



Consejería de Justicia y Administración Pública



Trew@ v1.5.0

Manual de operación

Versión: v01r00

Fecha: 04/09/2009

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.

 <p>JUNTA DE ANDALUCÍA</p>	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	 <p>Guadaltel</p>
---	---	--	--

HOJA DE CONTROL

Título	Trew@ v1.5.0		
Entregable	Manual de operación		
Nombre del Fichero	TRW001E_MIC_Manual de operación Trew@ v150_0100.doc		
Autor	Guadaltel, s.a.		
Versión/Edición	v01r00	Fecha Versión	04/09/2009
Aprobado por		Fecha Aprobación	-
		Nº Total Páginas	046

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio
v01r00	Nuevo entregable del producto	Guadaltel, s.a.	Trew@ v1.5.0	04/09/2009



CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Área	Nº Copias
Manuel Perera Domínguez	Jefe de Servicio	Servicio de Coordinación de Admón. Electrónica	1
Ignacio Rodríguez García		Servicio de Coordinación de Admón. Electrónica	1
Enrique Baleriola Salvo		Servicio de Coordinación de Admón. Electrónica	1
Francisco Mesa Villalba		Servicio de Coordinación de Admón. Electrónica	1
Manuel Escobar Montes		Servicio de Coordinación de Admón. Electrónica	1

ÍNDICE

1	INTRODUCCIÓN	6
1.1	Objeto.....	6
1.2	Alcance.....	6
2	Manual de operación, instalación y configuración	7
2.1	Instalación del motor Trew@	7
2.1.1	Pasos a seguir para instalar el motor v1.5.0 completo	7
2.1.2	Actualización desde un motor v1.3.X o v1.4.X existente (Oracle)	8
2.1.3	Instalación de un motor Trew@ v1.5.0 en PostgreSQL	8
2.1.4	Revisión de objetos y datos en el motor	9
2.1.4.1	Tablas necesarias	9
2.1.4.2	Secuencias	9
2.1.4.3	Constantes generales (tabla "GN_CONSTANTES").....	10
2.1.4.4	Datos de Sistemas por defecto	10
2.1.4.5	Datos de Componentes por defecto.....	10
2.1.4.6	Configuración de los sistemas (tabla "TR_CONSTANTES").....	11
2.1.4.7	Generación de documentos mediante Open Office	12
2.1.4.8	Generación de documentos a pdf mediante itext.....	12
2.1.4.9	Perfiles de usuario	12
2.1.5	Instalación de las pantallas de administración (J2EE).....	12
2.1.5.1	General.....	12
2.1.5.2	Actualización desde versiones anteriores	13
2.1.6	Creación de otros usuarios administradores distintos al propietario	13
2.1.7	Requerimientos para las aplicaciones "cliente" de Trew@.....	14
2.1.8	Despliegue y configuración de las J-TrAPIs.....	15
2.1.8.1	Despliegue de las J-TrAPIs	15
2.1.8.2	Despliegue en Tomcat.....	15
2.1.8.3	Despliegue en Jboss	15
2.1.8.4	Configuración de servlets	15
2.1.8.5	Configuración de perfiles de conexión.....	19
2.1.8.6	Configuración con archivo de properties	19
2.1.8.7	Formato del fichero de properties	20

2.1.8.8	Configuración con DataSources	23
2.1.8.9	Librerías necesarias	34
2.1.8.10	Trazabilidad con log4java	37
2.1.8.11	Configuración para WebOffice	39
2.1.8.12	Montaje de los Servicios Web de Trew@	40
2.1.8.13	Condiciones, acciones y variables Java	41
2.2	Mantenimiento del Motor	45
2.2.1	Base de datos	45
2.2.2	Pantallas de Administración de Trew@	45
2.2.3	Condiciones y acciones	45
2.2.4	Variables de los documentos	45
3	ANEXOS	46
4	GLOSARIO	46
5	BIBLIOGRAFÍA Y REFERENCIAS	46

 <p>JUNTA DE ANDALUCÍA</p>	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	 <p>Guadatel</p>
---	---	--	---

1 INTRODUCCIÓN

1.1 Objeto



El presente documento recoge los manuales de operación, instalación y datos de configuración de diversos componentes pertenecientes a un motor Trew@ v1.5.0.

1.2 Alcance

Este documento va dirigido a:

La dirección del proyecto w@ndA.

El colectivo de usuarios y desarrolladores de sistemas que usan Trew@ como motor de tramitación.

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

2 Manual de operación, instalación y configuración

2.1 Instalación del motor Trew@

En los siguientes apartados se describe cómo debe instalarse o actualizarse un motor Trew@ (en Oracle) a esta nueva versión v1.5.0 a partir de los ficheros ".dmp" y scripts suministrados.

Asimismo se indica dónde se puede encontrar la documentación necesaria para la instalación de Trew@ v1.5.0 sobre un motor de base de datos PostgreSQL.

2.1.1 Pasos a seguir para instalar el motor v1.5.0 completo

Creemos los tablespaces necesarios para los objetos de Trew@. Concretamente:

TS_TREWA → tablespace para datos, al menos de 100 Mb.

TS_TREWA_I → tablespace para índices, al menos de 500 Mb.

TS_TREWA_BLOB → tablespace para almacenar las columnas tipo BLOB, al menos de 20 Mb.

NOTA: Por ejemplo, para unos 100 procedimientos definidos, con unas 2000 fases posibles, unas 3500 transiciones entre fases, con 2000 expedientes, y unos 1000 documentos almacenados en blob, una estimación para el tamaño de los tablespaces sería:

TS_TREWA → sobre 700 Mb.

TS_TREWA_I → 1 Gb.

TS_TREWA_BLOB → 1 Gb.

Es decir, aquí se indica el mínimo necesario para la instalación, pero el administrador de base de datos debe tener en cuenta el crecimiento de estos tablespaces conforme se vayan incluyendo datos en la base de datos.



Creemos el usuario que va a ser propietario de los objetos, por ejemplo "TREWA", de aquí en adelante consideramos que se llame así (se suministran scripts de ejemplo).

Damos los permisos correspondientes al usuario, que serán (siempre que sea posible) al menos:

"alter user TREWA quota unlimited on TS_TREWA" (también en TS_TREWA_I y TS_TREWA_BLOB).

"grant connect to TREWA" (o en su defecto create y alter session).

"grant create table, create view, create synonym, create procedure, create role, create synonym, create sequence, create trigger to TREWA" (NOTA: téngase en cuenta que en el role de CONNECT van implícitos: create view, table, sequence, etc.).

 <p>JUNTA DE ANDALUCÍA</p>	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	 <p>Guadatel</p>
---	---	--	---

Opcionalmente (sí se decide en la organización en la que nos encontremos) crear los roles de Trew@ “TR_R_USUARIO” y “TR_R_ADMINISTRADOR”. Estos roles sólo se necesitan en caso de usar la comprobación de permisos sobre apis a través de roles de base de datos. Por defecto dicha comprobación se hace mediante los perfiles que el usuario tenga asignados en el modelo de datos de Trew@, es por ello que este apartado es opcional y no necesario (ver para más detalles el documento de “Notas sobre creación de usuarios”).

Lanzamos el import del fichero “trewa_v150_20090624.dmp” suministrado (subcarpeta “Base_de_datos\ Oracle\ fichero export limpio”). Hay que tener en cuenta que el fichero “.dmp” es un export de un esquema “TREWA_TST”, por lo que para la correcta creación de los objetos habrá que incluir en la instrucción del import algo así como “...fromuser=TREWA_TST touser=USUARIO_PROPIETARIO...”. Por ejemplo:

```
Imp userid=propietarioTrewa/clave@bd file=trewa_v150_20090624.dmp log=trewa_v150_imp.log buffer=10485760
fromuser=trewa_tst touser= TREWA
```

Comprobaríamos a continuación la correcta creación de los objetos (ver apartado 2.1.4).

En el caso que el usuario propietario (en este caso “TREWA” si así lo hemos creado) vaya a ser administrador de todos los sistemas deberemos hacer (bien así, bien modificando los datos directamente mediante alguna herramienta en las tablas correspondientes):

```
UPDATE GN_CONSTANTES SET V_CONSTANTE = 'TREWA' WHERE C_CONSTANTE ='TRUSUPROP';
```

```
UPDATE GN_USUARIOS SET C_USUARIO = 'TREWA' WHERE C_USUARIO ='TREWA_TST';
```

Si además no hemos creado o no hemos dado los roles del paso 4 al usuario propietario, para que podamos conectarnos como administrador desde las pantallas de administración de Trew@, debemos hacer:

```
INSERT INTO TR_USUARIOS_X_PERFILES_USUARIO (USUA_C_USUARIO, PEUS_X_PEUS) VALUES ('TREWA', 1);
```

Con esto damos al usuario “TREWA” (esquema propietario de los objetos), el perfil de usuario “TR_R_ADMINISTRADOR” sin necesidad de usar roles de base de datos.

