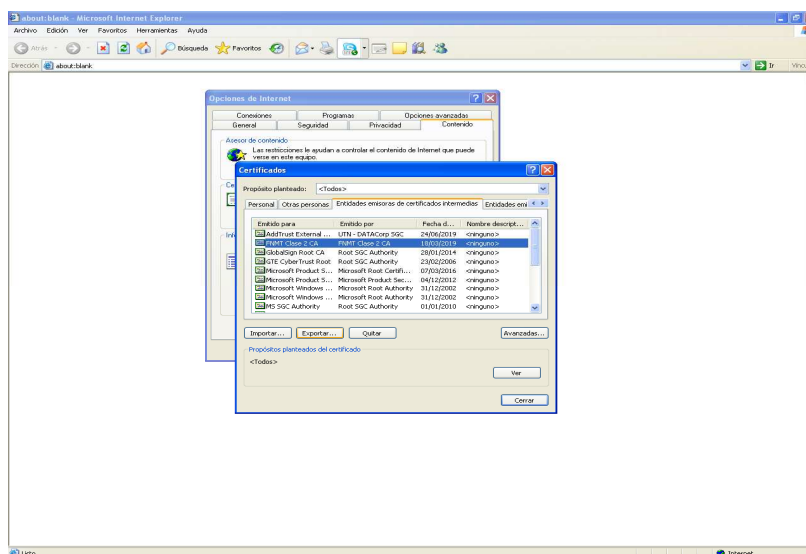
3.3.3.3 Almacén de certificados

Para poder establecer la comunicación SSL entre el servidor de aplicaciones de la Plataforma de Tramitación y el servidor de custodia de @firma, se debe importar la clave pública SSL de la FNMT (Fábrica Nacional de Moneda y Timbre) dentro del almacén de certificados de confianza de la JVM con la que se ejecuta el servidor de aplicaciones de PT.

Para ello, la forma más sencilla es obtener el certificado con la clave pública de FNMT desde el navegador web Internet Explorer de un microordenador que ya disponga de él, exportarlo y, seguidamente, importarlo desde el servidor de aplicaciones.

Para exportarlo, seguiremos el siguiente recorrido en los menús y opciones del navegador: Herramientas → Opciones de Internet... → Contenido → Certificados... → Entidades Emisoras Raíz de Confianza

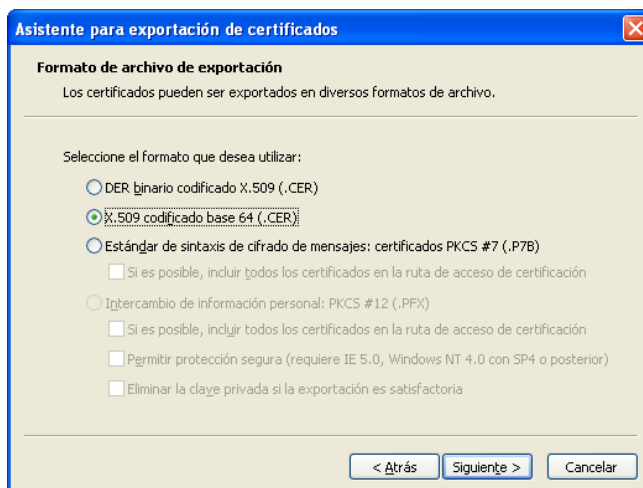
Llegamos a la siguiente pantalla:



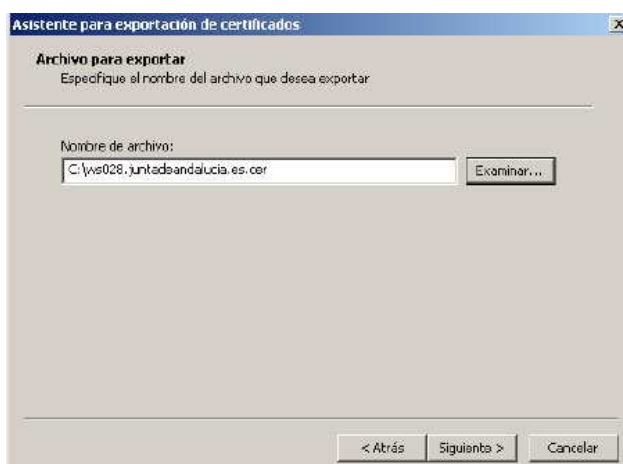
Marcaremos el certificado de la FNMT y pulsamos sobre "Exportar":





Pulsar sobre “Siguiente” y seleccionar “X.509 codificado base 64”:



A continuación, nos pedirá la ruta donde queremos guardarlo.



Tras ésto, ya tenemos el certificado público SSL de la FNMT, 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, localizado en el siguiente directorio:

	<p>Consejería de Hacienda y Administración Pública</p> <p>D.G. de Tecnologías para Hacienda y la Administración Electrónica</p>	<p>Plataforma de Tramitación w@ndA</p> <p>Manual de Instalación y Configuración</p>	
---	---	---	---

`${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 <ruta-fichero-fnmt> -
alias FNMT
```

3.3.3.4 Ampliación de memoria de la Máquina Virtual Java en JBOSS

En el caso de realizar una instalación por defecto de JBOSS 5.1 AS, se estará ejecutando la máquina virtual Java con un tamaño de memoria de 64 megas. Este límite pudiera resultar insuficiente para el correcto funcionamiento de PTw@ndA y resto de aplicaciones web desplegadas en el mismo.

Esto puede detectarse mediante la aparición de errores del tipo `java.lang.OutOfMemoryError` en la consola del servidor, unido a fallos en la ejecución de las aplicaciones instaladas. A fin de evitar dicho error, se habrá de añadir a la variable de sistema `JAVA_OPTS` un valor máximo de memoria de 256 MBytes.

Si se siguieran apreciando caídas (porque exista elevado número de aplicaciones desplegadas en el mismo servidor de aplicaciones o sean grandes consumidores de memoria), la recomendación obvia es seguir ampliando el valor de dicho parámetro en función a las características del servidor y recursos disponibles.

A continuación se explica cómo modificar este parámetro:

Es necesario que acceda al fichero `run.conf` ubicado en la ruta `{JBOSS}/bin` y lo edite, localizando para ello la asignación que se realiza a la variable `JAVA_OPTS`:

