



DOP Port@firmas



CONSEJERÍA DE JUSTICIA Y
ADMINISTRACIÓN PÚBLICA

Documento elaborado por: Francisco José Cantero Villar

Revisado por: Miguel J. Vázquez Rebollo

Versión: 1.1.1

Lugar y fecha: Sevilla, 3 de Febrero de 2006

Contenido:

1	Requisitos previos de Instalación	4
2	Procedimiento de migración de v1.0.X a v1.1.0	5
2.1	Revisión del usuario de base de datos.....	5
2.2	Script's de creación de tablas y objetos	6
2.3	Carga de procedimientos	6
2.3.1	Especificaciones de paquete	6
2.3.2	Cuerpos de paquete.....	7
3	Procedimiento de primera instalación	9
3.1	Creación usuario de base de datos	9
3.2	Script's de creación de tablas y objetos	9
3.3	Carga de datos	10
3.3.1	Carga de constantes binarias mediante TOAD.....	11
3.3.2	Carga de constantes binarias mediante SQLPLUS	13
3.4	Carga de procedimientos	14
3.4.1	Clase Base64.....	14
3.4.2	Especificaciones de paquete	14
3.4.3	Cuerpos de paquete.....	15
3.5	Tareas programadas en la base de datos	16
3.6	Creación de Rol.....	17
4	Procedimientos de configuración y despliegue.	18
4.1	Alta/Modificación de la aplicación Port@firmas en @firma	18
4.2	Despliegue en el servidor de aplicaciones Tomcat	21
4.2.1	Detector.js	23
4.2.2	Almacén de certificados	24
4.2.3	Ampliación del parámetro de memoria de la máquina virtual de Java....	27
4.2.4	Código de barras en servidores Unix/Linux	29
4.3	Actualizando Servidores de firma en Port@firmas	30
4.4	Actualizando las constantes de la aplicación	30
4.4.1	Configuración de los servicios de aviso por correo.....	31
4.5	Permisos Oracle a nuevos usuarios de aplicación	32



5	Historia de versiones	33
---	-----------------------------	----

1 Requisitos previos de Instalación

Para poder instalar y hacer funcionar correctamente [Port@firmas](#) con el servidor @firma que tengamos instalados debemos cumplir los siguientes requisitos:

- Base de datos Oracle 8i o 9i, y en el caso de 8i con las extensiones Java instaladas en el servidor.
- Servidor con JSDK 1.4.2_0_7 o superior, (la versión 1.4.2_0_3 tiene un “bug” en el SSL que se corrige de la 4 en adelante). Por reducir posibles “bug’s” de Java se aconseja tener la última versión.
- Librerías JCE 1.4 de Sun instaladas en el JSDK para la encriptación/descriptación "fuerte" requerida en los WebServices de @firma v4.0.
- Servidor de aplicaciones Tomcat 4.1.x ó 5.0.x. Se recomienda no usar el Tomcat 5.5.x al hacer uso de las JSE 1.5 de Java y no haberse testado la aplicación bajo la misma.
- (Opcional) Cuenta de Correo para los avisos por correo o bien acceso a @visor.

En cuanto a los requisitos en clientes, hay que cumplir los definidos por la propia plataforma @firma en su versión v4.0. Para más información acceder a Plutón, Área Técnica de @firma, y el “[Manual de Usuario de @FIRMA](#)” detalla los mismos.

2 Procedimiento de migración de v1.0.X a v1.1.0

Desde este punto se va a describir el proceso de migración necesario para pasar [Port@firmas](#) de la versión v1.0.X a la versión v1.1.0. Si no es su caso, y va a instalar una versión desde cero debe continuar directamente en el punto “**3. Procedimiento de primera instalación**” omitiendo todo lo contenido en este punto.

Los cambios del modelo de datos son mínimos pero necesarios para poder abordar la firma en bloque, y los paquetes de base de datos a su vez han sido ampliados con las nuevas funcionalidades ya definidas en el documento de novedades de [Port@firmas](#) V1.1.0.

El proceso de migración se puede resumir en tres pasos:

- 1) Modificación del modelo de datos y carga de los nuevos paquetes, que se explica en este punto.
- 2) Modificación de la configuración de @firma, el valor del *Generador*, punto 4.1.
- 3) Repliegue de la aplicación y despliegue del nuevo WAR de Port@firmas , 4.2.

Es recomendable parar la aplicación mientras se realizan estos cambios, para evitar que haya usuarios conectados mientras se actualizan las informaciones.

2.1 Revisión del usuario de base de datos

Partimos desde una instalación de [Port@firmas](#) v1.0.X, y por tanto de un usuario Oracle propietario del esquema con todos los permisos adecuados, siempre y cuando se haya seguido los requisitos de la instalación desde cero.

Para este usuario se tuvo que definir al menos tres “tablespaces”, que para el caso más genérico son:

- TS_PFIRMA_INDICES
- TS_PFIRMA_DATOS
- TS_PFIRMA_BLOB

Debemos asegurarnos que los “tablespaces” de índices y datos son correctos, si no, deberemos modificar los script’s para que apunten a los “tablespaces” definidos para nuestro sistema por el administrador de base de datos.

2.2 Script's de creación de tablas y objetos

El siguiente paso, una vez comprobado el usuario y los tablespaces es probar dicho usuario, conectando mediante un SQLPLUS o un cliente Oracle tipo TOAD o TORA.

Si la conexión es correcta podemos pasar a crear las nuevas entidades necesarias para el [Port@firmas](#) v1.1.0, en este caso, tablas, índices, constraints, secuencias y tipos en este orden.

Para ello pensemos a partir de ahora que tenemos el CD del kit Port@firmas v1.1.0 en la unidad D, y por tanto desde un SQLPLUS debemos poner:

```
SQL> @"D:\KitPortafirma\Scripts\Migración 1.0.X a 1.1.0\PFIRMA.tab"
```

Y pulsamos "intro", si todo va bien ira apareciendo diálogos indicando "tabla creada".

Ejecutamos el resto de los scripts en el siguiente orden:

```
SQL> @"D:\KitPortafirma\Scripts\Migración 1.0.X a 1.1.0\PFIRMA.ind"
```

```
SQL> @"D:\KitPortafirma\Scripts\Migración 1.0.X a 1.1.0\PFIRMA.con"
```

```
SQL> @"D:\KitPortafirma\Scripts\Migración 1.0.X a 1.1.0\PFIRMA.sqs"
```

```
SQL> @"D:\KitPortafirma\Scripts\Migración 1.0.X a 1.1.0\PFIRMA.typ"
```

Una vez terminado de ejecutar los scripts el modelo de datos ya está modificado en el usuario propietario. Estos mismos scripts se pueden lanzar desde un cliente TOAD o TORA sin problemas, teniendo solo cuidado en ejecutarlo en el mismo orden.

Se recomienda lanzar uno a uno los script's para poder detectar más rápidamente algún problema.

2.3 Carga de procedimientos

2.3.1 Especificaciones de paquete

Vamos primero a cargar las especificaciones de los paquetes, para ello en la ventana del SQLPLUS ejecutamos las siguientes instrucciones:

```
SQL>set scan off;
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_TYPE.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_API.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_TOOL.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_ACCION.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_ADMIN.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_BASE64.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_CORREO.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_DAEMON.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_EVENTO.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_FIRMA.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_LOOKUP.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_NOTIFICA.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_PETICION.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_REDIREC.PKS
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_WEB.PKS
```

Si todo va bien irán saliendo diálogos:

Paquete creado.

