



Eusko Jaurlaritzaren Informatika Elkarte
Sociedad Informática del Gobierno Vasco

Manual de tareas ant.

Fecha: 09/05/2008

Referencia:

EJIE S.A.
Mediterráneo, 14
Tel. 945 01 73 00*
Fax. 945 01 73 01
01010 Vitoria-Gasteiz
Posta-kutxatila / Apartado: 809
01080 Vitoria-Gasteiz
www.ejie.es

Este documento es propiedad de EJIE, S.A. y su contenido es confidencial. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE, S.A.. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE, S.A. no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.



Control de documentación

Título de documento:

Histórico de versiones

Código:

Versión: 1.0

Fecha: 18/02/2008

Resumen de cambios:

Cambios producidos desde la última versión

Primera versión.

Control de difusión

Responsable:

Aprobado por:

Firma:

Fecha:

Distribución:

Referencias de archivo

Autor: Unai Elorza – WLS810_NormasAlbergue.doc

Nombre archivo:

Localización:

Contenido

	Capítulo/sección	Página
	Introducción	1
	Sintaxis de ant	2
	Targets	3
1.1	Targets de compilación y preparación	3
1.2	Targets de despliegue	5
1.3	Targets de creación y eliminación de pooles y datasources	7
1.4	Targets de generación de ficheros jar	8
1.5	Targets de publicación de Web Services	10
1.6	Targets para test de la infraestructura en ejecución	13
1.7	Targets para configuración y creación de componentes JMS (ESTAS TAREAS SE ENCUENTRAN EN FASE DE VALIDACIÓN)	28
1.8	Targets para habilitar / deshabilitar aplicaciones	30
1.9	Targets para test de calidad	31
1.10	Targets para test de implantación	33
1.11	Targets para test de aplicaciones horizontales	41
1.12	Targets varios	48
	Anexo I: Registro de incidencias y situaciones comunes	50
	Incidencias relacionadas con las tareas de ant	51
	Incidencias relacionadas con PLATEA	55
	Incidencias relacionadas con los procesos batch, K31 y O75	61

BORRADOR

Introducción

El objetivo principal de este documento es informar sobre las diferentes tareas de ant que se pueden utilizar en el entorno de EJIE.

Este documento recoge parte de lo referente a las tareas ant que se encuentra en el documento "WLS810_NormasAlbergue.doc" y la especificación de las nuevas tareas ant creadas actualmente dentro de la propuesta de mejora del entorno de desarrollo planteado por el grupo de Consultoría de Areas de Conocimiento (CAC); estas nuevas tareas ant estarán en color azul.

Además, se han añadido otras tareas existentes y que no estaban reflejadas en el documento "WLS810_NormasAlbergue.doc"; estas tareas estarán en color verde.

Finalmente, hay que remarcar que no se trata de un documento definitivo y que según se vayan desarrollando las nuevas tareas se realizarán las modificaciones pertinentes.

Sintaxis de ant

Puesto que Ant necesita el fichero de configuración `build.xml`, para poder ejecutar las distintas operaciones de configuración hay que 'situarse' en el subdirectorio `java\javs` de la aplicación.

- **Sintaxis:**

```
ant [options] [target [target2 [target3] ...]]
```

- **Options**

- help**, imprime este mensaje.
- projecthelp**, imprime los targets del proyecto.
- version**, imprime la versión.
- diagnostics**, imprime información que puede ser útil para detectar problemas.
- quiet** o **-q**, imprime la información mínima.
- verbose** o **-v**, imprime información adicional.
- debug**, imprime información de debugging.
- emacs**, produce información de logging sin adornos.
- logfile file** o **-l file**, escribe el resultado de la ejecución en un fichero.
- logger classname**, fija la clase que va a implementar el logging.
- listener classname**, añade una instancia de clase como un proyecto de listener.
- buildfile filename** o **-file filename** o **-f filename**, utiliza el buildfile especificado.
- find file**, busca el fichero de configuración, `build.xml` por defecto, desde el directorio actual hacia sus directorios padre, usando el primero que encuentre.
- propertyfile name**, carga todas las propiedades del fichero con la propiedad `-D` que le precede.
- inputhandler class**, se indica la clase que manejará las peticiones de entrada.
- Dproperty=value**, define el valor de una propiedad para el proyecto.

Targets

1.1 Targets de compilación y preparación

- **help**, imprime los targets del proyecto.
- **clean**, borra los dos directorios que se crean durante la compilación. Por un lado borra el directorio dist, en el que se genera el fichero .ear de la aplicación y por otro lado borra el directorio exe, que tiene exactamente la misma estructura que el fichero .ear, pero en este directorio el ear se encuentra en modo exploded.
- **compile**, esta tarea compila las clases java de la aplicación. Si no se pasa ninguna opción se compilarán todas las clases de la aplicación. Opciones disponibles:
 - **path**, indica el subdirectorio a compilar.
 - **filej**, indica el fichero a compilar. Este parámetro no debe incluir la extensión del fichero (p.ej: aaaHelloWorld)
 - Combinación de ambos parámetros
- **create**, llama a la tarea appc que realiza diferentes acciones dependiendo del tipo de módulo sobre el que actúe. En los módulos war comprueba que la configuración del módulo sea válida y precompila los jsps. En los módulos EJB comprueba que la configuración del módulo sea válida y genera los stubs y los skeletons. En los módulos ear comprueba que la configuración sea correcta y actúa sobre cada uno de los módulos. Esta tarea es proactiva en su ejecución, por ejemplo si un jsp no ha sido modificado no intentará precompilarlo otra o si la interfaz home y remote de un EJB no han cambiado no intentará regenerar los stubs y los skeletons. Si no se pasa ninguna opción actuará sobre todos los módulos de la aplicación. Opciones disponibles:
 - **pathCreate**, indica el nombre del módulo sobre el que se ejecutará la tarea.
- **all**, esta tarea invoca las tareas clean, compile, y la tarea create.
- **all2**, realiza la misma función que la tarea all pero mejora el rendimiento ostensiblemente, se diferencia de la tradicional en que no realiza un limpiado previo de las clases compiladas ni hace una compilación de las JSPs sino que las copia directamente a la zona exe de la aplicación y se compilan en ejecución.
- **compileGeremua**, tarea que compila las clases java de una aplicación para Geremua versión 2
- **allGeremua**, tarea similar al all pero para aplicaciones con Geremua versión 2
- **prepare**, esta tarea realiza la preparación del ear para un Entorno en concreto (Internet, Intranet). Con esta tarea lo que se permite es de una aplicación en concreto desplegar partes diferentes en distintos entornos de producción. La tarea prepara un fichero .ear para el entorno en el que se la invoque en la carpeta de distribución.

Ejemplos de los Targets de compilación y preparación

- Imprime los targets del proyecto:

ant **-help**

- Borrado de los directorios creados durante la compilación:

ant **clean**

- Compilar las clases comunes:

ant -Dpath=classes **compile**

- Compilar las clases de un EJB:

ant -Dpath=aaaSaludaEJB **compile**

- Compilar las clases de un War:

ant -Dpath=aaaSaludaWar **compile**

- Compilar las clases de todos los Wars:

ant -Dpath=*War **compile**

- Compilar las clases de todos los EJBs:

ant -Dpath=*EJB **compile**

- Compilar una clase concreta de las comunes:

ant -Dpath=classes -Dfilej=aaaVarios **compile**

- Compilar únicamente la interfaz remota de un EJB:

ant -Dpath=aaaSaludaEJB -Dfilej=aaaSaluda **compile**

- Compilar todos los ficheros java que se encuentran en la aplicación:

ant **compile**

- Compilar todos los ficheros java que se encuentran en la aplicación desarrollada con Geremua versión 2:

ant **compileGeremua**

- Chequear la configuración de un EJB y generar los stubs y skeletons:

ant -DpathCreate=aaaSaludaEJB **create**

- Chequear la configuración de un War y recompilar los jsps modificados:

ant -DpathCreate=aaaSaludaWar **create**

- Chequear la configuración de la aplicación completa y preparar cada uno de los módulos:

ant **create**

- Compilar una aplicación completa (compilar todos los ficheros java, chequeo de las configuraciones, precompilación de los jsps, generación de stubs y skeletons...):

ant **all**

- Compilar una aplicación completa (compilar todos los ficheros java) de manera optimizada:

ant **all2**

- Compilar una aplicación completa desarrollada con Geremua versión 2

ant **allGeremua**

- Preparación del ear para el entorno Internet.

ant -Dproduccion=Internet **prepare**

1.2 Targets de despliegue

Estas tareas deben ser invocadas una vez que las tareas compile y create han sido ejecutadas satisfactoriamente. La única excepción a lo anteriormente comentado es la tarea undeploy.

- **deploy**, despliegue de la aplicación completa generada en la ruta /aplic/aaa/exe. Una vez que la aplicación se encuentra desplegada, al realizar cambios, será necesario invocar la tarea redeploy o redeployPartes para que los cambios surjan efecto.
- **deployPartes**, permite desplegar módulos de una aplicación, es decir, un módulo web (war) o un módulo EJB. Opciones:
 - **pathDeploy**, indica el nombre del módulo sobre el que se ejecutará la tarea.
- **rededeploy**, despliegue de la aplicación completa.
- **rededeployPartes**, redespiegue de una parte de la aplicación (opción redeployPartes). El redespiegue se puede hacer de toda la aplicación, de un módulo web o un EJB. Opciones:
 - **pathDeploy**, indica el nombre del módulo sobre el que se ejecutará la tarea.
- **undeploy**, undeploy de la aplicación completa.
- **desplegar**, esta tarea realiza un undeploy y un deploy. Para la mayoría de las situaciones será suficiente con incluir esta tarea en el fichero deploy.txt.
- **deployExploded**: Esta tarea desplegará la aplicación en modo exploded y nostage y su filesystem corresponderá con la carpeta exe de la aplicación.
- **rededeployExplodedAll**: Esta tarea redesplegará toda una aplicación que se encuentra desplegada con la tarea ant deployExploded.
- **rededeployExplodedModule**: Esta tarea redesplegará un modulo de una aplicación que se encuentra desplegada con la tarea ant deployExploded. Parámetros:
 - **pathDeploy**, indica el nombre del módulo sobre el que se ejecutará la tarea.
- **rededeployExplodedFile**: Esta tarea redesplegará un fichero o los todos los ficheros de una ruta relativa a un módulo de una aplicación que se encuentra desplegada con la tarea ant deployExploded. Parámetros:
 - **pathDeploy**, indica la ruta del fichero sobre el que se ejecutará la tarea.
- **appStruts**: Tarea utilizada para desplegar aplicaciones con Struts, primero se lanzará esta tarea y luego las utilizadas en aplicaciones normales.

Nota: Se recomienda lanzar las tareas undeploy + deploy cuando existan cambios en la configuración de la aplicación (p.ej: agregar un nuevo módulo ejb), en vez de lanzar únicamente la tarea redeploy.

Nota2: Cuando se vaya a realizar un despliegue en modo exploded y no-stage de una aplicación que ya se encuentra desplegada pero no en esos modos es preferible realizar previamente un undeploy de la aplicación.

Ejemplos de los Targets de despliegue

- Despliegue de una aplicación completa:

```
ant deploy
```

- Despliegue de una aplicación web perteneciente a la aplicación:

```
ant -DpathDeploy=aaaSaludaWar deployPartes
```

- Despliegue de un EJB perteneciente a la aplicación:

```
ant -DpathDeploy=aaaSaludaEJB deployPartes
```

- Redespliegue de la aplicación completa:

```
ant redeploy
```

- Redespliegue de una aplicación web:

```
ant -DpathDeploy=aaaSaludaWar redeployPartes
```

- Redespliegue de un EJB:

```
ant -DpathDeploy=aaaSaludaEJB redeployPartes
```

- Undeploy de la aplicación completa:

```
ant undeploy
```

- Se realizan muchos cambios en la aplicación, por lo que hay que hacer un undeploy y un deploy:

```
ant desplegar
```

- Despliegue de una aplicación en modo exploded y no-stage:

```
ant deployExploded
```

- Redespliegue de toda una aplicación que se encuentra desplegada con la tarea ant deployExploded

```
ant redeployExplodedAll
```

- Redespliegue de un módulo de una aplicación que se encuentra desplegada con la tarea ant deployExploded

```
ant -DpathDeploy=s73aUtilsWar redeployExplodedModule
```

- Redespliegue de un fichero o todos los ficheros de una ruta relativa a un módulo de una aplicación que se encuentra desplegada con la tarea ant deployExploded

```
ant -DpathDeploy=s73aUtilsWar/s73aTramitacionJSP/S73aPruebaRoc.jsp
```

```
redeployExplodedFile
```

- Despliegue de una aplicación con Struts, primero se lanzará esta tarea y luego las utilizadas en aplicaciones normales.

```
ant appStruts
```

1.3 Targets de creación y eliminación de pooles y datasources

- **createPool**, creación de un pool de conexiones jdbc. Parámetros requeridos:
 - **poolNameBD**, corresponde con el nombre del pool generado. Formato: aaaPool.
 - **userBD**, usuario del usuario en la Base de Datos. Formato: aaa.
 - **passBD**, password del usuario en la Base de Datos.
 - **conexionBD**, nombre generado en el tnsnames. Formato: aaad.
 - **usarCompatibilidad**, si se iguala a true añade la propiedad oracle.jdbc.V8Compatible=true; si se iguala a false o no se pone la propiedad será oracle.jdbc.V8Compatible=false.
- **createDataSource**, creación de un datasource asociado a un pool de conexiones JDBC. Parámetros requeridos:
 - **dataSource**, corresponde con el nombre del DataSource generado. Formato: aaaDataSource.
 - **poolNameBD**, corresponde con el pool sobre el que se generará el DataSource. Formato: aaaPool.
 - **jndiName**, nombre JNDI asociado al DataSource. Formato: aaa.aaaDataSource.
- **deletePool**, eliminación de un pool de conexiones JDBC.
- **deleteDataSource**, eliminación de un DataSource asociado a un pool de conexiones JDBC.

Ejemplos de los Targets de creación y eliminación de pooles y datasources

- Creación de un pool de conexiones JDBC:

```
ant -DpoolNameBD=aaaPool -DuserBD=aaa -DpassBD=aaa -DconexionBD=aaad -DusarCompatibilidad=true createPool
```

- Creación de un datasource asociado a un pool de conexiones JDBC:

```
ant -DpoolNameBD=aaaPool -DdataSource=aaaDataSource -DjndiName=aaa.aaaDataSource createDataSource
```

- Eliminación de un pool de conexiones JDBC:

```
ant -DpoolNameBD=aaaPool deletePool
```

- Eliminación de un datasource asociado a un pool de conexiones JDBC:

```
ant -DdataSource=aaaDataSource deleteDataSource
```

1.4 Targets de generación de ficheros jar

- **jar**, Permite generar un fichero .jar con las clases de una aplicación siguiendo un filtro. En caso de querer acceder desde una aplicación (aaa2) a una clase de otra aplicación (aaa) será necesario usar esta tarea. La idea es que desde la aplicación aaa se genere el fichero .jar con las clases compartidas y éste sea copiado en la carpeta lib de la aplicación aaa2. Parámetros requeridos:
 - **filesJar**, filtro que define las clases a incluir.
 - **nameJar**, parámetro opcional que permite indicar el nombre de salida del fichero .jar. En caso de no indicar este parámetro el fichero de salida se llamará aaa.jar.
- **deleteJar**, Tarea que borra el jar de aplicación, es decir el jar que está ubicado en /entorno/aplic/aaa/java/javs/aaa.jar. Permite que se le pase como parámetro el jar que se quiere borrar. Por ejemplo si no queremos borrar el jar de aplicación pero queremos borrar el jar xxx.jar ubicado bajo /entorno/aplic/aaa/lib/xxx.jar pondremos `ant -DPATH_JAR=/lib/xxx.jar deleteJar`.
- **ejb-stubs**, Permite generar un fichero .jar con las clases cliente del EJB. En caso de querer acceder desde una aplicación (aaa2) a un EJB de otra aplicación (aaa) será necesario usar esta tarea. La idea es que desde la aplicación aaa se genere el fichero .jar con las clases cliente del EJB y éste sea copiado en la carpeta lib de la aplicación aaa2. Parámetros requeridos:
 - **ejb**, nombre del ejb del que generar el fichero .jar con el cliente.
El fichero con el cliente será generado en la carpeta /aplic/aaa/java/javs y seguirá el formato `ejb_client.jar`.
- **mix-jars**, Permite unir ficheros .jar. Parámetros requeridos:
 - **filesJar**, lista de ficheros .jar.
 - **outputJar**, nombre del fichero .jar generado.
- **ear**, Permite crear el fichero de distribución aaa.ear. Lo que hace esta tarea es crear un jar de la carpeta /aplic/aaa/exe y dejar el fichero en /aplic/aaa/dist con extensión ear. En los pasos a pruebas y producción se envía el fichero ear en vez de la carpeta /aplic/aaa/exe, que estará disponible en la carpeta /explopr/aplic/aaa/dist o /explo/aplic/aaa.

