

# Documento de Operación y Despliegue



Documento elaborado por: Francisco José Cantero Villar

Revisado por: Miguel J. Vázquez Rebollo

Versión: 1.4.0
Lugar y fecha: Sevilla, 19 de Junio de 2007

## Contenido:

1	Requisitos previos de Instalación .....	4
1.1	Tabla de compatibilidades.....	4
2	Procedimiento de migración a v1.4.0.....	5
2.1	Carga de procedimientos .....	5
2.1.1	Especificaciones de paquete .....	5
2.1.2	Cuerpos de paquete .....	6
3	Procedimiento de primera instalación.....	8
3.1	Creación usuario de base de datos .....	8
3.2	Script's de creación de tablas y objetos .....	8
3.3	Carga de datos .....	9
3.4	Carga de procedimientos .....	10
3.4.1	Clase Base64 .....	10
3.4.2	Especificaciones de paquete .....	10
3.4.3	Cuerpos de paquete .....	11
3.5	Tareas programadas en la base de datos .....	12
3.6	Creación de Rol.....	13
4	Procedimientos de configuración y despliegue.....	14
4.1	Alta/Modificación de la aplicación Port@firmas en @firma .....	14
4.2	Despliegue en el Servidor de Aplicaciones Web .....	16
4.2.1	Apache Tomcat .....	16
4.2.2	Oracle OC4J.....	17
4.3	Configuración de la Aplicación .....	22
4.3.1	Propiedades de conexión a base de datos .....	22
4.3.2	Detector.js .....	23
4.3.3	Almacén de certificados.....	23
4.3.4	Ampliación de memoria de la máquina virtual de Java en Tomcat.....	27
4.3.5	Código de barras en servidores Unix/Linux .....	28
4.3.6	Servicio Web, publicación de datos de usuario.....	29
4.3.7	Informe de Firmas por defecto.....	30

4.3.8	Configuración de trazas mediante Log4j.....	31
4.3.9	Servicio web de firma remota .....	33
4.4	Actualizando Servidores de firma en Port@firmas .....	34
4.5	Actualizando las constantes de la aplicación .....	34
4.5.1	Configuración de los servicios de aviso por correo .....	35
4.6	Permisos Oracle a nuevos usuarios de aplicación .....	36
5	Historia de versiones .....	37

## 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 de Aplicaciones Web Tomcat / OC4J con JDK 1.4.2<sup>1</sup> o 1.5.0 con las extensiones JCE correspondientes a su versión necesarias para habilitar la encriptación “fuerte” requerida para atacar a los Servicios Web de @firma. (Ver tabla de compatibilidades definida a continuación)
- (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.

### 1.1 Tabla de compatibilidades

A continuación se detalla la relación de Servidores de Aplicaciones Web con los que se ha testeado y por tanto se garantiza el correcto funcionamiento de la aplicación.

Servidores de aplicaciones	JDK	Soportado	Observaciones
Tomcat 4.1.30	1.4.2_07	Sí	
Tomcat 5.0.28	1.4.2_07	Sí	
Tomcat 5.0.28	1.5.0_06	Sí	
Tomcat 5.5.16	1.5.0_06	Sí	
OC4J 10.1.3	1.4.2_10	Sí	Las librerías de JDBC tienen que ser las que trae el servidor de aplicaciones. Hay que utilizar el war de la versión correspondiente.
OC4J 10.1.3	1.5.0_06	Sí	

El servidor de aplicaciones OC4J 10.1.2 no se ha testeado al llevar una J2EE 1.3 y requerirse como mínimo una versión 1.4 para la descryptación 3DES.

<sup>1</sup> La versión 1.4.2 se recomienda que sea mínimo parche 7 o superior, al tener los parche 2 y 3 un bug interno que da problemas con las comunicaciones SSL.

## 2 Procedimiento de migración a v1.4.0

### 2.1 Carga de procedimientos

A continuación vamos a realizar la carga de paquetes de base de datos, para ello deberemos habernos conectados con el usuario propietario del esquema de Port@firmas mediante un cliente SQL tipo SQPLUS, TOAD o RAPTOR.

#### 2.1.1 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.**

Tras esto pasamos al siguiente punto.

### 2.1.2 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 o RAPTOR teniendo cuidado en respetar el orden de ejecución indicado.

Ahora pasaremos a replegar la aplicación en el Tomcat / OC4J para desplegar el nuevo WAR y revisar los parámetros de conexión a base de datos. Para ello debemos saltar al punto **“4. Procedimientos de configuración y despliegue”**.

## 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 MB, 64 MB y 128 MB 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.

### 3.4 Carga de procedimientos

Una vez que hemos cargado todas las tablas de codificación, vamos a proceder a cargar el resto de script's, 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\Base64plsjava.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\_NOTIFICA 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"**

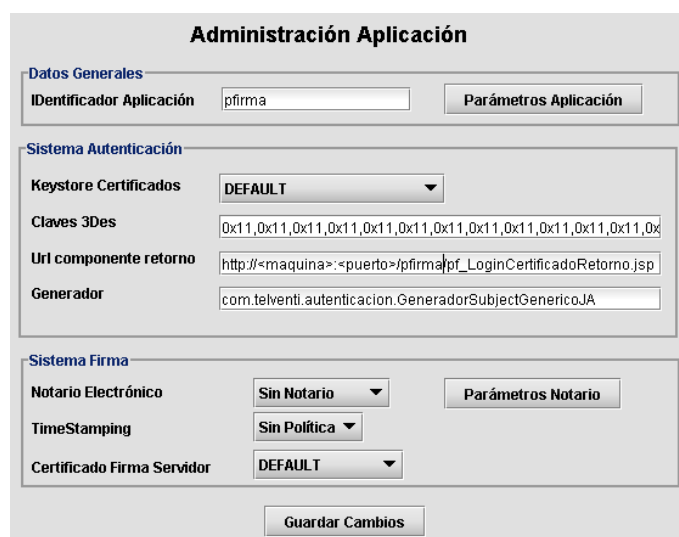
Existen en esta carpeta dos scripts más para ejecutar cuando queramos conceder este rol y el resto de permisos a un usuario de base de datos, así como la generación de sinónimos. En el punto 3.5 del presente documento se detalla más este apartado.