```
#
# Specify options to pass to the Java VM.
#
if [ "x${JAVA_OPTS}" = "x" ]; then
    JAVA_OPTS="-Xms128m -Xmx512m -XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true
    -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
fi
```

La próxima vez que se inicie el servidor, este arrancará con la configuración descrita anteriormente, habiendo resultado aumentada la memoria a los tamaños asignados.

Los valores especificados anteriormente dependerán de la memoria que disponga el servidor en el que se encuentra ubicado JBOSS, y del rendimiento y carga que vaya a tener la aplicación en el entorno que se despliegue: desarrollo, pruebas, formación, producción,... Para entornos de producción pueden especificarse valores de memoria mayores: 521, 1024, 2048, ...

3.3.3.5 Configuración de la conexión a Internet del servidor

PTw@ndA requiere conexión a Internet debido a que el componente *struts* necesita descargar la definición sintáctica DTD de sus ficheros XML de configuración.

En el caso de que el servidor disponga de conexión directa a <http://struts.apache.org>, no se requiere hacer ninguna tarea adicional de configuración.

Para el caso contrario, en el que no exista una conexión directa sino a través de servidor proxy, será necesario establecer la configuración del mismo en nuestra JVM, lo cual se consigue definiendo unos parámetros en la variable JAVA_OPTS definida en el fichero `{JBOSS}/bin/run.conf`.

Los valores a añadir dentro de la variable JAVA_OPTS serán:

```
-Dhttp.proxyHost=servidorProxy
-Dhttp.proxyPort=puertoProxy
-Dhttp.proxyUser=usuarioProxy           [si es necesario]
-Dhttp.proxyPassword=claveProxy        [si es necesario]
-Dhttp.nonProxyHosts=Filtro de direcciones locales.
```

Ejemplo:

```
JAVA_OPTS="-server -Xmx256m -Dfile.encoding=UTF-8
-Dhttp.proxyHost="proxy.organismo.junta-andalucia.es"
-Dhttp.proxyPort="8080"
-Dhttp.nonProxyHosts=localhost|*.junta-andalucia.es|*.juntadeandalucia.es"
```

3.3.4 Configuración de la Plataforma de Tramitación w@ndA

Una vez realizado el despliegue de la aplicación en el servidor de aplicaciones, se debe proceder a configurar la aplicación para su adecuado funcionamiento. Todas las conexiones a BBDD se realizan a través de fuentes de datos (DataSources) que serán definidos en el servidor de aplicaciones JBOSS. Se facilita a continuación un enlace a la documentación relativa (de lectura recomendable) a la configuración de fuentes de datos para JBOSS:



Enlace:

http://www.jboss.org/file-access/default/members/jbossas/freezone/docs/Administration_And_Configuration_Guide/5/html/ch13s03.html

Dicho esto, se detalla a continuación la forma de crear estas fuentes de datos para el caso particular de Plataforma de Tramitación w@ndA. Es oportuno recordar la necesidad de incluir en el servidor de aplicaciones la versión apropiada del driver JDBC acorde con la versión del SGBD a emplear.

3.3.4.1 Configuración de conexiones con Base de datos

- **Conexión a la Base de datos de Trew@:** la conexión con la base de datos de Trew@ se realizará mediante un pool de conexiones gestionado por el propio servidor de aplicaciones que, en el caso de JBOSS, se configura en el archivo de configuración `pt-wanda-oracle-ds.xml` ubicado en el directorio `{JBOSS}/server/all/deploy` tal y como se indica a continuación:

```
[...]
<local-tx-datasource>
  <jndi-name>TrewaDS</jndi-name>
  <use-java-context>false</use-java-context>
  <connection-url>jdbc:oracle:thin:@[url]:[puerto]:[instancia]</connection-url>
  <driver-class>[driver de la clase]</driver-class>
  <user-name>[usuario]</user-name>
  <password>[clave]</password>
  <min-pool-size>[número mínimo de conexiones]</min-pool-size>
  <max-pool-size>[número máximo de conexiones]</max-pool-size>
  <blocking-timeout-millis>[tiempo de espera en bloqueo]</blocking-timeout-millis>
  <idle-timeout-minutes>[tiempo]</idle-timeout-minutes>
  <metadata>
    <type-mapping>[Versión del SGBD. Ej: Oracle9i]</type-mapping>
  </metadata>
</local-tx-datasource>
<local-tx-datasource>
  <jndi-name>default</jndi-name>
  <use-java-context>false</use-java-context>
  <connection-url>jdbc:oracle:thin:@[url]:[puerto]:[instancia]</connection-url>
  <driver-class>[driver de la clase]</driver-class>
  <user-name>[usuario]</user-name>
  <password>[clave]</password>
  <min-pool-size>[número mínimo de conexiones]</min-pool-size>
  <max-pool-size>[número máximo de conexiones]</max-pool-size>
  <blocking-timeout-millis>[tiempo de espera en bloqueo]</blocking-timeout-millis>
  <idle-timeout-minutes>[tiempo]</idle-timeout-minutes>
  <metadata>
    <type-mapping>[Versión del SGBD. Ej: Oracle9i]</type-mapping>
  </metadata>
</local-tx-datasource>
[...]
```

- Conexión con la Base de datos de Plataforma:** la conexión con el modelo de datos propio de PTw@ndA se realizará configurando en el archivo *hibernate.properties* una referencia a un DataSource declarado en el servidor de aplicaciones. El usuario de BBDD vinculado a este DataSource deberá tener habilitados los permisos para crear tablas y para operar sobre éstas (insert, update, select y delete).

El archivo *hibernate.properties* debe quedar configurado tal y como se indica a continuación:

```
## HIBERNATE. Configuración de acceso al esquema propio de plataforma.
#DataSource de datos paramétricos de Plataforma (configurado en server.xml)
hibernate.connection.jndi=Plataforma
hibernate.connection.release_mode=auto
hibernate.dialect=org.hibernate.dialect.OracleDialect
hibernate.hbm2ddl.auto=update
hibernate.show_sql=false
transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory

#DataSource de datos no-paramétricos de Plataforma (configurado en server.xml)
hibernate.connection.jndi.datos=PlataformaDatos
hibernate.connection.release_mode.datos=auto
hibernate.dialect.datos=org.hibernate.dialect.OracleDialect
hibernate.hbm2ddl.auto.datos=update
hibernate.show_sql.datos=false
transaction.factory_class.datos=org.hibernate.transaction.JDBCTransactionFactory
java.naming.factory.initial.datos=org.jnp.interfaces.NamingContextFactory
```

No es necesario modificar este fichero a menos que se desee emplear un nombre JNDI distinto al valor por defecto para el DataSource, esto es, "*Plataforma o PlataformaDatos*".

También puede ser modificado para situaciones en las que el SGBD sea distinto a Oracle, ya que es ésta la configuración ofrecida por defecto. Caso de ser necesario modificar "hibernate.properties" por alguno de los motivos indicados, los parámetros que pueden modificarse son los siguientes:

Acceso a la base de datos de plataforma			
I	hibernate.connection.jndi	Nombre del datasource para el esquema paramétrico.	Plataforma
I	hibernate.dialect	Nombre del dialecto empleado para el esquema paramétrico.	org.hibernate.dialect.XXXXXXX
I	hibernate.connection.jndi.datos	Nombre del datasource para el esquema datos.	PlataformaDatos
I	hibernate.dialect.datos	Nombre del dialecto empleado para el esquema datos.	org.hibernate.dialect.XXXXXXX

Una vez haya realizado las modificaciones en el fichero de configuración de Hibernate hibernate.properties, debe añadir los orígenes de datos en el servidor de aplicaciones (pt-wanda-oracle-ds.xml), de forma similar a la realizada en el apartado anterior con Trew@. El fichero resultante debe tener la estructura siguiente:

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

<local-tx-datasource>
  <jndi-name>TrewaDS</jndi-name>
  <use-java-context>false</use-java-context>
  <connection-url>jdbc:oracle:thin:@[url]:[puerto]:[instancia]</connection-url>
  <driver-class>[driver de la clase]</driver-class>
  <user-name>[usuario]</user-name>
  <password>[clave]</password>
  <min-pool-size>[número mínimo de conexiones]</min-pool-size>
  <max-pool-size>[número máximo de conexiones]</max-pool-size>
  <blocking-timeout-millis>[tiempo de espera en bloqueo]</blocking-timeout-millis>
  <idle-timeout-minutes>[tiempo]</idle-timeout-minutes>
  <metadata>
    <type-mapping>[Versión del SGBD. Ej: Oracle9i]</type-mapping>
  </metadata>
</local-tx-datasource>
<local-tx-datasource>
  <jndi-name>default</jndi-name>
  <use-java-context>false</use-java-context>
  <connection-url>jdbc:oracle:thin:@[url]:[puerto]:[instancia]</connection-url>
  <driver-class>[driver de la clase]</driver-class>
  <user-name>[usuario]</user-name>
  <password>[clave]</password>
  <min-pool-size>[número mínimo de conexiones]</min-pool-size>
  <max-pool-size>[número máximo de conexiones]</max-pool-size>
  <blocking-timeout-millis>[tiempo de espera en bloqueo]</blocking-timeout-millis>
  <idle-timeout-minutes>[tiempo]</idle-timeout-minutes>
  <metadata>
    <type-mapping>[Versión del SGBD. Ej: Oracle9i]</type-mapping>
  </metadata>
</local-tx-datasource>
[continua en página siguiente...]
```

[sigue de página anterior...]

```

<local-tx-datasource>
  <jndi-name>Plataforma</jndi-name>
  <use-java-context>false</use-java-context>
  <connection-url>jdbc:oracle:thin:@[url]:[puerto]:[instancia]</connection-url>
  <driver-class>[driver de la clase]</driver-class>
  <user-name>[usuario]</user-name>
  <password>[clave]</password>
  <min-pool-size>[número mínimo de conexiones]</min-pool-size>
  <max-pool-size>[número máximo de conexiones]</max-pool-size>
  <blocking-timeout-millis>[tiempo de espera en bloqueo]</blocking-timeout-millis>
  <idle-timeout-minutes>[tiempo]</idle-timeout-minutes>
  <metadata>
    <type-mapping>[Versión del SGBD. Ej: Oracle9i]</type-mapping>
  </metadata>
</local-tx-datasource>
<local-tx-datasource>
  <jndi-name>PlataformaDatos</jndi-name>
  <use-java-context>false</use-java-context>
  <connection-url>jdbc:oracle:thin:@[url]:[puerto]:[instancia]</connection-url>
  <driver-class>[driver de la clase]</driver-class>
  <user-name>[usuario]</user-name>
  <password>[clave]</password>
  <min-pool-size>[número mínimo de conexiones]</min-pool-size>
  <max-pool-size>[número máximo de conexiones]</max-pool-size>
  <blocking-timeout-millis>[tiempo de espera en bloqueo]</blocking-timeout-millis>
  <idle-timeout-minutes>[tiempo]</idle-timeout-minutes>
  <metadata>
    <type-mapping>[Versión del SGBD. Ej: Oracle9i]</type-mapping>
  </metadata>
</local-tx-datasource>

```



El atributo `<type-mapping>` indica la versión del Sistema Gestor de Base de Datos utilizado, permitiendo optimizar al gestor del pool de conexiones de JBOSS las operaciones realizadas sobre el SGBD.

Este parámetro es opcional, aunque se recomienda su utilización para optimizar tiempos de respuesta en operaciones.

Evidentemente, se podrá añadir cualquier otra fuente de datos adicional que sea requerida por los desarrollos modulares verticales, adicionales a los que se incluyen por defecto en PTw@ndA, que se lleven a cabo por el implantador del sistema.

3.3.4.2 Primer inicio de PTw@ndA

Tras la configuración de Hibernate y la definición del DataSource asociado, el primer inicio del JBOSS con el WAR de PT desplegado implicará que Hibernate, automáticamente, cree todas las tablas necesarias para PT.

El sistema no estará aún completa y correctamente implantado, ya que Hibernate habrá, efectivamente, creado las tablas, como se indicó con anterioridad, pero no habrá insertado los valores de configuración necesarios en la tabla CONFIG para un correcto funcionamiento de PT.

Llegados a este punto, se requiere, por tanto, detener el servidor de aplicaciones JBOSS para, a continuación, lanzar sobre el esquema de BBDD PlataformaDatos el script 05.Datos_PTW_TPOWNER.sql localizado en el directorio 02.Scripts BBDD de la distribución de PT.



Para actualizaciones de versiones o para el despliegue de una instalación de PT en un servidor de aplicaciones vinculado a una BBDD de Plataforma ya existente, este paso NO es necesario. Los valores configurados previamente serán válidos para la nueva instalación o actualización.

3.3.4.3 Ejecución de scripts de BBDD

Como alternativa al punto anterior, a partir de esta versión de PTw@ndA, se proporcionan unos scripts de BBDD que permiten la creación del esquema, usuarios, tablas y datos paramétricos, sin necesidad de realizar el paso anterior.