2.1.2 Actualización desde un motor v1.3.X o v1.4.X existente (Oracle)

Lanzar el script “trewa_v13Xv14X_a_v150.sql” existente en la carpeta “scripts BD y datos (adaptación desde v13Xv14X)” suministrado.

Si procedemos de una versión 1.3.0 deberemos lanzar además los scripts “indices.drp” y “TR_V_SIM.sql” (no es necesario ejecutarlos si estamos actualizando un motor Trew@ 1.3.1 o superior). Además si no estamos usando las apis pl/sql o las pantallas de administración con Forms de Oracle o los reports de Oracle de generación de documentos, se aconseja eliminar los paquetes pl/sql y la función “GN_FU_VALOR” que existirán en el esquema propietario de los objetos.

2.1.3 Instalación de un motor Trew@ v1.5.0 en PostgreSQL

Para la instalación de Trew@ v1.5.0 sobre un motor de base de datos PostgreSQL, se remite al lector al documento de instalación específico para esta base de datos (“Manual de operación Trew@ v150 PostgreSQL”) que puede encontrarse en el entregable de la versión v1.5.0 en la subcarpeta “Documentación” (los script necesarios se encuentran en “Base_de_datos\PostgreSQL”).

2.1.4 Revisión de objetos y datos en el motor

Una vez hemos importado los objetos de Trew@ satisfactoriamente, podemos verificar si se han creado los objetos. Para ellos se puede ejecutar el siguiente select:

```
SELECT COUNT(*),STATUS,OBJECT_TYPE
FROM USER_OBJECTS GROUP BY STATUS,OBJECT_TYPE
ORDER BY COUNT(*) DESC;
```

El resultado de esta select debería ser:

COUNT(*)	STATUS	OBJECT_TYPE
324	VALID	INDEX
102	VALID	TRIGGER
98	VALID	VIEW
98	VALID	TABLE
69	VALID	SEQUENCE
14	VALID	LOB

2.1.4.1 Tablas necesarias



Podemos distinguir 2 tipos de tablas según su nombre:

Las que comienzan por "TR_", constituyen el "metamodelo" en el que se apoya Trew@ y debería estudiarse cualquier modificación en las mismas, ya que afectan a otros objetos necesarios para el correcto funcionamiento.

Las que comienzan por "GN_", que serían un conjunto de tablas necesarias para que Trew@ funcione, pero que no se consideran parte integrante del "metamodelo" de Trew@, sino que más bien se consideran tablas "generales" a todas las aplicaciones. Por ejemplo, se encuentran entre ellas aquellas tablas que guardan la estructura organizativa (Unidades Orgánicas de la Junta en nuestro caso), las que guardan los usuarios que utilizan Trew@, etc.

2.1.4.2 Secuencias

Secuencias necesarias para el buen funcionamiento de las pantallas de mantenimiento de Trew@ (los nombres de las mismas comienzan por "TR_S..."). También deben existir secuencias para las tablas generales (los nombres de las mismas comienzan por "GN_S...").

 <p>JUNTA DE ANDALUCÍA</p>	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	 <p>Guadatel</p>
---	---	--	---

2.1.4.3 Constantes generales (tabla "GN_CONSTANTES")

Debido a la desaparición desde la versión v1.4.0 del componente TRPQApi (api pl/sql de Trew@) necesarias además para las pantallas de administración con Forms de Oracle, los valores de la tabla de constantes generales "GN_CONSTANTES" se ha visto reducida en cuanto al número de valores, que cómo mínimo debe contener los siguientes:

"TRVERS" → VERSIÓN DE LAS PANTALLAS DE ADMINISTRACIÓN DE TREW@, que pondremos con el valor por defecto de "1.5.0". Este valor habrá que ir modificando en base según las actualizaciones que vayamos haciendo de los ejecutables de la aplicación.

"TRSTMA" → SISTEMA POR DEFECTO PARA INICIALIZACIÓN DEL MOTOR, que pondremos con el valor por defecto de "TREW@", que es el sistema creado por defecto al instalar Trew@. Este valor permite que al instanciarse la API se inicialice el motor con las constantes por defecto.

"TRUSUPROP" → USUARIO PROPIETARIO DE LOS OBJETOS DE TREW@, que debe contener como valor el nombre del esquema propietario de los objetos de Trew@.

En caso de necesitar el uso del componente PQTrApi o las pantallas de administración con Forms de Oracle o los reports Oracle de generación de documentos de versiones anteriores, se remite al lector a la documentación correspondiente a la versión de dichos componentes y los datos necesarios que deben existir en esta tabla.

2.1.4.4 Datos de Sistemas por defecto



Debe existir al menos un sistema (por defecto "TREW@") en la tabla "GN_SISTEMAS" para poder iniciar el motor con los valores por defecto. El nombre de este sistema se debe corresponder con la constante "TRSTMA" de la tabla "GN_CONSTANTES" descrita en el apartado anterior. Se recomienda que este sistema por defecto haga referencia al propio motor de tramitación, ya que así se define cierta información que por defecto es necesaria para poder trabajar con el motor, así como otros datos de los que pueden ser compartidos por otros sistemas por el hecho de estar asociada a este sistema por defecto.

2.1.4.5 Datos de Componentes por defecto

Debe existir al menos un componente (por defecto "TREW@") en la tabla "TR_COMPONENTES" para poder utilizar ciertas APIs orientadas al funcionamiento de Trew@ dentro de un entorno w@ndA, necesario para poder interactuar con otros componentes w@ndA. Esta información contendrá los datos del componente de tramitación que representa el Trew@ instalado así como otros datos de configuración contenidos en la tabla "TR_DATOS_COMPONENTE" necesarios para la integración con otros componentes. El nombre de este valor puede ser en principio cualquiera, y será etiquetado en w@ndA como un componente de tramitación más con un "código de identificación" en un entorno w@ndA.

De la misma forma en esta versión se incluyen datos por defecto de un componente "BUS" que representa el bus de comunicación en un entorno w@ndA y que tendremos que configurar para poder hacer uso de él (para más detalles sobre el funcionamiento en un entorno w@ndA se remite al lector al documento "Guía de integración w@ndA").

En la tabla "TR_CONSTANTES" asociado al sistema por defecto debe existir un valor "MOTOR_TRAMITA" que representa al nombre del componente de tramitación por defecto definido con la que ciertas APIs que ofrece Trew@ puede obtener la información del componente de tramitación definido por defecto, necesaria por ejemplo en el XML de exportación de expedientes.

 <p>JUNTA DE ANDALUCÍA</p>	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	 <p>Guadatel</p>
---	---	--	---

Además del componente por defecto también existirán como datos creados durante la instalación, otros componentes (así como datos asociados al mismo) tales como "W@RDA", para el caso en el que se quiera interactuar con dicho componente, o "WEBOFFICE", para el caso en el que se prefiera el tratamiento de documentos mediante Open Office (en vez de utilizar Report de Oracle) que ofrece desde la versión v1.0.0 de Trew@.

2.1.4.6 Configuración de los sistemas (tabla "TR_CONSTANTES")

Como se ha comentado anteriormente desde la versión v.1.4.0 dejan de existir los componente pl/sql de Trew@, por lo que la tabla que contiene datos de la configuración de sistemas se ha visto reducida en cuanto a los valores que se pueden usar. Por tanto, debe revisarse en este caso si existen las siguientes constantes asignadas al sistema por defecto (típicamente "TREW@", ver párrafo anterior):

"MOTOR_TRAMITA" → MOTOR DE TRAMITACIÓN. Por defecto a "TREW@" (ver apartado anterior).



"FORMATO_FECHA" → FORMATO DE FECHA UTILIZADO POR EL MOTOR, por ejemplo con el valor "DD/MM/YYYY". Este formato será el que utilicen todas las APIs de Trew@ en el caso de trabajar con fechas.

"COMP_PERMISOS_DOC" → PERMITE CONFIGURAR TREW@ PARA HABILITAR O NO LA COMPROBACIÓN DE LOS PERMISOS SOBRE DOCUMENTOS QUE UN USUARIO TIENE EN UNA DETERMINADA FASE DEL EXPEDIENTE, por defecto con el valor "N". Si ponemos esta constante como "S", Trew@ verificará que el usuario tenga el permiso sobre tareas de manipulación de escritos exactamente igual a como se haya definido en el procedimiento además de el perfil de tramitación adecuado. Es decir, si por ejemplo sólo se ha asignado a este tipo de tareas permisos de "Editar", las APIs destinadas a la edición de los párrafos o datos de documentos permitirán realizar la edición del mismo sólo en el caso de que el usuario tenga el perfil de tramitación al que se ha asignado este permiso, además de que no se permitirán llamadas a otras APIs que no sean para editar el documento (por ejemplo no podría generar documentos aunque sí editarlos). De esta forma un usuario que no tenga este perfil no podrá editar los datos del documento aunque tenga otros perfiles con otros permisos sobre el documento (por ejemplo de Firma o generación). En el caso de "N" Trew@ sólo verifica que el documento esté permitido al perfil de tramitación del usuario independientemente del permiso concreto asignado en la definición del procedimiento, todos los usuarios con el perfil adecuado tienen el mismo permiso sobre el documento, sin distinguir si es para "genera", "editar", "firmar", etc..