2.3.2 Cuerpos de paquete

Al igual que antes, ejecutamos las siguientes líneas en el SQLPLUS.

```
SQL>set scan off;
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_TYPE.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_API.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_TOOL.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_ACCION.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_ADMIN.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_BASE64.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_CORREO.PKB
```

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_DAEMON.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_EVENTO.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_FIRMA.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_LOOKUP.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_NOTIFICA.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_PETICION.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_REDIREC.PKB

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_WEB.PKB

Si al ver los diálogos de confirmación nos salta al llegar al paquete PF_PQ_NOTIFICA el mensaje:

Advertencia: Cuerpo del paquete creado con errores de compilación.

Esto quiere indicar que no tenemos acceso al avisador. Si no tenemos o no queremos usar el avisador en [Port@firmas](#) ejecutamos la siguiente línea.

SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_NOTIFICA_SIN_AVISADOR.PKB

Y con esto hemos terminado de cargar todos los objetos necesarios en la base de datos. Al igual que en los casos anteriores estos script's pueden ser lanzados con el TOAD teniendo cuidado en respetar el orden de ejecución indicado.

Ahora pasaremos a replegar la aplicación en el Tomcat para desplegar el nuevo WAR y revisar los parámetros de conexión a base de datos. Para ello debemos saltar al punto **“4.1 Alta/Modificación de la aplicación Port@firmas en @firma”**.

3 Procedimiento de primera instalación

Desde este punto se va a describir la instalación del sistema partiendo desde cero.

3.1 Creación usuario de base de datos

El primer requisito que debe cumplir es la versión de la base de datos. [Port@firmas](#) es en actualidad compatible con base de datos Oracle 8i y 9i en los que se hayan habilitado clases Java en el servidor (extensión Java en la base de datos).

En la instancia de base de datos donde queramos instalar el sistema, tenemos que dar de alta tres “tablespaces”:

- TS_PFIRMA_INDICES
- TS_PFIRMA_DATOS
- TS_PFIRMA_BLOB

Los tamaños de los mismos quedan a elección del administrador, siendo recomendables de partida 16, 64 y 128 respectivamente, con posibilidad de crecimiento automático. Por temas de rendimiento se recomienda además que el fichero de “tablespace” de índices esté en una unidad de disco física distinta de la de datos normales o columnas BLOB’s (documentos binarios).

Dar de alta un usuario Oracle, recomendamos “PFIRMAMG” pues será el que empleemos como referencia en el resto del documento, con los roles “CONNECT”, “RESOURCE” y de creación de roles. El “tablespace” por defecto del usuario debe ser “TS_PFIRMA_DATOS”, y el temporal el “TEMP”, o el definido en el sistema como temporal. Una vez creado el usuario, con el usuario “SYSTEM”, hay que concederle el permiso adicional, con la instrucción “GRANT CREATE TYPE TO PFIRMAMG”.

3.2 Script’s de creación de tablas y objetos

El siguiente paso, una vez creado el usuario y los tablespaces es probar dicho usuario, conectando mediante un SQLPLUS o un cliente oracle tipo TOAD o TORA.

Si la conexión es correcta podemos pasar a crear todas las entidades necesarias para el Port@firmas, en este caso, tablas, índices, constraints, secuencias y tipos en este orden.

Para ello pensemos a partir de ahora que tenemos el CD del kit Port@firmas en la unidad D, y por tanto desde un SQLPLUS debemos poner:

SQL>@D:\ KitPortafirma\Scripts\tablas\pfirmaGeneral.tab

Y pulsamos “intro”, si todo va bien ira apareciendo diálogos indicando “tabla creada”. Puede que al llegar a la tabla PF_POOL_PETICIONES, en ocasiones, de un mensaje de error, pero es simplemente que al tratar de crear la tabla temporal parece que tras crearla lo intente de nuevo. Puede ser debido a un error del SQLPLUS.

El administrador de base de datos puede revisar la creación de tablas para ajustar y afinar los parámetros de almacenamiento de columnas BLOB a las características propias de la base de datos sobre la que se vayan a instalar.

Ejecutamos el resto de los scripts en el siguiente orden:

SQL>@D:\ KitPortafirma\Scripts\tablas\pfirmaGeneral.ind

SQL>@D:\ KitPortafirma\Scripts\tablas\pfirmaGeneral.con

SQL>@D:\ KitPortafirma\Scripts\tablas\pfirmaGeneral.sqs

SQL>@D:\ KitPortafirma\Scripts\tablas\pfirmaGeneral.typ

Una vez terminado de ejecutar los scripts el modelo de datos ya está generado en el usuario propietario. Estos mismos scripts se pueden lanzar desde un cliente TOAD o TORA sin problemas, teniendo solo cuidado en ejecutarlo en el mismo orden.

3.3 Carga de datos

El siguiente paso es cargar los datos básicos necesarios en las tablas de codificación, el proceso es similar al anterior. Seguimos con el ejemplo en el caso de SQLPLUS. El orden en este caso es el siguiente:

SQL> @D:\KitPortafirma\Scripts\datos\PF_SERVIDORES.dat

Pulsamos intro y obtendremos el siguiente mensaje:

1 fila creada.

Validación terminada.

Procedemos con el resto de los scripts:

SQL> @D:\KitPortafirma\Scripts\datos\PF_APLICACIONES.DAT

SQL> @D:\KitPortafirma\Scripts\datos\PF_CONSTANTES.DAT

SQL> @D:\KitPortafirma\Scripts\datos\PF_ESTADOS.DAT

SQL> @D:\KitPortafirma\Scripts\datos\PF_EVENTOS.DAT

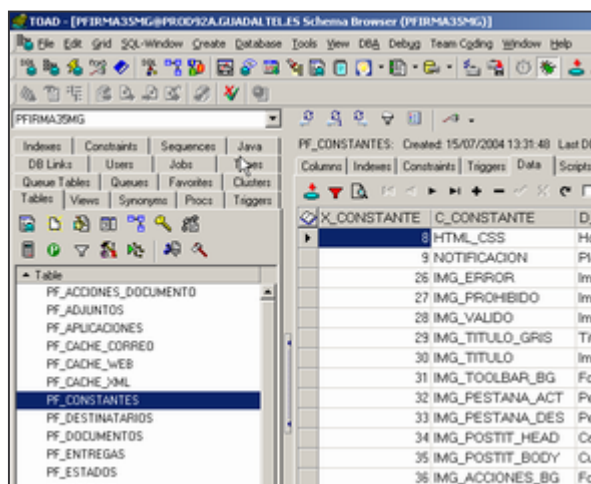
SQL> @D:\KitPortafirma\Scripts\datos\PF_TIPOS_DOCUMENTO.dat

Al igual que el caso anterior estos scripts se pueden lanzar como tales en el TOAD o similar.

