Manual de configuración de los servicios del Portafirmas móvil

Índice de contenidos

[1 Introducción 3](#_Toc129258567)

[2 Servicios del Portafirmas móvil 4](#_Toc129258568)

[2.1 Servicio Proxy 4](#_Toc129258569)

[2.1.1 Configuración del servicio 4](#_Toc129258570)

[2.1.2 Configuración de los logs 6](#_Toc129258571)

[2.2 Servicio de firma trifásica 8](#_Toc129258572)

[2.2.1 Configuración del servicio 8](#_Toc129258573)

# Introducción

El Portafirmas móvil es una *app* para dispositivos móviles que permite realizar las operaciones más destacadas del Portafirmas web del Mineco. Esta app sirve únicamente como interfaz del Portafirmas web, de tal forma que toda la lógica se mantiene en servidor a excepción de la operación de firma, que se realiza con el certificado que el usuario tiene instalado en el dispositivo y con el que accede a la aplicación.

Las operaciones que permite realizar el Portafirmas móvil es:

* Listado de solicitudes de firma (pendientes, firmadas y rechazadas).
* Visualización del detalle de las peticiones.
* Visualización de los documentos de las peticiones y de las firmas e informes de las peticiones ya firmadas.
* Firma, visto bueno y validación de peticiones.
* Rechazo de peticiones.
* Gestión de validadores y autorizaciones del usuario.

Para la ejecución de estas operaciones, el Portafirmas móvil requiere de una serie de servicios web. El presente manual trata de la configuración de estos servicios.

# Servicios del Portafirmas móvil

La solución de Portafirmas móvil se soporta en una serie de servicios que le permiten comunicarse con el Portafirmas web y realizar las distintas operaciones de firma. Estos son los servicios de proxy y de firma trifásica del Cliente @firma, respectivamente.

## Servicio Proxy

El servicio proxy del portafirmas web es el que se encarga de recoger las peticiones del Portafirmas móvil y redirigirlas al Portafirmas Web y/o al servicio de firma trifásica cuando se trate de operaciones de firma.

Este servicio se distribuye mediante el archivo afirma-signfolder-proxy.war.

### Configuración del servicio

El servicio de proxy se configura a través del fichero pfmovil.properties. El servicio buscará este fichero en el directorio configurado mediante la propiedad Java *pfmovil.config.path*. Si no se configurase esta propiedad o no se encontrase en el directorio establecido, se buscará dentro del WAR del servicio (afirma-signfolder-proxy.war), en el subdirectorio “WEB-INF/classes”.

Para establecer el directorio de configuración, se podría proporcionar la propiedad directamente al servidor de aplicaciones durante el arranque. Por ejemplo:

-Dpfmovil.config.path=/opt/usuarios/portafirmas/config

Las propiedades que se pueden configurar desde este fichero son:

* *signfolder.ws.url*
  + Indica cual es la URL de los servicios web del Portafirmas que dan soporte al Portafirmas móvil. La URL debe apuntar al WSDL publicado.
  + Ejemplo:
    - https://mihost.com/pf/servicesv2/MobileService?wsdl
* *proxy.server.url*
  + URL base en el que se despliega el proxy del portafirmas. Necesario para las redirecciones en las autenticaciones con Cl@ve.
  + Ejemplo:
    - https://mihost.com/pf/afirma-signfolder-proxy
* *triphase.server.url*
  + Indica la URL del servicio de firma trifásica.

**IMPORTANTE:** La ruta de acceso al servidor trifásico debería ser interna cuando ambos se encuentren en el mismo servidor, evitando que la comunicación entre ambos servicios deba salir de la red o atravesar cualquier proxy.

