Manual de Integración
PFI v1.4

Autor:	Víctor Sánchez Chueca
Tipo de Documento:	Manual de integración
Grupo de Trabajo:	PAEGA – Dirección
Versión:	1.4
Fecha:	11/06/2013
Fichero:	PAEGA_IN1_ManualIntegracion_PFI_v1.4

PROYECTO COFINANCIADO POR LA UNIÓN EUROPEA
FONDO EUROPEO DE DESARROLLO REGIONAL

Medidas susceptibles de financiación
plan avanza
eModel

Construyendo Europa desde Aragón

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

Historia del Documento			
Fecha	**Versión**	**Autor**	**Descripción**
14/02/2013	V1.0	Víctor Sánchez	Creación del documento.
21/02/2013	V1.1	Alberto Celma	Actualización del documento.
21/02/2013	V1.2	Diana Bellorín	Agregando los códigos de error y su descripción
22/02/2013	V1.3	Diana Bellorín	Agregando los tipos de datos
22/02/2013	V1.4	Diana Bellorín	Actualizando los tipos de datos
25/02/2013	V1.4	Alberto Celma	Actualizando casos de uso de integración
11/06/13	V1.4	Javier Martínez	Actualización del documento

Lista de Distribución	
Nombre	**Empresa/ Departamento**
Todo el personal	UTE - TelSoluciones-Iasoft-Tb-solutions
Personal DGA	DGA – Inspección y Servicios

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

Índice de contenido

1 Versión del servicio....5
2 Descripción funcional básica del servicio....6
3 Tipos de datos....7
3.1 BaseData....7
3.2 ParamSendDocument....7
3.3 ResultSendDocument....7
3.4 ParamGetDocumentState....8
3.5 ParamUpdateQuorumAndLimitDate....8
3.6 ResultUpdateQuorumAndLimitDate....8
4 Tipificación, codificación y descripción de errores....10
4.1 Errores servicios externos....10
4.2 Errores internos....16
4.3 Errores de conexión con servicios de PFI....21
4.4 Errores de conexión con servicios externos....21
4.5 Errores en la ejecución de las tareas....22
4.6 Errores de DWR (Direct Web Remoting)....23
4.7 Errores de firma....23
5 Servicios web....24
5.1 IActiveAuditorService....27
5.1.1 Descripción funcional básica....27
5.1.2 Métodos de integración con otras aplicaciones....27
5.1.3 Otros métodos....27
5.2 IAdviceService....28
5.2.1 Descripción funcional básica....28
5.2.2 Métodos de integración con otras aplicaciones....28
5.2.3 Otros métodos....28
5.3 IConfigurationService....29
5.3.1 Descripción funcional básica....29
5.3.2 Métodos de integración con otras aplicaciones....29
5.3.3 Otros métodos....29
5.4 IDocumentIntegrationService....30
5.4.1 Descripción funcional básica....30
5.4.2 Métodos de integración con otras aplicaciones....30
5.4.2.1 sendDocumentumDocument....30
5.4.2.1.1 Descripción....30
5.4.2.1.2 Parámetros petición....30
5.4.2.1.3 Parámetros de respuesta....31
5.4.2.1.4 Errores / Excepciones....31
5.4.2.2 updateQuorumAndLimitDateDocument....32
5.4.2.2.1 Descripción....32

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

5.4.2.2.2 Parámetros petición....33
5.4.2.2.3 Parámetros de respuesta....33
5.4.2.2.4 Errores / Excepciones....33
5.4.3 Otros métodos....34
5.5 IDocumentMetadataService....34
5.5.1 Descripción funcional básica....34
5.5.2 Métodos de integración con otras aplicaciones....34
5.5.3 Otros métodos....34
5.6 IDocumentService....35
5.6.1 Descripción funcional básica....36
5.6.2 Métodos de integración con otras aplicaciones....36
5.6.2.1 getDocumentState....36
5.6.2.1.1 Descripción....36
5.6.2.1.2 Parámetros petición....36
5.6.2.1.3 Parámetros de respuesta....37
5.6.2.1.4 Errores / Excepciones....37
5.6.3 Otros métodos....37
5.7 IDocumentumDocumentService....38
5.7.1 Descripción funcional básica....38
5.7.2 Métodos de integración con otras aplicaciones....39
5.7.3 Otros métodos....39
5.8 IHistoryService....39
5.8.1 Descripción funcional básica....39
5.8.2 Métodos de integración con otras aplicaciones....40
5.8.3 Otros métodos....40
5.9 ISignerService....40
5.9.1 Descripción funcional básica....41
5.9.2 Métodos de integración con otras aplicaciones....41
5.9.3 Otros métodos....41
6 Tecnología específica....42
6.1 Integración mediante CXF....42
6.1.1 Configuración de servicios para CXF:....42
7 Servicios en SOA....44
8 Casos de uso de integración....45
8.1.1 Enviar un documento al portafirmas....46
8.1.2 Consulta de cambio de estado de un documento....49
8.1.3 Actualizar quorum y fecha límite de preparación....52
9 Procedimiento de alta de una aplicación para integrarse con el servicio....54

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

1 Versión del servicio

El presente documento detalla la integración con la versión 1.4.2 del servicio de Portafirmas.

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

2 Descripción funcional básica del servicio

El servicio de Portafirmas proporciona una aplicación que unifica el firmado electrónico de documentos para personas de una misma organización.

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

3 Tipos de datos

A continuación enumeramos los tipos de datos utilizados para el intercambio de información con el sistema:

- ParamSendDocument
- ParamGetDocumentState
- ParamUpdateQuorumAndLimitDate
- ResultSendDocument
- ResultUpdateQuorumAndLimitDate

Notar que todos los tipos de datos extienden de BaseData.

3.1 BaseData

Contiene información del usuario y la aplicación que ejecutan el método del servicio web

Parámetro	Tipo	Obligatorio	Descripción
userNif	String	Si	Nif del usuario
applicationId	String	Si	Identificador de la aplicacion.

3.2 ParamSendDocument

Contiene el csv del documento a guardar.

Parámetro	Tipo	Obligatorio	Descripción
csv	String	Si	Csv del documento.

3.3 ResultSendDocument

Encapsula los parámetros de salida del resultado de guardar un documento.

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

Parámetro	Tipo	Descripción
errorCode	String	Código de error del servicio.
returnCode	String	Indica si la llamada ha finalizado correctamente (OK) o en error (ERROR)
errorMessage	String	Mensaje de error del servicio
exception	String	Información adicional de la excepción producida

3.4 ParamGetDocumentState

Encapsula el csv del documento del cual se desea obtener el estado.

Parámetro	Tipo	Obligatorio	Descripción
csv	String	Si	Csv del documento.

3.5 ParamUpdateQuorumAndLimitDate

Encapsula el csv del documento, el quorum de firmantes y la fecha límite de firma.

Parámetro	Tipo	Obligatorio	Descripción
Csv	String	Si	Csv del documento.
quorum	Integer	Si *	Quorum del firmante
quorumOnly	Boolean	Si *	Indica si se admiten tantas firmas como indica el quorum o no
date	Date	Si *	Fecha límite

*No es obligatorio introducir los campos **quorum** y **date** a la vez, se puede introducir uno u otro.

3.6 ResultUpdateQuorumAndLimitDate

Encapsula los parámetros de salida del resultado de informar del número de firmantes y de la fecha límite de firma de un documento.

Parámetro	Tipo	Descripción
errorCode	String	Código de error del servicio.
returnCode	String	Indica si la llamada ha finalizado correctamente (OK) o en error (ERROR)

errorMessage	String	Mensaje de error del servicio

4 Tipificación, codificación y descripción de errores

La aplicación PFI implementa una excepción propia llamada InternalCoreException. Esta excepción contiene unos campos adicionales con un código de error y una descripción del mismo, así como información relativa al origen de la excepción en el caso de que se haya producido por integraciones con agentes externos.