Ejemplos de los Targets de generación de ficheros jar

- Creación de un jar con las clases:

```
ant -DfilesJar=utiles/*,utilesWAR/* jar
```

- Borrado del jar prueba.jar

```
ant -DPATH_JAR=/lib/prueba.jar deleteJar
```

- Borrado del jar de aplicación (aaa.jar)

```
ant deleteJar
```

- Creación del cliente de un EJB:

```
ant -Dejb=aaaSaludaEJB ejb-stubs
```

- Creación de un jar con el contenido de dos jars existentes:

```
ant -DfilesJar=aaaClases.jar,aaaSaludaEJB_client.jar -DoutputJar=aaa.jar mix-jars
```

- Creación del ear de la aplicación:

```
ant ear
```

1.5 Targets de publicación de Web Services

En Weblogic 8 existe la posibilidad de publicar los métodos de una clase o de un EJB como un Web Service. La publicación se basa en la generación de un módulo war y un nombre de servicio asociado a una clase o a un EJB. Los nombres de los módulos war a utilizar como contenedores de Web Services deberán estar definidos en el fichero application.xml como un módulo mas y deberá existir la estructura en disco asociada. La url de acceso a los Web Services publicados estará formada por:

- nombre del sitio web
- nombre del war que contiene el Web Service
- nombre del Web Service

ej: <http://sitioweb/aaaWebService1War/aaaWebService1WS>

A ésta url habrá que realizar peticiones vía JAX-RPC, bien usando el cliente java generado dependiente de Weblogic, uno java independiente u otro cliente (p.ej: acceso desde Visual Basic).

De cara a generar un servicio web desde una clase Java o un EJB existe la posibilidad de que el proceso sea semiautomático (se genera el descriptor del servicio web *web-services.xml*) o que el fichero *web-services.xml* sea suministrado por el desarrollador.

En la Aplicación de Ejemplo existen ejemplos de cada una de las tareas anteriormente comentadas. Las tareas Ant a ejecutar se encuentran en el fichero webServices.sh de la carpeta /aplic/aaa/java/javs.

- Tareas semiautomáticas: el fichero web-services.xml es generado por Weblogic mediante introspección por lo que la aplicación deberá estar compilada. Es necesario haber ejecutado las tareas compile y create satisfactoriamente antes de ejecutar las tareas de publicación. Tras ejecutar las tareas de publicación será necesario invocar las tareas de despliegue, deploy o redeploy dependiendo de si es la primera vez, para que los Web Services estén disponibles.
 - **webServiceClaseAuto**, permite publicar una clase como un Web Service. Parámetros requeridos:
 - **war**, nombre del módulo war que contendrá el Web Service. Formato: **aaaNombreWar**.
 - **clases**, clase java a publicar como Web Service.
 - **serviceName**, nombre del servicio a generar. Formato **aaaNombreWS**.
 - **clientPackageName**, nombre del paquete sobre el que se generará el cliente java asociado.
 - **webServiceEJBAuto**, permite publicar un EJB como un Web Service. Parámetros requeridos:
 - **war**, nombre del módulo war que contendrá el Web Service. Formato: **aaaNombreWar**.
 - **ejb**, nombre del ejb a publicar como Web Service.

- **serviceName**, nombre del servicio a generar. Formato **aaaNombreWS**.
 - **clientPackageName**, nombre del paquete sobre el que se generará el cliente java asociado.
- Tareas manuales: se permite que el desarrollador aporte el fichero web-services.xml. Éste fichero deberá ser ubicado en la ruta /aplic/aaa/java/javs/aaaEAR/aaaNombreWar/WEB-INF/web-services.xml del war encargado de publicar el Web Service. Es necesario haber ejecutado las tareas compile y create satisfactoriamente antes de ejecutar las tareas de publicación. Tras ejecutar las tareas de publicación será necesario invocar las tareas de despliegue, deploy o redeploy dependiendo de si es la primera vez, para que los Web Services estén disponibles.
 - **webServiceClaseConf**, permite publicar una clase como un Web Service. Parámetros requeridos:
 - **war**, nombre del módulo war que contendrá el Web Service. Formato: **aaaNombreWar**.
 - **javacomponents**, clase java a publicar como Web Service. Formato: **package.aaaNombre** sin incluir la extensión .java.
 - **javaSource**, clase java a publicar como Web Service. Formato: **package/aaaNombre.java**.
 - **serviceName**, nombre del servicio a generar. Formato **aaaNombreWS**.
 - **clientPackageName**, nombre del paquete sobre el que se generará el cliente java asociado.
 - **webServiceEJBConf**, permite publicar un EJB como un Web Service. Parámetros requeridos:
 - **war**, nombre del módulo war que contendrá el Web Service. Formato: **aaaNombreWar**.
 - **javacomponents**, clase del ejb a publicar como Web Service. Formato: **package.aaaNombre** sin incluir la extensión .java.
 - **javaSource**, clase java a publicar como Web Service. Formato: **package/aaaNombre.java**.
 - **ejbName**, nombre del ejb a publicar como Web Service.
 - **serviceName**, nombre del servicio a generar. Formato **aaaNombreWS**.
 - **clientPackageName**, nombre del paquete sobre el que se generará el cliente java asociado.
 - En todos los casos se genera un cliente webService que se copia en /aplic/aaa/exe/APP-INF/lib para que pueda ser localizado en el classpath y en /aplic/aaa/exe/nombreWarWS/ para que el jar del cliente webService pueda ser descargado directamente por web (<http://maquina:puerto/aaaNombreWar/cliente.jar>). El nombre del fichero jar del cliente esta formado por el nombre del war, el nombre del servicio web y el sufijo _client. P.ej: aaaWebService1War_aaaSaludaWSClase_client.jar

- Para acceder al documento WSDL que describe el Web Service publicado la url será:
 - <http://sitioweb/aaaNombreWar/aaaNombreWS?WSDL>
- Weblogic Server 8.1 genera un cliente web para poder probar los Web Services. La url será:
 - <http://sitioweb/aaaNombreWar/aaaNombreWS>
- Otras tareas de Web Services:
 - **createWSClient**, tarea para generar un cliente de Web Service pasándole la ruta del WSDL

Ejemplos de los Targets de publicación de Web Services

- Publicación de una clase como un Web Service generando la configuración de forma automática:

```
ant -Dwar=aaaWebService1War -Dclases=utiles.aaaSaludaWS -DserviceName=aaaSaludaClase1WS -
DclientPackageName=webService1 webServiceClaseAuto
```

- Publicación de un EJB como un Web Service generando la configuración de forma automática:

```
ant -Dwar=aaaWebService2War -Dejb=aaaSaludaWSEJB -DserviceName=aaaSaludaEJB2WS -
DclientPackageName=webService2 webServiceEJBAuto
```

- Publicación de una clase como un Web Service suministrando la configuración:

```
ant -Dwar=aaaWebService3War -Djavacomponents=utiles.aaaSaludaWS -
DjavaSource=utiles/aaaSaludaWS.java -DserviceName=aaaSaludaClase3WS -
DclientPackageName=webService3 webServiceClaseConf
```

- Publicación de un EJB como un Web Service suministrando la configuración:

```
ant -Dwar=aaaWebService4War -Djavacomponents=aaaSaludaWSEJB.aaaSaludaWS -
DjavaSource=aaaSaludaWSEJB/aaaSaludaWS.java -DejbName=aaaSaludaWSEJB -
DserviceName=aaaSaludaEJB4WS -DclientPackageName=webService4 webServiceEJBConf
```

- Generación de un cliente Web Service pasándole la ruta del WSDL:

```
ant -DrutaWSDL=http://webldes55:7041/s73aUtilsWar/S73aGetSolicitud?WSDL -
Dpaquete=com.ejie.client -DnombreCliente=S73aGetSolicitud createWSClient
```


1.6 Targets para test de la infraestructura en ejecución

- **test**, tarea para recoger información útil para el desarrollador en cuanto a la situación en la que se encuentran todos los componentes que participan en una aplicación. Además esta tarea generará los correspondientes informes dependientes de los parámetros solicitados y visibles a través del dashboard. De esta manera, esta tarea se compone a su vez de dos subtareas, una la que genera el resultado del test por pantalla (testPantalla) y la otra la que genera el informe para el dashboard (informeTest) y que poseen los mismos parámetros de entrada que la tarea de test. Parámetros requeridos:

- **param**, indica el componente sobre el que se ejecutará la tarea. Los componentes posibles son los siguientes:

- **server**

El objetivo de esta tarea es presentar información relacionada con la instancia en la que está alojada una aplicación así como del estado en el que se encuentra dicha instancia y la máquina física que la alberga.

La información facilitada incluye:

- Instancia de la aplicación
- Dirección del cluster
- Máquinas activas del cluster
- Relación de máquinas que componen el cluster junto con su estado.
- Numero de procesadores
- Carga media de los procesadores (junto con su porcentaje sobre el total)
- Memoria física libre (junto con su porcentaje sobre el total)
- Memoria física usada (junto con su porcentaje sobre el total)
- Memoria física total
- Memoria libre de la máquina virtual (junto con su porcentaje sobre el total)
- Memoria usada de la máquina virtual (junto con su porcentaje sobre el total)
- Memoria total de la máquina virtual
- Versión de Java
- Numero actual de demonios java
- Numero actual de threads java (demonios y no demonios)

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] *****
[java] *****  SERVIDORES  *****
[java] Instancia de la aplicacion:  apps1
[java] Direccion del cluster:  webldes55:7041,webldes56:7041
[java] Maquinas activas del cluster:  2
[java]
[java] Servidor:  inter_apps_dpto_apps111 , Estado:  RUNNING
[java] Numero de procesadores:  8
[java] Carga media de los procesadores:  0.125  ( 12.5 %)
```

```

[java] Memoria fisica libre: 5236797440 ( 31 %)
[java] Memoria fisica usada: 11572547584 ( 68 %)
[java] Memoria fisica total: 16809365504
[java] Memoria libre de la maquina virtual: 57243224 ( 27 %)
[java] Memoria usada de la maquina virtual: 155626888 ( 74 %)
[java] Memoria total de la maquina virtual: 209715200
[java] Version de Java: 1.4.2_04
[java] Numero actual de demonios Java: 99
[java] Numero actual de threads Java (demonios y no demonios): 126
[java]
[java]
[java] Servidor: inter_apps_dpto_apps112 , Estado: RUNNING
[java] Numero de procesadores: 8
[java] Carga media de los procesadores: 0.125 ( 12.5 %)
[java] Memoria fisica libre: 5236801536 ( 31 %)
[java] Memoria fisica usada: 11572555776 ( 68 %)
[java] Memoria fisica total: 16809365504
[java] Memoria libre de la maquina virtual: 57200344 ( 27 %)
[java] Memoria usada de la maquina virtual: 152514856 ( 72 %)
[java] Memoria total de la maquina virtual: 209715200
[java] Version de Java: 1.4.2_04
[java] Numero actual de demonios Java: 97
[java] Numero actual de threads Java (demonios y no demonios): 124

```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones:

- Si el valor del estado de la máquina no es RUNNING, esto indica que la máquina no está arrancada correctamente y por lo tanto la aplicación no funcionará.
- Si cualquiera de las memorias, ya sea la referente a la física como a la de la máquina virtual, tiene un porcentaje de uso cercano al 100% eso significará que se está llegando al límite de capacidad de procesamiento de las peticiones y por lo tanto afectará negativamente a la aplicación.
- El resto de los datos, como la versión de Java y el número de procesadores son datos meramente informativos sobre la máquina analizada.

- app

El objetivo de esta tarea es presentar información general relacionada con los módulos que conforman una aplicación así como datos de si se ha realizado alguna modificación desde el último despliegue y cuándo se ha producido ese último despliegue.

La información facilitada incluye:

- Nombre de la aplicación
- Si la aplicación está desplegada o no

- Cuándo se ha realizado el último despliegue
- Si se ha modificado algo en el último despliegue
- Tipo de despliegue
- Datos de los componentes de aplicación por máquina del cluster junto con su estado.
- Datos de los componentes web-service por máquina del cluster junto con su estado.
- Datos de los componentes EJB por máquina del cluster junto con su estado

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] *****
[java] ***** APLICACION *****
[java]
[java] ***** Datos generales de aplicacion
*****
[java] Nombre de la aplicacion:  s73a
[java] Desplegado:  true
[java] Ultimo despliegue:  1193141672176
[java] Se ha modificado algo con el ultimo despliegue:  true
[java] Tipo de despliegue:  EAR
[java]
[java] ***** Datos de los componentes de aplicacion
*****
[java] Modulo:  s73aL3PrototipoWar  Servidor:  inter_apps_dpto_apps111  Estado:
Desplegado
[java] Modulo:  s73aL3PrototipoWar  Servidor:  inter_apps_dpto_apps112  Estado:
Desplegado
[java] Modulo:  s73aPasarelaWar  Servidor:  inter_apps_dpto_apps111  Estado:
Desplegado
[java] Modulo:  s73aPasarelaWar  Servidor:  inter_apps_dpto_apps112  Estado:
Desplegado
[java] Modulo:  s73aPruebasWar  Servidor:  inter_apps_dpto_apps111  Estado:
Desplegado
[java] Modulo:  s73aPruebasWar  Servidor:  inter_apps_dpto_apps112  Estado:
Desplegado
[java]
[java] ***** Datos de los componentes web service
*****
[java] Modulo:  s73aClienteTramitacionWar  Servidor:  inter_apps_dpto_apps111
Estado: Desplegado
[java] Modulo:  s73aClienteTramitacionWar  Servidor:  inter_apps_dpto_apps112
Estado: Desplegado
[java] Modulo:  s73aSHFWar  Servidor:  inter_apps_dpto_apps111  Estado:
Desplegado
[java] Modulo:  s73aSHFWar  Servidor:  inter_apps_dpto_apps112  Estado:
Desplegado
[java] Modulo:  s73aSistemasTramitacionWar  Servidor:  inter_apps_dpto_apps111
Estado: Desplegado
```

```

[java] Modulo: s73aSistemasTramitacionWar Servidor: inter_apps_dpto_apps112
Estado: Desplegado

[java] Modulo: s73aUtilsWar Servidor: inter_apps_dpto_apps111 Estado:
Desplegado

[java] Modulo: s73aUtilsWar Servidor: inter_apps_dpto_apps112 Estado:
Desplegado

[java]

[java] ***** Datos de los componentes EJB
*****

[java] Modulo: s73aEstudiantesSVTEJB Servidor: inter_apps_dpto_apps111
Estado: Desplegado

[java] Modulo: s73aEstudiantesSVTEJB Servidor: inter_apps_dpto_apps112
Estado: Desplegado

[java] Modulo: s73aL3PrototipoAccesoBDEJB Servidor: inter_apps_dpto_apps111
Estado: Desplegado

[java] Modulo: s73aL3PrototipoAccesoBDEJB Servidor: inter_apps_dpto_apps112
Estado: Desplegado

[java] Modulo: s73ajmsmdbEJB Servidor: inter_apps_dpto_apps111 Estado:
Desplegado

[java] Modulo: s73ajmsmdbEJB Servidor: inter_apps_dpto_apps112 Estado:
Desplegado

```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones:

- Si el valor del estado de un módulo para una instancia es "No desplegado", indicará que ese modulo para esa instancia no funciona y por lo tanto que es necesario su despliegue.
- El resto de los datos que se aportan en este test como por ejemplo, si está desplegada la aplicación o no, el timestamp de su último despliegue o si se ha realizado alguna modificación desde el último despliegue, son datos meramente informativos.