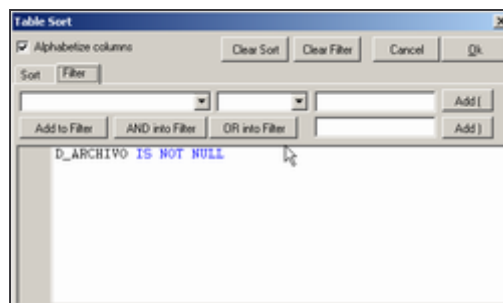
3.3.1 Carga de constantes binarias mediante TOAD

Hay una serie de constantes en el Port@firmas que tienen adjunto algo tipo de documento binario, según el caso, imágenes o documentos html como plantillas. Vamos a ver como se cargarían estos documentos mediante un cliente Oracle, en este caso el mas común, el TOAD.

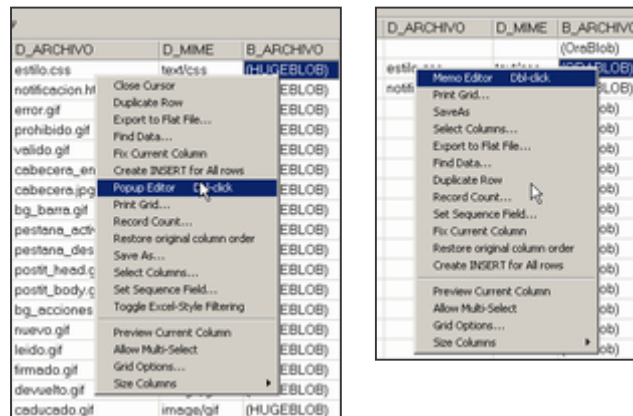
- Abrimos un navegador de objetos de TOAD y nos vamos a la pestaña de tablas, y seleccionamos la tabla constantes.



Pulsamos el siguiente botón  y introducimos el siguiente criterio de filtrado.



Pulsamos "OK" y solo nos aparecerán los registros en los que debemos incorporar los archivos. Nos situamos sobre la columna B_ARCHIVO y pulsamos el botón derecho del ratón, y seleccionaremos la opción








(Ojo, según la versión del TOAD puede que la entrada del menú varíe de nombre)

- Nos aparece la siguiente ventana

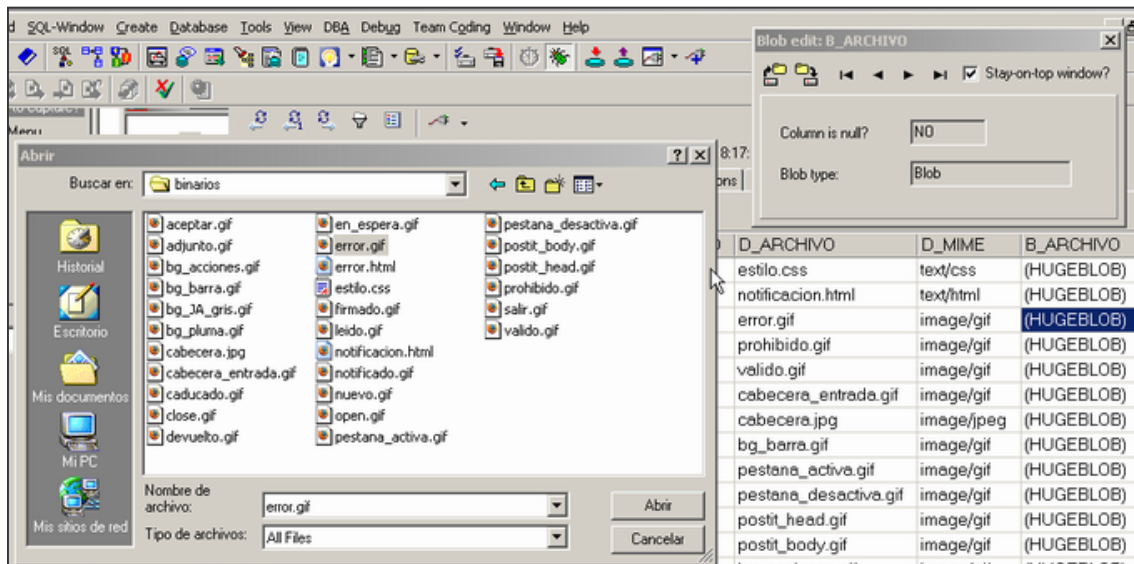


En este menú emergente que nos aparece usaremos los siguientes botones:

-  Para seleccionar un archivo y subirlo a la base de datos.
-  Navegar al primer registro.
-  Navegar al registro anterior.
-  Navegar al registro siguiente.
-  Navegar al último registro.

- Ahora iremos cargando uno a uno las imágenes y documentos existentes en la carpeta:

D:\KitPortafirma\Scripts\datos\binarios



3.3.2 Carga de constantes binarias mediante SQLPLUS

El proceso en este caso es más complejo por eso lo indicamos en el último caso dado que hay que subir a un directorio del servidor de base de datos el contenido de la carpeta **D:\KitPortafirma\Scripts\datos\binarios**, supongamos **/tmp**, y seguir los siguientes pasos.

- Conceder de forma temporal el permiso de creación de directorio al usuario PFIRMAMG con el usuario SYSTEM.

```
SQL> grant create any directory to pfirmamg;
```

- Tras esto nos conectaremos con el usuario PFIRMAMG, pondremos la siguiente instrucción:

```
SQL> create directory READDIR as '/tmp';
```

- Crear el siguiente procedimiento

```
CREATE OR REPLACE PROCEDURE CARGA_IMAGENES AS
CURSOR IMAGENES IS
SELECT X_CONSTANTE,D_ARCHIVO
FROM PF_CONSTANTES WHERE D_ARCHIVO IS NOT NULL;
V_IMAGENES IMAGENES%ROWTYPE;
AUX_B_ARCHIVO PF_CONSTANTES.B_ARCHIVO%TYPE;
FILS BFILE;
AMT INTEGER;
BEGIN
UPDATE PF_CONSTANTES SET B_ARCHIVO=EMPTY_BLOB() WHERE D_ARCHIVO IS NOT NULL;
COMMIT;
FOR V_IMAGENES IN IMAGENES LOOP
    FILS:= BFILENAME('READDIR',V_IMAGENES.D_ARCHIVO);
    DBMS_LOB.FILEOPEN(FILS, DBMS_LOB.FILE_READONLY);
    AMT:=DBMS_LOB.GETLENGTH(FILS);
    IF AMT>0 THEN
        SELECT B_ARCHIVO INTO AUX_B_ARCHIVO FROM PF_CONSTANTES WHERE
X_CONSTANTE=V_IMAGENES.X_CONSTANTE FOR UPDATE;
        DBMS_LOB.LOADFROMFILE(AUX_B_ARCHIVO, FILS, AMT);
        COMMIT;
    END IF;
    DBMS_LOB.FILECLOSE(FILS);
END LOOP;
END;
```

- Ejecutar el procedimiento anterior:

```
SQL> exec CARGA_IMAGENES;
```

- Eliminar el procedimiento y quitarle los permisos de creación de directorios al usuario.

3.4 Carga de procedimientos

Una vez que hemos cargado todas las tablas de codificación, vamos a proceder a cargar el resto de scripts, en este caso, las clases java y paquetes necesarios.