A continuación se presentan los códigos de error y su descripción. Estos códigos de error están recogidos en la clase "*ErrorCode.java"* y su descripción en *"ErrorMsg.java".*

4.1 Errores servicios externos

Código	Descripción
FIND_ROLES_SIU_01	No se ha recuperado la lista de cargos del Servicio de Identificación de Usuarios
FIND_USERS_SIU_01	No se recuperado la lista de usuarios del Servicio de Identificación de Usuarios
SIUINTEGRATION_SENDPROPOSEDSIGNERS	Error en el sistema de identificación. El firmante propuesto debe estar dado de alta en el sistema de identificación de usuarios (SIU)
SIUINTEGRATION_USERLISTBYANCESTRALORGANISM	Error al recuperar la lista de usuarios de la rama del organismo
SIUINTEGRATION_DESCENTORGANISMLIST	Error al buscar los organismos descendientes
SIUINTEGRATION_ORGANISMLISTBYORGANISMCODES	Error en la búsqueda de los organismos
SIUINTEGRATION_USERLISTBYFISCALANAGRAMLIST	Error en el sistema de identificación. Se ha producido un error al recuperar los usuarios a partir de su anagrama fiscal
SIUINTEGRATION_ROLELISTBYORGANISMDESCENDANTBRANCHINUSERLIST	Error en el sistema de identificación. Se ha producido un error al recuperar los roles a partir de la lista de usuarios

SIUINTEGRATION_GETANCESTRALCODE	No se ha podido recuperar el organismo padre raíz del árbol al que pertenece el organismo
SIUINTEGRATION_ORGANISMTREE	Error en el sistema de identificación. Se ha producido un error al recuperar el árbol de organismos
SIUINTEGRATION_ORGANISMASCENDANTBRANCHTREE	Error en el sistema de identificación. Se ha producido un error al recuperar la rama ascendente del árbol de organismos
SIUINTEGRATION_GETSTAMPLIST	Error en el sistema de identificación. Se ha producido un error al recuperar los sellos del usuario
SIUINTEGRATION_GETORGANISMLOGIN	Error en el sistema de identificación. Se ha producido un error al recuperar el organismo del usuario
SIUINTEGRATION_GETUSERORGANISM_01	Error en el sistema de identificación. Se ha producido un error al recuperar el rol del usuario
SIUINTEGRATION_GETUSERORGANISM_02	Error en el sistema de identificación. Se ha producido un error al recuperar los organismos del usuario
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_00	Error en el sistema de identificación. Error durante la consulta de los datos de organismos
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_01	Error en el sistema de identificación. El parámetro de entrada token es nulo
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_02	Error en el sistema de identificación. Ha ocurrido algún error al verificar la firma
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_03	Error en el sistema de identificación. El usuario no existe en la base de datos
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_04	Error en el sistema de identificación. El usuario ha sido dado de baja de la base de datos
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_05	Error en el sistema de identificación. El usuario no esta asignado a ningún organismo

SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_06	Error en el sistema de identificación. Ha ocurrido algún error al realizar la búsqueda
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_07	Error en el sistema de identificación. El objeto paramAncestralOrganismByUserList es nulo
SIUINTEGRATION_ANCESTRALORGANISMBYUSERLIST_08	Error en el sistema de identificación. El parámetro de entrada token original es nulo
CREATEADVICE_ADVICEMANAGER_01	Error de validación del aviso
CREATEADVICE_ADVICEMANAGER_02	No se pudo recuperar el correo del usuario
SENDADVICE_ADVICEMANAGER	Error al generar el aviso por correo
CCSV_ERROR_GETMETADATA	Error en el repositorio documental. Se ha producido un error leyendo metadatos.
DOCUMENTUMINTEGRATOR_GETDOCUMENTBYCSV	Error en el repositorio documental. Se ha producido un error recuperando el documento por csv.
DOCUMENTUMINTEGRATOR_GETDOCUMENTTYPELIST	Error en el repositorio documental. Se ha producido un error recuperando los tipos de documentos.
DOCUMENTUMINTEGRATOR_GETDSIGNDOCUMENTUM	Error en el repositorio documental. Se ha producido un error al recuperar la firma del documento.
DOCUMENTUMINTEGRATOR_GETDSIGNATUREDOCUMENTUM	Error en el repositorio documental. Se ha producido un error al recuperar la firma.
DOCUMENTUMINTEGRATOR_GETDOCUMENTMETADATALIST	Error en el repositorio documental. Se ha producido un error al recuperar la lista de metadatos.
DOCUMENTUMINTEGRATOR_INITIALIZEDOCUMENTMETADATA	Error en el repositorio documental. Se ha producido un error al inicializar los metadatos.
DOCUMENTUMINTEGRATOR_GETSIMPLEMETADATA	Error en el repositorio documental. Se ha producido un error al recuperar el metadato simple.
DOCUMENTUMINTEGRATOR_GETMETADATA	Error en el repositorio documental. Se ha

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

	producido un error al recuperar los metadatos.
DOCUMENTUMINTEGRATOR_DELETEDOCUMENT	Error en el repositorio documental. Se ha producido un error al borrar documento.
CCSV_SIGNATURE_ERROR	Error en el repositorio documental. Firma no valida, revocada o no verificada.
DOCUMENTUMINTEGRATOR_CREATESIGNATUTE	Error en el repositorio documental. Se ha producido un error creando la firma.
DOCUMENTUMINTEGRATOR_CREATESIGNATUTEANDVERIFY	Error en el repositorio documental. Se ha producido un error al crear y verificar la firma.
DOCUMENTUMINTEGRATOR_VERIFYDOCUMENTMETADATA	Error en el repositorio documental. Se ha producido un error verificando los metadatos.
DOCUMENTUMINTEGRATOR_CREATEDOCUMENT	Error en el repositorio documental. Se ha producido un error al crear el documento.
DOCUMENTUMINTEGRATOR_CREATERELATIONSHIP	Error en el repositorio documental. Se ha producido un error al crear las relaciones.
DOCUMENTUMINTEGRATOR_GETFICHERODOCUMENTUM_01	Error en el repositorio documental. Se ha producido un error al recuperar el contenido del documento.
DOCUMENTUMINTEGRATOR_GETFICHERODOCUMENTUM_02	Error en el repositorio documental. Se ha producido un generar el contenido del documento.
DOCUMENTUMINTEGRATOR_UPDATESIMPLEMETADATA	Error en el repositorio documental. Se ha producido un error al actualizar el metadato simple.
DOCUMENTUMINTEGRATOR_UPDATESTRUCTUREDMETADATA	Error en el repositorio documental. Se ha producido un error al actualizar el metadato estructurado.
DOCUMENTUMINTEGRATOR_UPDATEDOCUMENT	Error en el repositorio documental. Se ha producido un error al actualizar el documento.
DOCUMENTUMINTEGRATOR_CHANGESTATE	Error en el repositorio documental. Se ha producido un error al cambiar el estado del documento.

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

DOCUMENTUMINTEGRATOR_DELETERELATIONSHIP	Error en el repositorio documental. Se ha producido un error al eliminar la relación.
DOCUMENTUMINTEGRATOR_GETRELATEDDCOUMENTS	Error en el repositorio documental. Se ha producido un error al recuperar los documentos relacionados.
DOCUMENTUMINTEGRATOR_UPDATEDOCUMENTCONTENT	Error en el repositorio documental. Se ha producido un error al actualizar el contenido del documento.
DOCUMENTUMINTEGRATOR_GETDOCUMENTMETADATA	Error en el repositorio documental. Se ha producido un error al recuperar los metadatos del documento.
DOCUMENTUMINTEGRATOR_GETDOCUMENTSIGNATURES	Error en el repositorio documental. Se ha producido un error al recuperar las firmas del documento.
DOCUMENTUMINTEGRATOR_INITIALIZEDOCUMENTMETADATAEXTRAINFO	Error en el repositorio documental. Se ha producido un error al inicializar los metadatos extra.
DOCUMENTUMINTEGRATOR_DELETESIGNATURE	Error en el repositorio documental. Se ha producido un error al borrar la firma.
PAUINTEGRATION_ROLELIST	Error en el sistema de autentificación. Se ha producido un error al buscar los roles de PAU
PAUINTEGRATION_ASSIGNEDROLELIST	Error en el sistema de autentificación. Se ha producido un error al buscar los usuarios por el rol de PAU
PAUINTEGRATION_GETPROCEDUREFROMPAU	Error en el catalogo de procedimientos. Se ha producido un error al buscar el procedimiento desde PAU
PAUINTEGRATION_GETPROCEDUREATTRIBUTES	Error en el catalogo de procedimientos. Se ha producido un error al recuperar los atributos del procedimiento desde PAU
PAUINTEGRATION_USERBYORGANISMTREE	Error en el sistema de autentificación. Se ha producido un error al buscar los usuarios por rol y