Los pasos a realizar son los siguientes:

- i. Acceder con usuario system/sys y ejecutar los script ubicados en el directorio *02. Scripts BBDD * (previa modificación de las localizaciones de los tablespaces).
 - 01.Tablespace.sql
 - 02.Usuarios.sql (previa modificación de las constantes con los nombres de los tablespaces)
- ii. Acceder con usuario PTW_OWNER y ejecutar los scripts siguientes en el orden dado (previa modificación de las constantes con los nombres de los tablespaces creados):
 - 03.Tablas_PTW_OWNER.sql
 - 07.Secuencia_PTW_OWNER.sql
- iii. Acceder con usuario PTW_TPOWNER y ejecutar los scripts siguientes en el orden dado:
 - 04.Tablas_PTW_TPOWNER.sql (previa modificación de las constantes con los nombres de los tablespaces)
 - 05.Tablas_JMS (previa modificación de las constantes con los nombres de los tablespaces)
 - 06.Datos_PTW_TPOWNER.sql
 - 08.Secuencia_PTW_TPOWNER.sql
- iv. Acceder con usuario system/sys y ejecutar el script (previa modificación de las constantes con los nombres de los tablespaces creados):
 - 09.Usuarios_Trabajo.sql
- v. Acceder con usuario PTW_TPUSER y ejecutar el script
 - 10.Sinonimos_PTW_TPUSER.sql
- vi. Acceder con usuario PTW_USER y ejecutar el script
 - 11.Sinonimos_PTW_USER.sql
- vii. Modificar las propiedades relativas a la creación o sincronización de hibernate con el esquema de BBDD; para ello es necesario cambiar el valor de las siguientes propiedades del fichero `hibernate.properties`:


```
hibernate.hbm2ddl.auto=false
hibernate.hbm2ddl.auto.datos=false
```

Una vez realizado este punto, se configurará la aplicación a partir de lo descrito en el siguiente apartado, y al arrancar el servidor estará la aplicación PTw@ndA correctamente desplegada y configurada.

3.3.4.4 Configuración de la Aplicación

La configuración de PTw@ndA está centralizada en la tabla CONFIG del esquema propio de PTw@ndA (PlataformaDatos).

En la tabla se detallan parámetros de necesaria modificación en toda implantación de PTw@nda (tipo I), otros que pueden requerir, ocasionalmente, un cambio (tipo O) y aquellos otros que raramente deban cambiarse (tipo E):

Tipo	Parámetro	Descripción	Ejemplo	Obligatorio
Propiedades de Trew@				
I	TREWASISTEMA	Sistema Trew@ utilizado por defecto	TREW@	SI
I	TREWASISTEMAS	Diferentes Sistemas que pueda usar PTw@nda. Se definen separados por comas.	TREW@,CJAP	SI
I	TREWACONEXION	DataSource de conexion a la base de datos de Trew@	TrewaDS	SI
O	TREWAAPI_CACHESIZE	Número máximo de APIs instancias	25	SI
O	TREWAAPI_TIMEOUT	Máximo de inactividad de una API para cerrarla (milisegundos)	15000	SI
O	TREWAAPI_TIMEROTATION	Periodo de ejecución del demonio de API's para eliminar las caducadas (milisegundos)	10000	SI
	TREWA_DOC_FILTRARUSUARIO	A la hora de mostrar los documentos, mostrar solo los documentos elaborados por el usuario.	false	SI
	TREWA_TAREA_FILTRARUSUARIO	A la hora de mostrar las tareas, mostrar solo las tareas elaborados por el usuario.	false	SI
Propiedades del buscador				
O	BUSCADOR_SOLICITA	Indica si el buscador se debe integrar con solicita	false	SI
E	BUSCADOR_MODIFICADORES	Modificadores de la consulta	&stylesheet=&wt=xslt&tr=buscador.xsl &indent=on&hl=true&rows=10 &start=0	SI
login/@firma				
O	MODO_LOGIN	Indica la forma de logarse por defecto (2 = Autenticación con DNI, 1 = Autenticación con @firma 3 = Autenticación vía LDAP)	2	SI
E	USAR_FIRMA	Parámetro reservado. No tocar.	1	-
E	USAR_DNI	Parámetro reservado. No tocar.	2	-
E	USAR_LDAP	Parámetro reservado. No tocar.	3	-
I	SERVIDOR_FIRMA	IP o nombre DNS del servidor @firma	wsXXX.juntadeandalucia.es	SI
I	ID_APLICACION_FIRMA	Aplicación @firma a utilizar.	idPlataformaEnAfirma	NO
O	VERSION_FIRMA	La versión de @firma	5	NO

	USUARIO_FIRMA	El nombre de la aplicación dada de alta en @firma	USUARIO	NO
	LOGIN_FIRMA	El password de la aplicación dada de alta en @firma	PASSWORD	NO
	DIRECCION_VERIFICAR_FIRMA	Cadena utilizada para la formación del pie de página en la firma directa de documentos.		NO
port@firmas				
	ID_APLICACION_PORTA_FIRMA	Aplicación a utilizar en port@firma para el alta y consulta de peticiones.	idPlataformaEnPortafirmas	NO
	URL_PORTA_FIRMA	Acceso al informe de firma. Servidor port@firmas.	http[s]://servidor-portafirma[:puerto]/pfirma	SI
	CONVERSION_PDF_PORTAFIRMAS	Indica si hay que convertir a pdf los documentos antes de mandarlos a firmar.	true	SI
	PORTAFIRMAS_ENVIO_CORREO_AVISOS	Indica si se notificará por vía e-mail el envío de documentos a firmar.	false	NO
Conversión de documentos a PDF con OpenOffice				
	IP_OPENOFFICE	Servidor donde encontramos el servidor OpenOffice (solo es posible indicar localhost)	localhost	SI
	PUERTO_OPENOFFICE	Puerto donde debe escucha el servidor OpenOffice.	2002	SI
Sincronización				
	APLICACION_SINCRONISMO_TRAMITADOR	Si es "true", el sistema sincronizará la información en el esquema de datos propio de la plataforma con la información del sistema o sistemas Trew@ de forma automática tras el despliegue de la aplicación.	true	SI
	APLICACION_CREACION_INSTALACIONES_DEFECTO	Si es "true", el sistema creará la instalación "DEFECTO" de forma automática tras el despliegue de la aplicación.	true	SI
Configuración LDAP				
	LDAP_INITIAL_CONTEXT_FACTORY	Proveedor del servicio LDAP (Únicamente es necesario en caso de utilizar LDAP)	com.sun.jndi.ldap.LdapCtxFactory	NO
	LDAP_URL	URL de acceso al LDAP (Únicamente es necesario en caso de utilizar LDAP)	ldap://<servidor ldap>[:<puerto>]	NO