"URL_REPORT_JAVA" → GUARDA LA URL DONDE SE ENCUENTRAN LOS FICHEROS DE CONFIGURACIÓN PARA LA GENERACIÓN DE DOCUMENTOS JAVA PDF, por ejemplo "http://nombreMaquina:puerto/agendaTrewa/editor/config/". Usada por las Apis java generación de documentos .pdf mediante itext para localizar las plantillas .xml que formatean el documento.

"REPOSITORIO_DOC_EXT" → PERMITE MAYOR CONFIGURACIÓN DE UN REPOSITORIO EXTERNO DE DOCUMENTOS. Por defecto a "N", un valor a "S" indica a la api de Trew@ la eliminación de los documentos físicos almacenados en blobs en el modelo de datos del motor, siempre que se esté usando un gestor documental externo para el sistema como w@rdA o Alfresco.

NOTA: Aunque estas constantes deben existir por defecto asociadas al sistema "Trew@", se aconseja que cada sistema nuevo defina las suyas propias (concretamente todas excepto la 1ª) para así disponer de un "juego de configuración" concreto e (independiente del resto) para dicho sistema.

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

2.1.4.7 Generación de documentos mediante Open Office

Si se desea esta nueva funcionalidad disponible desde Trew@ v1.0.0, deben ajustarse los datos necesarios del componente "WEBOFFICE". Para más detalle sobre la instalación de los componentes necesarios para la utilización de esta funcionalidad ver la documentación relativa al componente WebOffice.

2.1.4.8 Generación de documentos a pdf mediante itext

Si se desea esta nueva funcionalidad disponible desde Trew@ v1.3.0, debe ajustarse para el sistema la constante "URL_REPORT_JAVA" que debe contener la ruta donde se encuentren los ficheros xml que contienen los datos para el formato a pdf de los documentos basados en párrafos, o lo que es lo mismo las "plantillas" para este tipo de generación.

De la misma forma, si se va a utilizar la generación de documentos que ofrece Trew@, la tabla "TR_TIPOS_PARRAFOS" debe contener los datos necesarios para la correcta visualización de los documentos generados. Estos datos permiten el formateo del documento, el estilo de los párrafos que componen el documento.

2.1.4.9 Perfiles de usuario

Deben existir por defecto 2 perfiles de usuarios asignados al sistema "TREW@" denominados "TR_R_USUARIO" y "TR_R_ADMINISTRADOR", necesarios en el caso de utilizar Trew@ con aplicaciones basadas en usuarios que no son de base de datos (aplicaciones que utilizan los mismos datos de conexión independientemente del usuario de la aplicación). Estos perfiles se darán a los usuarios definidos en el sistema a modo de perfiles de tramitación desde la pantalla de administración habilitada al efecto. Si un usuario que no es de base de datos no tiene alguno de estos perfiles no podrá ejecutar las APIs Trew@. Desde la versión v1.1.0 este mecanismo para dar permisos puede usarse también para usuarios "reales" de base de datos como alternativa a los roles de base de datos.

2.1.5 Instalación de las pantallas de administración (J2EE)

2.1.5.1 General

Para la instalación de este componente disponible desde la versión v1.1.0 de Trew@, es necesario disponer de un servidor de aplicaciones en el que desplegar el fichero "TrewaAdm.war" suministrado en la subcarpeta "Administración\J2EE". Estas pantallas hacen uso de las JTrApis de administración por lo que para más detalle se remite al lector al apartado "*Despliegue y configuración de las J-TrAPIs*" que más adelante se puede encontrar en este documento, sobre todo en lo que se refiere a "perfiles de conexión".

La ventana de conexión de esta aplicación permite obtener una lista de perfiles de conexión definidos previamente si se desea. Para que los perfiles nos aparezcan en la lista de perfiles debemos darlos de alta en el fichero *web.xml* de nuestra aplicación web. Para ello añadimos un parámetro del contexto cuyo nombre debe comenzar con el sufijo "TrPerfil_"; por ejemplo "TrPerfil_Default".

```

<!-- Perfiles para crear JTrAPIs Trew@ -->
<context-param>
  <param-name>TrPerfil_Default</param-name>
  <param-value>default</param-value>
  <description>Perfil por defecto para realizar la conexión</description>
</context-param>

<context-param>
  <param-name>TrPerfil_Postgresql</param-name>
  <param-value>postgresql</param-value>
  <description>Perfil para realizar la conexión a una base de datos postgresql</description>
</context-param>

<!-- Ejemplo de perfil usando un pool de conexiones en Tomcat -->
<context-param>
  <param-name>TrPerfil_PoolTomcat</param-name>
  <param-value>java:comp/env/jdbc/nombreDatasource</param-value>
  <description>Perfil para realizar la conexión mediante pool de tomcat</description>
</context-param>

```

Si se desea fijar uno de los perfiles de conexión basta con indicarlo en la url de arranque de la aplicación una vez desplegada (por ejemplo "<http://maquina.puerto/TrewaAdm/?perfil=perfilTrewa>").

El fichero "web.xml" de esta aplicación contiene otras entradas que permiten configurar y actuar sobre ciertos aspectos de funcionamiento, tales como la devolución de conexiones al pool cuando se usan estos, administración del sistema por defecto o la habilitación a otros usuarios administradores del forzado para la funcionalidad de eliminación de expedientes (ver para más detalles el contenido de web.xml).

2.1.5.2 Actualización desde versiones anteriores



Para actualizar este componente desde versiones anteriores ya instaladas, además de sustituir el fichero "trewa.jar" suministrado, se deben sustituir los ficheros contenidos en las distintas subcarpetas (según la versión anterior) que pueden encontrarse en "Administración\J2EE". También puede desplegarse directamente el .war facilitado para lo que se recomienda la copia previa del actual fichero web.xml y los posibles .properties de conexión definidos en la versión previamente instalada para poder actualizar y configurar los mismos en la nueva versión desplegada.

2.1.6 Creación de otros usuarios administradores distintos al propietario

Las pantallas de mantenimiento de Trew@, permiten desde la versión v1.0.1 utilizar distintos usuarios administradores, típicamente para separar la administración de los distintos sistemas que hagan uso del motor de tramitación. Esto permite que cada aplicación que use Trew@ pueda permitirse tener un usuario de conexión distinto al propietario y tener así organizadas las conexiones y mantenimiento de dichos sistemas. Para más detalle sobre la creación de usuarios para uso de Trew@ se remite al lector al documento "Notas sobre creación de usuarios.pdf".

Para crear estos usuarios administradores se pueden seguir los siguientes pasos:

Crear el usuario de base de datos (no más de 10 caracteres, pues debe darse de alta posteriormente en la tabla de usuarios de Trew@). Dar los permisos de "connect" y "create synonym" mínimo. Además debe darse permiso de "select" en la tabla "SYS.DBA_ROLE_PRIVS" o en su defecto asociarle posteriormente mediante las pantallas de administración uno de los perfiles de usuario del sistema por defecto "TREW@". Para la creación de este usuario se puede utilizar el script "Sentencia

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

creación usuarios BD Trew@.sql" suministrado en la subcarpeta "Base_de_datos/scripts usuarios BD" desde la versión v1.0.1.

Desde una conexión con el usuario propietario de los objetos de Trew@, lanzar el script "Permisos a usuario BD de Trew@ v150.sql" suministrado en la subcarpeta anteriormente mencionada, sustituyendo en dichos scripts los valores de los usuarios por los reales.

Desde una conexión con el usuario administrador de Trew@ creado, lanzar el script "Sinónimos de usuarios BD de Trew@ v150.sql" suministrado en la subcarpeta anteriormente mencionada, sustituyendo en dichos scripts los valores de los usuarios por los reales.

Si se desea, también se suministran los anteriores scripts en el caso de usar las vistas filtradas de Trew@ en la subcarpeta "scripts usuarios BD (vistas filtradas por sistema)".

Lanzar la aplicación conectándose con el usuario propietario de los objetos de Trew@. Entrar en la pantalla de mantenimiento "Usuarios del Tramitador" bajo el menú "Usuarios". Dar de alta el usuario administrador creado y asignarle los perfiles de usuario sobre los sistemas que vaya a administrar. Si los permisos se van a dar a través de roles en vez de perfiles de usuario, es necesario también (si así se desea) indicar el valor de la opción "Permiso" con el valor de "Administrador" (para esto es necesario que el propietario Trew@ tenga los roles Trew@ asignados con "ADMIN OPTION").