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

	organismos
PAUINTEGRATION_GETLASTCONNECTION	Error en el sistema de autentificación. Se ha producido un error al recuperar la fecha de la última conexión del usuario
PAUINTEGRATION_APPLICATIONLIST	Error en el sistema de autentificación. Se ha producido un error al recuperar la aplicación por su alias
PAUINTEGRATION_PROCEDURELIST	Error en el sistema de autentificación. Se ha producido un error al buscar la lista de procedimientos desde PAU
PAUINTEGRATION_PROCEDURELISTBYUSER	Error en el sistema de autentificación. Se ha producido el siguiente error al buscar la lista de procedimientos del usuario desde PAU
PAUINTEGRATION_METHODAPPLICATIONLIST	Error en el sistema de autentificación. Se ha producido un error al recuperar la métodos de la aplicación
PAUINTEGRATION_AUTHORIZEAPPLICATION	Error en el sistema de autentificación. La aplicación no está autorizada para realizar esta operación
ASFINTEGRATOR_GETCAS	Se ha producido un error al recuperar las CAs
ASFINTEGRATOR_SIGNDOCUMENTBYHASH	Error al firmar el documento de resumen con hash
ASFINTEGRATOR_GETCERTIFICATEDATA_01	Error al firmar una cadena de texto para obtener los datos de un certificado
ASFINTEGRATOR_GETCERTIFICATEDATA_02	Error al verificar la firma de una cadena de texto para obtener los datos de un certificado
ASFINTEGRATOR_OBTENERFIRMADOCUMENTO	Error al obtener la firma del documento
ASFINTEGRATOR_VERIFICARYELEVARFIRMA	Error al elevar y verificar la firma
ASFINTEGRATOR_VERYFYSIGNDETACHED	Error al verificar la firma

4.2 Errores internos

GENERIC_CORE_ERROR	Código de error genérico para errores no tratados
ACTIVE_AUDITOR_01	No se ha podido generar la lista de firmantes
ACTIVE_AUDITOR_02	No se ha podido generar la lista de revisores
PARSE_CREATE_BLOB	No se ha podido crear el objeto de base de datos para el binario correspondiente
PARSE_CREATE_CLOB	No se ha podido crear el objeto de base de datos para los caracteres correspondientes
EMPTY_LIST	La lista recuperada esta vacía
EMPTY_DOCUMENT	El archivo recuperado esta vacío
EMPTY_FILE_BEAN	El borrador recuperado esta vacío
CHECK_SUPERADMIN	No se ha podido comprobar que el usuario tenga roles para la aplicación
SAVE_ACTIVE_AUDITOR	Error almacenando el revisor
FIND_ACTIVE_AUDITOR	Error buscando revisor
DELETE_ACTIVE_AUDITOR	Error eliminando revisor
FIND_ACTIVE_AUDITOR_LIST	Error listando revisores activos
SAVE_DOCUMENT	Error almacenando documento
FIND_DOCUMENT	Error buscando documento
DELETE_DOCUMENT	Error eliminando documento. El documento ha sido dado de alta por una aplicación que no es Portafirmas
FIND_DOCUMENT_LIST	Error listando documentos
LOAD_DOCUMENT	Error cargando documentos
DELETE_FILE	Error eliminando fichero

FIND_PROPOSEDSIGNER	Error buscando el firmante propuesto
DELETE_PROPOSEDSIGNER	Error eliminando el firmante propuesto
FIND_PROPOSEDSIGNER_LIST	Error listando firmantes propuestos
LOAD_SIGN	Error cargando firma
DELETE_SIGN	Error eliminando firma
SAVE_HISTORY	Error almacenando historial
UPDATE_HISTORY	Error actualizando historial
FIND_HISTORY	Error buscando historial
FIND_ALL_HISTORY	Error buscando historiales
ADD_NOTE	Error añadiendo la nota
IS_REVISED	El firmante ya ha sido revisado
IS_REVISED_ALL	Los firmantes ya han sido revisados o firmado el documento
FIND_DOCUMENTMETADATA_LIST	Error listando los metadatas del documento
ADD_METADATA	Error al guardar metadato
FIND_METADATA	Error al recuperar el metadato
FIND_METADATA_VALUE	Error al recuperar el valor del metadato
DELETE_METADATA	Error al borrar el metadato
FIND_FILE	Error buscando el borrador
FIND_SIGN	Error buscando la firma
FIND_SIGN_LIST	Error listando firmas
SIGNDAOIMPL_DELETESIGN	Error borrando la firma
SIGNDAOIMPL_SAVESIGN	Error almacenando firma

SIGNDAOIMPL_FINDSIGN	Error recuperando la firma
PDFAUTHORIZATIONGENERATOR_CONFIGUREPDF	Error al generar el pdf
PDFAUTHORIZATIONGENERATOR_ADDPARAGRAPH	Error al añadir párrafo al documento de autorización
PDFAUTHORIZATIONGENERATOR_GENERATETEXT	Error al generar el contenido del documento de autorización
PDFSUMMARYGENERATOR_GENERATETEXT	Error al generar el contenido del documento de resumen
GENERATETEMPORALDOCUMENTWITHBANDANDSTAMP_01	Error al generar el documento temporal para firmar con banda y pie de firma
PDFINTEGRATION_ADDBLANKPAGE	Error al añadir la página en blanco
PDFINTEGRATION_GENERATEBLANKPAGE	Error al crear la página en blanco
GENERATEPDFTOSIGN_GENERATEPDF	Error al generar el documento a firmar con la banda y los pies de firma
PDFINTEGRATION_ISPDFBANNED	Error al intentar comprobar si el documento está permitido
PDFINTEGRATION_FINDTEXT_01	Error al cargar el documento donde se va a buscar el texto
PDFINTEGRATION_FINDTEXT_02	Error al buscar un texto en el documento
PDFINTEGRATION_REPLACETEXT	Error al reemplazar un texto en el documento
SIGNATUREFOLDERACTION_FILLPROCEDURELIST	Error al recuperar la lista de procedimientos
DRAFTDOCUMENTSACTION_FILLPROCEDURELIST	Error al recuperar la lista de procedimientos
TRAYDOCUMENTSACTION_FILLPROCEDURELIST	Error al recuperar la lista de procedimientos
SIGNEDDOCUMENTFOLDERACTION_FILLPROCEDURELIST	Error al recuperar la lista de procedimientos
SIGNATUREFOLDEROTHERSACTION_FILLPROCEDURELIST	Error al recuperar la lista de procedimientos
SIGNEDDOCUMENTFOLDEROTHERSACT	Error al recuperar la lista de procedimientos

ION_FILLPROCEDURELIST	
SIGNATUREFOLDERACTION_REVISEDORREJECTEDLIST_01	Error al revisar o rechazar los documentos
AUDITORFOLDERACTION_FILLPROCEDURELIST	Error al recuperar la lista de procedimientos
SIGNDOCUMENTSACTION_GETLISTUSERSTAMPS	Error al recuperar los sellos
GETUSERSTAMPS_SIGNDOCUMENTSACTION_01	Error al recuperar los roles del firmante
GETUSERSTAMPS_SIGNDOCUMENTSACTION_02	Error al recuperar los sellos del firmante
REJECTEDLIST_SIGNATUREFOLDEROTHERACTION	Error al rechazar los documentos pendientes de firma de la bandeja otros
REVISEDLIST_SIGNATUREFOLDEROTHERACTION	Error al revisar los documentos pendientes de firma de la bandeja otros
REJECTEDLIST_SIGNATUREFOLDERACTION	Error al rechazar los documentos pendientes de firma
REVISEDLIST_SIGNATUREFOLDERACTION	Error al revisar los documentos pendientes de firma
REVISEDLIST_AUDITORFODLERACTION	Error al revisar los documentos pendientes de firma de la bandeja de revisores
REJECTEDLIST_AUDITORFODLERACTION	Error al rechazar los documentos pendientes de firma de la bandeja de revisores
REVISEDORREJECTEDDOC_DOCUMENTDETAILACTION	Error al revisar o rechazar el documento
AUDITORFOLDERACTION_REVISEDORREJECTEDLIST_01	Error al revisar o rechazar los documentos
DOCUMENTMANAGERIMPL_UPDATEDOCUMENTMETADATAS	El estado del documento en el repositorio documental no es coherente con el de portafirmas
DOCUMENTMANAGERIMPL_GENERATEAUTHORIZATIONDOCUMENT_01	Error al guardar el documento de autorización
DOCUMENTMANAGERIMPL_GENERATEAUTHORIZATIONDOCUMENT_02	Error al inicializar los metadatos del documento de autorización