3.4.1 Clase Base64

Esta clase es necesaria para realizar operaciones de codificación y decodificación de BASE64 a binario y viceversa. Hacemos uso de ella debido a que en Oracle 8i no existen clases y paquetes a tal efecto.

Como en los casos anteriores, desde la ventana del SQLPLUS ponemos las siguientes instrucciones:

```
SQL> set scan off;
```

```
SQL> @D:\KitPortafirma\Scripts\base64\Base64plsqlijava.sql
```

3.4.2 Especificaciones de paquete

Vamos primero a cargar las especificaciones de los paquetes, para ello en la ventana

del SQLPLUS ejecutamos las siguientes instrucciones:

```
SQL>set scan off;
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_TYPE.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_API.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_TOOL.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_ACCION.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_ADMIN.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_BASE64.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_CORREO.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_DAEMON.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_EVENTO.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_FIRMA.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_LOOKUP.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_NOTIFICA.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_PETICION.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_REDIREC.PKS
```

```
SQL> @D:\KitPortafirma\Scripts\paquetes\PF_PQ_WEB.PKS
```

Si todo va bien irán saliendo diálogos:

Paquete creado.

3.4.3 Cuerpos de paquete

Al igual que antes, ejecutamos las siguientes líneas en el SQLPLUS.

```
SQL>set scan off;
```

```
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_TYPE.PKB
```

```
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_API.PKB
```

```
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_TOOL.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_ACCION.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_ADMIN.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_BASE64.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_CORREO.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_DAEMON.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_EVENTO.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_FIRMA.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_LOOKUP.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_NOTIFICA.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_PETICION.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_REDIREC.PKB
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_WEB.PKB
```

Si al ver los diálogos de confirmación nos salta al llegar al paquete PF_PQ_NOTIFICA el mensaje:

Advertencia: Cuerpo del paquete creado con errores de compilación.

Esto quiere indicar que no tenemos acceso al avisador. Si no tenemos o no queremos usar el avisador en [Port@firmas](#) ejecutamos la siguiente línea.

```
SQL>@D:\KitPortafirma\Scripts\paquetes\PF_PQ_NOTIFICA_SIN_AVISADOR.PKB
```

Y con esto hemos terminado de cargar todos los objetos necesarios en la base de datos. Al igual que en los casos anteriores estos script's pueden ser lanzados con el TOAD teniendo cuidado en respetar el orden de ejecución indicado.

Ahora pasaremos a dar de alta la aplicación en @firma y desplegar las aplicaciones en el Tomcat y terminar de configurar las mismas.

3.5 Tareas programadas en la base de datos

Hay un par de Job's definidos en el sistema. Uno se encarga de vaciar las tablas temporales periódicas, el otro, solo en el caso de tener activo y configurado el envío de

avisos por o bien a través de la plataforma @visor, del envío de avisos.

Para activar la tarea de borrado de tablas temporales ejecutar desde el SQLPLUS:

```
SQL> @D:\KitPortafirma\Scripts\jobs\SPOOL.sql
```

Si todo va bien debe salir la siguiente línea

Procedimiento PL/SQL terminado correctamente.:

Para activar la tarea de envío de avisos ejecutar:

```
SQL> @D:\KitPortafirma\Scripts\jobs\NOTIFICACION.sql
```

Si no se tiene configurado y activo el servicio de envío de correos esta tarea así como el paquete PF_PQ_NOTIFICACION puede aparecer como inválido según en cada caso. Por tanto si no se va a usar este servicio es mejor desactivar esta tarea de base de datos o incluso no cargar esta tarea.

Para asegurarnos que se guarden los cambios, hacemos un "commit":

```
SQL> commit;
```

Validación terminada.

3.6 Creación de Rol

Para controlar y limitar el acceso que puedan hacer terceras aplicaciones sobre las API's se ha definido un rol para adjudicar a los usuarios de base de datos y así tengan acceso únicamente a las API's que les interese.

El script de creación del rol debemos lanzarlo desde el SQLPLUS con la siguiente instrucción:

```
SQL> @"D:\KitPortafirma\Scripts\roles y permisos\CREACION ROL PF_R_PETICIONES.sql"
```

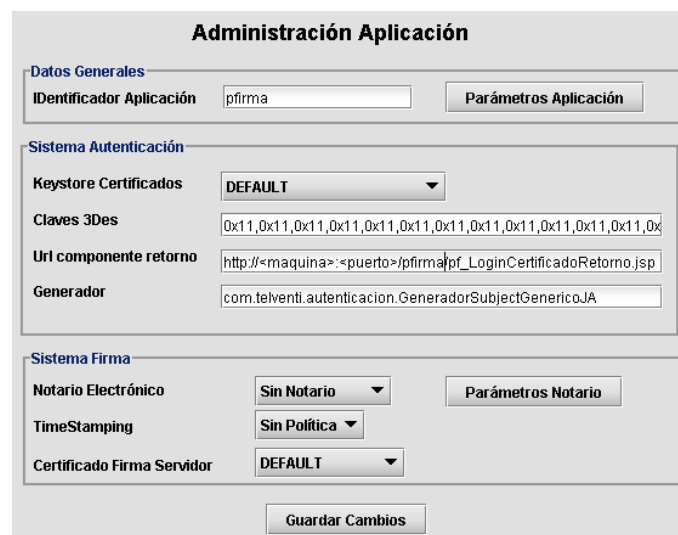
Existen en esta carpeta dos scripts más para ejecutar cuando queramos conceder este rol y el resto de permisos a un usuario de base de datos, así como la generación de sinónimos. En el punto 3.5 del presente documento se detalla más este apartado.

4 Procedimientos de configuración y despliegue.

4.1 Alta/Modificación de la aplicación Port@firmas en @firma

Vamos a ver el caso de que nosotros mismos demos de alta la aplicación en @firma por que fuésemos administradores. En el caso de usar un @firma donde no tengamos acceso directamente deberemos seguir el procedimiento de alta que nos indiquen sus administradores proporcionandoles los datos que vienen a continuación:

- 1) Abrir la herramienta de administración del servidor de firma:
- 2) Dar de alta una nueva aplicación “pfirma” si no existe, o bien modificar la existente.
 - Identificador de aplicación: `pfirma`
 - Clave 3des: proporcionada por el administrador o generada por la aplicación.
 - Componente de retorno: http://<maquina>:<puerto>/pfirma/pf_LoginCertificadoRetorno.jsp.
 - Generador: `com.telventi.autenticacion.GeneradorSubjectGenericoJA`¹
 - Personalizar los detalles del servicio de firma que se precise, en el ejemplo que viene a continuación se da el caso más sencillo y típico:



¹ **IMPORTANTE** Este valor ha cambiado de la v1.0.X a la v1.1.X, modificar en caso de migración.

3) Dar de alta una página firmable “pfirma” si no existe, o bien modificar la existente.

- Identificador de página: `pfirma`
- Keystore Certificado: `DEFAULT`
- CodeBase: <https://servidorFachada:444/firmadigital>
- Página firmable: <http://<maquina>:<puerto>/pfirma/descarga/>
- Página Error Certificado: (en blanco)
- Nombre Formulario: `form_firma`
- Botón Submit: `envia_firma`
- Aplicación: `pfirma`
- Página destino: `destino_pfirma.jsp`²
- Validación de página: `SÍ`
- Ceremonia de Firmado: (el valor por defecto)
- Página Destino Multifirma: http://<maquina>:<puerto>/pfirma/pf_confirmacion_multifirma.jsp
- Clave 3Des Multifirma: proporcionada por el administrador o generada por la aplicación.