- task

El objetivo de esta tarea es presentar información relacionada con aquellas tareas de la aplicación lanzadas en el weblogic que no han finalizado.

De esta manera se podrá comprobar si el weblogic ha procesado o no las tareas lanzadas y poder detectar así un posible mal funcionamiento de la instancia.

La información facilitada por tarea inacabada incluye:

- Nombre de la aplicación.
- Hora de inicio de la tarea.
- Descripción de la tarea.
- En qué estado se encuentra la tarea

Un ejemplo del resultado de la ejecución de esta tarea sería:

```

[java] *****
[java] ***** TAREAS INACABADAS ASOCIADAS A LA
APLICACION *****
[java] * Aplicacion: s73a
[java] * Hora Inicio: 1192613877499

```

```
[java] * Descripcion: [Deployer:149026]Deploy application s73a on appsl.  
[java] * Estado: Deploy Running
```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones:

- Si existen tareas inacabadas cuando ha pasado un tiempo prudencial desde el lanzamiento de esa tarea es un síntoma de que el weblogic no está procesando bien esas peticiones y por lo tanto la instancia no está funcionando debidamente. Por cada una de esas tareas inacabadas se presenta información para identificarla.

- **pool**

El objetivo de esta tarea es presentar información relacionada con el pool de conexiones de la aplicación y el estado en el que se encuentra.

La información facilitada por pool desplegado en cada máquina del cluster es:

- Nombre del pool.
- Estado del pool
- Número mínimo de conexiones físicas disponibles
- Número máximo de conexiones físicas que puede contener
- URL
- Conexiones activas
- Pico máximo de conexiones activas
- Conexiones disponibles
- Pico máximo de conexiones disponibles
- Conexiones no disponibles
- Pico máximo de conexiones no disponibles
- Peticiones de conexión en espera
- Pico máximo de peticiones de conexión en espera
- Conexiones no cerradas
- Numero de intentos fallidos de conexión
- Media de tiempo necesario para establecer una conexión (milisegundos)
- Numero de conexiones JDBC desde que el pool ha sido instanciado
- Numero de ocasiones en las que se ha usado una sentencia cacheada para hacer una consulta
- Numero de ocasiones en las que no se ha usado una sentencia cacheada para hacer una consulta

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] *****  
[java] ***** POOL *****  
[java] Pool: s73aPool  
[java] Numero minimo de conexiones fisicas disponibles: 1
```

```

[java] Numero maximo de conexiones fisicas que puede contener: 25
[java] URL: jdbc:oracle:thin:@s73d:1524:ede1
[java]
[java] Pool: inter_apps_dpto_apps111 , Estado: Running
[java] Conexiones activas: 0
[java] Pico maximo de conexiones activas: 1
[java] Conexiones disponibles: 1
[java] Pico maximo de conexiones disponibles: 1
[java] Conexiones no disponibles: 0
[java] Pico maximo de conexiones no disponibles: 0
[java] Peticiones de conexion en espera: 0
[java] Pico maximo de peticiones de conexion en espera: 0
[java] Conexiones no cerradas: 42
[java] Numero de intentos fallidos de conexion: 0
[java] Media de tiempo necesario para establecer una conexion (milisegundos):
71
[java] Numero de conexiones JDBC desde que el pool ha sido instanciado: 1
[java] Numero de ocasiones en las que se ha usado una sentencia cacheada para
hacer una consulta: 0
[java] Numero de ocasiones en las que no se ha usado una sentencia cacheada para
hacer una consulta: 2
[java]
[java] Pool: inter_apps_dpto_apps112 , Estado: Running
[java] Conexiones activas: 0
[java] Pico maximo de conexiones activas: 1
[java] Conexiones disponibles: 1
[java] Pico maximo de conexiones disponibles: 1
[java] Conexiones no disponibles: 0
[java] Pico maximo de conexiones no disponibles: 0
[java] Peticiones de conexion en espera: 0
[java] Pico maximo de peticiones de conexion en espera: 0
[java] Conexiones no cerradas: 196
[java] Numero de intentos fallidos de conexion: 0
[java] Media de tiempo necesario para establecer una conexion (milisegundos):
68
[java] Numero de conexiones JDBC desde que el pool ha sido instanciado: 1
[java] Numero de ocasiones en las que se ha usado una sentencia cacheada para
hacer una consulta: 13
[java] Numero de ocasiones en las que no se ha usado una sentencia cacheada para
hacer una consulta: 3

```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones:

- Si el valor de "Peticiones de conexión en espera" es mayor que cero, o bien el pico máximo de conexiones activas es igual al máximo de conexiones que el pool puede contener, eso nos indica que debido a las necesidades de nuestra aplicación es conveniente aumentar el número máximo de conexiones que el

pool puede contener y el nuevo valor debería ser aproximadamente la suma del valor anterior más el "Pico máximo de peticiones de conexión en espera".

- El valor de las conexiones que no han sido cerradas desde que se ha iniciado el pool debería ser cero, ya que en caso contrario, no se están cerrando las conexiones adecuadamente en el código.
- Otros datos que se muestran son informativos como la media de tiempo en realizar una conexión, las ocasiones en las que se usan o no sentencias cacheadas, etc...

- **webapp**

El objetivo de esta tarea es presentar información detallada sobre los componentes web de aplicación que se muestran en la tarea de app.

La información facilitada por componente web de aplicación incluye:

- Nombre del módulo.
- Servidor en el que se encuentra.
- Estado.
- Número total de sesiones abiertas de este componente.
- Pico máximo de sesiones abiertas de este componente desde que se ha iniciado el servidor.

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] *****
[java] ***** Componentes web de aplicacion
*****
[java] Modulo: s73aL3PrototipoWar Servidor: inter_apps_dpto_apps111 Estado:
Desplegado
[java] Numero total de sesiones abiertas de este componente: 1
[java] Pico maximo de sesiones abiertas de este componente desde que se
ha iniciado el servidor: 1
[java]
[java] Modulo: s73aL3PrototipoWar Servidor: inter_apps_dpto_apps112 Estado:
Desplegado
[java] Numero total de sesiones abiertas de este componente: 0
[java] Pico maximo de sesiones abiertas de este componente desde que se
ha iniciado el servidor: 0
[java]
[java] Modulo: s73aPasarelaWar Servidor: inter_apps_dpto_apps111 Estado:
Desplegado
[java] Numero total de sesiones abiertas de este componente: 0
[java] Pico maximo de sesiones abiertas de este componente desde que se
ha iniciado el servidor: 0
[java]
[java] Modulo: s73aPasarelaWar Servidor: inter_apps_dpto_apps112 Estado:
Desplegado
[java] Numero total de sesiones abiertas de este componente: 0
```

```

[java] Pico maximo de sesiones abiertas de este componente desde que se
ha iniciado el servidor: 0

[java]

[java] Modulo: s73aPruebasWar Servidor: inter_apps_dpto_apps111 Estado:
Desplegado

[java] Numero total de sesiones abiertas de este componente: 0

[java] Pico maximo de sesiones abiertas de este componente desde que se
ha iniciado el servidor: 0

[java]

[java] Modulo: s73aPruebasWar Servidor: inter_apps_dpto_apps112 Estado:
Desplegado

[java] Numero total de sesiones abiertas de este componente: 0

[java] Pico maximo de sesiones abiertas de este componente desde que se
ha iniciado el servidor: 0

[java]

```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones:

- Si el estado de un módulo es “No desplegado” será un indicativo de que ese módulo no se encuentra en funcionamiento y por lo tanto será necesario desplegarlo.
- El resto de datos son informativos como en el caso del numero de sesiones abiertas de un componente, que nos puede indicar en cuantas ocasiones se accede a un módulo concreto

- webservice

El objetivo de esta tarea es presentar información detallada sobre los web services desplegados en nuestra aplicación.

La información facilitada por web service incluye:

- Nombre del web service.
- URL del web service.
- Número de ocasiones que se ha accedido al web service.
- Número de ocasiones en las que se ha recibido una petición inválida.

Un ejemplo del resultado de la ejecución de esta tarea sería:

```

[java] *****
[java] ***** Web Services
*****

[java] Nombre: s73aProcessURL@inter_apps_dpto_apps111
[java] Url del WS: http://weblde55:7041/s73aSHFWar/s73aProcessURL
[java] Numero de ocasiones que se ha accedido al web service: 0
[java] Numero de ocasiones en las que se ha recibido una peticion invalida: 0
[java]

[java] Nombre: s73aProcessURL@inter_apps_dpto_apps112
[java] Url del WS: http://weblde56:7041/s73aSHFWar/s73aProcessURL

```



```
[java] Numero de ocasiones que se ha accedido al web service: 0
[java] Numero de ocasiones en las que se ha recibido una peticion invalida: 0
```

Todos los datos resultado de esta tarea son meramente informativos

- **ejb**

El objetivo de esta tarea es presentar información detallada sobre los EJBs desplegados en nuestra aplicación.

La información facilitada por EJB incluye:

- o Nombre del EJB.
- o Instancias de bean
- o Instancias de bean usadas
- o Instancias de bean disponibles
- o Veces que ha sido destruida una instancia de bean
- o Veces que ha fallado la creación de una instancia de bean
- o Threads que se han rechazado por time-out
- o Threads que están esperando a una instancia de bean
- o Pico máximo de threads que han estado esperando a una instancia de bean
- o Transacciones que han hecho commit
- o Transacciones que han hecho roll-back
- o Transacciones rechazadas por time-out
- o Beans activados
- o Beans pasivados
- o Beans que están actualmente en la caché
- o Intentos de acceso al bean desde la caché
- o Intentos de acceso al bean exitosos
- o Intentos de acceso al bean fallidos
- o Beans bloqueados
- o Intentos de obtener un bloqueo en un bean
- o Threads que esperan por un bloqueo en un bean
- o Pico máximo de threads que han esperado por un bloqueo en un bean
- o Threads rechazados por time-out esperando a un bloqueo en un bean

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] *****
[java] ***** EJB *****
[java] Elemento:
inter_apps_dpto_apps111_s73a_s73aEstudiantesSVTEJB_S73AEstudiantesSVT
```

```

[java]      Instancias de bean: 0
[java]      Instancias de bean usadas: 0
[java]      Instancias de bean disponibles: 0
[java]      Veces que se ha sido destruida una instancia de bean: 0
[java]      Veces que se ha fallado la creacion de una instancia de bean: 0
[java]      Threads que se han rechazado por time-out: null
[java]      Threads que estĩn esperando a una instancia de bean: null
[java]      Pico maximo de threads que han estado esperando a una instancia de
bean: null
[java]      Transacciones que han hecho commit: 0
[java]      Transacciones que han hecho roll back: 0
[java]      Transacciones rechazadas por time out: 0
[java]      Beans activados: null
[java]      Beans pasivados: null
[java]      Beans que estĩn actualmente en la cache: null
[java]      Intentos de acceso al bean desde la cache: null
[java]      Intentos de acceso al bean exitosos: null
[java]      Intentos de acceso al bean fallidos: null
[java]      Beans bloqueados: null
[java]      Intentos de obtener un bloqueo en un bean: null
[java]      Threads que esperan por un bloqueo en un bean: null
[java]      Pico maximo de threads que han esperado por un bloqueo en un bean:
null
[java]      Threads rechazados por time out esperando a un bloqueo en un bean:
null
[java]
[java]      Elemento:
inter_apps_dpto_apps111_s73a_s73aL3PrototipoAccesoBDEJB_s73aL3PrototipoAccesoBDEJB
[java]      Instancias de bean: 0
[java]      Instancias de bean usadas: 0
[java]      Instancias de bean disponibles: 0
[java]      Veces que se ha sido destruida una instancia de bean: 0
[java]      Veces que se ha fallado la creacion de una instancia de bean: 0
[java]      Threads que se han rechazado por time-out: null
[java]      Threads que estĩn esperando a una instancia de bean: null
[java]      Pico maximo de threads que han estado esperando a una instancia de
bean: null
[java]      Transacciones que han hecho commit: 0
[java]      Transacciones que han hecho roll back: 0
[java]      Transacciones rechazadas por time out: 0
[java]      Beans activados: null
[java]      Beans pasivados: null
[java]      Beans que estĩn actualmente en la cache: null
[java]      Intentos de acceso al bean desde la cache: null
[java]      Intentos de acceso al bean exitosos: null

```

```

[java] Intentos de acceso al bean fallidos: null
[java] Beans bloqueados: null
[java] Intentos de obtener un bloqueo en un bean: null
[java] Threads que esperan por un bloqueo en un bean: null
[java] Pico maximo de threads que han esperado por un bloqueo en un bean:
null
[java] Threads rechazados por time out esperando a un bloqueo en un bean:
null
[java]
[java] Elemento:
inter_apps_dpto_apps111_s73a_s73ajmsmdbEJB_s73ajmsmdbEJB_s73a.s73aJMSQueue@s73aJMSSer
ver_apps111
[java] Instancias de bean: null
[java] Instancias de bean usadas: null
[java] Instancias de bean disponibles: null
[java] Veces que se ha sido destruida una instancia de bean: null
[java] Veces que se ha fallado la creacion de una instancia de bean: null
[java] Threads que se han rechazado por time-out: null
[java] Threads que estãn esperando a una instancia de bean: null
[java] Pico maximo de threads que han estado esperando a una instancia de
bean: null
[java] Transacciones que han hecho commit: null
[java] Transacciones que han hecho roll back: null
[java] Transacciones rechazadas por time out: null
[java] Beans activados: null
[java] Beans pasivados: null
[java] Beans que estãn actualmente en la cache: null
[java] Intentos de acceso al bean desde la cache: null
[java] Intentos de acceso al bean exitosos: null
[java] Intentos de acceso al bean fallidos: null
[java] Beans bloqueados: null
[java] Intentos de obtener un bloqueo en un bean: null
[java] Threads que esperan por un bloqueo en un bean: null
[java] Pico maximo de threads que han esperado por un bloqueo en un bean:
null
[java] Threads rechazados por time out esperando a un bloqueo en un bean:
null
[java]
[java] Elemento:
inter_apps_dpto_apps112_s73a_s73aEstudiantesSVTEJB_S73AEstudiantesSVT
[java] Instancias de bean: 0
[java] Instancias de bean usadas: 0
[java] Instancias de bean disponibles: 0
[java] Veces que se ha sido destruida una instancia de bean: 0
[java] Veces que se ha fallado la creacion de una instancia de bean: 0
[java] Threads que se han rechazado por time-out: null

```

```

[java] Threads que estĩn esperando a una instancia de bean: null
[java] Pico maximo de threads que han estado esperando a una instancia de
bean: null
[java] Transacciones que han hecho commit: 0
[java] Transacciones que han hecho roll back: 0
[java] Transacciones rechazadas por time out: 0
[java] Beans activados: null
[java] Beans pasivados: null
[java] Beans que estĩn actualmente en la cache: null
[java] Intentos de acceso al bean desde la cache: null
[java] Intentos de acceso al bean exitosos: null
[java] Intentos de acceso al bean fallidos: null
[java] Beans bloqueados: null
[java] Intentos de obtener un bloqueo en un bean: null
[java] Threads que esperan por un bloqueo en un bean: null
[java] Pico maximo de threads que han esperado por un bloqueo en un bean:
null
[java] Threads rechazados por time out esperando a un bloqueo en un bean:
null
[java]
[java] Elemento:
inter_apps_dpto_apps112_s73a_s73aL3PrototipoAccesoBDEJB_s73aL3PrototipoAccesoBDEJB
[java] Instancias de bean: 0
[java] Instancias de bean usadas: 0
[java] Instancias de bean disponibles: 0
[java] Veces que se ha sido destruida una instancia de bean: 0
[java] Veces que se ha fallado la creacion de una instancia de bean: 0
[java] Threads que se han rechazado por time-out: null
[java] Threads que estĩn esperando a una instancia de bean: null
[java] Pico maximo de threads que han estado esperando a una instancia de
bean: null
[java] Transacciones que han hecho commit: 0
[java] Transacciones que han hecho roll back: 0
[java] Transacciones rechazadas por time out: 0
[java] Beans activados: null
[java] Beans pasivados: null
[java] Beans que estĩn actualmente en la cache: null
[java] Intentos de acceso al bean desde la cache: null
[java] Intentos de acceso al bean exitosos: null
[java] Intentos de acceso al bean fallidos: null
[java] Beans bloqueados: null
[java] Intentos de obtener un bloqueo en un bean: null
[java] Threads que esperan por un bloqueo en un bean: null
[java] Pico maximo de threads que han esperado por un bloqueo en un bean:
null

```