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

DOCUMENTMANAGERIMPL_GENERATEAUTHORIZATIONDOCUMENT_03	Error al verificar los metadatos del documento de autorización
DOCUMENTMANAGERIMPL_GENERATEAUTHORIZATIONDOCUMENT_04	Error al crear el documento de autorización
DOCUMENTMANAGERIMPL_GENERATEAUTHORIZATIONDOCUMENT_05	Error al verificar los metadatos del documento de autorización
DOCUMENTMANAGERIMPL_UPDATESIGNERS	El firmante del documento en el repositorio documental no es coherente con el de portafirmas
DOCUMENTMANAGERIMPL_CREATEZIP	Error creando el zip para descarga de documentos
DOCUMENTMANAGERIMPL_SIGNDOCUMENT_01	Error al crear la firma
DOCUMENTMANAGERIMPL_SIGNDOCUMENT_02	No se ha podido firmar el documento por estar en uso. Por favor, intente firmarlo de nuevo
DOCUMENTMANAGERIMPL_SIGNDOCUMENT_03	Ha ocurrido un error al verificar y elevar la firma
UTILSPRIV_GENERATEHASH_01	Error al generar código SHA-1
UTILSPRIV_CREATE_IMAGE_FROM_DOCUMENT	Error al generar la imagen del documento
SETIADVISEMANAGER_NOTESERVICE_01	Error al generar el aviso
ADDNOTE_NOTESERVICE	Error al crear la nota
SENDADVICE_NOTESERVICE	Error al enviar el aviso
SENDADVICEFINALIZE_DOCUMENTSERVICE_01	Error al generar el aviso al firmante
GETPRODUCERORGANISM_DOCUMENTSERVICEIMPL_01	Error al recuperar el organismo productor
GETPRODUCERORGANISM_DOCUMENTSERVICEIMPL_02	Error al recuperar el arbol de organismos
DOCUMENTSERVICEIMPL_SIGNDOCUMENT_01	Error al firmar el documento porque ya esta en uso
DOCUMENTSERVICEIMPL_SIGNDOCUMENT_02	El proceso de firma ha superado la fecha de firma del documento. Por favor, inténtelo de nuevo

DOCUMENTSERVICEIMPL_SIGNDOCUMENT_03	No se puede actualizar el contenido porque ya está en uso
DOCUMENTSERVICEIMPL_SIGNDOCUMENT_04	No se puede cambiar el estado porque ya está en uso
ADDNOTE_NOTEMANAGER	Error al generar la nota
UTILSCORE_GENERATEHASH_01	Error al generar codigo SHA-1
UTILSCORE_OBTAINCLOG_01	Error al leer el Clob
UTILSCORE_OBTAINCLOG_02	Error al recuperar el Clob
IS_SIGNED	El documento no puede ser rechazado o revisado porque ya ha sido firmado
PREPAREDOCUMENTFILE_GENERATETEMPORALFILE	Error crear el fichero temporal
UTILSCORE_GENERATETEMPORALDOCUMENT_01	Error al generar el documento temporal
UTILSCORE_GENERATETEMPORALDOCUMENT_02	Error al cargar el documento temporal
DOCUMENTDAOIMPL_LOCKDOCUMENT	Error al buscar el documento a bloquear o desbloquear en el proceso de firmado
SAVE_PROPOSEDSIGNER	Error al guardar el firmante propuesto

4.3 Errores de conexión con servicios de PFI

USERINTERCEPTORPRIV_INTERCEPT_01	Conexión errónea con el servidor, es posible que su sesión haya caducado
USERINTERCEPTORPRIV_INTERCEPT_02	El servicio de Portafirmas no puede atender su petición, inténtelo más adelante o contacte con el administrador.

4.4 Errores de conexión con servicios externos

SERVICEINTERCEPTORPRIV_INTERCEPT_01	Conexión errónea con un servidor externo, es

	posible que su sesión haya caducado
SERVICEINTERCEPTORPRIV_INTERCEPT_02	El servicio externo no puede atender su petición, inténtelo más adelante o contacte con el administrador.

4.5 Errores en la ejecución de las tareas

QUORUMTASK_DOCUMENTNONEXISTING	El documento con el CSV especificado no existe en la base de datos de PFI
QUORUMTASK_QUORUMWRONG	El valor de quorum indicado es incorrecto. Debe ser mayor que 0 y menor o igual que el número de firmantes propuestos del documento
QUORUMTASK_DATEWRONG	La fecha límite de firma del documento ha expirado
QUORUMTASK_QUORUMDATENULL	No se ha introducido un valor de fecha límite ni un valor de quorum, es necesario introducir al menos uno de ellos
QUORUMTASK_NOMULTIFIRMA	El documento no es multifirma
QUORUMTASK_QUORUMORDATEEXIST	El documento no se va a actualizar puesto que ya tenía valores de quorum y/o fecha límite
QUORUMTASK_SIGNED	El documento ya ha sido firmado por alguno de los firmantes
SCDETASK_INITIALIZEDOCUMENTMETADATA	Error al inicializar los metadatos del documento de resumen de la firma con sello
SCDETASK_VERIFYDOCUMENTMETADATA	Error al verificar los metadatos del documento de resumen de la firma con sello
SCDETASK_CREATEDOCUMENT	Error al crear el documento de resumen de la firma con sello
SCDETASK_CREATERELATIONSWITHAUTHORIZATION	Error al crear la relación entre el documento de resumen y el de autorización
SCDEINTEGRATION_GETSTATUS	Error al comprobar el estado de la operación
SCDEINTEGRATION_SINGDOCS	Error al firmar con sello de órgano

SCDETASK_SIGNSUMMARYDOCUMENT	Error al firmar el documento de resumen
SCDETASK_GETSUMMARYDOCUMENTHASH	Error al recuperar el hash del documento de resumen
SCDETASK_CREATESIGNATURE	Error al crear la firma del documento de resumen en el repositorio documental
SCDETASK_CHANGESTATE	Error al cambiar el estado del documento de resumen

4.6 Errores de DWR (Direct Web Remoting)

DWR_NO_SESSION	No hay sesión
DWR_CREATE_SIGNER_FOOT	Error al crear los pies de firma
DWR_REPLACE_DOCUMENT	Error al reemplazar el tag por las firmas
DWR_LOAD_PAGE_IMAGE	Error al cargar las páginas
DWR_GET_DOCUMENT_CONTENT	Error al cargar el documento

4.7 Errores de firma

SIGNCONTROLLER_01	Error al cargar el contenido del documento
SIGNCONTROLLER_02	Error al cargar el documento
SIGNCONTROLLER_03	Se ha superado la fecha límite establecida para firmar el documento
SIGNCONTROLLER_04	La firma con sello es obligatoria
SIGNCONTROLLER_05	El documento está siendo firmado en estos momentos. Por favor, inténtelo de nuevo más tarde
SIGNCONTROLLER_06	El proceso de firma ha superado la fecha de firma del documento. Por favor, inténtelo de nuevo
SIGNCONTROLLER_07	Error en el proceso de firma.

5 Servicios web

Una vez desplegado en el correspondiente entorno, el sistema PFI publica los siguientes servicios, que podemos observar accediendo a “http://**[entorno:puerto]**/pfi_core/services”:

IActiveAuditorServicePortType	
• findAuditor • saveActiveAuditor • findAuditorList • findSigner • deleteActiveAuditor • findSignerList	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IActiveAuditorService WSDL : {http://ws.core.pfi.aragon.es/}IActiveAuditorService Target namespace: http://ws.core.pfi.aragon.es/

IAdviceServicePortType	
• sendAdvice	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IAdviceService WSDL : {http://ws.core.pfi.aragon.es/}IAdviceService Target namespace: http://ws.core.pfi.aragon.es/

IConfigurationServicePortType	
• updateConfiguration • getConfigurationList • getConfiguration • getAllConfigurations	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/ConfigurationWS WSDL : {http://ws.core.admin.commons.paea.aragon.es/}IConfigurationService Target namespace: http://ws.core.admin.commons.paea.aragon.es/

IDocumentIntegrationServicePortType	

• sendDocumentumDocument • updateQuorumAndLimitDateDocument	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IDocumentIntegrationService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentIntegrationService Target namespace: http://ws.core.pfi.aragon.es/

IDocumentMetadataServicePortType	
• findDocumentMetadata • deleteDocumentMetadata • findDocumentMetadataList • saveDocumentMetadata	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IDocumentMetadataService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentMetadataService Target namespace: http://ws.core.pfi.aragon.es/

IDocumentServicePortType	
• reviseOrRejectDocument • deleteDocument • getProducerOrganism • findDocumentFolderList • findMetadataValue • findDocumentList1 • findDocumentList • signDocument • sendDocument • generateAuthorizationDocument • saveDocument • findDocument • saveMetadata • getDocumentState • downloadZip • findDocumentTray	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IDocumentService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentService Target namespace: http://ws.core.pfi.aragon.es/

• countFileList	

IDocumentumDocumentServicePortType	
• initializeDocumentum • findDocumentumMetadata • findDocumentumDocumentTypeList • findDocumentumDocumentMetadataList • saveDocumentumMetadata	Endpoint address: https:// **[entorno:puerto]**/pfi_core/services/IDocumentumDocumentService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentumDocumentService Target namespace: http://ws.core.pfi.aragon.es/

IHistoryServicePortType	
• saveLastSigners • updateShowDocument • deleteHistory • findHistoryList • findHistory • saveHistory	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IHistoryService WSDL : {http://ws.core.pfi.aragon.es/}IHistoryService Target namespace: http://ws.core.pfi.aragon.es/

ISignerServicePortType	
• findSignerByNif • saveProposedSigner • deleteProposedSigner • findProposedSignerList • findQuickSignerList • findSignerList	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/ISignerService WSDL : {http://ws.core.pfi.aragon.es/}ISignerService Target namespace: http://ws.core.pfi.aragon.es/

• findProposedSigner	

5.1 IActiveAuditorService

A continuación se detalla la interfaz utilizada para el tratamiento de usuarios con rol de revisor (auditores). Este interfaz aporta los siguientes servicios:

IActiveAuditorServicePortType	
• findAuditor • saveActiveAuditor • findAuditorList • findSigner • deleteActiveAuditor • findSignerList	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IActiveAuditorService WSDL : {http://ws.core.pfi.aragon.es/}IActiveAuditorService Target namespace: http://ws.core.pfi.aragon.es/

5.1.1 Descripción funcional básica

IActiveAuditorServicePortType dispone de métodos que permiten al usuario de la aplicación realizar la gestión de usuarios con rol de revisor; búsqueda de revisores, búsqueda de firmantes asociados a un revisor, y modificar las relaciones entre ambos.