## 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 proporcionándoles 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](http://<maquina>:<puerto>/pfirma/pf_LoginCertificadoRetorno.jsp).
  - Generador: [com.telventi.autenticacion.GeneradorSubjectGenericoJA](#)
  - 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:



- 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](#)<sup>2</sup>
  - Validación de página: [Sí](#)
  - Ceremonia de Firmado: (el valor por defecto)
  - Página Destino Multifirma: [http://<maquina>:<puerto>/pfirma/pf\\_confirmacion\\_multifirma.jsp](http://<maquina>:<puerto>/pfirma/pf_confirmacion_multifirma.jsp)
  - Clave 3Des Multifirma: proporcionada por el administrador o generada por la aplicación.
  - Personalizar el servicio de firma, en el ejemplo que viene a continuación se muestra el caso más sencillo.



<sup>2</sup> 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 Web

### 4.2.1 Apache Tomcat

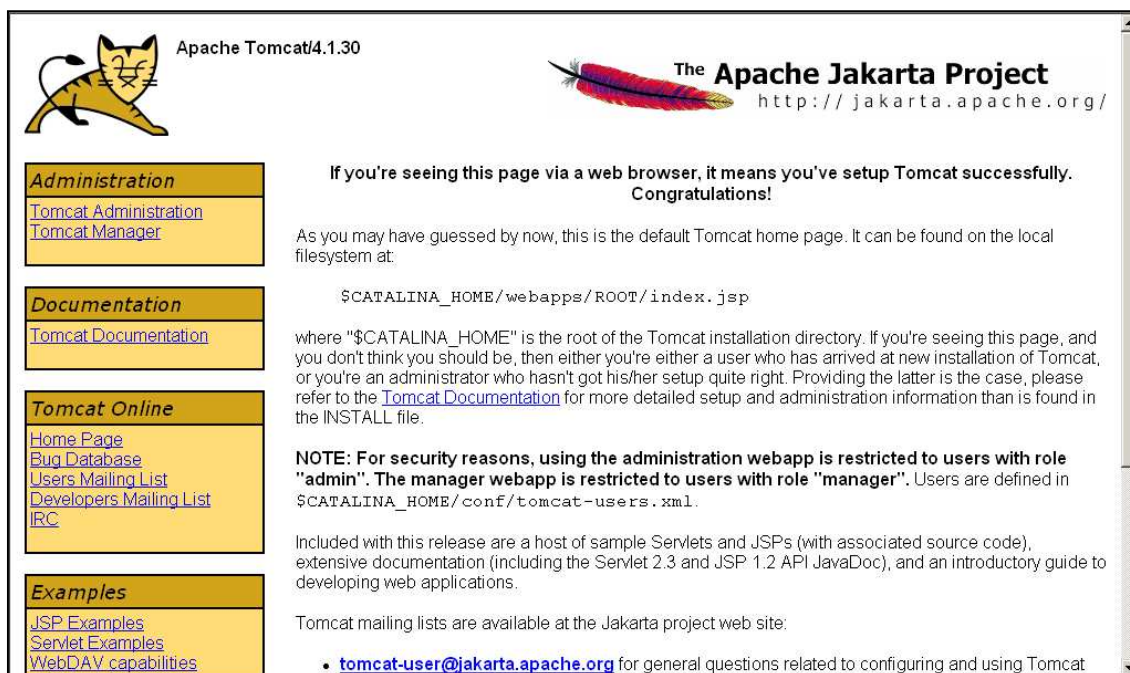
Seguidamente vamos a explicar como se despliega la aplicación de Port@firmas en el Tomcat, así como la herramientas de administración de la misma.

Partimos de la base de que disponemos de un Servidor de Aplicaciones Tomcat y una JDK en la máquina de los indicados en la tabla de compatibilidades. Para el caso del ejemplo veremos un Tomcat 4.1.30 con JDK1.4.2\_07.

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@firmas<sup>3</sup> para Tomcat y pulsamos “instalar” (install)

<sup>3</sup> En el caso de venir de una migración hacer previamente un “undeploy” o “remove” de la aplicación pfirma desde la herramienta Tomcat Manager para borrar la versión anterior.



**Upload a WAR file to install**

Select WAR file to upload

Hacemos igual con el WAR de la Herramienta de Administración<sup>4</sup>.

**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 <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/TrewaFidea	Escritorio Trewa (Versión para Fidea)	true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/acervobot		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/acervoweb		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/admin	Tomcat Administration Application	true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/adminpfirma		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/avisador		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/barbecue		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/demoautenticacion		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/emuladorAries		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/eperFirmaPrueba		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/examples	Tomcat Examples	true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/manager	Tomcat Manager Application	true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/notificador		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/ord_uni_sub		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/pdftoimage		true	0	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>
/pfirma		true	2	Start <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Remove</a>

Una vez desplegadas las dos aplicaciones, las paramos y pasamos al punto 4.3.