```

[java]      Threads rechazados por time out esperando a un bloqueo en un bean:
null
[java]
[java] Elemento:
inter_apps_dpto_apps112_s73a_s73ajmsmdbEJB_s73ajmsmdbEJB_s73a.s73aJMSQueue@s73aJMSSer
ver_apps112
[java]      Instancias de bean: null
[java]      Instancias de bean usadas: null
[java]      Instancias de bean disponibles: null
[java]      Veces que se ha sido destruida una instancia de bean: null
[java]      Veces que se ha fallado la creacion de una instancia de bean: null
[java]      Threads que se han rechazado por time-out: null
[java]      Threads que estigñn esperando a una instancia de bean: null
[java]      Pico maximo de threads que han estado esperando a una instancia de
bean: null
[java]      Transacciones que han hecho commit: null
[java]      Transacciones que han hecho roll back: null
[java]      Transacciones rechazadas por time out: null
[java]      Beans activados: null
[java]      Beans pasivados: null
[java]      Beans que estigñn actualmente en la cache: null
[java]      Intentos de acceso al bean desde la cache: null
[java]      Intentos de acceso al bean exitosos: null
[java]      Intentos de acceso al bean fallidos: null
[java]      Beans bloqueados: null
[java]      Intentos de obtener un bloqueo en un bean: null
[java]      Threads que esperan por un bloqueo en un bean: null
[java]      Pico maximo de threads que han esperado por un bloqueo en un bean:
null
[java]      Threads rechazados por time out esperando a un bloqueo en un bean:
null

```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones aunque hay que tener en cuenta que dependiendo del tipo y del uso que se haga de un EJB algunas de ellas vendrán informadas con el valor "null":

- Si el resultado de "Veces que ha fallado la creación de una instancia de bean" y "Intentos de acceso al bean fallidos" es superior a cero, habría que analizar más profundamente cuál es su causa.
- El resto de datos son informativos y sirven para poder medir el peso de los EJBs en la aplicación y tomar decisiones al respecto.

- **jms**

El objetivo de esta tarea es presentar información detallada sobre los servidores JMS asociados a nuestra aplicación.

La información facilitada por servidor JMS incluye:

- Nombre del servidor JMS.

- Estado en el que se encuentra.
- Bytes almacenados actualmente.
- Pico máximo de bytes almacenados.
- Bytes pendientes almacenados
- Bytes recibidos
- Mensajes almacenados actualmente
- Pico máximo de mensajes almacenados
- Mensajes pendientes almacenados
- Mensajes recibidos
- Sesiones instanciadas actualmente
- Pico máximo de sesiones instanciadas

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] *****
[java] ***** JMS *****
[java] Nombre: s73aJMSServer_apps111 , Estado: State:HEALTH_OK,ReasonCode:[
[java] Bytes almacenados actualmente: 0
[java] Pico maximo de bytes almacenados: 0
[java] Bytes pendientes almacenados: 0
[java] Bytes recibidos: 0
[java] Mensajes almacenados actualmente: 0
[java] Pico maximo de mensajes almacenados: 0
[java] Mensajes pendientes almacenados: 0
[java] Mensajes recibidos: 0
[java] Sesiones instanciadas actualmente: 0
[java] Pico maximo de sesiones instanciadas: 0
[java]
[java] Nombre: s73aJMSServer_apps112 , Estado: State:HEALTH_OK,ReasonCode:[
[java] Bytes almacenados actualmente: 0
[java] Pico maximo de bytes almacenados: 0
[java] Bytes pendientes almacenados: 0
[java] Bytes recibidos: 0
[java] Mensajes almacenados actualmente: 0
[java] Pico maximo de mensajes almacenados: 0
[java] Mensajes pendientes almacenados: 0
[java] Mensajes recibidos: 0
[java] Sesiones instanciadas actualmente: 0
[java] Pico maximo de sesiones instanciadas: 0
```

Sobre estos resultados se pueden sacar una serie de conclusiones e interpretaciones:

- Si el estado en el que se encuentra no es "HEALTH OK" entonces el servidor JMS no estará funcionando correctamente y como consecuencia se pueden estar encolando mensajes o no recibándose correctamente o algún otro tipo de error no controlado.
- Si el número de bytes-mensajes pendientes se incrementa exponencialmente eso significa que no se están enrutando bien los mensajes y se quedan en el servidor, por lo que se puede deducir que no se está dando un funcionamiento correcto del servidor JMS.
- El resto de datos que se ofrecen como resultado de la tarea ant son datos meramente informativos que ayudan a hacer un seguimiento de la aplicación.

Ejemplos de los Targets para test de la infraestructura en ejecución

- Testeo y extracción de información sobre los servidores de una aplicación

ant **-Dparam=server test**

- Testeo y extracción de información sobre los módulos de una aplicación

ant **-Dparam=app test**

- Testeo y extracción de información sobre las tareas iniciadas y no finalizadas de una aplicación

ant **-Dparam=task test**

- Testeo y extracción de información sobre el pool de conexiones de una aplicación

ant **-Dparam=pool test**

- Testeo y extracción de información sobre los componentes web de aplicación presentes en la aplicación

ant **-Dparam=webapp test**

- Testeo y extracción de información sobre los web services desplegados de una aplicación

ant **-Dparam=webservice test**

- Testeo y extracción de información sobre los EJBs desplegados de una aplicación

ant **-Dparam=ejb test**

- Testeo y extracción de información sobre los servidores JMS asociados a una aplicación

ant **-Dparam=jms test**

1.7 Targets para configuración y creación de componentes JMS (ESTAS TAREAS SE ENCUENTRAN EN FASE DE VALIDACIÓN)

- **createJMSServer**, tarea para crear un servidor JMS. La tarea internamente crea un servidor JMS por cada instancia de weblogic, en el cluster en el que se encuentra desplegada la aplicación.
- **createJMSFactory**, tarea para crear una factoría de conexiones JMS. La tarea crea un factoría JMS a través de la cual podrán crearse los diferentes Destinations, bien sean colas o topics.
- **createJMSQueue**, tarea para la generación de colas JMS. Estas colas se crearán asociadas a un servidor JMS previamente creado. Los parámetros necesarios para la creación de colas JMS son los siguientes:
 - **jmsqueue**, indica el nombre que tendrá la cola. El nombre completo generado por la tarea quedaría de esta forma:
 - **cod_app + jmsqueue + JMSQueue** (p.ej : t30bPasarelaJMSQueue)(el cod_app es un parámetro implícito a la tarea ant, no es necesario pasarlo)
 - **jmsserver**, indica el nombre del servidor JMS en que se ubicara la cola. Este servidor de JMS tiene que estar creado previamente para poder albergar las colas que se creen posteriormente.
- **createJMSTopic**, tarea para la generación de topics JMS. Estos topics, al igual que las colas, se crearán asociados a un servidor JMS previamente creado. Los parámetros necesarios para la creación de topics JMS son los siguientes:
 - **jmstopic**, indica el nombre que tendrá el topic. El nombre completo generado por la tarea quedaría de esta forma:
 - **cod_app + jmstopic + JMSTopic** (p.ej : t30bPasarelaJMSTopic)(el cod_app es un parámetro implícito a la tarea ant, no es necesario pasarlo)
 - **jmsserver**, indica el nombre del servidor JMS en que se ubicara el topic. Este servidor de JMS tiene que estar creado previamente para poder albergar las colas que se creen posteriormente.
- **createJMSDistributedQueue**, tarea para la generación de colas JMS distribuidas.
 - **jmsdistributedqueue**, indica el nombre que tendrá la cola distribuida. El nombre completo generado por la tarea quedaría de esta forma:
 - **cod_app + jmsdistributedqueue + JMSDistributedQueue** (p.ej : t30bPasarelaJMSDistributedQueue)
- **createJMSDistributedTopic**, tarea para la generación de topics JMS distribuidos.
 - **jmsdistributedtopic**, indica el nombre que tendrá la cola distribuida. El nombre completo generado por la tarea quedaría de esta forma:

- **cod_app + jmsdistributedtopic + JMSDistributedTopic** (p.ej : t30bPasarelaJMSDistributedTopic)
- **deleteJMSQueue**, tarea para el borrado de colas JMS. Los parámetros necesarios para lanzar la tarea son:
 - **jmsqueue**, es el nombre de la cola que queremos borrar.
- **deleteJMSTopic**, tarea para el borrado de topics JMS. Los parámetros necesarios para lanzar la tarea son:
 - **jmstopic**, es el nombre de la cola que queremos borrar.
- **deleteJMSDistributedQueue**, tarea para el borrado de colas JMS distribuidas. Los parámetros necesarios para lanzar la tarea son:
 - **jmsdistributedqueue**, es el nombre de la cola que queremos borrar.
- **deleteJMSDistributedTopic**, tarea para el borrado de topics JMS distribuidos. Los parámetros necesarios para lanzar la tarea son:
 - **jmsdistributedtopic**, es el nombre de la cola que queremos borrar.

Ejemplos de los Targets para la configuración y creación de componentes JMS

- Crear un servidor JMS

ant **createJMSServer**

- Crear una factoría de conexiones JMS

ant **createJMSFactory**

- Crear una cola JMS

ant **createJMSQueue** -Djmsqueue=Pasarela -Djmsserver=t30bJMSServer_apps111

- Crear un topic JMS

ant **createJMSTopic** -Djmstopic=Pasarela -Djmsserver=t30bJMSServer_apps111

- Creación de colas JMS distribuidas

ant **createJMSDistributedQueue** -Djmsdistributedqueue=Pasarela

- Creación de topics JMS distribuidos

ant **createJMSDistributedTopic** -Djmsdistributedtopic=Pasarela

- Borrado de colas JMS

ant **deleteJMSQueue** -Djmsqueue=Pasarela

- Borrado de topic JMS

ant **deleteJMSTopic** -Djmstopic=Pasarela

- Borrado de colas JMS distribuidas

ant **deleteJMSDistributedQueue** -Djmsdistributedqueue=Pasarela

- Borrado de topics JMS distribuidos

ant **deleteJMSDistributedTopic** -Djmsdistributedtopic=Pasarela

1.8 Targets para habilitar / deshabilitar aplicaciones

- **habilitar**, tarea para habilitar una aplicación. Esta tarea tomará como referencia un fichero de propiedades, el fichero `aaaApp.properties` sito en el mismo lugar que el `build.xml` (`/aplic/aaa/java/javs`) y en el que estarán indicados los datos necesarios de la aplicación como el usuario y contraseña del pool, todos aquellos datos referentes a las colas JMS si es el caso, etc....

Un ejemplo de este fichero se describe a continuación:

```
# Pool

usuarioPool=t29a

passPool=t29a

hostPool=t29d

# Colas JMS

servidorJMS=t29aJMSServer

colaJMS=t29aJMSQueue

topicJMS=t29aJMSTopic
```

- **deshabilitar**, tarea para deshabilitar una aplicación. Esta tarea tomará como referencia el mismo fichero de propiedades descrito en la tarea de habilitar.

Ejemplos de los Targets para test de calidad

- Habilitar una aplicación

```
ant habilitar
```

- Deshabilitar una aplicación

```
ant deshabilitar
```

1.9 Targets para test de calidad

- **junit**, tarea para lanzar la herramienta homologada para las pruebas unitarias JUnit. No requiere de parámetros pero si de otras consideraciones que se encuentran especificadas en el documento "Pruebas unitarias y de regresión. Manual de instalación en servidor".

Esta tarea genera los correspondientes informes de resultados que se empaquetarán en un zip y se depositarán en el directorio /datos/aaa/lis

- **checkstyle**, tarea para lanzar la herramienta homologada para las pruebas de calidad de código Checkstyle. No requiere de parámetros pero si de otras consideraciones como un fichero de reglas, etc... que se encuentran especificados en el documento "Pruebas unitarias y de regresión. Manual de instalación en servidor".

Esta tarea genera los correspondientes informes de resultados que se empaquetarán en un zip y se depositarán en el directorio /datos/aaa/lis

- **pmd**, tarea para lanzar la herramienta homologada para las pruebas de calidad de código PMD. No requiere de parámetros pero si de otras consideraciones como un fichero de reglas, etc... que se encuentran especificados en el documento "Pruebas unitarias y de regresión. Manual de instalación en servidor".

Esta tarea genera los correspondientes informes de resultados que se empaquetarán en un zip y se depositarán en el directorio /datos/aaa/lis

- **aus**, tarea para lanzar la herramienta homologada para identificar dependencias de API's de un producto en otro para crear un conjunto de detallados informes mostrando dónde se encuentran dichas dependencias y cómo están siendo usadas. No requiere de parámetros pero si de otras consideraciones como un fichero de reglas, etc... que se encuentran especificados en el documento "Aus. Manual de Usuario".

Esta tarea genera los correspondientes informes de resultados que se empaquetarán en un zip y se depositarán en el directorio /datos/aaa/lis

- **dashboard-ejie**, tarea para generar un entorno global con los resultados de todas las pruebas en servidor en formato HTML. No requiere de parámetros.

Ejemplos de los Targets para test de calidad

- Testeo de pruebas unitarias - JUnit

ant **junit**

- Testeo de pruebas de calidad de código - Checkstyle

ant **checkstyle**

- Testeo de pruebas de calidad de código - PMD

ant **pmd**

- Testeo de dependencias de APIs - AUS

ant **aus**

- Generación de un entorno global con los resultados de todas las pruebas.

ant **dashboard-ejie**

1.10 Targets para test de implantación

- **testImp**, tarea para lanzar el test de implantación que comprobará una serie de filtros que se especifican a continuación; además esta tarea generará el correspondiente informe visible a través del dashboard. De esta manera, esta tarea se compone a su vez de dos subtareas, una la que genera el resultado del test por pantalla (testImpPantalla) y la otra la que genera el informe para el dashboard (informeTestImp).
 - FILTRO 1: Chequeo de si existe el filtro de XLNets en todos los web.xml de los War de la aplicación
 - FILTRO 2: Chequeo de que están declarados todos los módulos de la aplicación en el application.xml
 - FILTRO 3: Se muestran todas las librerías que están puestas en la aplicación. De esta manera se podrán detectar aquellas librerías no homologadas, así como las necesarias para el correcto funcionamiento de Geremua versión 2. Estas librerías necesarias serían:
 - La librería **q702.jar**, única por aplicación, deberá encontrarse en la carpeta lib dentro de la aplicación correspondiente
 - La librería **q701.jar**, que existirá en cada uno de los módulos war de la aplicación, estará a nivel de WEB-INF/lib en cada war
 - FILTRO 4: Chequeo del fichero de configuración aaaConfig.properties. Se verificará que existen las propiedades del fichero dominio, adminServerURL y manejado.
 - FILTRO 5: Test para la verificación de la estructuración correcta de una aplicación con Geremua version 2. Se verificará que la estructura de proyectos de EJB debe constar de dos carpetas, una de ellas (META-INF), para los descriptores y otra (con el mismo nombre que el paquete) para los interfaces y el bean.
 - FILTRO 6: Test para la verificación del uso de Operaciones de Presentación. Se verificará la existencia física del directorio de **OperacionesPresentacion** ubicado dentro de la aplicación en cada módulo war dentro de la carpeta WEB-INF. También se comprobará que cada fichero de tipo **struts-xxx.xml** que se encuentra dentro del directorio OperacionesPresentacion está registrado en el archivo **web.xml**.
 - FILTRO 7: Test para la verificación del uso de Operaciones Internas. Se comprobará:
 - La existencia del fichero **configuraciónOI.xml** y que se encuentran registrados en él los nombres de los Ejes.
 - La existencia del fichero de mapas correspondiente **mapasAplicacion.xml**.
 - FILTRO 8: Test para la verificación de la configuración del servicio de Logs y Trazas. Se comprueba:
 - Que existe una ruta previamente creada en la ubicación **datos/aaa/log**, en la que Geremua en su arranque creará los diferentes ficheros para el volcado de logs y trazas.
 - Que en el fichero **aaa.properties** (config/aaa) existe la ruta **datos/aaa/log**, ya que esta ruta aparecerá en este fichero si se especifican ficheros de logs y trazas propios.