5.1.2 Métodos de integración con otras aplicaciones

No existentes

5.1.3 Otros métodos

findAuditor	Busca un usuario con rol revisor para unos parámetros de búsqueda dados
saveActiveAuditor	Guarda la asociación entre un usuario con rol de revisor y un firmante

findAuditorList	Busca la lista de usuarios que pueden revisar (auditores)
findSigner	Busca un usuario firmante para unos parámetros de búsqueda dados
deleteActiveAuditor	Elimina la asociación entre un usuario con rol de revisor y un firmante
findSignerList	Obtiene la lista de firmantes a los que puede revisar el usuario actual

5.2 IAdviceService

A continuación se detalla la interfaz utilizada para el envío de avisos cuando se produce un cambio en algún documento. Este interfaz aporta los siguientes servicios:

IAdviceServicePortType	
• sendAdvice	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IAdviceService WSDL : {http://ws.core.pfi.aragon.es/}IAdviceService Target namespace: http://ws.core.pfi.aragon.es/

5.2.1 Descripción funcional básica

IAdviceServicePortType dispone de un único método encargado de mandar avisos mediante SGA para informar de cambios en algún documento.

5.2.2 Métodos de integración con otras aplicaciones

No existentes.

5.2.3 Otros métodos

sendAdvice	Manda un aviso mediante SGA para informar de algún cambio en un documento

5.3 IConfigurationService

A continuación se detalla la interfaz utilizada para gestionar la configuración de la aplicación. Este interfaz aporta los siguientes servicios:

IConfigurationServicePortType	
• updateConfiguration • getConfigurationList • getConfiguration • getAllConfigurations	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/ConfigurationWS WSDL : {http://ws.core.admin.commons.paea.aragon.es/}IConfigurationService Target namespace: http://ws.core.admin.commons.paea.aragon.es/

5.3.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *IConfigurationServicePortType* para la gestión de los diferentes parámetros de configuración que almacenamos en la tabla PFI_CONFIGURATION de Portafirmas.

A través de sus métodos podremos recuperar y modificar cualquier parámetro configurable.

5.3.2 Métodos de integración con otras aplicaciones

No existentes.

5.3.3 Otros métodos

updateConfiguration	Actualiza el valor de un registro de configuración, en función de los valores que contiene el parámetro de tipo *ParamUpdateConfiguration* que pasamos como parámetro de entrada.
getConfigurationList	Obtiene una lista de configuraciones de un componente y tipo determinado que pasamos como parámetro de entrada.
getConfiguration	Obtiene el valor de un registro de configuración concreto, pasando como parametros de entrada el componente, tipo y propiedad que será modificada.
getAllConfigurations	Obtiene todas configuraciones de la aplicación.

5.4 IDocumentIntegrationService

A continuación se detalla la interfaz utilizada para enviar documentos a Portafirmas. Este interfaz aporta los siguientes servicios:

IDocumentIntegrationServicePortType	
• sendDocumentumDocument • updateQuorumAndLimitDateDocument	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IDocumentIntegrationService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentIntegrationService Target namespace: http://ws.core.pfi.aragon.es/

5.4.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *IDocumentIntegrationServicePortType* para el envío de documentos a Portafirmas. Además, permite actualizar el quorum (número mínimo de firmantes que deben firmar un documento) y fecha límite de preparación de un documento multifirma.

5.4.2 Métodos de integración con otras aplicaciones

5.4.2.1 *sendDocumentumDocument*

5.4.2.1.1 Descripción

ResultSendDocument *sendDocumentumDocument(*ParamSendDocument paramSendDocument)throws InternalCoreException;

Cuando una aplicación que integra Portafirmas almacena un documento en Documentum y desea que quede guardado en Portafirmas, invoca este servicio web para que posteriormente Portafirmas pueda hacer uso de dicho documento. El servicio web lanzara una InternalCoreException en el caso de que no se puedan realizar algunas de las siguientes acciones:

- Establecer conexión con Documentum
- Guardado del documento en PFI
- Guardado del los metadatas del documento en PFI

5.4.2.1.2 Parámetros petición

Pasamos como parámetro de entrada un objeto *ParamSendDocument* que encapsula el csv identificador del documento.

Ejemplo de invocación:

```
ParamSendDocument paramSendDocument=new ParamSendDocument();
paramSendDocument.setCsv("CSVDA6FLBW3PNAX01PFI");
ResultSendDocument resultSendDocument = pfiClient.sendDocumentumDocument(paramSendDocument);

if(ReturnCode.OK.equals(resultSendDocument.getReturnCode()){
log.info("El documento se ha guardado correctamente en PFI");
}else{
log.error(result.getErrorCode()+":"+result.getErrorMessage());
}
```

5.4.2.1.3 Parámetros de respuesta

El método devolverá un objeto del tipo *ResultSendDocument.* En la siguiente tabla se resume el tipo de dato y la información que contiene dicho objeto:

Nombre	Tipo	Descripción
errorCode	String	Código de error del servicio.
returnCode	String	Indica si la llamada ha finalizado correctamente (OK) o en error (ERROR)
errorMessage	String	Mensaje de error del servicio.
exception	String	Información adicional de la excepción producida

5.4.2.1.4 Errores / Excepciones

En la siguiente tabla se muestran los códigos de error y las descripciones de las InternalCoreException que puede lanzar el servicio web.

Código de error	Descripción
ADD_METADATA	Error al guardar metadato
CCSV_ERROR_GETMETADATA	Error leyendo metadatos de Documentum

5.4.2.2 *updateQuorumAndLimitDateDocument*

5.4.2.2.1 Descripción

ResultUpdateQuorumAndLimitDate *updateQuorumAndLimitDateDocument*(ParamUpdateQuorumAndLimitDate paramUpdateQuorumAndDate) throws InternalCoreException;

Este servicio es utilizado desde aplicaciones externas para actualizar el campo *quorum* (número mínimo de firmantes que deben firmar un documento) y fecha límite de preparación en los documentos multifirma. También se puede actualizar si se permiten más firmas que las mínimas obligatorias o no con el parámetro *quorumOnly*. Para ello, en el objeto de entrada al servicio *ParamUpdateQuorumAndLimitDate* se encapsulará el csv del documento que pretende actualizarse, el *quorum* y la fecha que definen el número de firmantes y fecha límite de firma del documento en cuestión respectivamente). El objeto del servicio será actualizarlos con los valores encapsulados en el objeto de entrada, siempre y cuando se cumplan los siguientes requisitos:

- El documento con el csv indicado debe existir en la base de datos de PFI.
- El documento que pretende actualizarse es multifirma.
- Al menos uno de los dos parámetros (quorum, date) será no nulo para que la llamada al servicio tenga sentido.
- En caso en que el parámetro de entrada quorum sea nulo se tomará, para la actualización del documento, el valor por defecto igual al número de firmantes propuestos del documento.
- En caso en que el parámetro de entrada quorum sea no nulo, se validará comprobando si el valor indicado es mayor que cero y menor o igual al número de firmantes propuestos del documento, en caso contrario, el resultado del servicio devolverá el código y mensaje de error correspondiente.
- En caso de que la fecha indicada en el parámetro de entrada sea nula, se tomará por defecto para la actualización del documento el valor correspondiente a la fecha actual más un número de días configurable en base de datos.
- En caso de que la fecha indicada en el parámetro de entrada sea no nula, se validará si no ha expirado. En caso contrario el resultado del servicio devolverá el código y mensaje de error correspondiente.
- En el caso de que el valor de *quorumOnly* sea nulo se tomará por defecto el valor falso, indicando que se permiten más firmas que las mínimas obligatorias.
- Si alguno de los firmantes propuestos del documento ya había firmado dicho documento, el servicio devolverá un código y mensaje de error que indique en esta caso que ya no es posible realizar la actualización del documento.