#### 4.2.2 Oracle OC4J

A continuación vamos a ver como desplegar la aplicación Port@firmas con la herramienta de administración de OC4J.

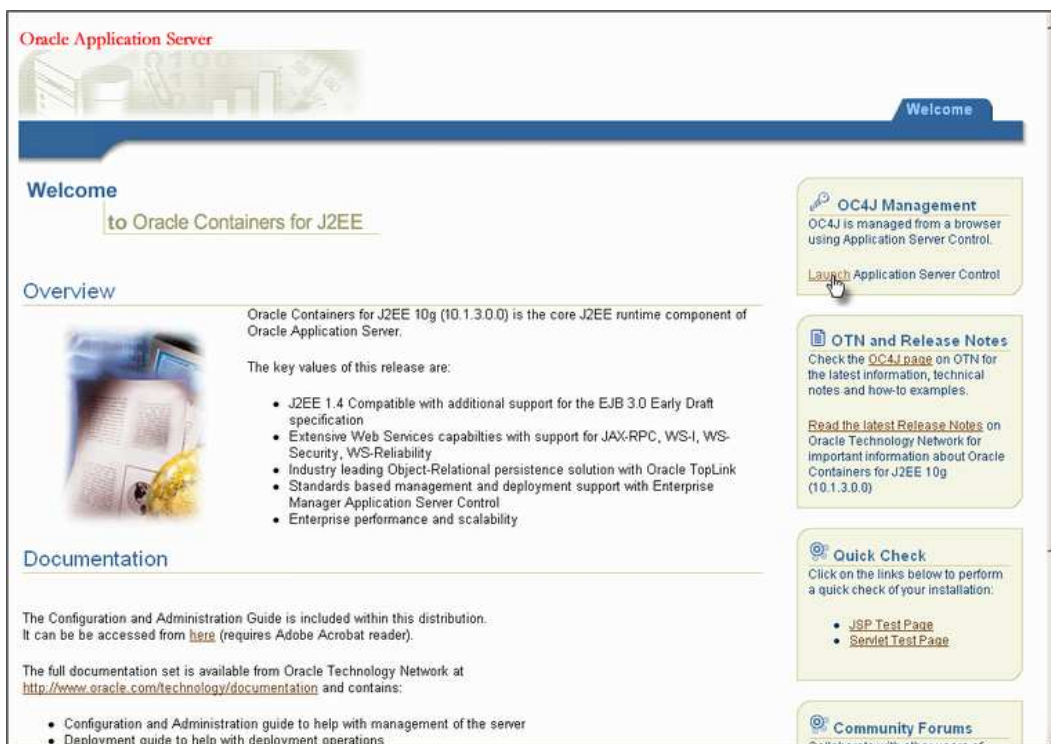
Para ello deberemos entrar en el “manager” accesible desde la portada del servidor de aplicaciones en la siguiente dirección:

<http://servidorAplicaciones:8888>

Con lo que nos saldrá la siguiente pantalla:

---

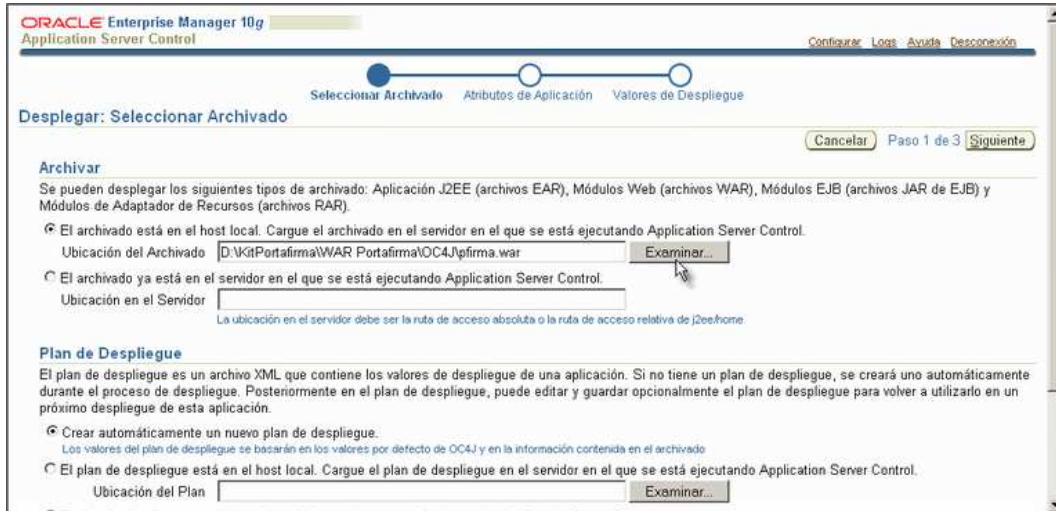
<sup>4</sup> Si ya tenemos la herramienta de administración desplegada no es necesario replegarla y desplegarla de nuevo.



Accedemos al “Application Server Control” como indica la imagen y nos pedirá a continuación el usuario y clave de administración que se facilitó en la instalación del servidor de aplicaciones. Tras esto, nos saldrá la pantalla principal de administración y accederemos al detalle de aplicaciones desplegadas. Desde esta pantalla pulsamos en el botón “Desplegar” e iniciaremos el proceso de instalación de la aplicación.



Primeramente nos pide la ruta al fichero WAR, donde deberemos escoger el existente para OC4J.