- La existencia de los ficheros **configuracionLogs_aplicacion.xml** y **configuracionTraza_aplicacion.xml** en la ubicación **config/aaa**.
- Que en el fichero **GestorArranqueConfiguracion_aplicacion.xml** se encuentran las entradas correspondientes a los componentes de Logs y Trazas. Para ello, se verificará en primer lugar la existencia de este fichero y luego la entrada **FicheroConfiguracion** que tendrá el valor de los ficheros propios de la aplicación para la gestión de Logs y Trazas (**configuracionLogs_aplicacion.xml** y **configuracionTraza_aplicacion.xml**)
- FILTRO 9: Test para la verificación de los ficheros de propiedades de la aplicación. Se comprueba la existencia de los siguientes ficheros:
 - **configuracionCache_aplicacion.xml** -> la aplicación tiene una configuración propia para la gestión de la caché .
 - **configuracionExcepciones.xml** -> la aplicación configura su propio tratamiento de excepciones.
 - **configuracioni18n.xml** -> configuración de la internacionalización de mensajes.
 - **configuracionLogs_aplicacion.xml** -> configuración del servicio de Logs a nivel de aplicación.
 - **configuracionMapeador.xml** -> configuración del fichero de Mapas de la aplicación.
 - **configuracionOI.xml** -> la aplicación dispone de Operaciones Internas para invocar a las diferentes funciones de Negocio.
 - **configuracionTablas_aplicacion.xml** -> para la configuración de catálogos de aplicación
 - **configuracionTraza_aplicacion.xml** -> configuración del servicio de Trazas de la aplicación.
 - **contexto-aplicacion.xml** -> configuración del contexto de aplicación, datos compartidos entre diferentes Operaciones de Presentación de la aplicación para un mismo usuario.
 - **contextoGlobalAplicacion_t30b.xml** -> configuración del contexto global de aplicación (información de la aplicación para todos los usuarios de esa aplicación).

- **GestorArranqueConfiguracion_aplicacion.xml** -> para indicar los componentes aplicación arrancables por parte de Geremua.
- **mapasAplicacion.xml** -> configuración de los mapas de aplicación.
- FILTRO 10: Test de verificación del uso de componentes de Presentación. Se comprobarán los diferentes aspectos:
 - **ejieFramework-html.tld:** Comprobación de que el taglib ejieFramework-html.tld se encuentra en la ruta correcta para poder ser utilizado desde los diferentes wars que compongan la aplicación. Este fichero debe encontrarse en el **WEB-INF/tld** dentro de cada módulo war y es necesario para el uso de componentes como son las Listas Enlazadas, Validador Remoto, etc.
 - **Display Table Tag:** Comprobación informativa de si se está utilizando o no la librería displaytag-1.0-b2.jar ubicada en el WEB-INF/lib de cada módulo war de la aplicación.
 - **Tiles:** Confirmación de si existe o no el fichero necesario para el uso de Tiles con Geremua: **configuracionTiles.xml** (en la carpeta WEB-INF en cada módulo web).
 - **Menus:** Verificar si se hace uso de menús comprobando si existe en el directorio WEB-INF de cada war de la aplicación el fichero **configuracionMenus.xml**.
- FILTRO 11: Test de verificación de la configuración del servicio JMS de Geremua. Se comprobarán los diferentes aspectos:
 - **Q70JMSServer:** la tarea verificaría la propiedad **Name** del componente tipo *JMSServer* para comprobar que existe un servidor JMS con este nombre (Q70JMSServer). Para comprobar si el servidor JMS está activo o no, se verificaría en *JMSServerRuntime* que exista también.
 - **Q70JMSFactory:** componente de tipo *JMSConnectionFactory*. Habría que verificar la propiedad **Name** en *JMSConnectionFactory* para verificar que está creada una factoría de conexiones JMS con el nombre Q70JMSFactory.
 - **q70contextoGlobalJMSTopic:** componente de tipo *JMSTopic*. Habría que verificar estas propiedades:
 - **Name:** existe un topic con este nombre, comprobando la propiedad.
 - **Parent:** También habría que testear la propiedad *Parent*, para verificar que el topic pertenece al servidor JMS Q70JMSServer.

- **q70desconexionUsuarioJMSTopic:** componente de tipo JMSTopic. Habría que verificar las siguientes propiedades:
 - **Name:** para verificar que existe un Topic con el nombre q70desconexionUsuarioJMSTopic *Name*.
 - **Parent:** habría que verificar que el topic pertenece al servidor JMS Q70JMSServer.
- **Q70ForeignJMSServer:** componente de tipo ForeignJMSServer. Habrá que verificar las siguientes propiedades:
 - **Name:** dentro de los elementos de este componente que exista uno en el que la propiedad Name sea *Q70ForeignJMSServer*.
 - **ConnectionFactory:** Habrá que verificar también el atributo ConnectionFactories para comprobar que existe asociada al servidor JMS la factoría Q70ForeignJMSFactory.
 - **Destinations:** para verificar que están incluidos los dos topics necesarios: Q70contextoGlobalForeignJMSTopic y Q70desconexionUsuarioForeignJMSTopic.
 - **ConnectionURL:** para verificar que existe una URL que el servidor weblogic empleará para conectar con el proveedor del servicio JMS. Se podría mostrar como información resultante del testeo el valor de esta propiedad.
- **Q70ForeignJMSFactory:** componente de tipo JMSForeignConnecionFactory. Se comprobarán estas propiedades:
 - **Name:** que exista una factoría con el nombre de Q70ForeignJMSFactory.
 - **Parent:** este atributo tendrá que contener el nombre del servidor JMS al que pertenece esta factoría (Q70ForeignJMSServer).
 - **RemoteJNDIName:** factoría de conexiones remota (en el servidor de la *ConnectionURL*).
- **q70contextoGlobalForeignJMSTopic:** componente de tipo ForeignJMSDestination. Se verificarán estas propiedades:

- **Name:** que exista un elemento de tipo ForeignJMSDestination con este nombre (q70contextoGlobarForeignJMSTopic).
 - **Parent:** tendrá que contener el nombre del servidor JMS en el que está ubicado este topic (Q70ForeignJMSServer).
 - **RemoteJNDIName:** habrá que comprobar que la referencia JNDI del árbol en el servidor remoto (ConnectionURL) se corresponda con el valor de esta propiedad.
- **q70desconexionUsuarioForeignJMSTopic:** componente de tipo ForeignJMSDestination. Se verificarán las siguientes propiedades:
 - **Name:** que exista un elemento de tipo ForeignJMSDestination con este nombre (q70contextoGlobarForeignJMSTopic).
 - **Parent:** tendrá que contener el nombre del servidor JMS en el que está ubicado este topic (Q70ForeignJMSServer).
 - **RemoteJNDIName:** habrá que comprobar que la referencia JNDI en el árbol en el servidor remoto (ConnectionURL) se corresponda con el valor de esta propiedad (q70.q70desconexionUsuarioJMSTopic)

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
testImp:
[testImpTask] ***** FILTRO 1: Chequeo de si existe el filtro de XLNets en todos los
web.xml de los War de la aplicacion
[testImpTask] El fichero: /aplic/t29a/java/javs/t29aEAR/t29aTestingWar/WEB-
INF/web.xml NO tiene el filtro de XLNets
[testImpTask] El fichero: /aplic/t29a/java/javs/t29aEAR/t29aWar/WEB-INF/web.xml NO
tiene el filtro de XLNets
[testImpTask]
[testImpTask] ***** FILTRO 2: Chequeo de que estan declarados todos los modulos de la
aplicacion en el application.xml
[testImpTask] El fichero: /aplic/t29a/java/javs/t29aEAR/META-INF/application.xml SI
tiene el modulo t29aTestingWar
[testImpTask] El fichero: /aplic/t29a/java/javs/t29aEAR/META-INF/application.xml SI
tiene el modulo t29aWar
[testImpTask] El fichero: /aplic/t29a/java/javs/t29aEAR/META-INF/weblogic-
application.xml SI tiene el modulo t29aTestingWar
[testImpTask] El fichero: /aplic/t29a/java/javs/t29aEAR/META-INF/weblogic-
application.xml SI tiene el modulo t29aWar
[testImpTask]
[testImpTask] ***** FILTRO 3: Se muestran todas las librerias que estan puestas en la
aplicacion
```

```
[testImpTask] Libreria: /aplic/t29a/java/javs/t29aEAR/t29aTestingWar/WEB-INF/lib/junitee.jar
[testImpTask] Libreria: /aplic/t29a/lib/junitee-anttask.jar
[testImpTask] Libreria: /aplic/t29a/lib/junit.jar
[testImpTask]
[testImpTask] ***** FILTRO 4: Chequeo del fichero de configuracion
aaaConfig.properties
[testImpTask] El fichero t29aConfig.properties SI contiene los atributos dominio,
adminServerURL y manejado
[testImpTask]
[testImpTask] ***** FILTRO 5: Test para verificacion de estructuracion correcta de
una aplicacion con Geremua version 2
[testImpTask] Para el EJB t29aPetStoreEJB NO existe un subdirectorio t29aPetStoreEJB
[testImpTask] Para el EJB t29aPetStoreEJB SI existe un subdirectorio META-INF
[testImpTask]
[testImpTask] ***** FILTRO 6: Test para verificacion de uso de Operaciones de
Presentacion
[testImpTask] NO existe la carpeta de operaciones de presentacion para el WAR
t29aTestingWar
[testImpTask] SI existe la carpeta de operaciones de presentacion para el WAR t29aWar
[testImpTask] El struts-config.xml SI existe en el web.xml del WAR t29aWar
[testImpTask] El struts-shop.xml SI existe en el web.xml del WAR t29aWar
[testImpTask]
[testImpTask] ***** FILTRO 7: Test para verificacion de uso de Operaciones Internas
[testImpTask] SI existe el fichero configuracionOI.xml
[testImpTask] El EJB t29aPetStoreEJB NO esta registrado en el fichero
configuracionOI.xml
[testImpTask] SI Existe el fichero de mapasAplicacion.xml
[testImpTask]
[testImpTask] ***** FILTRO 8: Test para verificacion de configuracion de servicio de
Logs y Trazas
[testImpTask] SI existe el fichero t29a.properties
[testImpTask] NO existe la ruta datos/t29a/log en el fichero t29a.properties
[testImpTask] NO Existe el fichero de configuracionLogs_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracionTraza_aplicacion.xml
[testImpTask] NO existe la entrada configuracionLogs_aplicacion.xml correspondiente
al componente de logs en el fichero GestorArranqueConfiguracion_aplicacion.xml
[testImpTask] NO existe la entrada configuracionTraza_aplicacion.xml correspondiente
al componente de trazas en el fichero GestorArranqueConfiguracion_aplicacion.xml
[testImpTask]
[testImpTask] ***** FILTRO 9: Test para verificacion de ficheros de propiedades de la
Aplicacion
[testImpTask] NO Existe el fichero de configuracionCache_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracionExcepciones_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracioni18n_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracionLogs_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracionMapeador_aplicacion.xml
```

```

[testImpTask] NO Existe el fichero de configuracionOI_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracionTablas_aplicacion.xml
[testImpTask] NO Existe el fichero de configuracionTraza_aplicacion.xml
[testImpTask] NO Existe el fichero de contexto-aplicacion_aplicacion.xml
[testImpTask] SI Existe el fichero de contextoGlobalAplicacion_t29a.xml
[testImpTask] SI Existe el fichero de GestorArranqueConfiguracion_aplicacion.xml
[testImpTask] SI Existe el fichero de mapasAplicacion.xml
[testImpTask]

[testImpTask] ***** FILTRO 10: Test de verificacion de uso de componentes de
Presentacion

[testImpTask] NO Existe el fichero ejieFramework-html.tld en el WAR t29aTestingWar
[testImpTask] SI Existe el fichero ejieFramework-html.tld en el WAR t29aWar
[testImpTask] NO Existe el fichero displaytag-1.0-b2.jar en el WAR t29aTestingWar
[testImpTask] NO Existe el fichero displaytag-1.0-b2.jar en el WAR t29aWar
[testImpTask] NO Existe el fichero configuracionTiles.xml en el WAR t29aTestingWar
[testImpTask] SI Existe el fichero configuracionTiles.xml en el WAR t29aWar
[testImpTask] NO Existe el fichero configuracionMenus.xml en el WAR t29aTestingWar
[testImpTask] NO Existe el fichero configuracionMenus.xml en el WAR t29aWar


[java] wslsh [not connected]> @ connect

[java] wslsh inter_apps_dpto:/> if $CONNECTED then
[java]
[java]
[java] ***** FILTRO 11: Test de verificacion de configuracion de servicio JMS de
Geremua
[java]
[java] SI existe q70JMSServer
[java] q70JMSServer SI esta activo
[java]
[java] SI existe q70JMSFactory
[java]
[java] SI existe q70contextoGlobalJMSTopic
[java] Parent:  inter_apps_dpto:Name=q70JMSServer,Type=JMSServer
[java]
[java] SI existe q70desconexionUsuarioJMSTopic
[java] Parent:  inter_apps_dpto:Name=q70JMSServer,Type=JMSServer
[java]
[java] SI existe Q70ForeignJMSServer
[java] ConnectionFactories:
[inter_apps_dpto:ForeignJMSServer=Q70ForeignJMSServer,Name=Q70ForeignJMSFactory,Type=
ForeignJMSConnectionFactory]
[java] Destinations:
[inter_apps_dpto:ForeignJMSServer=Q70ForeignJMSServer,Name=q70contextoGlobalForeignJM
STopic,Type=ForeignJMSDestination,

```

```
inter_apps_dpto:ForeignJMSServer=Q70ForeignJMSServer,Name=q70desconexionUsuarioForeig
nJMSTopic,Type=ForeignJMSDestination]

[java] ConnectionURL:  t3://webldes57:8003
[java]
[java] SI existe Q70ForeignJMSFactory
[java] Parent:  inter_apps_dpto:Name=Q70ForeignJMSServer,Type=ForeignJMSServer
[java] RemoteJNDIName:  q70.q70JMSFactory
[java]
[java] SI existe q70contextoGlobalForeignJMSTopic
[java] Parent:  inter_apps_dpto:Name=Q70ForeignJMSServer,Type=ForeignJMSServer
[java] RemoteJNDIName:  q70.q70contextoGlobalJMSTopic
[java]
[java] SI existe q70desconexionUsuarioForeignJMSTopic
[java] Parent:  inter_apps_dpto:Name=Q70ForeignJMSServer,Type=ForeignJMSServer
[java] RemoteJNDIName:  q70.q70desconexionUsuarioJMSTopic
[java]

[java] wslsh [not connected]> quit
[java] bye!