5.4.2.2.2 Parámetros petición

Pasamos como parámetro de entrada un objeto *ParamUpdateQuorumAndLimitDate* que encapsula el csv identificador del documento, el quorum de firmantes y la fecha límite de firma.

Ejemplo de invocación:

```
ParamUpdateQuorumAndLimitDate paramCheckQuorumAndDate = new
ParamUpdateQuorumAndLimitDate();
//indicamos el csv del documento que pretende actualizarse
paramCheckQuorumAndDate.setCsv("CSVP55RYQS7R9AF01PFI");
paramCheckQuorumAndDate.setQuorum(2); //quorum que queremos actualizar
paramCheckQuorumAndDate.setQuorumOnly(Boolean.TRUE); //sólo se permiten 2 firmas
//queremos de fecha límite por ejemplo el día de hoy más una semana
Calendar cal = Calendar.getInstance();
cal.setTime(new Date());
Integer incFecha=7;
cal.add(Calendar.DATE, incFecha);
paramCheckQuorumAndDate.setDate(cal.getTime());
ResultUpdateQuorumAndLimitDate resultCheckQuorumAndDate =
iDocumentIntegrationService.updateQuorumAndLimitDateDocument(paramCheckQuorumAndDate);}
```

5.4.2.2.3 Parámetros de respuesta

El método devolverá un objeto del tipo *ResultUpdateQuorumAndLimitDate*. En la siguiente tabla se resume el tipo de dato y la información que contiene dicho objeto:

Nombre	Tipo	Descripción
errorCode	String	Código de error del servicio.
returnCode	String	Indica si la llamada ha finalizado correctamente (OK) o en error (ERROR)
errorMessage	String	Mensaje de error del servicio.

5.4.2.2.4 Errores / Excepciones

En la siguiente tabla se muestran los códigos de error y las descripciones de las InternalCoreException que puede lanzar el servicio web.

Código de error	Descripción
QUORUMTASK_DOCUMENTNONEXISTING	El documento con el CSV especificado no existe en la base de datos de PFI
QUORUMTASK_QUORUMWRONG	El valor de quorum indicado es incorrecto. Debe ser mayor que 0 y menor o igual que el número de firmantes

	propuestos del documento
QUORUMTASK_DATEWRONG	La fecha límite de firma del documento ha expirado
QUORUMTASK_QUORUMDATE NULL	No se ha introducido un valor de fecha límite ni un valor de quorum, es necesario introducir al menos uno de ellos
QUORUMTASK_NOMULTIFIRMA	El documento no es multifirma
QUORUMTASK_SIGNED	El documento ya ha sido firmado por alguno de los firmantes

5.4.3 Otros métodos

No existentes.

5.5 IDocumentMetadataService

A continuación se detalla la interfaz utilizada para la gestión de metadatos de un documento. Este interfaz aporta los siguientes servicios:

IDocumentMetadataServicePortType	
• findDocumentMetadata • deleteDocumentMetadata • findDocumentMetadataList • saveDocumentMetadata	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IDocumentMetadataService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentMetadataService Target namespace: http://ws.core.pfi.aragon.es/

5.5.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *IDocumentMetadataServicePortType* para el envío/recuperación/edición de metadatos de documentos en Portafirmas.

5.5.2 Métodos de integración con otras aplicaciones

No existentes.

5.5.3 Otros métodos

findDocumentMetadata	Busca un metadato de un documento en PFI. Se le pasa

	un objeto de tipo *DocumentMetadataSearch* como parámetro de entrada para filtrar la búsqueda
DeleteDocumentMetadata (Obsoleto)	Elimina un metadato de un documento en PFI. Se le pasa un objeto de tipo *DocumentMetadataSearch* como parámetro de entrada para filtrar la búsqueda.
FindDocumentMetadataList (Obsoleto)	Busca lista de metadatas de un documento en PFI. Se le pasa un objeto de tipo *DocumentMetadataSearch* como parámetro de entrada para filtrar la búsqueda
SaveDocumentMetadata (Obsoleto)	Guarda un metadato de un documento en PFI. Como parámetro de entrada pasamos un objeto *DocumentMetadata* donde irán encapsulados los valores que queremos almacenar.

5.6 IDocumentService

A continuación se detalla la interfaz utilizada para la gestión de documentos en Portafirmas. Este interfaz aporta los siguientes servicios:

IDocumentServicePortType	
• reviseOrRejectDocument • deleteDocument • getProducerOrganism • findDocumentFolderList • findMetadataValue • findDocumentList1 • findDocumentList • signDocument • sendDocument • generateAuthorizationDocument • saveDocument • findDocument • saveMetadata	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IDocumentService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentService Target namespace: http://ws.core.pfi.aragon.es/

• getDocumentState • downloadZip • findDocumentTray • countFileList	

5.6.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *IDocumentServicePortType* para la gestión de documentos, que nos permite a través de sus métodos acciones como firmar, revisar, rechazar o consultar el estado de un documento. También nos permite recuperar documentos y metadatos de los mismos.

5.6.2 Métodos de integración con otras aplicaciones

5.6.2.1 *getDocumentState*

5.6.2.1.1 Descripción

Integer *getDocumentState*(ParamGetDocumentState paramGetDocumentState) throws InternalCoreException;

Este servicio es utilizado desde aplicaciones externas para obtener el estado actual de un documento. Para ello, pasaremos un objeto del tipo ParamGetDocumentState como parámetro de entrada, donde se encapsulará el csv del documento del se que pretende obtener su estado.

5.6.2.1.2 Parámetros petición

Pasamos como parámetro de entrada un objeto *ParamGetDocumentState* que encapsula el csv identificador del documento.

Ejemplo de invocación:

```
ParamGetDocumentState paramGetDocumentState =  new ParamGetDocumentState();
//indicamos el csv del documento que pretende actualizarse
paramGetDocumentState.setCsv("CSVP55RYQS7R9AF01PFI");
Integer documentState =
IDocumentService.getDocumentState(paramGetDocumentState);}
```

5.6.2.1.3 Parámetros de respuesta

Como respuesta obtenemos un entero que se corresponde con un determinado estado de un documento:

Valor entero	Descripción estado
0	Borrador
1	Pendiente de firma
2	Firmado
3	Firmando asíncronamente
4	Eliminado
5	Incompleto
6	Firmando en paralelo

5.6.2.1.4 Errores / Excepciones

No existentes.

5.6.3 Otros métodos

reviseOrRejectDocument	Método que revisa o rechaza un documento añadiendo la nota correspondiente para el mismo
deleteDocument	Método que elimina un documento
getProducerOrganism	Método que obtiene el organismo productor a partir del método getProcedureAttributes de PAU
findDocumentFolderList	Método que devuelve un listado de bloques o carpetas a partir del objeto de búsqueda del documento
findMetadataValue	Busca el valor de un metadato para un documento
findDocumentList1	Obtiene una lista de documentos, pasando como parámetro de entrada un objeto de tipo *DocumentSearch*
findDocumentList	Obtiene una lista de documentos a partir de un listado de objetos contenedores de parámetros de búsqueda
signDocument	Método utilizado para firmar un documento
sendDocument	Método para el envío de un documento a Documentum

generateAuthorizationDocument	Método que genera el documento de autorización en los casos de firma con sello
saveDocument	Método que guarda un documento en base de datos
findDocument	Obtiene un documento, pasando como parámetro de entrada un objeto de tipo *DocumentSearch*
saveMetadata	Método que guarda y actualiza los metadatos de un documento
downloadZip	Método que permite la descarga de un archivo comprimido
findDocumentTray	Método que permite la búsqueda de un listado de documentos para las bandejas
countFileList	Método que cuenta el conjunto de elementos tipo File que forman parte de una lista

5.7 IDocumentumDocumentService

A continuación se detalla la interfaz utilizada para la obtención y actualización de información relativa a documentos en Documentum. Este interfaz aporta los siguientes servicios:

IDocumentumDocumentServicePortType	
• initializeDocumentum • findDocumentumMetadata • findDocumentumDocumentTypeList • findDocumentumDocumentMetadataList • saveDocumentumMetadata	Endpoint address: https:// **[entorno:puerto]**/pfi_core/services/IDocumentumDocumentService WSDL : {http://ws.core.pfi.aragon.es/}IDocumentumDocumentService Target namespace: http://ws.core.pfi.aragon.es/

5.7.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *IDocumentumDocumentServicePortType* que nos permite tanto almacenar/extraer metadatos de un documento en Documentum, como obtener los distintos tipos de documentos existentes en Documentum.