² Hay que situar este fichero que se encuentra dentro del kit en `D:\KitPortafirma\Recibo` en el directorio `FormBackEnds` del servidor fachada de `@firma` y revisar que la URL de retorno apunte al servidor donde tenemos desplegada la aplicación.

4.2 Despliegue en el servidor de aplicaciones Tomcat

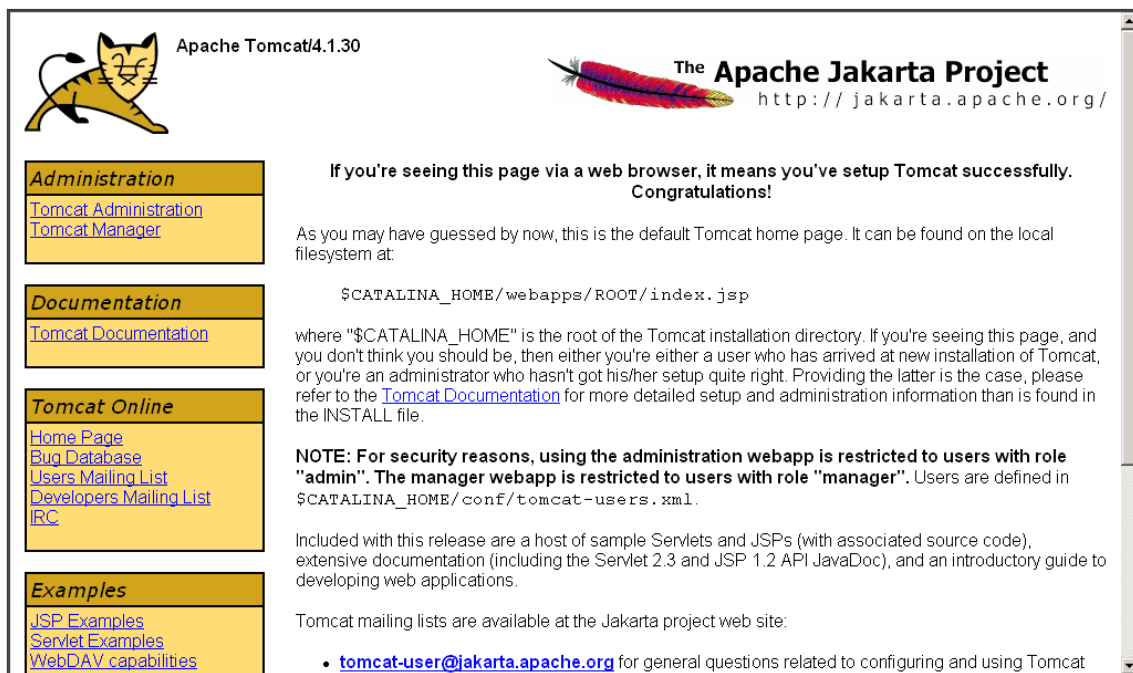
Seguidamente vamos a explicar como se despliega la aplicación de [Port@firmas](#) en el Tomcat, así como la herramientas de administración de la misma.

Partimos de la base de que disponemos de un servidor de aplicaciones Tomcat 4.1 o 5.0 y JDK1.4.2.0_7 o superior en nuestro servidor de aplicaciones (Tomcat 5.5 no es válido al necesitar la JSDK 1.5.0).

Procedemos a desplegar los wars, para ellos desde una ventana del navegador escribimos la dirección:

<http://servidorAplicaciones:8080/>

Con lo que nos saldrá la siguiente ventana:



Entramos en “Tomcat Manager” y tras indicar el usuario y clave administrador del Tomcat con el que dimos de alta el servidor, accederemos al módulo de configuración de aplicaciones. Nos vamos a la parte de Instalación (Install) y pulsamos el botón examinar, escogemos el fichero WAR del Port@firmas³ y pulsamos “instalar” (install)

³ En el caso de venir de una migración hacer previamente un “undeploy” o “remove” de la aplicación pñirma desde la herramienta Tomcat Manager para borrar la versión anterior. No interesa conservar el archivo de propiedades dado que hay cambios respecto a la versión anterior.

Upload a WAR file to install

Select WAR file to upload

Hacemos igual con el WAR de la Herramienta de Administración⁴.

Upload a WAR file to install

Select WAR file to upload

Si todo ha ido bien nos saldrá las dos nuevas aplicaciones en la lista de aplicaciones de la parte superior de la pantalla.

Applications				
Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	1	Start Stop Reload Remove
/TrewaFidea	Escritorio Trewa (Versión para Fidea)	true	0	Start Stop Reload Remove
/acervobot		true	0	Start Stop Reload Remove
/acervoweb		true	0	Start Stop Reload Remove
/admin	Tomcat Administration Application	true	0	Start Stop Reload Remove
/adminpfirma		true	0	Start Stop Reload Remove
/avisador		true	0	Start Stop Reload Remove
/barbecue		true	0	Start Stop Reload Remove
/demoautenticacion		true	0	Start Stop Reload Remove
/emuladorAries		true	0	Start Stop Reload Remove
/eperFirmaPrueba		true	0	Start Stop Reload Remove
/examples	Tomcat Examples	true	0	Start Stop Reload Remove
/manager	Tomcat Manager Application	true	0	Start Stop Reload Remove
/notificador		true	0	Start Stop Reload Remove
/ord_uni_sub		true	0	Start Stop Reload Remove
/pdftoimage		true	0	Start Stop Reload Remove
/pfirma		true	2	Start Stop Reload Remove

Una vez desplegadas las dos aplicaciones, las paramos y accedemos al directorio

\$ruta_tomcat\webapps\pfirma\WEB-INF\classes

Y modificamos el archivo de propiedades “conn_pfirma.properties” con los datos de la conexión:

```

nombreMaquina=maquinaBD
puerto=PuertoBD (1521 por lo general)
nombreBD=IdentificadorDeBaseDeDatos (SID)
nombreUsuario=usuarioBD
clave=claveUsuarioBD
avisador=AVISAMG. (Usuario de BD de @visor en el caso de tenerlo instalado)
tituloHTML=Port@firmas v1.1.0 - Consejería de XXX - Junta de Andalucía
conexionesMaximas = 50
# El siguiente parametro determina cuantos seg. despues de hacer un close()
# se cierra la conexion fisica
# -1 == infinito

```

⁴ Si ya tenemos la herramienta de administración desplegada no es necesario replegarla y desplegarla de nuevo.

```
inactivityTimeout = 15  
# El siguiente parametro determina cuantos seg. tarda una conexion que no se usa  
# en hacer un auto close(), con lo cual no es necesario hacer close() en el codigo  
# -1 == infinito (recomendado)  
timeToLiveTimeout = 3000
```

Como se puede apreciar, en el archivo de propiedades existe información para configurar el pool de conexiones habilitado para las descarga de los documentos, los servicios web's o el informe de firmas. Según sea necesario y tengamos recursos disponibles se pueden modificar el número máximo de conexiones, el tiempo máximo de inactividad o el tiempo de cierre físico de conexiones.