BUILD SUCCESSFUL
```

Ejemplos de los Targets para test de implantación

- Lanzar el test de Implantación
ant **testImp**

1.11 Targets para test de aplicaciones horizontales

- **testLote4**, Tarea para recoger información sobre el estado de los servidores que alberga las aplicaciones de lote4; si están desplegadas dichas aplicaciones y sus pools de conexiones así como el estado y datos sobre los servidores JMS y las colas asociadas.

La información presentada en esta tarea se da por cada uno de los dominios en los que están alojadas las aplicaciones de lote4; de esta manera y por cada dominio se presenta la siguiente información:

- Por cada una de las máquinas del cluster:
 - Nombre de la máquina y su estado.
 - Numero de procesadores
 - Carga media de los procesadores (junto con su porcentaje sobre el total)
 - Memoria física libre (junto con su porcentaje sobre el total)
 - Memoria física usada (junto con su porcentaje sobre el total)
 - Memoria física total
 - Memoria libre de la máquina virtual (junto con su porcentaje sobre el total)
 - Memoria usada de la máquina virtual (junto con su porcentaje sobre el total)
 - Memoria total de la máquina virtual
 - Versión de Java
 - Numero actual de demonios java
 - Numero actual de threads java (demonios y no demonios)
- Por cada una de las aplicaciones:
 - Nombre de la aplicación y si está desplegada
 - Nombre del pool y se está desplegado por cada una de las instancias.
- Por cada servidor JMS:
 - Nombre del servidor y estado
 - Conexiones actuales
 - Pico máximo de conexiones
 - Conexiones totales
- Por cada cola JMS
 - Nombre de la cola y estado
 - Bytes almacenados actualmente
 - Pico máximo de bytes almacenados
 - Bytes pendientes almacenados

- Bytes recibidos
- Mensajes almacenados actualmente
- Pico máximo de mensajes almacenados
- Mensajes pendientes almacenados
- Mensajes recibidos
- Sesiones instanciadas actualmente
- Pico máximo de sesiones instanciadas

Un ejemplo del resultado de la ejecución de esta tarea sería:

```
[java] ***** INTER_TELETRAMITACION
*****

[java] * Maquinas activas del cluser: 2
[java] Servidor: inter_teletramitacion_apps111 , Estado: RUNNING
[java] Numero de procesadores: 8
[java] Carga media de los procesadores: 0.03453596247068028 ( 3.4535962470680284 %)
[java] Memoria fisica libre: 7993020416 ( 24 %)
[java] Memoria fisica usada: 24681611264 ( 75 %)
[java] Memoria fisica total: 32674631680
[java] Memoria libre de la maquina virtual: 428608032 ( 79 %)
[java] Memoria usada de la maquina virtual: 108262880 ( 20 %)
[java] Memoria total de la maquina virtual: 536870912
[java] Version de Java: 1.4.2_10
[java] Numero actual de demonios Java: 99
[java] Numero actual de threads Java (demonios y no demonios): 110
[java]
[java] Servidor: inter_teletramitacion_apps112 , Estado: RUNNING
[java] Numero de procesadores: 8
[java] Carga media de los procesadores: 0.034544254317692445 ( 3.4544254317692444 %)
[java] Memoria fisica libre: 7993020416 ( 24 %)
[java] Memoria fisica usada: 24681611264 ( 75 %)
[java] Memoria fisica total: 32674631680
[java] Memoria libre de la maquina virtual: 131995936 ( 24 %)
[java] Memoria usada de la maquina virtual: 404874976 ( 75 %)
[java] Memoria total de la maquina virtual: 536870912
[java] Version de Java: 1.4.2_10
[java] Numero actual de demonios Java: 100
[java] Numero actual de threads Java (demonios y no demonios): 111
[java]
[java] ***** APLICACIONES *****
[java]
[java] * Aplicacion: r02f , Desplegado: true
```

```

[java]
[java] * Aplicacion: r02k ,   Desplegado: true
[java]
[java] * Aplicacion: r02n ,   Desplegado: true
[java]
[java] ***** SERVIDORES JMS *****
[java] Servidor: inter_teletramitacion_apps111.jms , Estado:
State:HEALTH_OK,ReasonCode:[
[java] Conexiones actuales: 0
[java] Pico maximo de conexiones: 0
[java] Conexiones totales: 0
[java]
[java] Servidor: inter_teletramitacion_apps112.jms , Estado:
State:HEALTH_OK,ReasonCode:[
[java] Conexiones actuales: 0
[java] Pico maximo de conexiones: 0
[java] Conexiones totales: 0
[java]
[java]
[java] ***** COLAS JMS *****

[java] wlsh [not connected]> quit
[java] bye!

[java] wlsh [not connected]> @ connect

[java] wlsh intra_teletramitacion:/> if $CONNECTED then
[java]
[java] ***** INTRA_TELETRAMITACION
*****
[java] * Maquinas activas del cluster: 2
[java] Servidor: intra_teletramitacion_apps111 , Estado: RUNNING
[java] Numero de procesadores: 8
[java] Carga media de los procesadores: 0.2023564049226158 ( 20.23564049226158 %)
[java] Memoria fisica libre: 13916573696 ( 38 %)
[java] Memoria fisica usada: 21831237632 ( 61 %)
[java] Memoria fisica total: 35747811328
[java] Memoria libre de la maquina virtual: 281375112 ( 26 %)
[java] Memoria usada de la maquina virtual: 792366712 ( 73 %)
[java] Memoria total de la maquina virtual: 1073741824
[java] Version de Java: 1.4.2_10
[java] Numero actual de demonios Java: 100
[java] Numero actual de threads Java (demonios y no demonios): 122
[java]

```

```
[java] Servidor:  intra_teletramitacion_apps112 , Estado:  RUNNING
[java] Numero de procesadores:  8
[java] Carga media de los procesadores:  0.20234870465476903  ( 20.234870465476902 %)
[java] Memoria fisica libre:  13916573696  ( 38 %)
[java] Memoria fisica usada:  21831237632  ( 61 %)
[java] Memoria fisica total:  35747811328
[java] Memoria libre de la maquina virtual:  346995336  ( 32 %)
[java] Memoria usada de la maquina virtual:  726746488  ( 67 %)
[java] Memoria total de la maquina virtual:  1073741824
[java] Version de Java:  1.4.2_10
[java] Numero actual de demonios Java:  100
[java] Numero actual de threads Java (demonios y no demonios):  122
[java]
[java] ***** APLICACIONES *****
[java]
[java] * Aplicacion:  r02d ,    Desplegado:  true
[java] Nombre del pool:  r02dPool
[java] Pool:  intra_teletramitacion_apps111 , Estado: Running
[java] Pool:  intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion:  r02e ,    Desplegado:  true
[java] Nombre del pool:  r02ePool
[java] Pool:  intra_teletramitacion_apps112 , Estado: Running
[java] Pool:  intra_teletramitacion_apps111 , Estado: Running
[java]
[java] * Aplicacion:  r02f ,    Desplegado:  true
[java]
[java] * Aplicacion:  r02g ,    Desplegado:  true
[java] Nombre del pool:  r02gPool
[java] Pool:  intra_teletramitacion_apps112 , Estado: Running
[java] Pool:  intra_teletramitacion_apps111 , Estado: Running
[java]
[java] * Aplicacion:  r02h ,    Desplegado:  true
[java] Nombre del pool:  r02hPool
[java] Pool:  intra_teletramitacion_apps111 , Estado: Running
[java] Pool:  intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion:  r02i ,    Desplegado:  true
[java] Nombre del pool:  r02iPool
[java] Pool:  intra_teletramitacion_apps111 , Estado: Running
[java] Pool:  intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion:  r02j ,    Desplegado:  true
```



```
[java] Nombre del pool: r02jPool
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java]
[java] * Aplicacion: r02l , Desplegado: true
[java] Nombre del pool: r02lPool
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion: r02m , Desplegado: true
[java] Nombre del pool: r02mPool
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java]
[java] * Aplicacion: r02o , Desplegado: true
[java] Nombre del pool: r02oPool
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion: r02p , Desplegado: true
[java] Nombre del pool: r02pPool
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion: r02q , Desplegado: true
[java]
[java] * Aplicacion: r02r , Desplegado: true
[java] Nombre del pool: r02rPool
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java]
[java] * Aplicacion: r02s , Desplegado: true
[java] Nombre del pool: r02sPool
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion: r02t , Desplegado: true
[java] Nombre del pool: r02tPool
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion: r02u , Desplegado: true
[java]
```

```
[java] * Aplicacion: r02v ,   Desplegado: true
[java]
[java] * Aplicacion: r02w ,   Desplegado: true
[java] Nombre del pool: r02wPool
[java] Pool: intra_teletramitacion_apps111 , Estado: Running
[java] Pool: intra_teletramitacion_apps112 , Estado: Running
[java]
[java] * Aplicacion: r02y ,   Desplegado: true
[java]
[java] ***** SERVIDORES JMS *****
[java] Servidor: intra_teletramitacion_apps111.jms , Estado:
State:HEALTH_OK,ReasonCode:[
[java] Conexiones actuales: 11
[java] Pico maximo de conexiones: 16
[java] Conexiones totales: 11268
[java]
[java] Servidor: intra_teletramitacion_apps112.jms , Estado:
State:HEALTH_OK,ReasonCode:[
[java] Conexiones actuales: 13
[java] Pico maximo de conexiones: 14
[java] Conexiones totales: 12946
[java]
[java]
[java] ***** COLAS JMS *****
[java] Nombre: r02JMSServer_apps111 , Estado: State:HEALTH_OK,ReasonCode:[
[java] Bytes almacenados actualmente: 1610
[java] Pico maximo de bytes almacenados: 2305537
[java] Bytes pendientes almacenados: 0
[java] Bytes recibidos: 15287536
[java] Mensajes almacenados actualmente: 7
[java] Pico maximo de mensajes almacenados: 9
[java] Mensajes pendientes almacenados: 0
[java] Mensajes recibidos: 671
[java] Sesiones instanciadas actualmente: 0
[java] Pico maximo de sesiones instanciadas: 0
[java]
[java] Nombre: r02JMSServer_apps112 , Estado: State:HEALTH_OK,ReasonCode:[
[java] Bytes almacenados actualmente: 690
[java] Pico maximo de bytes almacenados: 57112
[java] Bytes pendientes almacenados: 0
[java] Bytes recibidos: 3054919
[java] Mensajes almacenados actualmente: 3
[java] Pico maximo de mensajes almacenados: 5
[java] Mensajes pendientes almacenados: 0
```

```
[java] Mensajes recibidos: 218
[java] Sesiones instanciadas actualmente: 0
[java] Pico maximo de sesiones instanciadas: 0
[java]

[java] wslsh [not connected]> quit
[java] bye!
```

```
BUILD SUCCESSFUL
Total time: 19 seconds
```

La interpretación de estos resultados sería la misma que la detallada anteriormente en los tests de infraestructura en ejecución en lo referente a las máquinas, las aplicaciones y las colas jms.

1.12 Targets varios

- **copiarLibApp**, Permite gestionar dependencias con terceras aplicaciones en los traspasos a pruebas y producción. Esta tarea lo que hace es copiar la librería /(explopr o explo)/aplic/bbb/java/javs/bbb.jar a la carpeta lib de la aplicación que se está traspasando. Parámetros requeridos:
 - **dependeApp**, código de aplicación del que depende la aplicación que está siendo traspasada.
- **console**, Muestra información de la aplicación: estado de la instancia, módulos de la aplicación desplegados, pool y dataSource.
- **ficherosexplo**, Tarea para traspasar uno o varios ficheros a la carpeta explopr. Parámetros requeridos:
 - **files**, lista de ficheros a traspasar.
- **copiarFicheros**, Tarea para copiar los ficheros de una extensión determinada de una ruta origen a una ruta destino.
 - **rutaOrigen**, Ruta origen de los ficheros a copiar
 - **rutaDestino**, Ruta destino de los ficheros a copiar
 - **extensión**, Extensión de los ficheros a copiar
- **cvs**, Tarea que extrae la aplicación correspondiente del repositorio CVS.
- **svn**, Tarea que extrae la aplicación correspondiente del repositorio Subversion.
- **maven**, Tarea que permite ejecutar tareas maven
 - **tarea**, Tarea de maven a lanzar
- **build-xmlbean**, Tarea que permite crear un jar con los objetos xmlbeans a partir de un esquema; los esquemas deberán estar situados en la carpeta /aplic/aaa/html/xmlbean_schemas de la propia aplicación y el jar resultante se depositará dentro de la carpeta /aplic/aaa/java/javs
- **copiar_build**, Tarea que copia el build.xml con las tareas actualizadas de aaa a la aplicación desde donde se lanza. Si se quiere realizar una primera copia del build.xml (debido a que no existe el target copiar_build en el build.xml de la aplicación) se podrá ejecutar el script copiar_build.sh que se encuentra en la carpeta /aplic pasándole como parámetro el código de aplicación donde se quiere copiar de la siguiente manera: ./copiar_build.sh cod_aplic (Ej: ./copiar_build.sh s73a). Se realizará a su vez una copia del build.xml antiguo renombrándolo como build.xml.ant
- **copiar_testing**, Tarea que copia el aaaTestingWar genérico para el lanzamiento de pruebas a la aplicación desde la que se lance cambiándole el aaa por el código de aplicación y que modifica el application.xml de la aplicación añadiéndole el modulo copiado (el aaaTestingWar). Se controlará si

ya existe el módulo en la aplicación y en el fichero application.xml. Se realizará a su vez una copia del application.xml antiguo renombrándolo como application.xml.ant.

Ejemplos de los Targets Varios

- Paso a pruebas o producción de una aplicación aaa que hace uso de un API generado por la aplicación bbb:

```
ant -DdependeApp=bbb copiarLibApp
```

Nota: esta tarea está pensada para ser incluida en el fichero compile.txt durante los traspasos antes de lanzar ninguna tarea de compilación (ant all). En Desarrollo no podrá ser usada ya que se estaría intentado crear una dependencia con una librería no estable.

- acceder a la información de la consola en modo texto:

```
ant console
```

- Traspaso de dos ficheros a la carpeta explor:

```
ant -Dfiles=aaaSaludaWar/presentacionJSP/aaaSaluda.jsp,aaaSaludaWar/WEB-INF/web.xml ficherosexplo
```

- Copia de ficheros .class de una ruta origen a una ruta destino:

```
ant -DrutaOrigen=exe/APP-INF/classes/com/ejie/s73a/jws -DrutaDestino=exe/APP-INF/classes/com/ejie/s73a/dao -Dextension=class copiarFicheros
```

- Extracción de una aplicación del repositorio de CVS:

```
ant cvs
```

- Extracción de una aplicación del repositorio de Subversion:

```
ant svn
```

- Ejecución de una tarea maven

```
ant -Dtarea=ejie-dist maven
```

- Generar un jar con los objetos xmlbeans a partir de sus esquemas.