I	LDAP_DN	Entrada a la raíz del árbol LDAP. (Únicamente es necesario en caso de utilizar LDAP)	<code>ou=<nombre ou>,dc=<organismo>,dc=es</code>	NO
I	LDAP_ID	Identificador del usuario de LDAP que se utiliza para realizar la autenticación (Únicamente es necesario en caso de utilizar LDAP)	Ej. <code>uid</code>	NO
I	LDAP_ATRIBUTO	Atributo del usuario LDAP que contiene el NIF de la persona para realizar la autorización a la Plataforma de Tramitación al verificar que el usuario está dado de alta en Trew@ con los perfiles correctos. (Únicamente es necesario en caso de utilizar LDAP)	Ej. <code>nif</code>	NO
Registro @ries de Salida				
I	REGISTRO_URL	Url del servicio Web utilizado para realizar el registro de salida	<code>http://ariesdes.cjap.junta-andalucia.es:8081/WSariesRTL/services/RegistroTelematicoSOAP</code>	NO
I	REGISTRO_USUARIO	Usuario dado de alta en @ries	<code>USUARIO</code>	NO
I	REGISTRO_PASSWORD	Password del usuario dado de alta en @ries	<code>PASSWORD</code>	NO
I	REGISTRO_ASUNTO	Asunto dado de alta en @ries	<code>COD_ASUNTO</code>	NO
I	REGISTRO_CODORIGEN	En su defecto, código de unidad orgánica de origen	<code>ORGANISMO_ORIGEN</code>	NO
I	REGISTRO_CODDESTINO	En su defecto, código de unidad orgánica de destino	<code>ORGANISMO_DESTINO</code>	NO
Documento				
O	DOCUMENTO_TIPO_INCORPORADO_NO_DEFINIDO	Nombre del tipo de documento de Trew@ que se utilizará para incorporar documentos no definidos en los flujos de tramitación, por ejemplo en la utilidad de incorporación de documentos.	<code>NoDefinido</code>	SI
O	DOCUMENTO_TIPO_INCORPORADO_NO_DEFINIDO_INFORMABLE	Nombre del tipo de documento de Trew@ que se utilizará para incorporar documentos no definidos en los flujos de tramitación, por ejemplo en la utilidad de incorporación de documentos y a que además son informables.	<code>NoDefInf</code>	SI
O	DOCUMENTO_SUSTITUIRFIRMADOS	Para insertar los documentos firmados en trewa con cajetín		SI
Perfil				
E	ADMINISTRACION_PERFIL	Perfil mínimo que debe tener un usuario para hacer uso de la herramienta de administración.	<code>TR_R_USUARIO</code>	SI

E	ESCRITORIO_PERFIL	Perfil mínimo que debe tener un usuario para hacer uso del escritorio.	TR_R_USUARIO	SI
I	USUARIO_SERVICIO_WEB	Código de usuario del tramitador definido en trew@ que se utilizará para realizar las operaciones desde los servicios WEB.	PL_WEBUSR	SI
Otras				
I	DEBUG	Si es "true", habilita el servidor en modo debug.	false	SI
I	URL_ESCRITORIO	URL DE ACCESO AL ESCRITORIO	http[s]://servidor[:puerto]/ruta-despliegue-plataforma	SI
I	RUTA_FICHEROS_AYUDA_CONTEXTUAL	Ruta donde encontrar los ficheros para la ayuda contextual. Esta ayuda está asociada a la utilidad AyudaContextual, mostrándose los ficheros alojados en el directorio citado.	C:\\ayudaContextual	NO
O	TAREASPENDIENTES_RESERVADOS	Si es "true", añade a la lista de tareas pendientes, las de aquellos expedientes reservados por el usuario.	true	SI
E	CONEXION_TRAMITADOR_FIJA	Si es "true", habilita las conexiones fijas Trew@ para el escritorio de tramitación. Indicar solo "true" o "false".	false	SI
E	CONEXION_ADMINISTRACION_FIJA	Si es "true", habilita las conexiones fijas Trew@ para la herramienta de administración. Indicar solo "true" o "false".	false	SI
I	CACHEO_API_NUMERO	Es el número máximo de API's asignadas a un solo usuario.	1	SI
I	CACHEO_API	Indica si queremos que se haga un cacheado de la API de TREW@	true	SI
I	DESCARGAR_MODULOS	Indica si queremos instalar los módulos verticales almacenados en BBDD,	true	SI
I	INSTALACION_DEFECTO	Indica la instalación por defecto. Si está configurado, se salta la pantalla de selección de instalación a la hora de logarse en la aplicación.		NO
I	SISTEMA_DEFECTO	Indica el sistema por defecto. Si está configurado, se salta la pantalla de selección de sistema a la hora de logarse en la aplicación.		NO
I	VISUALIZAR_PANTALLA_INTERESADOS_EXPEDIENTE	Si es "false", deshabilita la pantalla de asociar interesados en el alta de un expediente.	true	SI

I	NOMBRE_COMPONENTE_NUMERA	Nombre del componente Trew@ definido para la asignación automática de números de expediente a la hora de crearlos.		NO
I	EVITAR_FILTROS_FASE_EXPEDIENTE	Si es "true", no se diferencia entre diferentes instancias de la misma fase para el cálculo de documentos y tareas.	false	SI

3.3.4.5 Configuración del motor de indexación y búsqueda

El motor de indexación y búsqueda de la Plataforma de Tramitación w@ndA está compuesto en su núcleo por el proyecto **Solr de Apache Lucene**. Las características de dicho sistema son:

- Avanzadas capacidades de búsqueda *full-text*.
- Optimizado para soportar un volumen de tráfico elevado.
- Basado en interfaces abiertas, como XML y http.
- Escalable.
- Flexible y parametrizable en base a archivos de configuración en formato XML.
- Arquitectura extensible en base a *plug-ins*.

Este componente de la Plataforma de Tramitación habilitará búsquedas sobre tres sistemas de información de forma integrada y transparente para el usuario tramitador:

- Trew@.
- Gestor documental (compatible con Trew@, W@rdA y Alfresco). (nota: se utiliza **LIUS** para la extracción de texto de los documentos).
- Solicit@.

Para el correcto funcionamiento de Solr, independientemente del número de nodos utilizados para el despliegue de PTw@ndA, se ha optado por una solución basada en JMS.