Por último para la herramienta de Administración accederemos también al directorio

```
$ruta_tomcat$webapps\adminpfirma\WEB-INF\classes
```

Y comprobaremos que los datos de conexión a la base de datos sean correctos. Los datos más importantes son la máquina y el puerto de la base de datos, dado que el usuario y clave se le solicita al usuario al entrar.

Una vez hecho estos cambios, arrancamos de nuevo las aplicaciones y listo.

4.2.1 Detector.js

Dentro del directorio "servicio" de la carpeta creada al desplegar la aplicación en el Tomcat, **\$ruta_tomcat\$webapps\pfirma\servicio**, se encuentra el fichero "Detector.js" que hay que revisar la variable de retorno en caso de que falten componentes para que apunte al servidor de fachada.

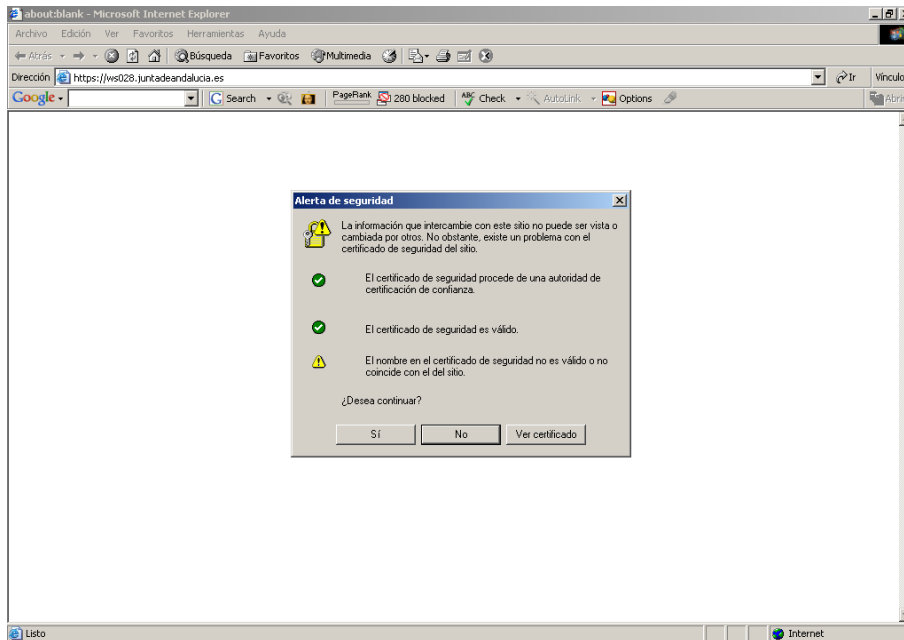
```
// ----- Comienzo Variables a modificar -----  
  
// Url for the next page when all the necessary components are detected  
var nextUrl = "../pf_peticion.jsp";  
  
// Url base of digital sign se  
var url_base = "https://<serviforFachada>:<puerto>/firmadigital";  
  
// ----- Fin variables a modificar -----
```

Deberemos cambiar el valor marcado en rojo para que apunte al servidor de fachada, por ejemplo:

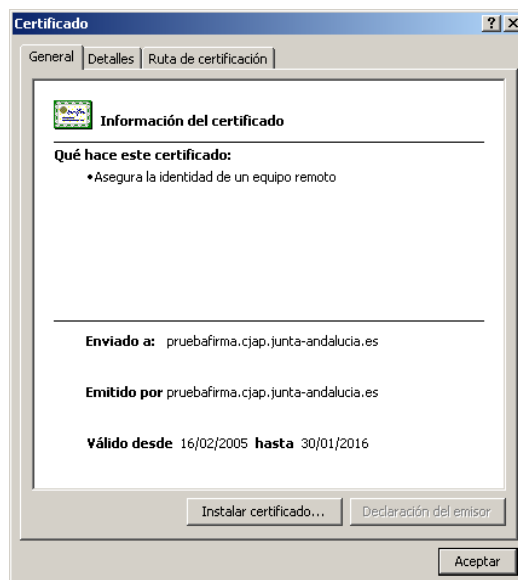
```
var url_base = "https://ws017.juntadeandalucia.es:444/firmadigital";
```

4.2.2 Almacén de certificados

Para poder establecer la comunicación SSL entre nuestro servidor de aplicaciones y el servidor de custodia de @firma debemos cargar la clave pública del mismo dentro del almacén de certificados de nuestro servidor de aplicaciones. Para ello, la forma más sencilla es conectarnos con un navegador al servidor de custodia, ejemplo:



Pulsamos “Ver certificado”.

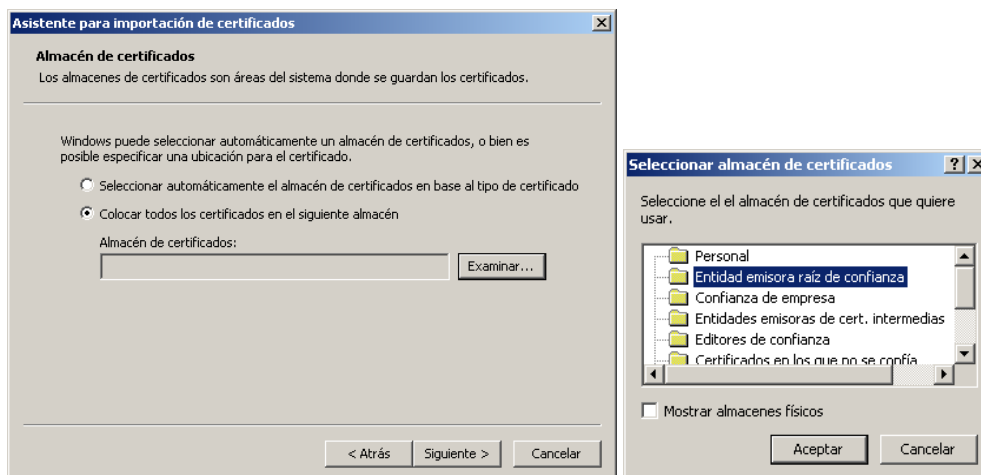


Pulsamos “Instalar certificado”.



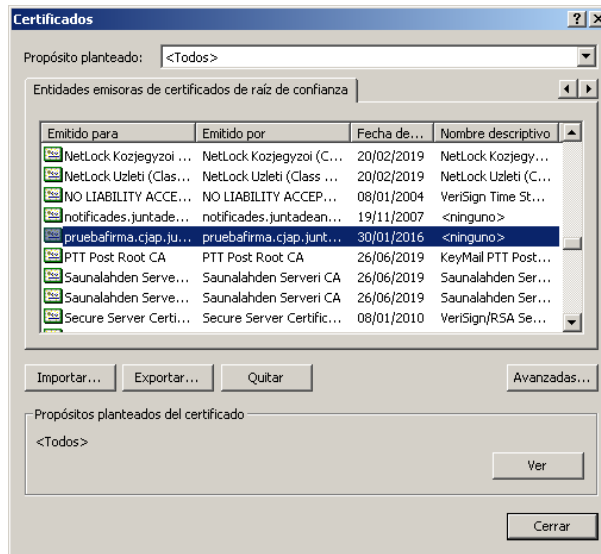
Nos sale un asistente mediante el cual podemos incorporar el certificado del servidor, pulsamos siguiente para continuar, nos preguntará si queremos incorporar a un almacén automáticamente o bien si queremos escogerlo nosotros directamente.