* + En caso de que el servicio cambiase de puerto en cada ejecución del servidor de aplicaciones, es posible utilizar el símbolo “${tomcat.httpport}” para cargar el puerto establecido en la variable de entorno Java “tomcat.httpport”.
  + Ejemplo:
    - http://192.168.1.24:${tomcat.httpport}/afirma-server-triphase-signer/SignatureService
* *forced.extraparams*
  + Permite establecer parámetros de configuración que se establecerán para TODAS las firmas que se realicen a través del Portafirmas móvil. Estas propiedades se agregarán a las especificadas para cada documento y sustituirán a aquellas que se estableciese en origen.
  + Los parámetros se establecerán como una cadena de pares “clave=valor”. Las distintas propiedades se separarán con el carácter punto y coma (‘;’).
  + Por defecto:
    - mode=implicit (Todas las firmas será implícitas)
* *cache.enabled*
  + Habilita la cache para conservar los documentos entre las operaciones de prefirma y postfirma en lugar de tener que recuperarlo del Portafirmas en cada ocasión. Habilitar esta opción habitualmente reducirá la velocidad del servicio y reducirá el tráfico de red entre el servicio proxy y el Portafirmas. Su desempeño dependerá de la implementación del sistema de caché.
  + Por defecto:
    - false
* *cache.system.expirationtime*
  + Número de milisegundos que deben transcurrir para considerar caducado un fichero en cache. A partir de ese momento el servicio podrá borrarlo.
  + Por defecto:
    - 60000 (1 minuto)
* *cache.system.class*
  + Nombre de la clase que gestiona la cache. Sólo se utiliza si la cache está habilitada.
  + Opciones disponibles:
    - es.gob.afirma.signfolder.server.proxy.FileSystemDocumentCache
      * Almacena los documentos temporalmente en el sistema de ficheros. En caso de desplegarse el servicio en alta disponibilidad, se debería utilizar una unidad compartida por todos los nodos.
  + Por defecto:
    - es.gob.afirma.signfolder.server.proxy.FileSystemDocumentCache
* *cache.filesystem.dir*
  + Directorio para el guardado de datos en cache. Esta propiedad solo se utiliza cuando la cache está habilitada y se utiliza el sistema *FileSystemDocumentCache*. Si se despliega el servicio en varios nodos, debería ser un directorio compartido por todos ellos.
  + Si no se indica, se almacenarán en el directorio temporal del usuario.
* *share.sessions.enable*
  + Permite habilitar la compartición de sesiones en despliegues con varios nodos. Por defecto, esto sólo habilita al sistema para el uso de certificados en la nube. Para habilitarlo también para certificados locales hay que habilitar la propiedad "share.sessions.withcertificate.enable".
  + Por defecto:
    - false
* *share.sessions.withcertificate.enable*
  + Habilitar la compartición de sesiones para el uso de certificados locales con varios nodos. Requiere que se habilite también la propiedad "share.sessions.enable".
  + Por defecto:
    - true
* *share.sessions.dir*
  + Ruta absoluta del directorio en el que se almacenaran las sesiones compartidas. Debe ser accesible por todos los nodos del despliegue.
  + Si no se indica, se almacenarán en el directorio temporal del usuario.
* *share.sessions.requeststoclean*
  + Número de peticiones que se admiten hasta iniciar la limpieza del almacén de sesiones compartidas.
  + Por defecto:
    - 1000
* *signfolder.ws.username*
  + Nombre de usuario para el acceso al *web service* del Portafirmas en caso de que se le haya establecido un usuario/contraseña del acceso.
  + Si no se indica, no se agregará la cabecera de seguridad para el acceso al servicio.
* *signfolder.ws.password*
  + Contraseña de acceso al *web service* del Portafirmas en caso de que se le haya establecido un usuario/contraseña del acceso.
  + Si no se indica, no se agregará la cabecera de seguridad para el acceso al servicio.

### Configuración de los logs

El servicio proxy utiliza la biblioteca SLF4J como fachada para la gestión de los logs e incluye las bibliotecas de Logback para imprimir los logs a través de este *framework*.

Por defecto los logs de la aplicación se imprimirán junto al resto de logs del servidor de aplicaciones, pero se puede alterar este comportamiento configurando el *framework* de logs. En el caso de Logback, se puede utilizar un fichero externo en el que se establezca la configuración que se debe emplear. Se puede indicar a la aplicación cual es el fichero de configuración proporcionando durante el arranque del servicio la propiedad “logback.configurationFile” con la ruta del fichero en cuestión. Por ejemplo:

-Dlogback.configurationFile=/opt/usuarios/portafirmas/proxy\_logback.xml

Un ejemplo de fichero de configuración de Logback sería:

|  |
| --- |
| <configuration>  <!-- Configuración del log -->    <appender name="ROLLING" class="ch.qos.logback.core.rolling.RollingFileAppender">      <!-- Ruta del fichero de log -->      <file>RUTA\_LOGS/proxy\_log.txt</file>      <!-- Rotado diario o por tamaño -->      <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">        <!-- Nombre del fichero de rotado -->        <fileNamePattern> proxy\_log -%d{yyyy-MM-dd}.%i.txt</fileNamePattern>         <!-- Tamaño maximo de fichero de 50MB -->         <maxFileSize>50MB</maxFileSize>         <!-- Máximo de 20 ficheros totales -->         <maxHistory>20</maxHistory>         <!-- Tamaño total no superior a 1GB -->         <totalSizeCap>1GB</totalSizeCap>      </rollingPolicy>      <!-- Formato de las entradas de log -->      <encoder>        <pattern> %d %-5level [%thread] %logger{0}: %msg%n </pattern>      </encoder>    </appender>    <!-- Nivel de log y configuración a aplicar -->    <root level="info">      <appender-ref ref="ROLLING" />    </root>  </configuration> |

Consulte la documentación de Logback para saber las opciones de configuración que ofrece este framework: <https://logback.qos.ch/>

#### Configuración de los logs

Si un integrador tuviese interés en utilizar un *framework* de gestión de logs distinto a Logback (Java Logging API, Log4J 2...), puede modificar el WAR del servicio proxy eliminando las bibliotecas de Logback y agregando las del *framework* que desee utilizar y la biblioteca puente correspondiente para SLF4J. Consulte la documentación de SLF4J para más información: <https://www.slf4j.org/>

## Servicio de firma trifásica

El servicio de firma trifásica permite realizar los procesos de firma del portafirmas móvil. Este servicio se encarga de obtener la información que el usuario cifrara con su clave privada en su dispositivo y compondrá la firma en servidor con el formato adecuado.

### Configuración del servicio

El servicio de firma trifásica se distribuye ya configurado para que funcione correctamente con el Portafirmas móvil. Existe la posibilidad, sin embargo, de configurar varias propiedades para ajustar su comportamiento a nuestro despliegue. Para configurar el servicio se utiliza el fichero tps\_config.properties. Ese fichero se buscará en el directorio configurado a través de la variable de entorno *clienteafirma.config.path*. Si no se proporciona esta variable, se buscará dentro del propio WAR del servicio (afirma-server-triphase-signer.war), en el subdirectorio “WEB-INF/classes”.

Para establecer el directorio de configuración, se podría proporcionar la propiedad directamente al servidor de aplicaciones durante el arranque. Por ejemplo:

-Dclienteafirma.config.path=/opt/usuarios/portafirmas/properties

De entre todas las propiedades de configuración que admite este servicio, para los despliegues del Portafirmas móvil sólo se recomienda modificar las siguientes:

* *Access-Control-Allow-Origin*
  + Permite establecer el origen permitido de las peticiones. El servicio de firma trifásica agregará el valor de esta propiedad en las respuestas del servicio.
  + Si se establece como valor un asterisco (‘\*’), se indica que se pueden realizar peticiones desde cualquier dominio.
  + Valor por defecto: \*
* *alternative.xmldsig*
  + Permite habilitar el modo de compatibilidad con bibliotecas de firma XML que puedan encontrarse en el *classpath* del servidor de aplicaciones. Este tipo de bibliotecas pueden interferir con las que incluye el propio Oracle Java e impedir realizar firmas XAdES. Ejemplos de bibliotecas que provocan estos errores son XERCES/XALAN.
  + Al indicar el valor true, se habilitará el modo de compatibilidad con estas bibliotecas, lo que obligará al servicio a buscar diversos paquetes de clases para encontrar un modo de completar las firmas.
  + No se garantiza la compatibilidad con todas las versiones de estas bibliotecas.
  + Valor por defecto: false

**IMPORTANTE:** Será necesario establecer a true esta propiedad cuando se desplieguen los servicios sobre un servidor JBoss 6 o superior.

[Creative Commons License](#Licencia_Creative_Commons)

Esta obra está bajo una licencia [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported](#Licencia_Creative_Commons).