Hacer doble-click sobre el campo "Usuario" para abrir la funcionalidad de "Test de clave". Introducir la clave deseada para conexión con las pantallas de administración y se calculará la clave que debe actualizarse al usuario de base de datos. Una vez obtenida actualizar dicha clave de base de datos.

A partir de entonces el usuario administrador creado podrá entrar con las pantallas de administración de Trew@, pero sólo podrá administrar los sistemas en los que se haya dado permiso. Además este usuario podría utilizarse como conexión de las aplicaciones J2EE que hagan uso de las J-TrApis, para lo que deberán cambiar los datos de conexión (Datasource del servidor de aplicaciones o datos del fichero ".properties" que se utilice como "perfil" de conexión) utilizando la clave "encriptada" obtenida en la pantalla de usuarios.

2.1.7 Requerimientos para las aplicaciones "cliente" de Trew@



Todas las aplicaciones que usen Trew@ deben estar dadas de alta en la tabla "GN_SISTEMAS".

Para poder ejecutar las APIs de Trew@ los usuarios de cada aplicación deben estar dados de alta en la tabla "GN_USUARIOS" y tener asignado uno de los roles de Trew@ (bien el de usuario o bien el de administrador) ya sea directamente a través de las pantallas de administración (Mantenimiento de usuarios) o bien a través de un role de aplicación que ya contenga el role de Trew@. Como alternativa a los roles y a partir de la versión v1.1.0 existe la posibilidad de dar los permisos a través de perfiles de usuario del sistema por defecto "TREW@".

Si trabajamos con una aplicación basada en usuarios que no son de base de datos, no podremos asignar a un usuario concreto el role de base de datos correspondiente. En este caso es necesario asociar al usuario el perfil de usuario correspondiente (TR_R_USUARIO o TR_R_ADMINISTRADOR) como si de un perfil de tramitación se tratara.

Como ya se ha citado anteriormente, también se puede establecer para cada aplicación un "juego" de constantes. Si no existen se tomarán las definidas en el sistema por defecto.

Además cada aplicación "cliente" de Trew@ debe dar de alta una serie de datos si se utiliza la generación de documentos mediante reports de Oracle. Como se ha dicho anteriormente se suministran ficheros de scripts para crear estos datos (ver el apartado "Generación de documentos mediante reports de Oracle").

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

2.1.8 Despliegue y configuración de las J-TrAPIs

2.1.8.1 Despliegue de las J-TrAPIs

Para que las J-TrAPIs puedan estar accesibles desde diferentes aplicaciones, deben estar desplegadas en el mismo servidor de aplicaciones donde dichas aplicaciones residan. El despliegue de las mismas variará en función del servidor de aplicaciones donde se vaya a realizar y en todo caso también se tendrá que realizar una configuración para habilitar el acceso a los servlets.

2.1.8.2 Despliegue en Tomcat

Incluir la librería **trewa.jar** en el directorio **%Tomcat%\common\lib**, o desplegar de forma completa las clases en el directorio **%Tomcat%\common\classes**.

2.1.8.3 Despliegue en Jboss

Incluir la librería **trewa.jar** en el directorio **%JBOSS%\server\default\lib** para que todas las aplicaciones desplegadas a nivel default puedan acceder a ella.

2.1.8.4 Configuración de servlets

Para que una aplicación pueda acceder a los servlets de subida y descarga de documentos (TrAPIUI), para los de subida y descarga de XML (TrAPIUTL) y para los de subida de documento WebOffice y descarga de XML WebOffice, debe existir un paso previo de configuración a nivel del servidor de aplicaciones. Básicamente la configuración consistirá en modificar el archivo "web.xml" del servidor para registrar la información de los servlets.

Se editará el archivo web.xml y en la primera parte "Built In Servlet Definitions" definiremos los servlets de la siguiente forma:



```
<servlet>

  <servlet-name>TrAPIUIDescarga</servlet-name>
  <servlet-class>trewa.bd.trapi.trapiui.servlet.DescargaDocumento</servlet-class>
</servlet>

<servlet>

  <servlet-name>TrAPIUISubida</servlet-name>
  <servlet-class>trewa.bd.trapi.trapiui.servlet.SubidaDocumento</servlet-class>
</servlet>

<servlet>

  <servlet-name>TrAPIUTLDescarga</servlet-name>
  <servlet-class>trewa.bd.trapi.trapiutl.servlet.DescargaXML</servlet-class>
</servlet>

<servlet>

  <servlet-name>TrAPIUTLSubida</servlet-name>
  <servlet-class>trewa.bd.trapi.trapiutl.servlet.SubidaXML</servlet-class>
</servlet>

<servlet>

  <servlet-name>TrAPIUTLSubidaZip</servlet-name>
  <servlet-class>trewa.bd.trapi.trapiutl.servlet.SubidaXMLzip</servlet-class>
</servlet>

<!-- Servlets para WebOffice -->

<servlet>
```



```

<servlet-name> TrAPIUISubidaOO</servlet-name>

<servlet-class>trewa.bd.trapi.trapiui.servlet.OOSubidaDocumento</servlet-class>

</servlet>

<servlet>

<servlet-name> TrAPIUIDescargaXMLOO</servlet-name>

<servlet-class>trewa.bd.trapi.trapiui.servlet.OODescargaXML</servlet-class>

</servlet>

```

Además se debe añadir información referente a desde dónde se puede acceder a los servlets. Esta información se denomina “Servlet Mappings”, y para configurarla se incluirán las siguientes líneas en el apartado “Built In Servlet Mappings”:

```

<servlet-mapping>

<servlet-name> TrAPIUIDescarga </servlet-name>

<url-pattern> / trewa / DescargaDoc </url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name> TrAPIUISubida </servlet-name>

<url-pattern> / trewa / SubidaDoc </url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name> TrAPIUTLDescarga </servlet-name>

```



```
<url-pattern>/trewa/DescargaXML</url-pattern>  
  
</servlet-mapping>  
  
<servlet-mapping>  
  
  <servlet-name>TrAPIUTLSubida</servlet-name>  
  
  <url-pattern>/trewa/SubidaXML</url-pattern>  
  
</servlet-mapping>  
  
<servlet-mapping>  
  
  <servlet-name>TrAPIUTLSubidaZip</servlet-name>  
  
  <url-pattern>/trewa/SubidaXMLzip</url-pattern>  
  
</servlet-mapping>  
  
<servlet-mapping>  
  
  <servlet-name>TrAPIUISubidaOO</servlet-name>  
  
  <url-pattern>/trewa/OOSubidaDoc/*</url-pattern>  
  
</servlet-mapping>  
  
<servlet-mapping>  
  
  <servlet-name>TrAPIUIDescargaXMLOO</servlet-name>  
  
  <url-pattern>/trewa/OODescargaXML/*</url-pattern>  
  
</servlet-mapping>
```

La ubicación por defecto del archivo “**web.xml**” variará en función del servidor de aplicaciones:

- En **Tomcat** el archivo “web.xml” se encuentra en la carpeta **%Tomcat%\conf** .

- En **JBoss** el archivo “web.xml” se encuentra en la carpeta **%JBoss%\server\default\deploy\jbossweb-tomcat50.sar** .

Nota: Estas ubicaciones son a modo orientativo, en todo caso se recomienda recurrir a la documentación técnica del servidor de aplicaciones en uso para garantizar una correcta instalación de las J-TrAPIs.

2.1.8.5 Configuración de perfiles de conexión

Un perfil de conexión para las J-TrAPIs no es más que una configuración de la conexión a una base de datos.

En la actualidad las J-TrAPIs permiten dos posibles usos de perfil de conexión:

- Configuración básica con archivo de properties
- Configuración con DataSources a nivel del servidor de aplicaciones

A nivel de aplicación el uso de un perfil u otro va a ser totalmente transparente y no se va a a conocer exactamente donde se establece la conexión. Lo único que van a necesitar las aplicaciones será el **nombre del perfil** que deberá ser aportado por el administrador del servidor de aplicaciones donde se desplieguen las J-TrAPIs.

2.1.8.6 Configuración con archivo de properties

En este caso el nombre del perfil va a coincidir con el nombre del archivo de properties donde se realice la configuración. Así por ejemplo si el administrador crea un archivo **trOracle.properties**, el nombre del perfil será **trOracle**.

Posibles ubicaciones para el archivo de perfiles

Los perfiles deben ubicarse en el paquete **trewa.conf.perfiles** para que las J-TrAPIs los puedan localizar. En esa carpeta debe existir configurado por lo menos un perfil **default** (default.properties) que es el que intentarán usar las J-TrAPIs cuando no se le indique perfil alguno.