JMS es la solución de Sun para los sistemas de mensajes. JMS se sitúa como middleware de la comunicación de varias aplicaciones (distintos nodos), concretamente se hará uso de la comunicación Publisher/Subscriber. En este tipo de comunicación, la aplicación A publica su mensaje en el servicio JMS y lo reciben todas las aplicaciones que estén suscritas al servicio JMS al que se envió el mensaje.

Esta solución está pensada para despliegues de PTw@ndA sin almacenamiento compartido, de tal forma que cualquier operación realizada sobre el almacenamiento de índices de un nodo, sea replicada al resto de almacenamientos.

Se estima que la configuración por defecto de Solr es válida para la mayoría de las instalaciones de PTw@ndA. Por defecto, el índice del motor de búsqueda se creará dentro del directorio ".solr/data", relativo al directorio desde el que se lanza el servidor de aplicaciones. Alternativamente, se puede establecer una ubicación distinta sin necesidad de modificar contenidos propios del WAR de la aplicación.

Esta acción se llevará a cabo mediante la especificación de un parámetro de sistema a la JVM que ejecuta JBOSS modificando el valor de la variable **JAVA_OPTS** del fichero {JBOSS}/bin/run.conf.

El nuevo parámetro a incluir será `-Dsolr.data.dir=<ruta-directorio-solr>`.

Una configuración avanzada del motor de búsqueda requiere la edición de los archivos de configuración propios de Solr. Se insiste nuevamente en que la configuración por defecto será válida para la mayoría de las instalaciones de PT:

- `[APLICACIÓN]\WEB-INF\CLASSES\schema.xml`
- `[APLICACIÓN]\WEB-INF\CLASSES\solrconfig.xml`

Puede consultar información adicional de **Solr** y de la configuración de los ficheros descritos en el siguiente enlace:



Enlace: <http://lucene.apache.org/solr/>

[Página oficial]

<http://wiki.apache.org/solr/>

[Documentación]



Se ha detectado un bug del servidor de aplicaciones JBOSS cuando se despliega sobre Solaris. El bug se localiza en la librería **server/log/jgroups.jar** y se soluciona con la versión 2.6.15GA que lo corrige, la cual se puede localizar aquí:

<http://sourceforge.net/projects/javagroups/files/>

Esta versión es la última 2.6.X, las versiones 2.7.X y posteriores **NO SON COMPATIBLES con JBOSS 5.1GA**.



Es imprescindible realizar el despliegue de la Plataforma en el server **all** de JBoss, debido a que se requieren determinados servicios para la indexación que sólo se arrancan para la configuración all de jboss.

Para una correcta configuración del sistema JMS deben realizarse los siguientes pasos:

- **Creación de la cola de mensajes.**

Definimos un nuevo "topic" que será el que use PT-WANDA para intercambiar mensajes. Este topic se define en el fichero `${JBOSS_HOME}/server/all/deploy/messaging/destinations-service.xml`

```
<mbean code="org.jboss.jms.server.destination.TopicService"
      name="jboss.messaging.destination:service=Topic,name=SOLR-PTWANDA-TOPIC" xbean-dd="xmdesc/Topic-xmbean.xml">
  <depends optional-attribute-name="ServerPeer">jboss.messaging:service=ServerPeer</depends>
  <depends>jboss.messaging:service=PostOffice</depends>
  <attribute name="Clustered">true</attribute>
  <attribute name="SecurityConfig">
    <security>
```

```

        <role name="topicSOLR" read="true" write="true" create="true"/>
    </security>
</attribute>
</mbean>

```

Esta cola es nombrada como **SOLR-PTWANDA-TOPIC**. Este nombre no debe ser editado debido a que se realizan referencias a él desde el propio núcleo de la Plataforma.

Definimos un rol **topicSOLR** que deberá ser asignado a cada uno de los nodos de nuestra arquitectura.

- **Eliminación del fichero hsqldb-persistence-service.xml.**

Eliminar el fichero **hsqldb-persistence-service.xml** localizado en la siguiente ruta:

```

${JBOSS_HOME}/server/all/deploy/messaging/hsqldb-persistence-service.xml

```

- **Añadir fichero oracle-persistence-service.xml.**

Añadir al directorio `${JBOSS_HOME}/server/all/deploy/messaging` el fichero "oracle-persistence-service.xml" que se distribuye con la Plataforma de Tramitación. Este fichero se encuentra en la ruta `/05.Software base/solr/` dentro del CD de distribución de Plataforma.

Dentro de este fichero aparece repetidas veces una referencia a un DataSource (SolrDS) que será el esquema Oracle donde se asegura la persistencia de los mensajes. Debe definirse un DS dentro de `${JBOSS_HOME}/server/all/deploy` con los valores necesarios, ya sea en un XML diferenciado o dentro de otro XML donde se definen más DS.

El nuevo DataSource especificado puede apuntar a uno de los 2 esquemas de la aplicación, a Plataforma o bien a PlataformaDatos, sin necesidad de crear otro esquema para el nuevo DS definido. Es importante destacar en este punto la necesidad de declarar la propiedad **<use-java-context>** a **true**, dado que en el fichero *oracle-persistence-service.xml* se declara el datasource nuevo con la cadena **java:/**.

Dentro de este fichero se define una etiqueta `<Clustering>`.

```

<!-- This post office is non clustered. If you want a clustered post office then set to true -->
<attribute name="Clustered">true</attribute>

```

Este atributo determinará si un mensaje enviado desde un nodo a la cola de mensajes, será reenviando a los demás nodos.

En el caso de tener un almacenamiento diferenciado para cada nodo se establecerá un valor "true", de esta forma cualquier operación sobre el almacenamiento de índices de un nodo será replicada en los restantes almacenamientos. Por el contrario, si se hace uso de un almacenamiento compartido se establecerá un valor "false". Con esto evitamos que se reenvien mensajes al resto de los nodos, impidiendo que se realice la misma operación sobre el mismo almacenamiento de índices.

- **Modificar fichero messaging-jboss-beans.xml.**

Modificar el fichero messaging-jboss-beans.xml ubicado en la ruta `${JBoss_HOME}/server/all/deploy/messaging`. Dentro de la etiqueta `<module-option name="dsJndiName">` se debe configurar el DS que se haya seleccionado en el paso anterior. Ejemplo:

```
<!-- messaging application-policy definition -->
<application-policy xmlns="urn:jboss:security-beans:1.0" name="messaging">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
      <module-option name="unauthenticatedIdentity">guest</module-option>
      <module-option name="dsJndiName">java:/SolrDS</module-option>
      <module-option name="principalsQuery">SELECT PASSWD FROM JBM_USER WHERE USER_ID=?</module-option>
      <module-option name="rolesQuery">SELECT ROLE_ID, 'Roles' FROM JBM_ROLE WHERE USER_ID=?</module-option>
    </login-module>
  </authentication>
</application-policy>
```

- **Configuración de los nodos jboss.**

Todos los servidores de aplicación JBoss con PTw@ndA 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, por lo que 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 pertenece. 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:

Nodo 1: `-Djboss.partition.name=ptw -Djboss.messaging.ServerPeerID=1`

Nodo 2: `-Djboss.partition.name=ptw -Djboss.messaging.ServerPeerID=2`

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 PTw@ndA pertenecientes a diversos entornos, siempre que se les asignen distintos nombres de partición: `ptwDesa`, `ptwPru`, `ptwPro`, ...