Escogemos esta última tal y como se ve en las siguientes imágenes.

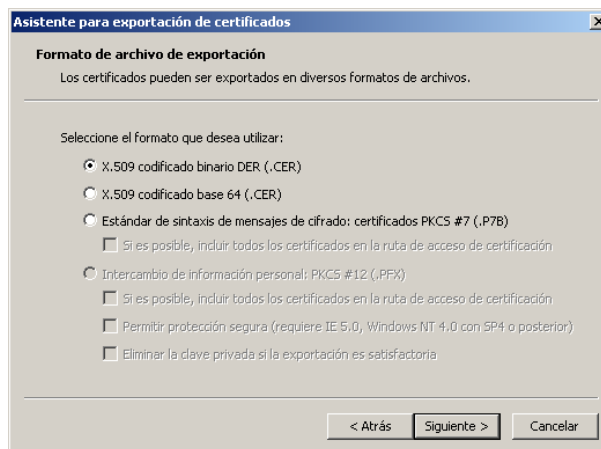


Seguimos hacia delante en el asistente y si todo va bien nos mostrará un diálogo de confirmación de operación realizada con éxito. Tras esto accederemos al menú “Herramientas”, “Opciones de Internet”, “Contenido” pulsamos el botón “Certificados...” de forma que nos saldrá un nuevo asistente.

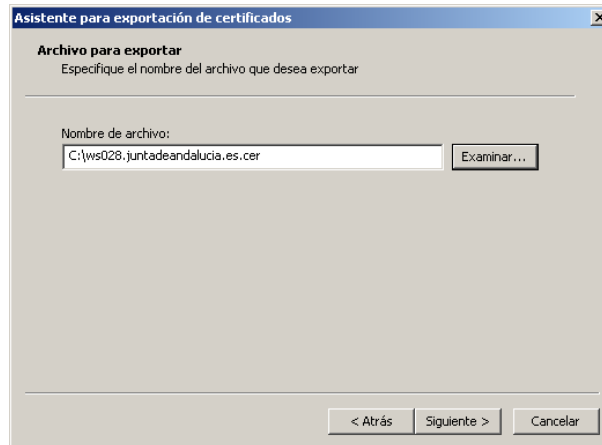
Nos vamos a la última pestaña y buscamos el certificado recién importado.



Seleccionamos el certificado y pulsamos “Exportar...”, nos mostrará un nuevo asistente en donde después de una pantalla de bienvenida, nos preguntará por el formato de exportación.



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



Tras esto ya tenemos el certificado del servidor, que podremos importar mediante la herramienta “keytool” de java sobre el almacén por defecto, “cacerts” (la clave de este almacén es “changeit”) del runtime de java, que por lo general se encuentra en el directorio:

```
%JAVA_HOME%/jre/lib/security
```

Para incluirlo deberemos importarlo mediante el siguiente comando, que siguiendo con el ejemplo sería:

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

Tras esto solo debemos recordar la ubicación del almacén de certificados para un paso más adelante.

4.2.3 Ampliación del parámetro de memoria de la máquina virtual de Java

Si tenemos una instalación típica de Tomcat, este se estará ejecutando con un límite de memoria RAM de 64 megas, las cuales se quedan cortas para poder funcionar correctamente con [Port@firmas](#) y resto de aplicaciones web desplegadas en el mismo produciendo pasado un tiempo que se den errores “java.lang.OutOfMemoryError”.

Recomendamos añadir por tanto a la variable JAVA_OPTS el valor máximo de memoria a 256 Megas.

A continuación se explica como modificar este parámetro con lo que conseguimos pasar de 64 megas a un máximo de 256. Si se siguen apreciando caídas por que existan gran número de aplicaciones desplegadas en el mismo servidor de

aplicaciones, seguir ampliando dicho parámetro en función a las características del servidor y recursos disponibles.

Para más información es mejor acceder a la FAQ de Apache/Tomcat y buscar en relación a este tema.

4.2.3.1 Servidor de Aplicación Windows

En este caso debemos editar el archivo “catalina.bat” que se encuentra en el directorio “\TomcatX\bin”. Buscamos el siguiente trozo de código e incluimos las partes marcadas en rojo:

```
...
# JAVA_HOME    Must point at your Java Development Kit installation.
#
# JAVA_OPTS    (Optional) Java runtime options used when the "start",
#              "stop", or "run" command is executed.
#
# JPDA_TRANSPORT (Optional) JPDA transport used when the "jpda start"
#              command is executed. The default is "dt_socket".
#
# JPDA_ADDRESS  (Optional) Java runtime options used when the "jpda start"
#              command is executed. The default is 8000.
#
# JSSE_HOME    (Optional) May point at your Java Secure Sockets Extension
#              (JSSE) installation, whose JAR files will be added to the
#              system class path used to start Tomcat.
#
# CATALINA_PID (Optional) Path of the file which should contains the pid
#              of catalina startup java process, when start (fork) is used
#
# $Id: catalina.sh,v 1.13.2.1 2004/08/21 15:49:49 yoavs Exp $
# -----
set JAVA_OPTS="-server -Xmx256m"

# OS specific support. $var _must_ be set to either true or false.
cygwin=false
os400=false
case "`uname`" in
CYGWIN*) cygwin=true;;
OS400*) os400=true;;
...

```

Esto sería para el caso de arrancar el Tomcat con el “startup.bat”.

Para que los cambios se lleven a efecto deberemos reiniciar el Tomcat.

4.2.3.2 Servidor de Aplicaciones Linux/Unix

Para el caso de estos servidores hay que editar el archivo “catalina.sh” que se encuentra en el directorio “\TomcatX\bin”. Buscamos el siguiente trozo de código e incluimos las partes marcadas en rojo:

```
...
# JAVA_HOME    Must point at your Java Development Kit installation.
#
# JAVA_OPTS    (Optional) Java runtime options used when the "start",
#              "stop", or "run" command is executed.
#
# JPDA_TRANSPORT (Optional) JPDA transport used when the "jpda start"

```

```
#          command is executed. The default is "dt_socket".
#
# JPDA_ADDRESS  (Optional) Java runtime options used when the "jpd start"
#                command is executed. The default is 8000.
#
# JSSE_HOME    (Optional) May point at your Java Secure Sockets Extension
#                (JSSE) installation, whose JAR files will be added to the
#                system class path used to start Tomcat.
#
# CATALINA_PID (Optional) Path of the file which should contains the pid
#                of catalina startup java process, when start (fork) is used
#
# $Id: catalina.sh,v 1.13.2.1 2004/08/21 15:49:49 yoavs Exp $
# -----
JAVA_OPTS="-server -Xmx256m"

# OS specific support. $var _must_ be set to either true or false.
cygwin=false
os400=false
case "`uname`" in
CYGWIN*) cygwin=true;;
OS400*) os400=true;;
...

```

Hay que tener cuidado que la forma de definir las variables varía de Windows a este sistema.

Para que los cambios se lleven a efecto deberemos reiniciar el Tomcat.