En todo caso será conveniente analizar varias posibles situaciones para asegurarse de que el perfil sea detectado por las TrAPIs. En la tabla adjunta se muestra una comparativa de la ubicación del archivo trewa.jar y del archivo de configuración indicándose si la localización se llevará a cabo o no:

Ubicación del archivo trewa.jar en el servidor de aplicaciones	Ruta default.properties en el servidor de aplicaciones	Observaciones
TOMCAT		
common/lib	common/classes/trewa/conf/perfiles	El perfil es localizado por las TrAPIs
common/lib	webapps/<APLICACIÓN>/classes	El perfil no es localizado por las TrAPIs

	/trewa/conf/perfiles	
webapps/<APLICACIÓN>/lib	common/classes/trewa/conf/perfiles	El perfil es localizado por las TrAPIs
webapps/<APLICACIÓN>/lib	webapps/<APLICACIÓN>/classes /trewa/conf/perfiles	El perfil es localizado por las TrAPIs
JBOSS		
server/default/lib	deploy/<APLICACIÓN.war>	El perfil no es localizado por las TrAPIs
deploy/<APLICACIÓN.war>	deploy/<APLICACIÓN.war>	El perfil es localizado por las TrAPIs

2.1.8.7 Formato del fichero de properties

La configuración del archivo de properties se realiza editando el archivo y publicando los siguientes campos:

Campo	Ob.	Descripción
tipo	Sí	Tipo de base de datos a la que se va a acceder. Por defecto ORACLE
nombreMaquina	Sí	Máquina
puerto	Sí	Puerto de acceso
nombreBD	Sí	Nombre de la base de datos
nombreUsuario	No	Nombre de usuario
claveUsuario	No	Clave de acceso
usaPool	No	Indica si se quiere hacer uso del pool interno que ofrecen las Jtrapis. Posibles valores: true ó false
maxConexionesActivas	No	Máximo de conexiones abiertas posibles
maxEsperaConexion	No	Tiempo máximo de espera (en milisegundos) para obtener una conexión del pool.
maxConexionesOciosas	No	Máximo de conexiones ociosas.

Un ejemplo de archivo de properties sería:

```
#Archivo de propiedades

tipo=ORACLE

#Datos para realizar la conexion

nombreMaquina=desa.ja.es

puerto=1521

nombreBD=bdOracle

nombreUsuario=

claveUsuario=

#Indica si se quiere crear la conexión usando pool de conexiones

usaPool=false

#Máximo de conexiones abiertas

maxConexionesActivas=4



#Máximo de tiempo de espera por conexión

maxEsperaConexion=5000

#Máximo de conexiones ociosas

maxConexionesOciosas=2
```

Si no se especifica usuario, este será exigido a través de las J-TrAPIs para establecer la conexión. En este caso se dejaría el campo en el archivo pero sin valor asignado.

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

A partir de la versión 1.2.0 de Trew@ se permite indicar los valores de nombreMaquina, puerto y nombreBD en uno solo, formando así la url de conexión JDBC. Para ello basta con indicar la url de conexión JDBC en el atributo nombreMaquina. De este modo se permiten realizar conexiones remotas tan solo indicando la url de conexión JDBC.

Por ejemplo:

#Archivo de propiedades

tipo=ORACLE

#Datos para realizar la conexion

```
nombreMaquina=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(Host=nombreMaquina)(Port=puerto))(ADDRESS=(PROTOCOL = TCP)(Host=nombreMaquina)(Port = puerto)))(CONNECT_DATA=(SID=nombreBD))(SOURCE_ROUTE=YES))
```

puerto=

nombreBD=

nombreUsuario=

claveUsuario=

A partir de la versión 1.4.1 de Trew@ se permite indicar en el properties si se hace uso del pool de conexiones que ofrecen las JtrApis, para ello se definen los atributos usaPool, maxConexionesActivas, maxEsperaConexion y maxConexionesOciosas en el fichero properties. Al habilitar este pool de conexiones interno tenemos los mismos beneficios que si usásemos una conexión con Datasource, con la ventaja añadida que es más rápida su configuración.

2.1.8.8 Configuración con DataSources

En este caso se intenta aprovechar la configuración de DataSources que aportan la mayoría de los servidores de aplicaciones, ganando así la posibilidad de habilitar un pool de conexiones que será en todo caso el administrador del servidor el que se encargue de gestionar debidamente.

Esta gestión a través del DataSource va a redundar en beneficio del rendimiento de la base de datos. Estableciéndose una serie de conexiones (configuradas por el administrador) al inicio y reutilizándose las mismas, se reducirá el número de conexiones y desconexiones a la base de datos, reduciéndose así el uso de recursos de la base de datos. Es preferible en muchos casos tener un número de sesiones predefinidas abiertas que no ir abriendo y cerrando sesiones a la base de datos.

En los apartados siguientes se muestra una orientación básica de cómo configurar los DataSources en diferentes servidores de aplicaciones, remitiendo al lector a los manuales de configuración del servidor de aplicaciones en uso para cuestiones de configuración.

Dependiendo del servidor de aplicaciones que usemos debemos utilizar un prefijo para utilizarlo como perfil de conexión a la hora de crear cualquier interfaz de las JtrApis.

Servidor aplicaciones	de	Prefijo	Nombre DataSource	Perfil de conexión para la creación de cualquier JtrApis
Tomcat		java:comp/env/	pruebads	java:comp/env/pruebads
Jboss		java:/	pruebads	java:/pruebads
OC4J			Pruebads	pruebads

2.1.8.8.1 Configuración de DataSource en Tomcat (5.0.18)



NOTA: Se recomienda crear los DataSource por cada contexto (aplicación web) para no modificar el archivo server.xml. Para ello ver el apartado *"DataSource para una aplicación cliente desde Tomcat."*

Para crear un DataSource en Tomcat se debe modificar el archivo **server.xml** que está ubicado en **%TOMCAT%\conf**, incluyendo la información referente al DataSource y a la aplicación que lo usará.

Entre las etiquetas `<GlobalNamingResources>` y `</GlobalNamingResources>` se debe aportar la información del DataSource:

Etiqueta Resource		
Parámetro	Ob.	Descripción
name	Sí	Nombre del DataSource

type	Sí	Clase del DataSource
Etiqueta ResourceParams		
Parámetro	Ob.	Descripción
name	Sí	Nombre del DataSource
url	Sí	Cadena de conexión JDBC
password	Sí	Contraseña
maxActive	No	Máximo número de conexiones activas
maxWait	No	Máximo número de conexiones en espera
driverClassName	Sí	Clase que implementa el driver de conexión
maxIdle	No	Máximo de conexiones que se dejan abiertas en espera de ser utilizadas
username	Sí	Usuario



	Consejería de Justicia y Administración Pública Dirección General de Modernización e Innovación de los Servicios Públicos	Trew@ v1.5.0 Manual de operación	
---	---	-------------------------------------	---

La estructura para las etiquetas *Resource* y *ResourceParams*, en el *server.xml* quedaría:

```

<Resource name="trOracle" type="javax.sql.DataSource"/>
<ResourceParams name="trOracle">
  <parameter>
<name>url</name>    <value>jdbc:oracle:thin:@desa.ja.es:1521:bdOracle</value>
  </parameter>
  <parameter>
<name>password</name>
    <value>Pwd</value>
  </parameter>
  <parameter>
<name>maxActive</name>
    <value>10</value>
  </parameter>
  <parameter>
<name>maxWait</name>
<value>5000</value>
  </parameter>
  <parameter>
<name>driverClassName</name>
    <value>oracle.jdbc.driver.OracleDriver</value>
  </parameter>
  <parameter>
    <name>maxIdle</name>
    <value>2</value>
  </parameter>
  <parameter>
<name>username</name>
    <value>Usuario</value>
  </parameter>

```

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

Para completar la configuración, hay que indicar la aplicación (o aplicaciones) que va a usar el DataSource definido anteriormente. Para ello se debe añadir entre las etiquetas <Host> y </Host> la siguiente información (un contexto por aplicación):

```
<Context path="/AppWeb" docBase="AppWeb" debug="0">
    <ResourceLink name="trOracle" global="trOracle" type="javax.sql.DataSource"/>
</Context>
```

En el caso anterior se indica que la aplicación **AppWeb**, que se encuentra en el directorio **%TOMCAT%/webapps/AppWeb**, va a hacer uso del DataSource definido como **trOracle**.