```
ant build-xmlbean
```

- Copiado del build.xml:

```
ant copiar_build
```

- Copiado del aaaTestingWar y añadir el módulo en el application.xml de la aplicación:

```
ant copiar_testing
```

Anexo I: Registro de incidencias y situaciones comunes

En este anexo se detallará un registro de incidencias y situaciones comunes en el que se especificará una descripción de la incidencia junto con la causa de que se produzca y las correspondientes medidas de actuación.

Las incidencias que se detallan a continuación son las incidencias más habituales que se suelen producir aunque no todas las que se han producido; el registro de todas las incidencias y consultas que se han enviado al grupo de Consultoría de Areas de Conocimiento (CAC) se encuentran disponibles en el repositorio de conocimiento Mantis.

En este repositorio de conocimiento se categorizan todas las incidencias recibidas por el grupo de Consultoría de Areas de Conocimiento y en el que se puede consultar el error que se está produciendo filtrando mediante un buscador, categoría, etc...

Este repositorio de conocimiento es de acceso público, la URL de acceso es la siguiente:

http://mantis.ejiedes.net/mantis/login_page.php

El usuario y contraseña para acceder son los siguientes:

Usuario: ejie

Contraseña: ejie

En un principio este anexo de registro se detallará de forma secuencial pudiéndose complementar a futuro con un mecanismo más ágil de explotación de los datos generados.

Este anexo también pretende ser un complemento a cómo actuar ante un error producido; lo primero que se debe de hacer una vez detectado un error es revisar las incidencias relacionadas en este anexo así como el repositorio de Mantis descrito anteriormente.

En el caso de que el error no se encuentre reflejado y si no se encuentra una solución se deberá reportar la correspondiente M53 al grupo de Consultoría de Areas de Conocimiento.

Incidencias relacionadas con las tareas de ant

A la hora de analizar una posible incidencia ocasionada por una tarea de ant hay que tener en cuenta que para detectarla es preciso revisar toda la información que desprende la propia tarea; esto es debido a que a pesar de que el resultado sea un “Build Successfull” no tiene por qué haberse ejecutado bien el propósito de la tarea sino que lo que se ha ejecutado “bien” es la tarea, es decir, no ha habido nada que rompa su ejecución pero no ha hecho lo que debería hacer.

Este tipo de incidencias son unas de las más frecuentes en lo referente a este tipo de incidencias.

A continuación, se presentará de una manera secuencial las incidencias más significativas relacionadas con la ejecución de las tareas de ant.

1 – Fallo en el despliegue o borrado de una aplicación

La tarea de ant para el despliegue de aplicaciones acaba su ejecución con un “Build Successfull” pero la aplicación no se ha desplegado correctamente.

Si se revisa la ejecución de la tarea observamos el siguiente mensaje:

```
Undeploy/Remove/Deploy request for "codAplicación" on cluster "cluster"
deferred because no servers in the cluster are reachable.
```

Lo que nos está informando con este mensaje es que no se ha realizado el despliegue/borrado debido a que no hay servidores activos (por lo menos uno) para desplegar/borrar la aplicación del cluster.

Existe la posibilidad de consultar el estado de las instancias donde se encuentra nuestra aplicación lanzado la tarea de testeo de los servidores donde aparte de informar de si están o no activos los servidores se informará de otros datos como la memoria, threads activos, etc...

Esta tarea se lanza de la siguiente manera:

```
ant -Dparam=server test
```

En el caso de que alguna de las instancias (o las dos) no se encuentre arrancada será preciso llamar al CAU para que procedan a arrancarla.

Otro de los errores de los que se tiene constancia es si se despliega en modo no exploded, que no pueda generar bien el .ear, esto es debido habitualmente a un problema de permisos de la carpeta que contiene el .ear o del propio fichero y el error que aparece es habitualmente:

```
[java] Deployment failed on Cluster appsl
[java] Exception:weblogic.management.ApplicationException: Cannot
create JarFile for/aplic/q91/dist/q91.ear
```

Por otro lado, también se produce un error en el despliegue si el DataSource o el pool de conexiones al que se hace referencia en la aplicación (el nombre JNDI) no existen o están corruptos o no disponibles, la descripción del error es del tipo:

```

[java] Initiated Task: [94] [Deployer:149026]Deploy application u06t on
apps2.
[java] Task 94 failed: [Deployer:149026]Deploy application u06t on
apps2.
[java] Deployment failed on Cluster apps2
[java] weblogic.management.ManagementException:
weblogic.management.DeploymentException:
[java] Exception:weblogic.management.ApplicationException: activate
failed for u06tEntityEJB
[java] Module: u06tEntityEJB Error: Exception activating module:
EJBModule(u06tEntityEJB,status=PREPARED)

[java] Unable to deploy EJB: u06tTemaConsultaEntity from
u06tEntityEJB:

[java] [EJB:011028]The DataSource with the JNDI name:
u06t.u06tDataSource could not be located. Please ensure that the
DataSource has been deployed successfully and that the JNDI name in your
EJB Deployment descriptor is correct.

[java] weblogic.management.ManagementException:
weblogic.management.DeploymentException:
[java] Exception:weblogic.management.ApplicationException: activate
failed for u06tEntityEJB
[java] Module: u06tEntityEJB Error: Exception activating module:
EJBModule(u06tEntityEJB,status=PREPARED)

[java] Unable to deploy EJB: u06tTemaConsultaEntity from
u06tEntityEJB:

[java] [EJB:011028]The DataSource with the JNDI name:
u06t.u06tDataSource could not be located. Please ensure that the
DataSource has been deployed successfully and that the JNDI name in your
EJB Deployment descriptor is correct.

```

La forma de solucionar este error es creando el pool y el DataSource en el caso de que no existan, y en el caso de que existan regenerarlos borrándolos y creándolos de nuevo.

Existe la posibilidad de consultar el estado de los pools de conexiones lanzando la tarea de testeo de los pools donde aparte de informar de si están o no activos se informará de otros datos como conexiones no cerradas, etc...

Esta tarea se lanza de la siguiente manera:

```
ant -Dparam=pool test
```

En algunas ocasiones a la hora de desplegar una aplicación se da un error poco común pero problemático, el error es el siguiente:

```

[java] Initiated Task: [550] [Deployer:149026]Deploy application u76t on
apps2.
[java] Task 550 running: [Deployer:149026]Deploy application u76t on

```



```

apps2.
[java] Deployment failed on Cluster apps2
[java] weblogic.management.ManagementException:
weblogic.management.DeploymentException:
[java] Exception:weblogic.management.ApplicationException: start() failed.
[java] Module: u76tWar Error: weblogic.management.DeploymentException -
with nested exception:
[java] [java.lang.NullPointerException]
[java] weblogic.management.ManagementException:
weblogic.management.DeploymentException:
[java] Exception:weblogic.management.ApplicationException: start() failed.
[java] Module: u76tWar Error: weblogic.management.DeploymentException -
with nested exception:
[java] [java.lang.NullPointerException]

[java] Java Result: 1

```

El error es debido a que se ha desincronizado el cluster donde se aloja la aplicación con el administrativo.

Para que se corrija el error es necesario reiniciar el administrativo, una vez reiniciado la aplicación se puede desplegar perfectamente.

2 – El despliegue o borrado de una aplicación no termina de finalizar

Esta situación sucede cuando el weblogic no es capaz de resolver las peticiones que se le asignan y se produce un bloqueo que hace que se encolen las tareas que se envían.

La única solución a este problema es que se reinicie la instancia que administra el dominio haciendo la correspondiente petición al CAU.

Es inútil hacer los habituales control+d y control+c porque a pesar de que nos devuelve a la línea de comandos la tarea se sigue ejecutando internamente.

Existe la posibilidad de consultar las tareas asociadas a nuestra aplicación y que no han sido finalizadas; esto se hace mediante la tarea de test con el parámetro task.

Esta tarea se lanza de la siguiente manera:

```
ant -Dpath=task test
```

Con esta tarea dispondremos de suficiente información para determinar si las tareas que se están lanzando finalizan o no.

En el caso de que efectivamente se estén encolando las peticiones, la única manera de solucionarlo es reiniciando el servidor administrativo del dominio y para ello será preciso llamar al CAU.

3 – Fallo al crear un pool de conexiones

La creación de un pool de conexiones puede fallar por diversas causas, entre ellas que ya exista, que algún parámetro para su creación esté mal informado o que las instancias estén caídas.

Cuando el pool de conexiones ya existe se produce un error de este tipo:

```

file:/dominio_wls8/j2se/jakarta-ant-1.5.3-1/ejie/build-ejie.xml:519:
Unable to create mbean: weblogic.management.MBeanCreationException: -
with nested exception:

```

```
[javax.management.InstanceAlreadyExistsException:  
inter_apps_dpto:Name=t29aPool,Type=JDBCConnectionPool]
```

Otro de los errores más habituales es que algún parámetro de la creación del pool esté mal informado, en tal caso algunos de los errores que se producen se muestran a continuación:

```
file:/dominio_wls8/j2se/jakarta-ant-1.5.3-1/ejie/build-  
ejie_softbase8.xml:469: Error invoking MBean command:  
java.lang.IllegalArgumentException: Property Name and value not valid for  
the MBean. Value apps8 for  
parameter[Targets].java.lang.IllegalArgumentException:  
javax.management.MalformedObjectNameException: Domain name either missing  
or null
```

```
file:/usr/j2se/jakarta-ant-1.5.3-1/ejie/build-ejie.xml:561: Property not  
set: ${poolId}
```

Otro posible caso de error se produce cuando el listener de la base de datos sobre la que se crea el pool de conexiones está caído, es un error poco habitual pero puede suceder, la descripción del error que da es el siguiente:

```
file:/dominio_wls8/j2se/jakarta-ant-1.5.3-1/ejie/build-ejie.xml:519: Error  
invoking MBean command: weblogic.management.ManagementRuntimeException:  
weblogic.management.DistributedManagementException: Distributed Management  
[1 exceptions]  
Error creating connection pool r21bPool: 0:Could not create pool  
connection. The DBMS driver exception was: Listener refused the connection  
with the following error:  
ORA-12514, TNS:listener does not currently know of service requested in  
connect descriptor  
The Connection descriptor used by the client was:  
r21d
```

La manera de solucionar este error es haciendo la consiguiente petición a Soporte BBDD para que levanten el listener y además suele ser habitual que se tengan que relanzar las instancias del weblogic donde se quieren crear esos pools.

También se puede dar el caso de que las instancias donde se tienen que desplegar el pool de conexiones estén caídas, en ese caso el error que se muestra es el siguiente:

```
file:/dominio_wls8/j2se/jakarta-ant-1.5.3-1/ejie/build-ejie.xml:465:  
Failed to connect to admin server: javax.naming.CommunicationException  
[Root exception is java.net.ConnectException: t3://webldes49:8002: [^]  
Destination unreachable; nested exception is:
```

```
java.net.ConnectException: connect: Connection refused; No  
available router to destination]
```

Incidencias relacionadas con PLATEA

A la hora de analizar una incidencia relacionada con PLATEA existen diferentes situaciones según sea la funcionalidad de PLATEA que se esté analizando, es por ello que las incidencias relacionadas con PLATEA se van a categorizar en función de las diferentes aplicaciones que lo conforman.

Una vez detectada una incidencia lo primero que hay que hacer si se desconoce su origen es lanzar la tarea para analizar el estado de PLATEA, la tarea ant de testLote4.

Esta tarea se lanza de la siguiente manera:

ant testLote4

Con esta tarea dispondremos de información relativa al estado de las aplicaciones que conforman PLATEA, sus pools de conexiones, sus colas JMS, los dominios y servidores que las albergan, etc...

Además, existe tanto este anexo como el repositorio de Mantis descritos anteriormente para poder consultar todas las incidencias que se han producido con las aplicaciones que utilizan PLATEA y así detectar si es un error que ya se ha producido y conocer su causa y resolución.

1 – Toolkit de Formularios

Cuando se está trabajando con el toolkit de formularios se pueden dar diferentes errores que se pueden clasificar de la siguiente manera:

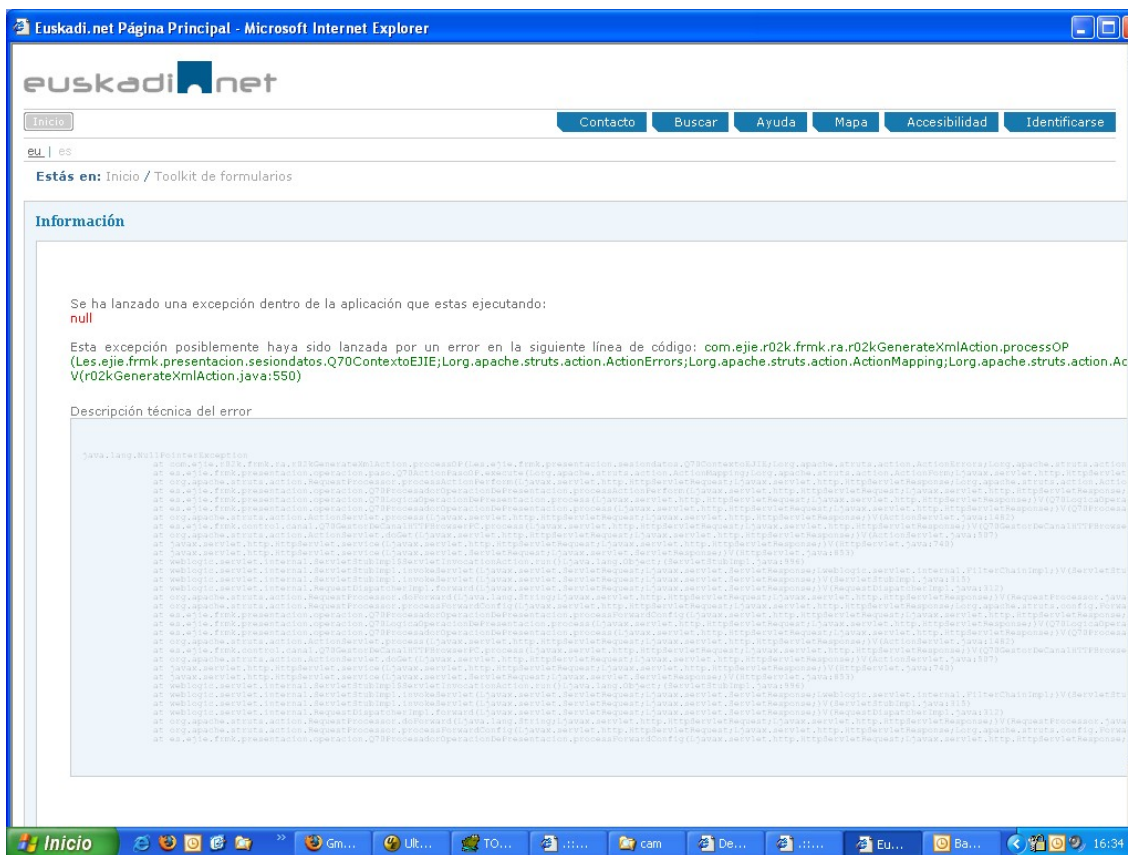
- Antes de visualizar el formulario:

Los errores que se dan antes de visualizar el formulario pueden ser debidos a varias situaciones, las más habituales son que el xml que conforma el formulario sea incorrecto, que la precarga del formulario (si la tiene) se esté haciendo mal (el xml que devuelve el SVT es incorrecto) y que la validación de entrada no devuelva correctamente la respuesta (el SVT).

En los casos en los que se depende de un SVT no tiene por qué ser solo debido a que se devuelve mal la respuesta sino que puede ser debido a que la aplicación donde se aloja el SVT no esté desplegada, el servlet o servicio web que lo conforma no esté bien realizado o bien que la URL del servicio web no sea la correcta en el Catálogo de Servicios, etc...

Una manera de asegurarse de que el xml de una precarga es correcto es haciendo un pequeño truco, este XML se puede sacar haciendo una solicitud con el toolkit sin el SVT de precarga (para que aparezca el formulario con los campos vacíos) se rellenan los que se quieren precargar y cuando hay que firmarla se hace botón derecho-ver código fuente, se busca el ejgvdocument (que es un xml) y se coge la parte que va desde el Applicantinfo hasta el detailInfo; en ese xml vendrán los datos introducidos en el formulario y es donde se pondrán los datos que se quieren precargar porque éste es el xml que puede interpretar el toolkit.

Para la mayoría de los casos el error que sale es el que se adjunta a continuación:



- Una vez visualizado el formulario:

Los errores que se suelen producir una vez visualizado el formulario pueden ser debidos a varias causas, entre ellas, que el xml del formulario a pesar de ser un xml válido no posee algún elemento indispensable para continuar, que no se carguen los combos del formulario, etc...

Para el caso de que a pesar de ser un xml válido no posee algún elemento indispensable, el error que da es:

Se ha lanzado una excepción dentro de la aplicación que estas ejecutando:
La solicitud no es correcta :{Debe existir el elemento XXX=}

Esta excepción posiblemente haya sido lanzada por un error en la siguiente línea de código:

```
com.ejie.r02k.utilities.R02kComprobarSolicitud.comprobarContextSubmission(
Ljava.lang.String;Ljava.lang.String;)Ljava.lang.String;(R02kComprobarSolic
itud.java:276)
```

Para el error de que no se carga los combos, siempre es debido a que el SVT asociado a esa carga del combo no está devolviendo correctamente el xml de carga del combo. Este error siempre se identifica porque además de que no se ven los elementos en el combo, se muestra un mensaje en rojo que indica que no se ha podido cargar dicho combo.

Otro de los errores que se suelen dar es que no esté dado de alta el modelo de registro de vuestro organo, etc... en el libro de registro; este error se da a la hora de presentar la solicitud cuando se intenta registrar en el libro de registro. El error que da es el siguiente:

```
Se ha lanzado una excepción en la aplicación n6li de código
TM.getModelCode:
n6li-TM.getModelCode # n6li-TM.getModelCode # n6li-TM.getModelCode # [WLI-
Core:489214]ProcessControl invocation failed[n6li-TM.getModelCode #
Unexpected exception in
com.ejie.ejb.N6liModelCodeSystemBean.getRegistryModelInfo():
java.lang.Exception: Error. No existe el centro orgánico: 331 at
com.ejie.ejb.N6liModelCodeSystemBean.getDescriptionModelByCodeNumber (Ljava
.lang.String;)Lcom.ejie.classes.model.N6liDescriptionModel;
```

2 – Cliente de Tramitación

Cuando se está trabajando con el cliente de tramitación se pueden producir diferentes errores que habitualmente se producen o bien a la hora de sugerir una tarea o bien a la hora de ejecutarla:

- Errores producidos al sugerir una tarea.

Estos errores se suelen producir por una mala implementación del xml de la tarea o bien porque su ejecución implica la llamada a uno o varios SVTs que no funcionan correctamente.

- Errores producidos al ejecutar una tarea.

Estos errores se producen habitualmente cuando no se proveen los datos necesarios para ejecutar una tarea o bien los datos proporcionados no son del tipo o tamaño requerido.

Por ejemplo un error del tipo en que algún dato no tiene el tamaño requerido y se excede de su máximo es:

```
Se ha lanzado una excepción en INTEGRACIÓN en el método
storePostalNotification de la aplicación q99s: q99s-
storePostalNotification # q99s-storePostalNotification # [WLI-Core:489214]
ProcessControl invocation failed [R02S-10001008 # Error SQL {ORA-
12899:value too large for column "R02I"."R02I00T00"- "PROCEDURENAME_NO"
(actual: 319, maximum: 256)}]
```

3 – Llamadas a los servicios de integración

Cuando se realizan llamadas a los servicios se pueden producir diferentes errores debidos a diferentes motivos, habitualmente las causas se deben a situaciones como que esté mal la URL del servicio, que el token de sesión que se envía al servicio esté caducado, que el token de sesión no sea el válido para llamar al servicio o que los parámetros sean incorrectos.

La mayoría de los errores que se producen en las llamadas de los servicios son errores genéricos devueltos por la plataforma de integración. El error es un "Unhandled process exception" cuya respuesta completa es la siguiente:

```
<?xml version="1.0" encoding="utf-8"?><SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" [^]
xmlns:xsd="http://www.w3.org/2001/XMLSchema" [^]
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> [^]
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
```

```

        <faultcode
xmlns:fc="http://www.bea.com/2003/04/jwFaultCode/">fc:JWSError</faultcode>
[ ^]
        <faultstring>
            Unhandled process exception
        </faultstring>
        <detail>
            <jwErr:jwErrorDetail
xmlns:jwErr="http://www.bea.com/2002/04/jwErrorDetail/"> [ ^]
                Unhandled process exception
            </jwErr:jwErrorDetail>
        </detail>
    </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Para poder consultar el error concreto que está dando realmente (el que se da en la plataforma de integración) podemos consultarlo en la siguiente URL:

<http://www.tramita.jakina.ejiedes.net/r02sLogViewerWar/erroresJSP/r02slnit.do>

En esta aplicación podremos consultar el error que se está dando por servicio y hora y de esta manera determinar mejor cuál es la causa del error.

- Mal la URL del servicio:

Si en las trazas aparece un “Error 404 not found” cuando se está llamando a un servicio eso significa que la URL del servicio no es correcta y por lo tanto hay que revisarla y sustituirla por la correcta.

- Token caducado o no válido para llamar a ese servicio:

La traza que aparece en este caso es la traza genérica descrita anteriormente, pero la traza que se ve si se entra a la página de logs descrita anteriormente es la siguiente:

```

<SOAP-ENV:Fault>
  <faultcode
xmlns:fc="http://www.bea.com/2003/04/jwFaultCode/">fc:JWSError</faultcode>
[ ^]
  <faultstring>
    q99e-getAuthorizationException # [WLI-Core:489214]ProcessControl
    invocation failed[q99e-getAuthorizationException # Autorizacion denegada
    .]
  </faultstring>
  <detail>
    <jwErr:jwErrorDetail
xmlns:jwErr="http://www.bea.com/2002/04/jwErrorDetail/"> [ ^]
      q99e-getAuthorizationException # [WLI-Core:489214]ProcessControl
      invocation failed[q99e-getAuthorizationException # Autorizacion denegada
      .]
    </jwErr:jwErrorDetail>
  </detail>
</SOAP-ENV:Fault>

```

```
</detail>
</SOAP-ENV:Fault>
```

- Parámetros incorrectos:

Uno de los errores habituales es que no se estén enviando los parámetros correctos a los servicios, por ejemplo, si un parámetro tiene que ser un array de bytes en Base64 y se le pasa únicamente como array de bytes, etc...

Habitualmente lo que devuelve son respuestas vacías con el nombre del método llamado, por ejemplo:

```
<ns:createEjgvDocumentWSResponse xmlns:ns="http://www.openuri.org/"> [^]
</nscreateEjgvDocumentWSResponse>
```

Aunque no siempre se devuelve así, por lo que se trata de uno de los errores más difíciles de detectar y que solo se puede solucionar revisando bien el contenido requerido y el enviado.

A la hora de verificar una firma también se han detectado errores debidos al encoding del array de bytes que se pasa como parámetro, la descripción de este error es el siguiente:

```
Aplicación q99m Servicio IS.q99mPublishNotification Error q99m-
publishNotification - q99d-publishNotification # [WLI-
Core:489214]ProcessControl invocation failed[q99s-
completeNotificationDocument # q99s-completeNotificationDocument # [WLI-
Core:489214]ProcessControl invocation failed[R02F-0 # [1204] Error en el
proceso de verificación de la firma, en el vector de códigos de error
tendrá información más concreta: [1202] Error en la verificación
matemática de la firma (1202)]] Fecha 01/04/08 11:01 Reglas suscripción
{ErrorCategory=ERROR, reglas=467}
```

4 – SVTs:

Los errores mas habituales de los SVTs vienen relacionados casi siempre con las respuestas que envían y los parámetros de entrada, algunos de estos errores ya se han comentado en puntos anteriores.

Otro de los focos importantes de problemas tiene que ver con que no se ponga o no se tenga en cuenta como primer parámetro de los SVTs el parámetro integrationToken, este parámetro es necesario para todos los SVTs.

Además, en los SVTs que son Web Services es necesario que el nombre del parámetro que se devuelve sea siempre el nombre del método seguido de Result, nombremetodoResult.

Por otra parte cuando una aplicación es la primera en su instancia en utilizar web services es habitual que cuando se envían datos con caracteres especiales como "ñ", etc... se de este error:

```
Problem reading the XML: Malformed UTF-8 char -- is an XML encoding
declaration missing?
```

Este error es debido a que hay que cambiar el charset (porque no reconoce alguno de los caracteres devueltos) para las llamadas a los Web Services en los arranques de weblogic que suministren y consuman ese Web Service.

Para arreglar el problema hay que pedir a implantación que pongan en los arranques de weblogic de la instancia en cuestión esta línea:

```
-Dweblogic.webservice.il8n.charset=utf-8
```

BORRADOR

Incidencias relacionadas con los procesos batch, K31 y O75

Los errores que se suelen dar suelen ser por los siguientes motivos:

- No se poseen los permisos adecuados para ejecutarse en la O75:

El error que suele salir en estos casos en la consola de la O75 es **shm open fail** y se solventa poniendo los permisos que le corresponde correctamente mediante una petición a Implantación.

- Las carpetas en las que el proceso batch (desde Oracle) tiene que escribir (por ejemplo /file /lis /log /tmp) no tienen los permisos adecuados de escritura, en este caso una descripción habitual del error es:

```
ERROR at line 1:
ORA-29283: invalid file operation
ORA-06512: at "SYS.UTL_FILE", line 475
ORA-29283: invalid file operation
ORA-06512: at "K432.UP_BED_SINFEC", line 36
ORA-06512: at line 1
```

- Al lanzar un ejecutable que está en /cade desde la O75 lo va a buscar al /exe en vez de al /cade y dice que no lo encuentra, dando el error **/aplic/aaa/exe/xxx.sh: not found**; este error es debido a que a la hora de dar de alta el proceso para que coja el ejecutable de /cade en vez del /exe en el campo "runtime" hay que poner "sh".

Incidencias relacionadas con la presencia en Internet (Lote 3)

Los errores relacionados con las aplicaciones que tienen presencia en Internet suelen estar relacionados con las grabaciones y catalogaciones de contenidos así como con el buscador y el no poder visualizar algunos elementos de la aplicación.

Estos errores habituales se detallan a continuación:

- Se produce un error al grabar un contenido, ya sea una petición de publicación, la catalogación o un dato básico:

Para este error existen varias causas posibles siendo las más habituales:

1- Existe una petición de publicación que no tiene OID en el XML de configuración de contenido en la Base de Datos.

Se debe localizar en la tabla R01D01T01 el contenido, localizar en la R01D12T01 sus documentos, y en la R01D17T01 las peticiones de publicación de dichos documentos. Se localizan las peticiones en el XML, y se comprueba cuál es la que no tiene el tag <publicationOid>. El método más rápido es eliminar dicha petición tanto del XML como de la tabla R01D17T01, y crear una nueva petición desde el propio Gestor.

2- Existe una publicación que no tiene OID en el XML de configuración de contenido en la Base de Datos.

Se debe localizar en la tabla R01D01T01 el contenido, localizar en la R01D12T01 sus documentos, y en la R01D19T01 las publicaciones de dichos documentos. Se localizan las publicaciones en el XML, y se comprueba cuál es la que no tiene el tag <publicationOid>. El método más rápido es eliminar dicha publicación tanto del XML como de la tabla R01D19T01, y publicar el documento de nuevo, desde el propio Gestor. Antes de publicar, es necesario pedir confirmación al WebMaster.

3- El contenido tiene vacío algún campo obligatorio: Fecha, Autor, Responsable.

Sólo ocurre con contenidos migrados. Normalmente el campo que falta es el de Responsable. Se rellena desde el propio Gestor, y se comunica al WebMaster para que actualice el campo con un valor adecuado.

4- No existe archivo de inicio.

Ocurre con contenidos migrados únicamente. Se detecta gracias a que todos los archivos de un documento aparecen en azul. Es necesario consultar con el WebMaster cuál es el archivo de inicio. Una vez identificado, se localiza el archivo tanto en la Base de Datos (tabla R01D20T01) y en el XML descriptor del contenido (tabla R01D01T01). En la tabla R01D20T01, se actualiza la columna INITIAL_PAGE_20 con el valor 1 (debería ser 0 si se ha dado esta situación). En el XML, se localiza la tag <initialPage>, y se le asigna el valor 1.

5- Error en la catalogación

Un error en la catalogación puede causar que no se pueda modificar un contenido. Para comprobarlo, lo idóneo es abrir la ventana de catalogación, y pulsar Aceptar, con lo que se provoca que se recatalogue el contenido, pero con las mismas etiquetas. Si se produce un error, la causa es una etiqueta o eje en mal estado. Es preciso borrar el tag <labels> del XML descriptor del contenido (tabla R01D01T01) y todos los registros de etiqueta asociados a dicho contenido en la tabla R01D09T01. Posteriormente, es necesario avisar al WebMaster para que proceda a la catalogación del contenido.

- Se produce algún error al catalogar un contenido desde el Gestor:

Para este error existen varias causas posibles siendo las más habituales:

1- Fallo al grabar la catalogación.

Se trataría de la incidencia descrita anteriormente

2- Al pulsar sobre el botón de Catalogar, se produce un error **Internal Server 500**

La causa de éste error es una etiqueta que, aunque está bien asignada al contenido, y pertenece al eje donde se ha catalogado; su rol (nivel dentro del eje) ha cambiado desde que se asignó al contenido. Este error es muy habitual cuando se cambia la catalogación por defecto de una WorkArea, o cuando hay algún cambio en la estructura de etiquetas a nivel global en el Gestor de Etiquetas. Aunque se puede localizar el rol corrupto, la solución más rápida es eliminar la catalogación, y pedir al WebMaster que la restaure posteriormente.

- Se produce algún error en el buscador:

Para este error existen varias causas posibles siendo las más habituales:

1- Un contenido sale duplicado en Euskadi.net

Si el contenido ha sido publicado, posteriormente despublicado y por último vuelto a publicar, es posible que salga dos veces. Para solucionar esto, la mejor solución es localizar la despublicación en la tabla R01D19T01, y en el descriptor XML del contenido, y borrarla. El efecto no es inmediato, hay que esperar a que la vista materializada R01K04V00 se refresque.

2- Un contenido todavía no publicado sale en Euskadi.net

Una publicación programada puede salir en Euskadi.net. Cuando esto ocurre lo mejor es borrar la publicación, y emplear la publicación inmediata.

3- Un contenido no aparece en las búsquedas

Cuando se crea y modifica un contenido, o cualquiera de sus componentes, se inserta en la Base de Datos del Buscador (esquema R01K). Si no aparece un contenido en Euskadi.net, lo más probable es que no se haya insertado correctamente en la R01K. Es necesario comprobar que el contenido, sus documentos y sus archivos estén en las tablas R01K02T00 y R01K03T00 (el contenido), R01K04T00 (los documentos) y en la R01K05T00 (los archivos). Si no están en éstas tablas, la solución más fácil es modificar la descripción del elemento que no aparece (contenido, documento o archivo) y se indexará de nuevo. En caso de emergencia, se pueden insertar estos registros a mano.

3- Un procedimiento administrativo no aparece en las búsquedas, o aparece Cerrado cuando debería estar Abierto.

Primero se debe seguir el comprobar el paso anterior. Si todo está bien, hay que revisar la tabla R01K19T00. Si no aparece, o aparece con campos en blanco, es necesario re-indexar, cambiando la descripción de alguno de los ficheros del contenido. Éste caso se puede identificar fácilmente si el contenido aparece en la búsqueda, pero al final de los resultados, en vez de al principio.

- No se genera HTML desde un DCR correctamente:

Para este error existen varias causas posibles siendo las más habituales:

1-No se genera HTML

Se identifica este error gracias a que cuando se pulsa a “Grabar”; desde la pantalla de captura de Interwoven, no aparece el mensaje de “Se ha generado el archivo”. Para resolver esto, es necesario localizar el archivo datacapture.cfg correspondiente a la plantilla que se está empleando, que se encuentra en la carpeta templatedata, dentro de cada workarea. En este archivo, debajo de la tag <ruleset>, se debe insertar ésta línea:

```
<script language="javascript" location="webserver" src="/contenidos/scripts/r01dOnSaveFile.js"/>
```

2- No se previsualiza bien un contenido, a pesar del mensaje de que se ha generado.

Cuando se de ésta situación, hay que comprobar si el código HTML se ha generado correctamente. Se puede descargar el archivo HTML de Interwoven y visualizarlo en un navegador. Normalmente estará bien generado, así que el error se produce al pulsar Preview. Este error es un bug de Interwoven. Cuando un archivo contiene en su nombre un “_” seguido de 8 o más caracteres, se produce este error. La solución habitual es cambiar el nombre y eliminar el “_”. Esto influye únicamente sobre la previsualización, si el archivo se publica, se visualizará correctamente desde Euskadi.net.

- En una aplicación integrada en portal no se visualizan algunos elementos en el navegador.

Este error es debido a que el elemento se intenta cargar con la cabecera y pie de página de portales por lo que debería cargarse sin ellas.

Para que una URL no se cargue como si fuese en portal se debe poner en la URL el parámetro:
R01HNoPortal=true