5.7.2 Métodos de integración con otras aplicaciones

No existentes

5.7.3 Otros métodos

initializeDocumentum	Método para la inicialización de un documento en Documentum.
findDocumentumMetadata	Busca metadatos para un documento a través de la integración con CCSV
findDocumentumDocumentTypeList	Método que recupera la lista de tipos de documentos de Documentum
findDocumentumDocumentMetadataList	Busca la lista de metadatos para un documento a través de la integración con CCSV
saveDocumentumMetadata	Almacena los metadatos del documento en un objeto serializado en BD. Si ya existe dicho objeto, lo actualiza

5.8 IHistoryService

A continuación se detalla la interfaz utilizada para la obtención y actualización de los últimos firmantes propuestos por un preparador. Este interfaz aporta los siguientes servicios:

IHistoryServicePortType	
• saveLastSigners • updateShowDocument • deleteHistory • findHistoryList • findHistory • saveHistory	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/IHistoryService WSDL : {http://ws.core.pfi.aragon.es/}IHistoryService Target namespace: http://ws.core.pfi.aragon.es/

5.8.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *IHistoryServicePortType* que a través de sus métodos, nos permitirá recuperar o actualizar información de la tabla `PFI_HISTORY` de Portafirmas, donde almacenamos

información sobre los últimos firmantes propuestos por un preparador.

5.8.2 Métodos de integración con otras aplicaciones

No existentes.

5.8.3 Otros métodos

saveLastSigners	Método encargado de la actualizacion de los últimos firmantes propuestos por un preparador
updateShowDocument	Función encargada de grabar si se debe mostrar el documento incrustado en el detalle o no
deleteHistory	Elimina un registro de la tabla PFI_HISTORY borrando así los últimos firmantes propuestos seleccionados por un preparador
findHistoryList	Busca una lista de registros de la tabla PFI_HISTORY con los últimos firmantes propuestos seleccionados por un preparador
findHistory	Busca un registro de la tabla PFI_HISTORY con los últimos firmantes propuestos seleccionados por un preparador
saveHistory	Almacena un registro en la tabla PFI_HISTORY para guardar los últimos firmantes propuestos seleccionados por un preparador

5.9 ISignerService

A continuación se detalla la interfaz utilizada para el tratamiento de firmantes de un documento. Este interfaz aporta los siguientes servicios:

ISignerServicePortType	
• findSignerByNif • saveProposedSigner • deleteProposedSigner • findProposedSignerList • findQuickSignerList	Endpoint address: https://**[entorno:puerto]**/pfi_core/services/ISignerService WSDL : {http://ws.core.pfi.aragon.es/}ISignerService Target namespace: http://ws.core.pfi.aragon.es/

• findSignerList • findProposedSigner	

5.9.1 Descripción funcional básica

El agente de acceso ofrece un interfaz *ISignerServicePortType* que a través de sus métodos, nos permitirá buscar un firmante o lista de firmantes para un documento, así como el guardado o borrado de los mismos.

5.9.2 Métodos de integración con otras aplicaciones

No existentes.

5.9.3 Otros métodos

findSignerByNif	Busca un firmante atendiendo a unos parámetros de búsqueda establecidos.
saveProposedSigner	Guarda un firmante propuesto para un documento
deleteProposedSigner	Elimina un firmante propuesto para un documento
findProposedSignerList	Busca una lista de firmantes propuestos para un documento atendiendo a unos criterios de búsqueda
findQuickSignerList	Busca una lista de firmantes compuesta por: - el propio usuario - su jefe - el jefe de su jefe - los cinco últimos firmantes en su histórico
findSignerList	Busca una lista de firmantes atendiendo a unos parámetros de búsqueda
findProposedSigner	Busca un firmante propuesto para un documento atendiendo a unos criterios de búsqueda

6 Tecnología específica

6.1 Integración mediante CXF

6.1.1 Configuración de servicios para CXF:

Mediante sintaxis xml podemos definir fácilmente la configuración del cliente CXF que acceda a los servicios de PFI.
Inicialmente deberemos definir la referencia a la interfaz del servicio, que se encuentra dentro del cliente suministrado, así como a la dirección que apunta al wsdl levantado.

IDocumentIntegrationService:

```
<simple:client id="pfiClient"
        serviceClass="es.aragon.pfi.core.ws.IDocumentIntegrationService"
        address="http://[entorno:puerto]/pfi_core/services/IDocumentIntegrationServiceWS"
        serviceName="s:IDocumentIntegrationService" xmlns:s="http://ws.core.pfi.aragon.es/ "
        endpointName="s:IDocumentIntegrationServiceWS">
        <simple:outInterceptors>
                <ref bean="authorizeInterceptor" />
        </simple:outInterceptors>
</simple:client>
```

Como se puede observar en la definición XML del cliente CXF, se deberán configurar las urls donde se ubica el servicio *IDocumentIntegrationServiceWS*, para ello habrá que sustituir **entorno** y **puerto** por los valores correctos.

Las aplicaciones que se integran con PFI deben proporcionar en todas
las peticiones el código de aplicación. Para ello, utilizando los denominados interceptores, es
posible rellenar este valor automáticamente en todas las peticiones. Como se ve en la definición del cliente se ha definido un interceptor de salida que entra en funcionamiento en el momento de construir la petición SOAP y que inserta el valor del código de aplicación en los parámetros de todos los métodos de PFI invocados.

```
<simple:outInterceptors>
        <ref bean="authorizeInterceptor" />
</simple:outInterceptors>
```

La definición al bean authotizeInterceptor que se incluye a continuación, permite indicarle el código de la aplicación, de esta manera el interceptor se encarga de establecer dicho valor en todos los parámetros de las llamadas a los servicios PFI

```
<bean id="authorizeInterceptor" class="es.aragon.pfi.core.util.interceptors.AuthorizeOutInterceptor">
        <property name="applicationId" value="[código aplicación llamante]"/>
</bean>
```

La clase con el interceptor de salida *AuthorizeOutInterceptor* se proporciona junto con el cliente CXF de integración con PFI.

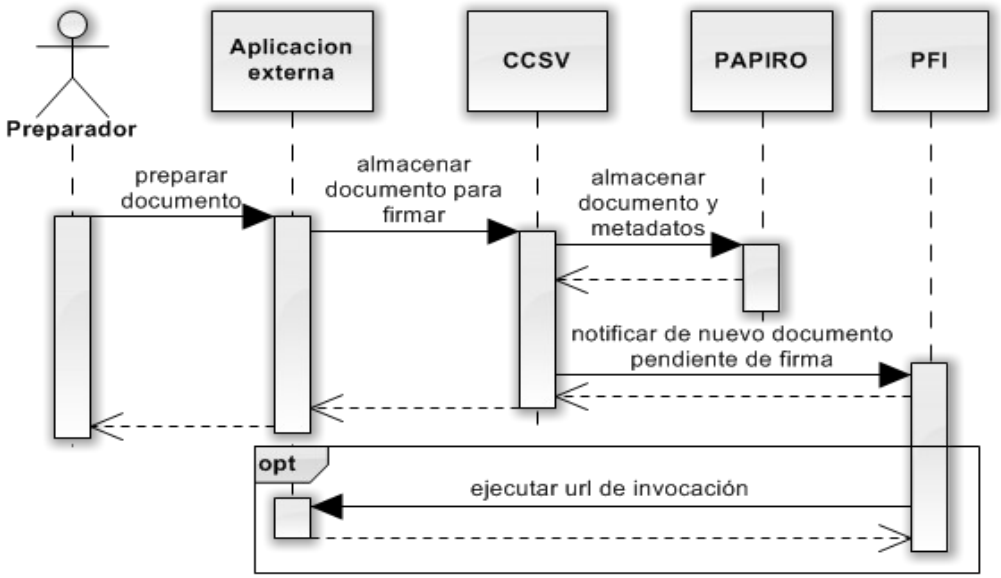
7 Servicios en SOA

La aplicación expone sus servicios web en SOA.

8 Casos de uso de integración

PFI ofrece tres casos de uso para la integración de aplicaciones externas. Por un lado se puede dejar un documento en la bandeja de firma de un usuario en particular, por otro lado se puede conocer el estado de ese documento, y además, se puede actualizar el quorum y fecha limite de firma del mismo.

El diagrama de la secuencia típica sería el siguiente:



Como se observa, en el momento de almacenar en el repositorio documental sería CCSV, que actúa como intermediario con dicho repositorio, el encargado de comunicar a PFI que un documento está preparado para firmar. Para ello, la aplicación externa necesita introducir los siguientes metadatos del documento en el repositorio:

- "Firmante propuesto": entidad agente que identifica el usuario destinatario del documento en PFI.
- Estado de elaboración: el estado del documento debe ser pendiente de firma ("P").

Además, y opcionalmente, las aplicaciones pueden definir un *servlet* al que invoque PFI cuando cambie el estado del documento, aunque recomendamos consultar directamente al repositorio para

EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

garantizar el último estado real. Dicho *servlet* se invocará con tres parámetros mediante *querystring*:

- idApp='PFI'.
- CSV del documento.
- 'estado' en que ha dejado el documento, borrador (1) u original (0).