El archivo **server.xml**, una vez modificado, quedaría del siguiente modo:

```
<?xml version='1.0' encoding='utf-8'?>

<Server>

    <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener"/>

    <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>

    <GlobalNamingResources>

        <Environment name="simpleValue" type="java.lang.Integer" value="30"/>

        <Resource auth="Container" description="User database that can be updated and saved" name="UserDatabase"
type="org.apache.catalina.UserDatabase"/>

        <ResourceParams name="UserDatabase">

            <parameter>

                <name>factory</name>

                <value>org.apache.catalina.users.MemoryUserDatabaseFactory</value>

            </parameter>

            <parameter>

                <name>pathname</name>

                <value>conf/tomcat-users.xml</value>

            </parameter>



        </ResourceParams>

    </GlobalNamingResources>

</Server>
```



```
</ResourceParams>  
  
<Resource name="trOracle" type="javax.sql.DataSource"/>  
  
<ResourceParams name="trOracle">  
  
  <parameter>  
  
    <name>url</name>  
  
    <value>jdbc:oracle:thin:@desa.ja.es:1521:bdOracle</value>  
  
  </parameter>  
  
  <parameter>  
  
    <name>password</name>  
  
    <value>Pwd</value>  
  
  </parameter>  
  
  <parameter>  
  
    <name>maxActive</name>  
  
    <value>10</value>  
  
  </parameter>  
  
  <parameter>  
  
    <name>maxWait</name>  
  
    <value>5000</value>  
  
  </parameter>  
  
  <parameter>  
  
    <name>driverClassName</name>  
  
    <value>oracle.jdbc.driver.OracleDriver</value>  
  
  </parameter>  
  
  <parameter>  
  
    <name>maxIdle</name>  
  
    <value>2</value>  
  
  </parameter>
```

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

```

<parameter>

  <name>username</name>

  <value>Usuario</value>

</parameter>

</ResourceParams>

</GlobalNamingResources>

<Service name="Catalina">

  <Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true" port="8080"
  redirectPort="8443">

    </Connector>

    <Connector                                port="8009"                                protocol="AJP/1.3"
    protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler" redirectPort="8443">

      </Connector>

    <Engine defaultHost="localhost" name="Catalina">

      <Host appBase="webapps" name="localhost">

        <Logger className="org.apache.catalina.logger.FileLogger" prefix="localhost_log." suffix=".txt"
        timestamp="true"/>

        <Context path="/AppWeb" docBase="AppWeb" debug="0">

          <ResourceLink name="trOracle" global="trOracle" type="javax.sql.DataSource"/>

        </Context>

      </Host>

      <Logger className="org.apache.catalina.logger.FileLogger" prefix="catalina_log." suffix=".txt"
      timestamp="true"/>

      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"/>

    </Engine>

  </Service>

</Server>

```

2.1.8.8.2 *Datasource para una aplicación cliente desde Tomcat*

Se puede definir un DataSource asociado a la aplicación desde las pantallas de administración de Tomcat. La configuración definida quedará registrada en el archivo:

```
<rutaTomcat>\conf\Catalina\localhost\<nombreApp>.xml
```

Los pasos a seguir para crear el DataSource son los siguientes:

Accedemos a la Administración de Tomcat.

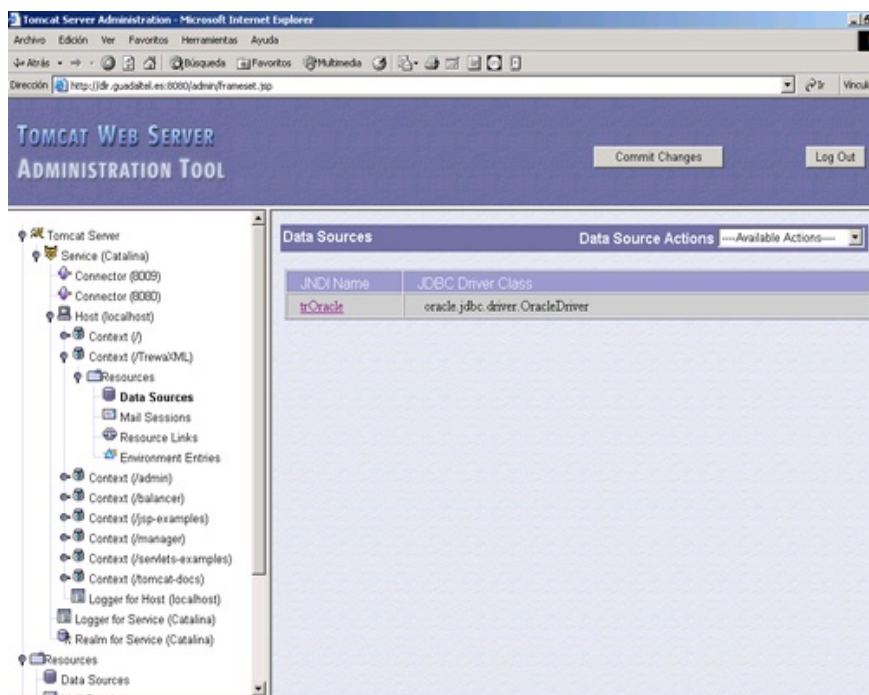
Desplegamos el nodo "Tomcat Server" hasta llegar al contexto de nuestra aplicación.



Seleccionamos Data Sources (Fuentes de Datos), desplegamos la lista de opciones que tenemos a la derecha y seleccionamos (CreaCrear Nueva Fuente de Datos)

Rellenamos los datos para realizar la conexión.

Pulsamos sobre el botón aplicar cambios.

Ya podemos usar este DataSource (con la nomenclatura adecuada) como un perfil de conexión para crear cualquier interfaz de las JTrApis



	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<Context displayName="Seervicios Web de Trewa" docBase="C:/Servidores/Tomcat50N/webapps/trewaws"
path="/trewaws">
```

```
<Resource name="jdbc/trewaws" type="javax.sql.DataSource"/>
```

```
<ResourceParams name="jdbc/trewaws">
```

```
<parameter>
```

```
<name>maxWait</name>
```

```
<value>5000</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>maxActive</name>
```

```
<value>25</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>password</name>
```

```
<value>pwd</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>url</name>
```

```
<value>jdbc:oracle:thin:@máquina:1521:bd</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>driverClassName</name>
```

```
<value>oracle.jdbc.driver.OracleDriver</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>maxIdle</name>
```

```

<value>1</value>

</parameter>

<parameter>

  <name>username</name>

  <value>user</value>

</parameter>

</ResourceParams>

</Context>

```

2.1.8.8.3 Configuración de DataSource en JBoss (4.0.0/4.0.1)

Desde JBoss es necesario modificar o crear el archivo **oracle-ds.xml**. Este archivo debe estar ubicado en **%JBOSS%\server\default\deploy** y para el DataSource que se vaya a crear se tendrá que aportar la siguiente información:

Etiqueta	Ob.	Descripción
<jndi-name>	Sí	Nombre del DataSource
<connection-url>	Sí	Cadena de conexión JDBC
<driver-class>	Sí	Clase que implementa el driver de conexión
<user-name>	No	Nombre de usuario
<password>	No	Contraseña
<min-pool-size>	No	Mínimo número de conexiones en el pool
<max-pool-size>	No	Máximo número de conexiones en el pool
<type-mapping>	Sí	Tipo de base de datos

La estructura del oracle-ds.xml vendría a ser:

```

<datasources>

  <local-tx-datasource>

    <jndi-name>[Nombre del DataSource]</jndi-name>

    <connection-url>[Cadena de conexión jdbc]</connection-url>

    <driver-class>[Clase java que implementa el driver de conexión]</driver-class>

    <user-name>[Nombre de usuario]</user-name>

    <password>[Contraseña]</password>

    <exception-sorter-class-name>

      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter

    </exception-sorter-class-name>

    <min-pool-size>[mínimo de conexiones en el pool]</min-pool-size>

    <max-pool-size>[máximo de conexiones en el pool]</max-pool-size>

    <metadata>

      <type-mapping>Oracle9i</type-mapping>

    </metadata>

  </local-tx-datasource>

  <local-tx-datasource>

    .....

  </local-tx-datasource>

</datasources>

```

El caso descrito en la configuración con archivos de properties quedaría en este caso del siguiente modo:

La estructura del oracle-ds.xml sería:

```

<datasources>

  <local-tx-datasource>

    <jndi-name>trOracle</jndi-name>

    <connection-url>jdbc:oracle:thin:@desa.ja.es:1521:bdOracle</connection-url>

    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>

    <user-name>UsrComun user-name>

    <password>Pwd</password>

    <exception-sorter-class-name>

      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter

    </exception-sorter-class-name>

    <min-pool-size>5</min-pool-size>

    <max-pool-size>20</max-pool-size>

    <metadata>

      <type-mapping>Oracle9i</type-mapping>

    </metadata>

  </local-tx-datasource>

  <local-tx-datasource>



    .....