ORACLE Enterprise Manager 10g Application Server Control

Configurar Logs Ayuda Desconexión

Selección Archivado Atributos de Aplicación Valores de Despliegue

Desplegar: Seleccionar Archivado Cancelar Paso 1 de 3 Siguiente

**Archivar**

Se pueden desplegar los siguientes tipos de archivado: Aplicación J2EE (archivos EAR), Módulos Web (archivos WAR), Módulos EJB (archivos JAR de EJB) y Módulos de Adaptador de Recursos (archivos RAR).

El archivado está en el host local. Cargue el archivado en el servidor en el que se está ejecutando Application Server Control.

Ubicación del Archivado:  Examinar...

El archivado ya está en el servidor en el que se está ejecutando Application Server Control.

Ubicación en el Servidor:

La ubicación en el servidor debe ser la ruta de acceso absoluta o la ruta de acceso relativa de j2ee.home

**Plan de Despliegue**

El plan de despliegue es un archivo XML que contiene los valores de despliegue de una aplicación. Si no tiene un plan de despliegue, se creará uno automáticamente durante el proceso de despliegue. Posteriormente en el plan de despliegue, puede editar y guardar opcionalmente el plan de despliegue para volver a utilizarlo en un próximo despliegue de esta aplicación.

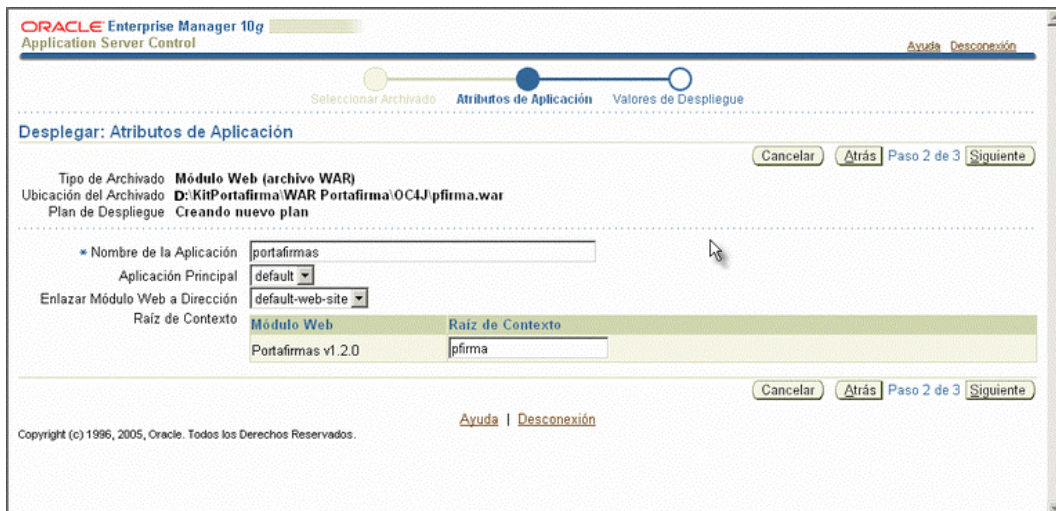
Crear automáticamente un nuevo plan de despliegue.

Los valores del plan de despliegue se basarán en los valores por defecto de OC4J y en la información contenida en el archivado

El plan de despliegue está en el host local. Cargue el plan de despliegue en el servidor en el que se está ejecutando Application Server Control.

Ubicación del Plan:  Examinar...

Tras esto pulsamos siguiente y rellenamos los siguientes datos.



ORACLE Enterprise Manager 10g Application Server Control

Ayuda Desconexión

Selección Archivado Atributos de Aplicación Valores de Despliegue

Desplegar: Atributos de Aplicación Cancelar Atrás Paso 2 de 3 Siguiente

Tipo de Archivado: **Módulo Web (archivo WAR)**

Ubicación del Archivado: **D:\KitPortafirma\WAR Portafirma\OC4J\pfirma.war**

Plan de Despliegue: **Creando nuevo plan**

\* Nombre de la Aplicación:

Aplicación Principal:

Enlazar Módulo Web a Dirección:

Raíz de Contexto:

Módulo Web	Raíz de Contexto
Portafirmas v1.2.0	<input type="text" value="pfirma"/>

Cancelar Atrás Paso 2 de 3 Siguiente

Copyright (c) 1996, 2005, Oracle. Todos los Derechos Reservados. [Ayuda](#) | [Desconexión](#)

Seguidamente nos aparece un diálogo mostrando el resumen de las operaciones que se van a realizar.



Pulsamos el botón “Desplegar” y comienza el proceso de despliegue, tras lo cual nos sale la siguiente pantalla.



Si todo ha ido bien, nos sale el botón "Volver" con el que accedemos a la lista principal de aplicaciones donde ya aparece la de Port@firmas como se aprecia a continuación.



Repetimos el proceso para la herramienta de administración, y una vez hecho, continuamos con el siguiente punto.

## 4.3 Configuración de la Aplicación

### 4.3.1 Propiedades de conexión a base de datos

Accedemos al directorio donde se encuentre desplegada la aplicación dentro del servidor de aplicaciones, lo cual varía según el tipo de servidor