Finalmente, para cada nodo JBoss habrá que editar el siguiente fichero:

`${JBoss_HOME}/server/all/deploy/jbossweb.sar/server.xml`

Será necesario localizar el elemento `Engine` dentro del documento XML y establecer un atributo `jvmRoute` con el nombre de nodo asignado, el cual se recupera del establecido en el fichero `run.conf`:

```
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="${jboss.messaging.ServerPeerID}">
...
</Engine>
```

- **Creación de usuarios y asignación de roles.**

Cada uno de los servidores JBOSS que se vayan a utilizar deben darse de alta en el sistema de cola de mensajes JMS. Para ello es necesario crear un usuario por cada servidor JBOSS. Posteriormente a cada usuario se le asigna el rol especificado a la hora de crear la cola (fichero destinations-service.xml). En este caso se creó el rol **topicSOLR**.

Tenemos que editar el fichero oracle-persistence-service.xml que se proporciona con la distribución de PTw@ndA. El último mbean creado en este fichero hace referencia a los usuarios y roles que se insertarán en la correspondiente tabla en el esquema SolrDS. En el atributo **SqlProperties** se especifica la sintaxis SQL de creación de las tablas usuarios y roles. Habrá que añadir tantos usuarios como servidores JBOSS vayan a formar parte del cluster. En el ejemplo proporcionado se hace uso de 2 servidores:

```
<mbean code="org.jboss.jms.server.plugin.JDBCJMSUserManagerService"
  name="jboss.messaging:service=JMSUserManager"
  xmbean-dd="xmdesc/JMSUserManager-xmbean.xml">

  <depends>jboss.jca:service=DataSourceBinding,name=SolrDS</depends>

  <depends optional-attribute-name="TransactionManager">jboss:service=TransactionManager</depends>

  <attribute name="DataSource">java:/SolrDS</attribute>

  <attribute name="CreateTablesOnStartup">>true</attribute>

  <attribute name="SqlProperties"><![CDATA[
CREATE_USER_TABLE=CREATE TABLE JBM_USER (USER_ID VARCHAR2(32) NOT NULL, PASSWD VARCHAR2(32) NOT NULL, CLIENTID VARCHAR2(128), PRIMARY KEY(USER_ID))
CREATE_ROLE_TABLE=CREATE TABLE JBM_ROLE (ROLE_ID VARCHAR2(32) NOT NULL, USER_ID VARCHAR2(32) NOT NULL, PRIMARY KEY(USER_ID, ROLE_ID))
SELECT_PRECONF_CLIENTID=SELECT CLIENTID FROM JBM_USER WHERE USER_ID=?
POPULATE.TABLES.1 = INSERT INTO JBM_USER (USER_ID, PASSWD, CLIENTID) VALUES ('NODO-1', 'NODO-1', 'DurableSubscriberExample')
POPULATE.TABLES.2 = INSERT INTO JBM_USER (USER_ID, PASSWD, CLIENTID) VALUES ('NODO-2', 'NODO-2', 'DurableSubscriberExample')
POPULATE.TABLES.3 = INSERT INTO JBM_ROLE (ROLE_ID, USER_ID) VALUES ('topicSOLR', 'NODO-1')
POPULATE.TABLES.4 = INSERT INTO JBM_ROLE (ROLE_ID, USER_ID) VALUES ('topicSOLR', 'NODO-2')
]]></attribute>
</mbean>
```

Las líneas POPULATE.TABLES.1 y POPULATE.TABLES.2 insertan los usuarios en la tabla **JBM_USER**. El campo USER_ID debe ser completado con la sintaxis **NODO-** y a continuación el identificador especificado en el parámetro - Djboss.messaging.ServerPeerID. El campo PASSWD y CLIENTID pueden albergar cualquier cadena.

Las líneas POPULATE.TABLES.3 y POPULATE.TABLES.4 asocian los usuarios creados con el rol.

La configuración en modo clúster es obligatoria realizarla independientemente del número de jboss empleados. A continuación se detalla cómo realizar la configuración de los diferentes nodos si se desea utilizar una arquitectura basada en alta disponibilidad:

3.3.4.5.1 Alta Disponibilidad

3.3.4.5.1.1 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 haga entrar en tal circunstancia en funcionamiento alguno de los nodos pasivos.

- Se dispone de dos o más nodos activos que se reparten (mediante un mecanismo hardware o software) la carga del trabajo. Con esta arquitectura, un sistema externo a estos nodos detecta los posibles problemas que puedan presentar; en tal caso, el nodo afectado dejará de recibir trabajo hasta su recuperación.

De las dos posibilidades planteadas es preferible, con diferencia, la segunda, ya que además de lograr la alta disponibilidad y, por tanto, la continuidad del servicio, se logra un mayor aprovechamiento de los recursos (no quedan recursos ociosos como en el primer caso) y se producen mejoras de rendimiento y su progresiva escalabilidad con la incorporación de más nodos activos al sistema. Por todo ello, en este documento se explicará cómo conseguir una infraestructura de este tipo para el despliegue del sistema PTw@ndA.

3.3.4.5.1.2 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 PTw@ndA, no existe ningún problema con esta casuística.

- 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 mala gestión del reparto de la carga del trabajo o por la caída no controlada de nodos del sistema.

3.3.4.5.1.3 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 misma sesión serán redirigidas siempre a un único nodo.

Del mismo modo, se deberá detectar la creación de nuevas sesiones para asignar nodo de ejecución 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 la reasignación a otro nodo de los usuarios conectados, implicando la pérdida de sesión sobre los mismos.

- 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 éstos se ven afectados por la caída de alguno de ellos. Como contrapartida, esta solución implica una mayor carga de trabajo y tráfico de red para los servidores de aplicaciones.