  </local-tx-datasource>

</datasources>

```

En este caso las aplicaciones accederán usando el perfil **trOracle** ya que así se ha definido por parte del administrador y no tendrían que validarse pues en la propia especificación del DataSource lleva incluido usuario y password.

	Consejería de Justicia y Administración Pública Dirección General de Modernización e Innovación de los Servicios Públicos	Trew@ v1.5.0 Manual de operación	
---	--	-------------------------------------	---

2.1.8.9 Librerías necesarias

2.1.8.9.1 *JDK*

→ Para el correcto funcionamiento de las TrAPIs se precisa de la **versión 1.4.2 del JDK/JRE** o posterior.

2.1.8.9.2 *JDBC Oracle*

→ Las TrAPIs hacen uso de la librería **ojdbc14.jar** que contiene las clases de **JDBCDrivers Oracle10g**. Para que las TrAPIs hagan uso de estas clases, dicha librería puede estar en uno de estos directorios:

2.1.8.9.3 *Despliegue en Tomcat*

Incluir la librería **ojdbc14.jar** en el directorio **%Tomcat%\common\lib**.

2.1.8.9.4 *Despliegue en Jboss*

Incluir la librería **ojdbc14.jar** en el directorio **%JBOSSE%\server\default\lib**.

El uso de la librería **ojdbc14.jar** permite, entre otras cosas, el uso de **Savepoint** y **Rollback**.



IMPORTANTE: Si durante la ejecución de la aplicación JSP aparece el siguiente error:

```
ERROR: java.lang.AbstractMethodError: java.sql.Savepoint
       java.sql.Connection.setSavepoint()
```

la causa puede ser que no haya reconocido la librería **ojdbc14.jar**. Esto puede ser por varios motivos: por que no esté colocada en el directorio correcto, o porque en el directorio donde esté colocada exista la librería **clases12.jar** que también incluye los drivers de Oracle pero no soporta Savepoint ni Rollback. Una posible solución a este último caso es renombrar la librería **ojdbc14.jar** con un nombre menor que **clases12.jar** para que encuentre antes la librería **ojdbc14.jar** que la **clases12.jar**. Se podría renombrar la librería **ojdbc14.jar** a **aojdbc14.jar** por ejemplo.

2.1.8.9.5 *Despliegue en OC4J*

Para asegurar que en oc4j se carga inicialmente el jdbc y otras clases necesarias se debe incluir junto con la aplicación el archivo orion-web.xml con el siguiente contenido:

	Consejería de Justicia y Administración Pública Dirección General de Modernización e Innovación de los Servicios Públicos	Trew@ v1.5.0 Manual de operación	
---	--	-------------------------------------	---

```
<?xml version = '1.0' encoding = 'windows-1252'?>

<!DOCTYPE orion-web-app PUBLIC "-//Evermind//DTD Orion Web Application 2.3//EN"
"http://xmlns.oracle.com/ias/dtds/orion-web.dtd">

<orion-web-app servlet-webdir="/servlet/">

  <web-app-class-loader search-local-classes-first="true"/>
</orion-web-app>
```

Este archivo se ubicará en la carpeta WEB-INF de la aplicación.

2.1.8.9.6 Drivers Oracle Database 10g (10.1.0.4) - Versión 10.1.0.2.0 driver jdbc

Archivos

<i>ojdbc14.jar</i>	Clases para usar con JDK 1.4
<i>orai18n.jar</i>	Clases NLS (National Language Support) para usar con JDK 1.2, 1.3 y 1.4. Este jar reemplaza a los antiguos nls_charset jar/zip. NLS permite obtener e insertar información en la base de datos en cualquier juego de caracteres que soporte Oracle.
<i>ocrs12.jar</i>	Clases que implementan las interfaces javax.sql.rowset CachedRowSet y WebRowSet para ser usadas con JDK 1.2, 1.3 y 1.4



2.1.8.9.7 Drivers Oracle Database 10g Release 2 - Versión 10.2.0.1.0 driver jdbc

Archivos

<i>ojdbc14.jar</i>	Clases para usar con JDK 1.4 y JDK 1.5
<i>orai18n.jar</i>	Clases NLS (National Language Support) para usar con JDK 1.2, 1.3, 1.4 y 1.5 . Este jar reemplaza a los antiguos nls_charset jar/zip. NLS permite obtener e insertar información en la base de datos en cualquier juego de caracteres que soporte Oracle.

2.1.8.9.8 Otras librerías necesarias

Para el correcto funcionamiento de Trew@ se precisa una serie de librerías adicionales que se aportan junto con la distribución y que se describen a continuación:

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

Librerías adicionales

Jdbc PostgreSQL 8

Librería para acceso a base de datos con postgresQL.
Necesaria si se monta la base de datos sobre PostgreSQL

Jdom 1.0

Librería para parseo de XML.
Necesaria para habilitar la importación y exportación de definición de procedimientos, expedientes y entidades del sistema.
(Ver APIUTL)

Log4java 1.2.9

Librería para trazabilidad de errores.
Necesaria si se quiere habilitar la trazabilidad de la aplicación para detectar posibles errores.
(Ver Trazabilidad con log4java)

Cientes webServices

Cientes necesarios para habilitar el acceso a @visor, notific@dor, port@firmas, w@rda y trew@

Axis 1.1

Librerías necesarias para desplegar los servicios web de Trew@
(Ver Montaje de servicios Trew@)

2.1.8.10 Trazabilidad con log4java

Para habilitar la trazabilidad de log4java en trewa, bastará con modificar dos archivos:

1.- <web_app>/WEB-INF/classes/trewa/conf/trewaconf.properties

En este archivo se tendrá que especificar la ruta del archivo de configuración log4java:

#Ruta del fichero de configuración para Log4Java

log_4j_configuration_file=/trewa/log4j.properties

Campo	Ob.	Descripción
log_4j_configuration_file	No	Ruta donde está ubicado el fichero de configuración para log4java

2.- El archivo log4j.properties

A través de este archivo se podrá configurar el nivel de trazabilidad que se desea habilitar sobre trewa.

#Logger para aplicación TREWA

log4j.logger.trewa=DEBUG, FILE_LOG_TREWA_APPENDER

log4j.additivity.trewa=false

#Appender usado por aplicación TREWA

log4j.appender.FILE_LOG_TREWA_APPENDER=org.apache.log4j.RollingFileAppender



log4j.appender.FILE_LOG_TREWA_APPENDER.File=/trewa.log

log4j.appender.FILE_LOG_TREWA_APPENDER.layout=org.apache.log4j.PatternLayout

log4j.appender.FILE_LOG_TREWA_APPENDER.layout.ConversionPattern=%d{[yyyy-MM-dd HH:mm:ss,SSS]} %5p: %m%n

log4j.appender.FILE_LOG_TREWA_APPENDER.MaxFileSize=10MB

log4j.appender.FILE_LOG_TREWA_APPENDER.MaxBackupIndex=10

	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	
---	---	--	---

Para cambiar la configuración de los niveles de trazabilidad sobre Trew@, se modificará la entrada `log4j.logger.trewa` estableciendo algún nivel de los posibles que se describen a continuación:

DEBUG: Se utiliza para escribir mensajes de depuración, este log no debe estar activado cuando la aplicación se encuentre en producción.

INFO: Se utiliza para mensajes similares al modo "verbose" en otras aplicaciones.

WARN: Se utiliza para mensajes de alerta sobre eventos que se desea mantener constancia, pero que no afectan el correcto funcionamiento del programa.

ERROR: Se utiliza en mensajes de error de la aplicación que se desea guardar, estos eventos afectan al programa pero lo dejan seguir funcionando.

FATAL: Se utiliza para mensajes críticos del sistema, generalmente luego de guardar el mensaje el programa abortará.

Adicionalmente a estos niveles de log, existen 2 niveles extras que solo se utilizan en el archivo de configuración, estos son:

ALL: este es el nivel más bajo posible, habilita todos los logs.

OFF: este es el nivel más alto posible, deshabilita todos los logs.

`log4j.logger.trewa=OFF, FILE_LOG_TREWA_APPENDER`

2.1.8.11 Configuración para WebOffice

Para el correcto funcionamiento de WebOffice (versiones anteriores a v3.0.0) se precisa el establecimiento de un perfil para el mismo. Dicho perfil se especificará a través del archivo de **weboffice.properties** que se ubicará en el paquete **trewa.conf** de la aplicación cliente con el siguiente formato:

perfil=default

Campo	Ob.	Descripción
perfil	No	Perfil de acceso para WebOffice

2.1.8.12 Montaje de los Servicios Web de Trew@

Para el correcto funcionamiento de los servicios web de Trew@ se precisa realizar el despliegue de una aplicación con las siguientes características:

1.- Definición de los perfiles de acceso. Para esto se creará el archivo **trewaws.properties** en el paquete **trewa.conf**, donde se especificarán los perfiles a usar por los diferentes servicios. El archivo *trewaws.properties* tiene el siguiente formato:

perfil=java:comp/env/jdbc/trewaws

perfil_trewa=java:comp/env/jdbc/trewaws1

perfil_modelo=java:comp/env/jdbc/trewaws2

perfil_warda=java:comp/env/jdbc/trewaws3

Campo	Ob.	Descripción
perfil	No	Perfil por defecto para todos los servicios
perfil_trewa	No	Perfil por defecto para el servicio de trewa. Si no se especifica esta entrada se tomará la que se indique en la entrada <i>perfil</i> .
perfil_modelo	No	Perfil por defecto para el servicio de actualización de codificadoras. Si no se especifica esta entrada se tomará la que se indique en la entrada <i>perfil</i> .
perfil_warda	No	Perfil por defecto para el servicio de warda. Si no se especifica esta entrada se tomará la que se indique en la entrada <i>perfil</i> .

2.- Inclusión en la carpeta **public_html\WEB-INF** de la aplicación los archivos **web.xml** y **server-config.wsdd**. Ambos archivos se aportan junto con la plantilla de ejemplo para montar el servicio web.

3.- Inclusión de las librerías necesarias.

2.1.8.13 Condiciones, acciones y variables Java

Una de las características que aporta la interfaz de acceso a Trew@ es la posibilidad de ejecutar clases java desplegadas en el servidor de aplicaciones. Para poder habilitar esta característica se deben tener en cuenta los siguientes puntos:



1.- Se debe definir alguna condición/acción y/o variable en un procedimiento con <<Tipo de implementación>> = "Java". Dicha definición debe llevar el nombre de la clase, que incluirá el paquete completo al que pertenezca (miApp.trewa.condiciones.ClaseCondiciones), y el nombre de la función a ejecutar. Dicha función debe aparecer definida como *public* en la clase de forma que pueda ser accesible por trew@.

2.- Los métodos asociados a condiciones o acciones deben devolver un valor numérico (objeto java.lang.Integer,...) ,**0** si no se cumple la condición y **1** si se cumple la condición. Los parámetros en este caso deben llevar el siguiente orden y deben ser del tipo que corresponda según la siguiente tabla:

Parámetro en Trew@	Parámetro en método
Id Expediente	java.math.BigDecimal
Id Transición	java.math.BigDecimal
Id Documento permitido	java.math.BigDecimal
Id Expediente en fase	java.math.BigDecimal
Id del procedimiento	java.math.BigDecimal
Fecha	java.sql.Timestamp
Usuario	java.lang.String
Id Fase	java.math.BigDecimal
Id del tipo de documento	java.math.BigDecimal

3.- Los métodos asociados a variables deben devolver un java.lang.String. Los parámetros en este caso deben ser equivalentes a los definidos en los parámetros asociados a la variable, siendo la correspondencia la indicada en la siguiente tabla:

Tipo de parámetro en Trew@	Tipo de parámetro en método
NUMÉRICO	java.lang.Integer
CADENA	java.lang.String
FECHA	java.sql.Timestamp

 <p>JUNTA DE ANDALUCÍA</p>	<p>Consejería de Justicia y Administración Pública</p> <p>Dirección General de Modernización e Innovación de los Servicios Públicos</p>	<p>Trew@ v1.5.0</p> <p>Manual de operación</p>	 <p>Guadatel</p>
---	---	--	---

4.- La clase debe aparecer accesible a las clases de Trew@. Esto se puede conseguir de alguna de las formas siguientes (se muestra ejemplo para Tomcat):

Si trewa.jar está a nivel de aplicación

(%TOMCAT%\webapps\AplicacionJSP\WEB-INF\lib\trewa.jar)

En este caso las clases de condiciones y/o variables podrán ubicarse en:

- %TOMCAT%\webapps\AplicacionJSP\WEB-INF\lib (empaquetadas en un jar)
- %TOMCAT%\webapps\AplicacionJSP\WEB-INF\classes (los .class)
- %TOMCAT%\common\lib (empaquetadas en un jar)
- %TOMCAT%\common\classes (los .class)

Si trewa.jar está a nivel de servidor

(%TOMCAT%\common\lib\trewa.jar)

En este caso las clases de condiciones y/o variables podrán ubicarse en:

- %TOMCAT%\common\lib (empaquetadas en un jar)
- %TOMCAT%\common\classes (los .class)



5.- Si se desea que la función a ejecutar en la condición o variable reutilice la interfaz TrAPIUI, la clase a la que pertenezca se tendrá que derivar de la clase

trewa.ext.TrAccesoUI

Heredando de la clase TrAccesoUI se tendrá acceso al método **getApiUI()** a través del cual se podrá obtener la interfaz TrAPIUI y reutilizar la conexión actual.

Además al extender de la clase TrAccesoUI tendremos acceso al método **getNombre()** con el que podemos obtener el nombre de la variable, condición u acción que se esté ejecutando.

IMPORTANTE: Hasta la versión 1.4.2 de Trew@ los atributos apiUI y nombre de la clase TrAccesoUI eran static, a partir de esta versión no son static para corregir algunas causísticas. Al actualizar de una versión anterior o igual a la 1.4.2 a una superior se recomienda revisar el código de las variables, condiciones y acciones.

	Consejería de Justicia y Administración Pública Dirección General de Modernización e Innovación de los Servicios Públicos	Trew@ v1.5.0 Manual de operación	
--	---	-------------------------------------	--