- **Apache Tomcat**

`$CATALINA_HOME\webapps\pfirma\WEB-INF\classes`

- **Oracle OC4J**

`$ORACLE_HOME\j2ee\home\applications\portafirmas\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 @visorador en el caso de tenerlo instalado)
tituloHTML=Port@firmas v1.3.0 - Consejería de XXX - Junta de Andalucía
conexionesMaximas = 50
# El siguiente parametro determina cuantos seg. despues de hacer un close()
# se cierra la conexion fisica
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
timeToLiveTimeOut = 30
```

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 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, paramos y arrancamos de nuevo las aplicaciones y listo.

### 4.3.2 Detector.js

Dentro del directorio “servicio” de la carpeta **lpfirma** de la aplicación, se encuentra el fichero “Detector.js”. En este fichero hay que revisar la variable de retorno en caso de que falten componentes para que apunte al servidor de fachada.

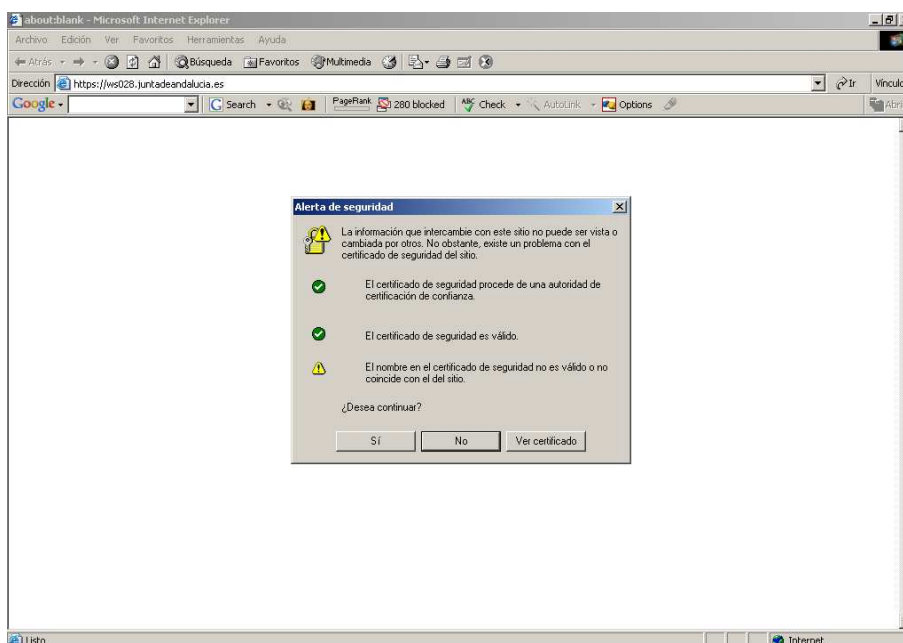
```
// ----- Comienzo Variables a modificar -----  
  
// Url for the next page when all the necessary components are detected  
var nextUrl = "../pf_peticion.jsp";  
  
// Url base of digital sign se  
var url_base = "https://<serviforFachada>:<puerto>/firmadigital";  
  
// ----- Fin variables a modificar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

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

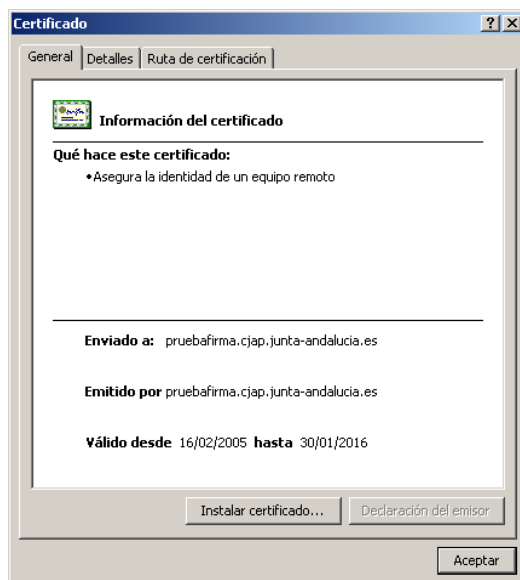
```
var url_base = "https://ws017.juntadeandalucia.es:444/firmadigital";
```

### 4.3.3 Almacén de certificados

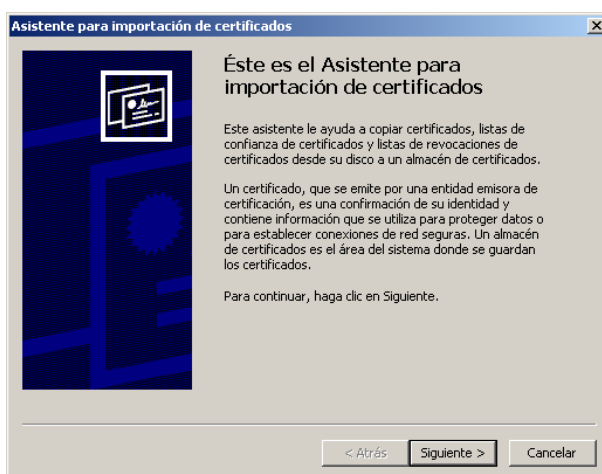
Para poder establecer la comunicación SSL entre nuestro servidor de aplicaciones y el servidor de custodia de @firma debemos cargar la clave publica del mismo dentro del almacén de certificados de nuestro servidor de aplicaciones. Para ello, la forma más sencilla es conectarnos con un navegador al servidor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



Pulsamos “Ver certificado”.