4.2.4 Código de barras en servidores Unix/Linux

La librería "Barbecue" de java para la generación de imágenes de códigos de barra necesita que, o bien se tenga configurada un entorno XWindows en dicha máquina, o bien hay que decirle a la JVM de forma explícita que las haga de forma interna.

Para este último caso, al igual que antes, tendremos que añadir un parámetro en el JAVA_OPTS del "catalina.sh" del Tomcat.

```
# CATALINA_PID (Optional) Path of the file which should contains the pid
#                of catalina startup java process, when start (fork) is used
#
# $Id: catalina.sh,v 1.13.2.1 2004/08/21 15:49:49 yoavs Exp $
# -----
JAVA_OPTS="-server -Xmx256m -Djava.awt.headless=true"

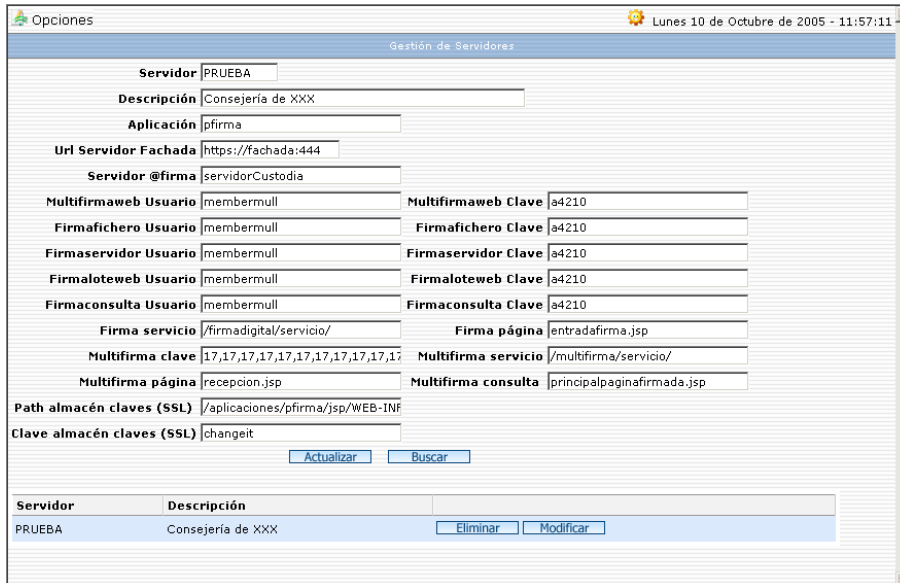
# OS specific support. $var _must_ be set to either true or false.
cygwin=false
os400=false
case "`uname`" in
CYGWIN*) cygwin=true;;

OS400*) os400=true;;

```

4.3 Actualizando Servidores de firma en [Port@firmas](#)

Una vez desplegadas las aplicaciones, accederemos a la herramienta de administración para volcar la información que habíamos indicado en el servidor de @firma en la configuración del servidor de firma de [Port@firmas](#) así como la ruta al almacén de certificados.



Servidor	Descripción	Eliminar	Modificar
PRUEBA	Consejería de XXX		

4.4 Actualizando las constantes de la aplicación

Debido a que las constantes del sistema son muy “sensibles” ante cambios inadecuados, no se ha dejado por ahora que estas se puedan modificar desde la herramienta de administración. Por tanto tendremos que modificarla a mano:

- FIRMA_AUTEN: Dirección URL de autenticación

<https://servidorFachada/cjap/servlet/com.telventi.autenticacion.ServletAutenticacion>

- FIRMA_APLIC: Nombre de la aplicación de autenticación `pfirma`
- FIRMA_AUTEN_PASS: Clave de autenticación en decimal.
- SERVIDOR_PRIMARIO: Servidor primario por defecto.
- SERVER_WEB: con el servidor de aplicaciones: <http://<maquina>:<puerto>/pfirma/>

4.4.1 Configuración de los servicios de aviso por correo

Si queremos habilitar los mecanismo de aviso por correo electrónico a los destinatarios de las petición o bien a los emisores de la misma deberemos disponer de una cuenta de correo y la información de acceso a los servidores POP3 y SMTP de correo. Este servicio está deshabilitado por defecto, para cambiarlo hay que editar la siguiente variable:

- NOTIF_EMAIL: Valor S o N para habilitarlo o no.

En caso de activarlo tenemos que revisar las siguientes variables:

- EMAIL: Dirección de correo de Port@firmas.
- EMAIL_HELO: Servidor de correo.
- EMAIL_NOMBRE: Nombre como remitente.
- EMAIL_POP3: Servidor POP3.
- EMAIL_SMTP: Servidor SMTP.
- EMAIL_USER: Usuario de correo.
- EMAIL_PASS: Clave de correo.

Para que el servicio de avisos por correo funcione correctamente, debe estar creado el JOB de base de datos indicado en el punto 3.5 de este documento.

4.5 Permisos Oracle a nuevos usuarios de aplicación

Para permitir el acceso a las API's de [Port@firmas](#), existe un Rol así como una serie de permisos adicionales sobre objetos necesarios para resolver la comunicación vía PLSQL o base de datos de [Port@firmas](#) con el resto de aplicaciones.

Al igual que en pasos anteriores, hay que hacer los siguientes pasos desde un SQLPLUS:

Para la creación del rol es necesario haber lanzado previamente el siguiente script:

```
SQL> @"D:\KitPortafirma\Scripts\roles y permisos\CREACION ROL PF_R_PETICIONES.sql"
```

La creación del rol se realiza una única vez en el sistema. El resto de pasos si hay que repetirlos por cada usuario de aplicación que vaya a emplear las API's.

Para asignarle los permisos a un usuario hay que lanzar el script:

```
SQL>@"D:\KitPortafirma\Scripts\roles y permisos\PERMISOS USUARIO PETICIONES.SQL" USUAPLIC
```

Para crearle los sinónimos, con el usuario final lanzar el siguiente script:

```
SQL>@"D:\KitPortafirma\Scripts\roles y permisos\SINONIMOS USUARIO PETICIONES.SQL" USUAPLIC
```

Tras esto el usuario Oracle tendrá todo lo necesario para probar la demo de integración desde PLSQL o bien atacar directamente las API's.⁵

⁵ Si se quiere crear nuevos paquetes PLSQL almacenados que ataquen a las API's de [Port@firmas](#), el usuario propietario no le bastará solo el Rol sino necesitará también los permisos específicos de ejecución sobre los mismos. Se puede modificar el script de creación del rol y concederle estos mismos permisos al usuario de destino.

5 Historia de versiones

VERSION	FECHA	AUTOR	DESCRIPCIÓN
1.0.0	27/06/2005	FJCV	Primera versión del documento, Port@firmas v1.0.0
1.0.1	30/09/2005	FJCV	Revisión requisitos, almacén de certificados, gestión de memoria en Linux/Unix y configuración de avisos por correo.
1.0.2	21/10/2005	FJCV	Redirección en detección de componentes a servidor Fachada.
1.1.0	28/12/2005	FJCV	Migración v1.0.X a v1.1.0 y revisión de la documentación.
1.1.1	03/02/2006	FJCV	Inclusión del punto 4.2.4 que aclara como configurar los códigos de barras en servidores Unix/Linux.