3.3.4.5.1.4 Solución ejemplo basada en Apache

Se plantea a continuación una solución al problema expuesto basada en el servidor web APACHE como elemento software balanceador de carga entre nodos. En nuestro caso, los distintos servidores de aplicaciones JBOSS con PTw@ndA instalado.

Aunque no se detallará en este documento, es también posible la configuración de varios servidores web APACHE 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 (igualmente, de la Fundación Apache).

3.3.4.5.1.4.1 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, el puerto 8009.

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

# Se definirá un worker de tipo host para cada instalación de JBOSS
# Definimos un primer nodo JBOSS
worker.ptw1.host=nodo1.midominio.es
worker.ptw1.port=8009
worker.ptw1.type=ajp13
worker.ptw1.lbfactor=1
worker.ptw1.socket_timeout=5
worker.ptw1.recycle_timeout=10

# Definimos un segundo nodo JBOSS
worker.ptw2.host=nodo2.midominio.es
worker.ptw2.port=8009
worker.ptw2.type=ajp13
worker.ptw2.lbfactor=1
worker.ptw2.socket_timeout=5
worker.ptw2.recycle_timeout=10

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

# Definimos el worker global ptw de tipo lb load balancing
worker.ptw.type=lb
worker.ptw.balance_workers=ptw1,ptw2
worker.ptw.sticky_session=false
worker.ptw.session_cookie=JSESSIONID
```


3.3.4.5.1.4.2 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 del contexto de PTw@ndA. 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 /PTw@ndA* ptw
```

3.3.4.5.1.4.3 Configuración de los nodos JBoss

Todos los servidores de aplicación JBoss con PTw@ndA 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, por lo que 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 pertenece. 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:

```
Nodo 1: -Djboss.partition.name=ptw -Djboss.messaging.ServerPeerID=1
```

```
Nodo 2: -Djboss.partition.name=ptw -Djboss.messaging.ServerPeerID=2
```

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 PTw@ndA pertenecientes a diversos entornos, siempre que se les asignen distintos nombres de partición: ptwDesa, ptwPru, ptwPro, ...

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

```
${JBOSS_HOME}/server/all/deploy/jbossweb.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="nodoX">  
...  
</Engine>
```

Puede obtener información adicional en el siguiente enlace de la página oficial de JBOSS que explica en detalle la solución relativa a clustering de aplicaciones (JBoss AS 5.1 Clustering Guide. High Availability Enterprise Services with JBoss Application Server Clusters):



Enlace: <https://www.jboss.org/community/docs/DOC-12928>

3.3.4.6 Configuración de trazas mediante Log4j

Para configurar la generación de mensajes de información, avisos y error del sistema mediante el sistema Log4j de Apache, debe editar el archivo `log4j.properties` que se encuentra dentro de la carpeta `[APLICACIÓN] \ WEB - INF \ CLASSES \`.

Se considera que la configuración distribuida por defecto es válida para la mayoría de los entornos y no son requeridas tareas de configuración. En caso de que la configuración por defecto no satisfaga las necesidades de una instalación particular, este fichero puede ser adaptado, facilitándose a continuación un enlace a un manual de configuración del fichero `log4j.properties`.



Enlace: <http://logging.apache.org/log4j/1.2/manual.html>

3.4 Instalación de Módulos Funcionales. Cambio de Versión.

Se ha previsto un mecanismo para simplificar las actualizaciones a futuras versiones de PTw@ndA. Así, para las nuevas versiones no será necesario volver a desplegar todos los componentes verticales instalados en versiones o implantaciones previas, sino que éstos se descargarán automáticamente en la nueva implantación desde el esquema de BBDD de PTw@ndA preexistente.

Para que esta funcionalidad se lleve a cabo es necesario tener configurados 2 propiedades de la aplicación (más información en el apartado 3.3.4.4):

DESCARGAR_MODULOS = true.

APLICACION_SINCRONISMO_TRAMITADOR = true.

Una vez configurado ésto, la primera vez que arranque la nueva implantación, una vez desplegado el WAR y configurados los DataSource, se accederá a BBDD, comprobándose los módulos instalados previamente e instalándose aquellos que no se encuentren.

También se ha implementado un control de versiones de tal forma que a la hora de desplegar PTw@ndA, si los módulos que se encuentran en BBDD no coinciden en nombre y tamaño con los desplegados se sustituirán por los existentes en BBDD.



Cada vez que se actualice o instale un módulo, será necesario reiniciar el servidor de aplicaciones para que los cambios tengan efecto y no quede inestable el contexto de la aplicación.

Esta funcionalidad es también útil para el despliegue de PT en situaciones de granjas de servidores en la que los nuevos nodos a incorporar desplegarán de modo automatizado todos los módulos verticales disponibles para el resto de los nodos que comparten el mismo esquema de base de datos de PT.

3.5 Definición de Menús.

En esta nueva versión de PTw@ndA, se ha incluido una facilidad de mantenimiento de menús, mediante la cual definir un menú/submenú que permitirá gestionar los accesos mostrados desde la pantalla principal.

Tras una nueva instalación de la versión 2.0 de PTw@ndA, no se mostrará ningún menú en la pantalla principal, por lo que será necesaria su definición tras el correcto despliegue de la aplicación.



Para más información referente al mantenimiento de Menús, consultar el punto “3.3 Mantenimiento de Menús” y “3.4 Mantenimiento de Instalaciones” del *Manual de Administración de PTw@ndA*.

4 REFERENCIAS

Referencia	Documento
[1]	Manual de Instalación de Trew@
[2]	Manual de Instalación de @firma
[3]	Manual de Instalación de Port@firmas
[4]	Manual de Instalación de Avis@dor
[5]	Manual de Instalación de Solicit@
[6]	Manual de Administración de PTw@ndA

5 ENLACES

Referencia	Enlace
Hibernate	http://www.hibernate.org/80.html
Soporte a la Administración Electrónica Junta de Andalucía	https://ws024.juntadeandalucia.es/
MADEJA	http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/
@firma	https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/afirma.jsp
@ries	https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/aries.jsp
Notific@	https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/notifica.jsp
Port@firmas	https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/portafirmas.jsp
Solicit@	https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/solicita.jsp
Trew@	https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/trewa.jsp
OpenOffice	http://es.openoffice.org/
JBOSS Instalación	https://www.jboss.org/community/docs/DOC-12923
JBOSS Administración	https://www.jboss.org/community/docs/DOC-12927
JBOSS Clustering	https://www.jboss.org/community/docs/DOC-12928
Struts	http://struts.apache.org
Solr	http://lucene.apache.org/solr/
Log4J	http://logging.apache.org/log4j/1.2/manual.html