```
import trewa.ext.TrAccesoUI;
```

```
public class ClaseVariable extends TrAccesoUI
```

```
{
```

```
    // Método de variable sin parámetros
```

```
    public String metodoVar( java.lang.String val ) throws TrException
```

```
    {
```

```
        String valRet = "";
```

```
        .....
```

```
        return str;
```

```
    }
```

```
}
```

```
import trewa.ext.TrAccesoUI;

public class ClaseCondicion extends TrAccesoUI
{
    // Método de condición
    public Integer existenDatos( BigDecimal idExpediente ) throws TrException
    {
        Integer intRet = new Integer(0);

        apiUI = getApiUI();



        if ( apiUI == null ) return intRet;

        ....

        return str;
    }
}
```

En el caso de condiciones complejas, se permite que la condición/es que componen dicha condición compleja devuelvan un String. Con esto podremos comparar el valor de una condición con un valor numérico o una cadena para decidir si la condición se cumple o no. Para ello debemos usar el prefijo `$$VAL$$` si queremos que se compare por el valor de la condición en lugar de su resultado lógico.

Por ejemplo, suponemos que tenemos definidas dos condiciones, `COND_COMPLEJA` y `COND_SIMPLE`. La `COND_SIMPLE` estará definida como una condición de implementación java que devolverá un String con el valor "2000". La `COND_COMPLEJA` estará definida como una condición compleja de implementación java cuya expresión podría ser `$$VAL$$COND_SIMPLE$$ > 1000`, esto quiere decir que si el valor de `COND_SIMPLE` es mayor que 1000 se cumplirá la condición.

	Consejería de Justicia y Administración Pública Dirección General de Modernización e Innovación de los Servicios Públicos	Trew@ v1.5.0 Manual de operación	
---	--	-------------------------------------	---

2.2 Mantenimiento del Motor

2.2.1 Base de datos

La base de datos de Trew@ no requiere especial mantenimiento, aparte de las tareas típicas de administración de base de datos. Basta con ir actualizando cuando sea necesario a futuras nuevas versiones.

2.2.2 Pantallas de Administración de Trew@

Estas pantallas no sólo nos van a permitir definir procedimientos en Trew@, sino que también nos van a permitir mantener los datos de configuración del motor (tablas de constantes nombradas anteriormente).

Mediante estas pantallas podremos por ejemplo definir el "juego" de constantes que va a utilizar cada sistema (siempre que sea necesario) así como modificar los permisos y perfiles asociados a cada usuario.

Además nos van a permitir la importación al sistema y exportación desde el sistema de ficheros de definición de procedimientos en XML.

2.2.3 Condiciones y acciones

Cabe destacar en este apartado que las condiciones y acciones pl/sql que se definen en el procedimiento a nivel de transiciones y de tareas, deben estar definidas como funciones PL/SQL que devuelven un valor numérico (0 para cuando no se cumplan y 1 cuando se ejecuten correctamente). También deben darse permisos de ejecución sobre las mismas al usuario propietario del motor Trew@ (o al usuario de conexión en caso de utilizar usuarios distintos al propietario).

2.2.4 Variables de los documentos

Al igual que las condiciones las variables que se definen en los documentos, deben estar definidas como funciones que devuelven un valor cadena de caracteres (varchar2 o string). También en el caso de variables pl/sql deben darse permisos de ejecución sobre las mismas al usuario propietario del motor Trew@ (o al usuario de conexión en caso de utilizar usuarios distintos al propietario).

Para que Trew@ haga la sustitución de variables por el valor de la función que representan, éstas deben ponerse en los párrafos de los documentos como "\$\$NOMBRE_VARIABLE\$".

En el caso de utilizar Web Office, además de cumplir esta norma en el texto de la plantilla del documento, las variables se deben asociar al tipo de documento independientemente de si están implementadas en pl/sql o java.

3 ANEXOS

No hay anexos.

4 GLOSARIO

No hay glosario, no se cree necesario.

5 BIBLIOGRAFÍA Y REFERENCIAS

Referencia	Título	Código
	Dominio semántico w@ndA	
	Guías de referencia TrApis de Trew@	
	Manual de operación Trew@ para PostgreSQL	
	Notas sobre generación de documentos	
	Notas sobre creación de usuarios	
	Guía de integración w@ndA	