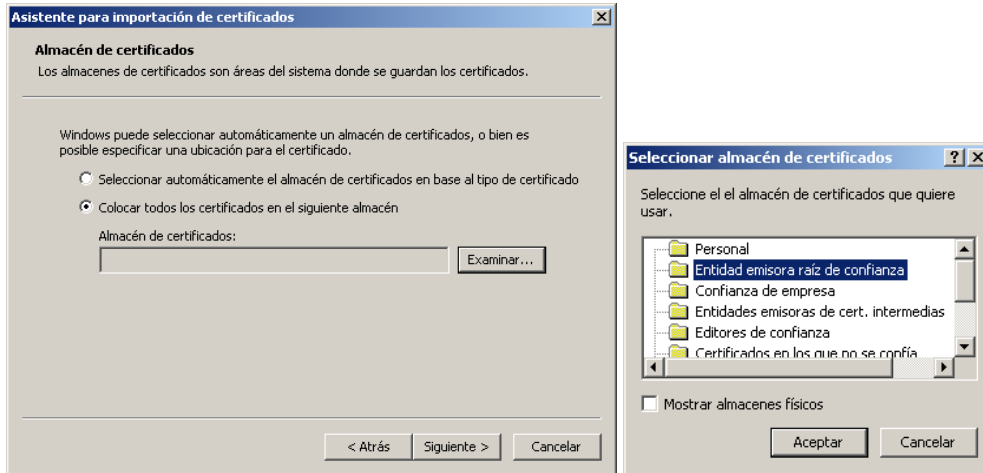
Pulsamos “Instalar certificado”.



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.

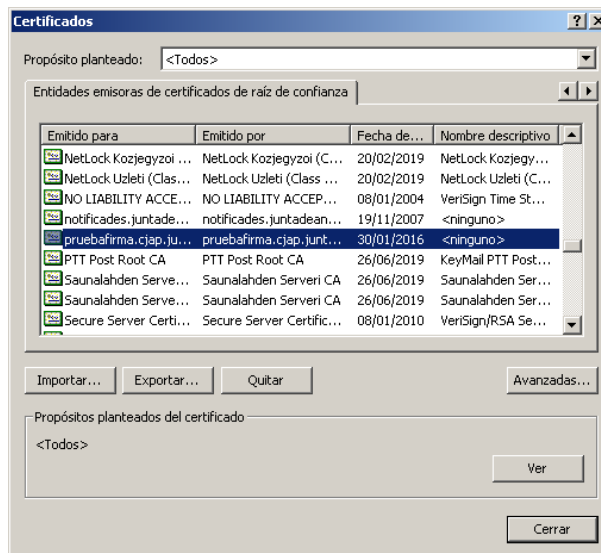


Escogemos esta última tal y como se ve en las siguientes imágenes.

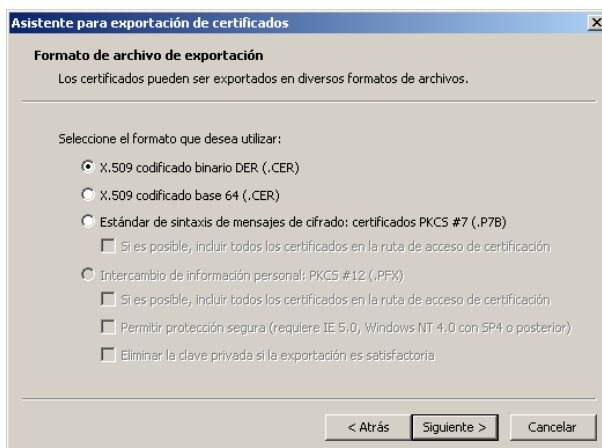


Seguimos hacia delante en el asistente y si todo va bien nos mostrará un diálogo de confirmación de operación realizada con éxito. Tras esto accederemos al menú “Herramientas”, “Opciones de Internet”, “Contenido” pulsamos el botón “Certificados...” de forma que nos saldrá un nuevo asistente.

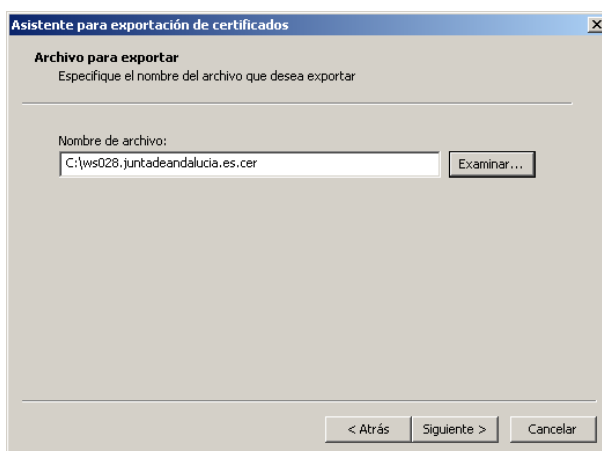
Nos vamos a la última pestaña y buscamos el certificado recién importado.



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.

#### 4.3.4 Ampliación de memoria de la máquina virtual de Java en Tomcat

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.

##### 4.3.4.1 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 "jpd 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 $
# -----
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.

#### 4.3.4.2 Servidor de Aplicaciones Linux/Unix

Para el caso de estos servidores hay que editar el archivo “catalina.sh” que se encuentra en el directorio “\Tomcat\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 $
# -----
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.

#### 4.3.5 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, 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;;
```

En el caso de OC4J, debemos modificar el script de arranque del mismo.

```
tart_oc4j()
{
  echo "Starting OC4J from $J2EE_HOME ..."

  CMDARGS="-config $SERVER_XML"

  while [ $# -ge 1 ]
  do

    case $1 in
      *)
        echo ""
        echo "Error: The option \"$1\" was not recognized."
        EXIT=1
        return
        ;;
      esac
    done

    if [ "$VERBOSE" = "on" ]
    then
      echo "Executing: $JAVA_HOME/bin/java $JVMARGS -jar $OC4J_JAR $CMDARGS"
    fi

    $JAVA_HOME/bin/java -Djava.awt.headless=true $JVMARGS -jar $OC4J_JAR $CMDARGS
  }
}
```

#### 4.3.6 Servicio Web, publicación de datos de usuario

Se ha añadido en esta nueva versión un nuevo método que lista todos los usuarios existentes en Port@firmas por requerimiento de terceras aplicaciones. Debido al carácter de estas informaciones, no en todos los entornos es recomendable la publicación (se recomienda securizar los servicios de Port@firmas siguiendo el documento de “**Securización Port@firmas**” para asegurar la confidencialidad de estas informaciones).

Si no se va a hacer uso de este servicio en su entorno, este se puede inhabilitar de forma sencilla editando mínimamente el fichero de descripción del servicio. Para ello debe acceder al fichero “server-config.wsdd” que se encuentra dentro de la carpeta

/pfirma/WEB-INF/ donde haya desplegado la aplicación.

Debe buscar dentro de dicho fichero el siguiente trozo de código XML marcado en rojo.

```
<operation name="eliminarPetición" qname="operNS:eliminarPetición" xmlns:operNS="http://DefaultNamespace"
returnQName="eliminarPeticiónReturn" returnType="rtns:long" xmlns:rtns="http://www.w3.org/2001/XMLSchema"
soapAction="" >
  <parameter qname="petición" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
</operation>
<operation name="obtenerListaUsuarios" qname="operNS:obtenerListaUsuarios"
xmlns:operNS="http://DefaultNamespace" returnQName="obtenerListaUsuariosReturn"
returnType="rtns:ArrayOf_tns1_UsuarioWS" xmlns:rtns="http://client.pfirma" soapAction="" >
  </operation>

<parameter name="allowedMethods" value="obtenerListaUsuarios eliminarPetición ...
```

Eliminamos el método “obtenerListaUsuarios” de los métodos permitidos del servicio, así como la operación en sí. Quedando esta entrada en el XML de la siguiente forma:

```
<operation name="eliminarPetición" qname="operNS:eliminarPetición" xmlns:operNS="http://DefaultNamespace"
returnQName="eliminarPeticiónReturn" returnType="rtns:long" xmlns:rtns="http://www.w3.org/2001/XMLSchema"
soapAction="" >
  <parameter qname="petición" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
</operation>

<parameter name="allowedMethods" value="eliminarPetición ...
```

Tras esto reiniciaremos la aplicación en el Tomcat/OC4J y habremos inhabilitado el método del servicio.

Si luego interesa restaurarlo sólo tenemos que recuperar el WSDD del WAR original de la aplicación y sobrescribir el actual modificado.

#### 4.3.7 Informe de Firmas por defecto

En esta nueva versión se ha incluido un nuevo tipo de Informe de Firmas que incluye un nuevo pie de firma digital con el código de transacción cifrado y que hace referencia a la herramienta de Verificación de Firmas.

Configurar Port@firmas para que al invocar al informe de firmas nos genere uno u otro es sencillo, solo tendremos que editar un nuevo archivo de propiedades “informe.properties” que se encuentra dentro de la carpeta pfirma/WEB-INF/classes/.


```
tipo=1
tipo2.url=https://ws050.juntadeandalucia.es/verificarfirma
tipo3.url=https://ws050.juntadeandalucia.es/verificarfirma
```

Indicando valores “1”, “2” ó “3” cambiamos el informe a generar.

Tipos de pie de firma:

ID. FIRMA	scfirma 1 / 16829		
FECHA Y HORA	17/05/2006 13:22:51	PÁGINA	1 / 1
FIRMADO POR	CANTERO VILLAR FRANCISCO JOSE	CERT. EXPEDIDO	CERT. CADUCIDAD
		17/09/2006	17/09/2006
 scfirma 1/16829			

Tipo 1

Código Seguro de verificación: 16829. Permite la verificación de la integridad de una copia de este documento electrónico en la dirección: <a href="https://ws050.juntadeandalucia.es/verificarfirma">https://ws050.juntadeandalucia.es/verificarfirma</a>			
FIRMADO POR	CANTERO VILLAR FRANCISCO JOSE	FECHA Y HORA	17/05/2006 13:22:51
ID. FIRMA	scfirma 1	16829	PÁGINA 1 / 1
 16829			

Tipo 2

Código Cifrado de verificación: wGtdhuYDc1U=. Permite la verificación de la integridad de una copia de este documento electrónico en la dirección: <a href="https://ws070.juntadeandalucia.es/verifica/">https://ws070.juntadeandalucia.es/verifica/</a>			
FIRMADO POR	CANTERO VILLAR FRANCISCO JOSE	FECHA Y HORA	14/11/2006 19:15:12
ID. FIRMA	svrpuafirma.cice.junta-andalucia.es	wGtdhuYDc1U=	PÁGINA 1 / 1
 wGtdhuYDc1U=			

Tipo 3

La variable “tipo2.url” o “tipo3.url”, permite definir la ruta donde está desplegada la aplicación de Verificación de Firmas. Si no se tiene o no se va a usar, el valor de esta variable es indiferente.

Por defecto y por mantener compatibilidades con las versiones anteriores el valor inicial en el WAR de la aplicación es el tipo 1.

De todas formas, las aplicaciones clientes pueden pedir el informe de forma explícita pasando un nuevo parámetro opcional al Servlet que devuelve el Informe de Firmas.

<http://servidor:puerto/pfirma/informeFirma/?docHASH=códigoDocumento&petHASH=códigoPetición&tipo=X>

### 4.3.8 Configuración de trazas mediante Log4j

La nueva versión de Port@firmas tiene una mejor gestión de los mensajes de Log que hasta la actualidad solo se mostraban por la consola del servidor de aplicaciones. Ahora se puede configurar para que vuelque los mensajes de error a un archivo así como la cantidad de información a mostrar para una mejor depuración de las incidencias.

Para ello debemos buscar el fichero “log4j.properties” que se encuentra dentro de la carpeta pfirmas/WEB-INF/classes/ el cual contiene inicialmente el siguiente contenido:

```
#Declaracion del Logger
log4j.rootLogger = ERROR, fichero
#, pantalla
log4j.additivity.fichero=false

#Configuracion del Logger
log4j.appender.fichero=org.apache.log4j.RollingFileAppender
log4j.appender.fichero.layout=org.apache.log4j.PatternLayout
# con nombre de metodo, nombre de clase y linea
log4j.appender.fichero.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m (%M [%C:%L])%n
# solo nombre de clase
#log4j.appender.fichero.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m (%C)%n
# sin nombre de metodo, sin nombre de clase y sin linea
#log4j.appender.fichero.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m%n
log4j.appender.fichero.MaxFileSize=10MB
log4j.appender.fichero.MaxBackupIndex=10
# ruta en la que se va a crear el log
log4j.appender.fichero.File=c:/pfirma.log

# pantalla
#log4j.appender.pantalla=org.apache.log4j.ConsoleAppender

# patron de pantalla
#log4j.appender.pantalla.layout=org.apache.log4j.PatternLayout
#log4j.appender.pantalla.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m (%M [%C:%L])%n
```

Por defecto solo muestra trazas por un fichero de log ubicado en la ruta C:/pfirma.log, por tanto debemos revisar este valor para que la ruta sea la correcta dentro de nuestro de servidor de aplicaciones.

Si queremos modificar las trazas para que también salgan por pantalla deberemos hacer los siguientes cambios marcados en verde.

```
#Declaracion del Logger
log4j.rootLogger = ERROR, fichero, pantalla
#, pantalla
log4j.additivity.fichero=false

#Configuracion del Logger
log4j.appender.fichero=org.apache.log4j.RollingFileAppender
log4j.appender.fichero.layout=org.apache.log4j.PatternLayout
# con nombre de metodo, nombre de clase y linea
log4j.appender.fichero.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m (%M [%C:%L])%n
# solo nombre de clase
#log4j.appender.fichero.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m (%C)%n
# sin nombre de metodo, sin nombre de clase y sin linea
#log4j.appender.fichero.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m%n
log4j.appender.fichero.MaxFileSize=10MB
log4j.appender.fichero.MaxBackupIndex=10
# ruta en la que se va a crear el log
log4j.appender.fichero.File=D:/Java/Tomcat5.0/logs/pfirma.log

# pantalla
log4j.appender.pantalla=org.apache.log4j.ConsoleAppender

# patron de pantalla
log4j.appender.pantalla.layout=org.apache.log4j.PatternLayout
log4j.appender.pantalla.layout.ConversionPattern=%-5p:%d{[yyyy-MM-dd HH:mm:ss,SSS]} %m (%M [%C:%L])%n
```

Cuando cambiemos la configuración del Log, para que la aplicación tome los mismos, deberemos reiniciar la aplicación desde la herramienta de administración del servidor de aplicaciones.



### 4.3.9 Servicio web de firma remota

En esta versión se ha incluido un nuevo servicio web para realizar firmas remotas a través de Port@firmas. El servicio está activo por defecto. Para desactivarlo sólo tenemos que comentar las siguientes líneas en el fichero **/pfirma/WEB-INF/server-config.wsdd**:

```
...
<service name="FirmaRemotaWS" provider="java:RPC" style="rpc" use="encoded">
  <parameter name="wsdlTargetNamespace" value="http://firmaremotas.ws.pfirma"/>
  <parameter name="wsdlServiceElement" value="FirmaRemotaWSService"/>
  <parameter name="wsdlServicePort" value="FirmaRemotaWS"/>
  <parameter name="className" value="FirmaRemotaWS"/>
  <parameter name="wsdlPortType" value="FirmaRemotaWS"/>
  <parameter name="typeMappingVersion" value="1.2"/>
  <operation name="iniciarFirma" qname="operNS:iniciarFirma" xmlns:operNS="http://DefaultNamespace"
returnQName="iniciarFirmaReturn" returnType="rtns:string" xmlns:rtns="http://www.w3.org/2001/XMLSchema"
soapAction="" >
  <parameter qname="hashDoc" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
  <parameter qname="idUsuario" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
  </operation>
  <operation name="finalizarFirma" qname="operNS:finalizarFirma" xmlns:operNS="http://DefaultNamespace"
returnQName="finalizarFirmaReturn" returnType="rtns:double" xmlns:rtns="http://www.w3.org/2001/XMLSchema"
soapAction="" >
  <parameter qname="hashDoc" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
  <parameter qname="datosFirmados" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
  <parameter qname="idUsuario" type="tns:string" xmlns:tns="http://www.w3.org/2001/XMLSchema"/>
  </operation>

  <parameter name="allowedMethods" value="iniciarFirma finalizarFirma"/>
</service>
...
```



#### 4.5.1 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.

## 4.6 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.<sup>5</sup>

---

<sup>5</sup> 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.
1.2.0	23/06/2006	FJCV	Migración v1.1.0 a v1.2.0, tabla de compatibilidades, despliegue en OC4J, inhabilitar el servicio y configuración de Informes de Firma y Log4j.
1.3.0	20/12/2006	FJCV	Nuevo proceso de migración y eliminación de la carga de imágenes en la tabla de constantes.
1.4.0	19/06/2007	MJVR	Nuevo proceso de migración e introducción de servicio web de firma remota.