Para que PFI pueda invocar a dicha url, debe estar asociada a la aplicación a través de la plataforma PAU, en el atributo *responseUrl* correspondiente a la aplicación.

A continuación se detallarán con ejemplos de código estos tres casos de uso de la integración.

8.1.1 Enviar un documento al portafirmas

Como se ha dicho previamente, para dejar un documento como pendiente de firma y que le aparezca al destinatario en su bandeja de firma es necesario modificar ciertos metadatos del documento.

Antes de nada, definiremos algunas constantes de metadatos necesarias:

```
	// constantes de metadatos
	public static final String PAEGA_FIRMANTE_PROPUESTO = "firmante_propuesto";
	public static final String PAEGA_AGENTE_NTI_TIPO_ENTIDAD =
"agente_nti_tipo_entidad";
	public static final String PAEGA_AGENTE_NTI_TIPO_CATEGORIA =
"agente_nti_categoria";
	public static final String PAEGA_AGENTE_IDENTIFICACION_TIPO_IDENTIFICACION =
"agente_ident_tipo_identif";
	public static final String PAEGA_AGENTE_IDENTIFICACION_IDENTIFICACION =
"agente_ident_identificacion";
	public static final String PAEGA_AGENTE_DESCRIPCION_NOMBRE = "agente_desc_nombre";
	public static final String PAEGA_FIRMANTE_PROPUESTO = "firmante_propuesto";
	public static final String PAEGA_FIRMANTE_PROPUESTO_TEX_DELEGA =
"firmante_prop_texto_delega";
	public static final String AGENTE = "Agente";
	public static final String FUNCIONARIO = "Empleado Público";
	public static final String NIF = "N";
```

En primer lugar, creamos el metadato de firmante propuesto para añadir al documento. Dicho firmante debe estar dado de alta en SIU para poder acceder a su bandeja de firma:

```
	// inicializamos el metadato
```

```
	ParamInitializeStructuredMetadata paramInitializeStructuredMetadata = new
ParamInitializeStructuredMetadata(PAEGA_FIRMANTE_PROPUESTO);
	ResultInitializeStructuredMetadata resultInitializeStructuredMetadata;
	try {
	 resultInitializeStructuredMetadata =
documentumWS.initializeStructuredMetadata(paramInitializeStructuredMetadata);
	} catch (Exception e) {
		...
	}

	if (!resultInitializeStructuredMetadata.getReturnCode().equals(ReturnCode.OK)) {
		...
	}

	// rellenamos los valores del firmante propuesto
	es.aragon.ccsv.core.ws.data.documentum.Document firmantePropuesto =
resultInitializeStructuredMetadata.getDocument();
	HashMap<String,Object> firmantePropuestoMetadata = firmantePropuesto.getMetadata();
	firmantePropuestoMetadata.put(PAEGA_AGENTE_NTI_TIPO_ENTIDAD, AGENTE);
	firmantePropuestoMetadata.put(PAEGA_AGENTE_NTI_TIPO_CATEGORIA, FUNCIONARIO);
	firmantePropuestoMetadata.put(PAEGA_AGENTE_IDENTIFICACION_TIPO_IDENTIFICACION,
NIF);
	firmantePropuestoMetadata.put(PAEGA_AGENTE_IDENTIFICACION_IDENTIFICACION,
signerNif);
	firmantePropuestoMetadata.put(PAEGA_AGENTE_DESCRIPCION_NOMBRE, signerName + " " +
signerSurname1 + " " + signerSurname2);
	// en caso de ser necesario
	firmantePropuestoMetadata.put(PAEGA_FIRMANTE_PROPUESTO_TEX_DELEGA, "Texto de
delegacion");
	firmantePropuesto.setMetadata(firmantePropuestoMetadata);
```

Después recuperamos los metadatos del documento en cuestión, para actualizar el firmante propuesto:

```
	//Recuperamos el documento de papiro mediante el CSV
	Document docPapiro = getDocumentData(document.getCsv());
	//Actualizamos el firmante propuesto en los metadatos del documento recuperado
	docPapiro.getMetadata().put(PAEGA_FIRMANTE_PROPUESTO, firmantePropuesto);
	// Actualizamos metadatos en Documentum
	ParamUpdateDocumentMetadata paramUpdateDocumentMetadata = new
```


EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

```
ParamUpdateDocumentMetadata();
	paramUpdateDocumentMetadata.setDocument(docPapiro);
	ResultUpdateDocumentMetadata resultUpdateDocumentMetadata;
	LOG.info("Añadimos el firmante al documento con csv: " + document.getCsv());
	try {
		resultUpdateDocumentMetadata =
	documentumWS.updateDocumentMetadata(paramUpdateDocumentMetadata);
	} catch (Exception e) {
		...
	}

	if (!resultUpdateDocumentMetadata.getReturnCode().equals(ReturnCode.OK)) {
		...
	}
```

Por último, actualizamos el estado del documento para pasarlo a pendiente de firma:

```
	// actualizamos el estado en Documentum
	ParamChangeState paramChangeState = new ParamChangeState();
	paramChangeState.setApplicationId(appId);
	paramChangeState.setCsv(document.getCsv());
	paramChangeState.setState(state);
	ResultChangeState resultChangeState = new ResultChangeState();

	try {
		resultChangeState = documentumWS.changeState(paramChangeState);
	} catch (Exception e) {
		LOG.error("Error al cambiar el estado del documento",e);
		throw new InternalCoreException(SCECoreConstants.DOCUMENTUM_CCSV_ERROR, e);
	}

	if (!resultChangeState.getReturnCode().equals(ReturnCode.OK)) {
		...
	}
```

8.1.2 Consulta de cambio de estado de un documento

Mediante el caso de uso de consulta del estado del documento PFI notifica a la aplicación que crea un documento del cambio del estado en ese documento, debido principalmente a que haya sido firmado.

Para comunicar dicho cambio de estado, PFI invoca a un servlet cuya url debe estar asociada a la aplicación origen a través del atributo *responseUrl* dentro PAU.

Un ejemplo de implementación de dicho servlet podría ser el siguiente:

```
      public class StateDocumentPFIImpl extends HttpServlet {

      private static final long serialVersionUID = 5954532668702585433L;
      private static final Logger LOG = CoreConfig.getLog(StateDocumentPFIImpl.class);
      private static final String INVALID_PARAMETERS = "Alguno de los parámetros
necesarios está vacío";
      private static final String INVALID_STATE = "Estado del documento no es válido";
      private static final String UPDATESTATE_METHOD = "updateStateByPFI";
      private static final String APPID_PARAM = "appId";
      private static final String CSV_PARAM = "csv";
      private static final String STATE_PARAM = "state";

      private String appId;
      private IInfoService infoServicesPAUWS;
      private IDocumentManager documentManager;
      private ICCSVDocument ccsvManager;

      public void init() {
WebApplicationContext applicationContext =ContextLoader.getCurrentWebApplicationContext();
      appId = "SCE";
      infoServicesPAUWS = (IInfoService) applicationContext.getBean("infoServicesPAUWS");
      documentManager = (IDocumentManager) applicationContext.getBean("documentManager");
      ccsvManager = (ICCSVDocument) applicationContext.getBean("ccsvManager");
      }

      public void doGet(HttpServletRequest request, HttpServletResponse response)
                  throws ServletException, IOException {
            String applicationId = (String) request.getParameter(APPID_PARAM);
            String csv = (String) request.getParameter(CSV_PARAM);
            String state = (String) request.getParameter(STATE_PARAM);
```

```
            String ip = request.getRemoteAddr();

            try {
                  validateParameters(applicationId, csv, state, ip);
                  authorizeRequest(applicationId, ip, UPDATESTATE_METHOD);
                  updateStateByPFI(csv, state);
            } catch (InternalCoreException e) {
                  throw new ServletException(e.getMsg(), e);
            }
            response.setStatus(HttpServletResponse.SC_OK);
      }

      private void validateParameters(String applicationId, String csv,
                  String state, String ip) throws ServletException {
            if (applicationId == null || applicationId.equals("")) {
                  throw new ServletException(INVALID_PARAMETERS);
            }
            if (csv == null || csv.equals("")) {
                  throw new ServletException(INVALID_PARAMETERS);
            }
            if (state == null || state.equals("")) {
                  throw new ServletException(INVALID_PARAMETERS);
            }
            if (ip == null || ip.equals("")) {
                  throw new ServletException(INVALID_PARAMETERS);
            }
      }

      private void authorizeRequest(String remoteApplicationId, String ip,
                  String methodName) throws InternalCoreException {
            LOG.info("BEGIN authorizeRequest " + remoteApplicationId + "-" + appId
                        + "-" + ip);
ResultAuthorizeApplication resultAuthorizeApplication = new ResultAuthorizeApplication();
                  try {
                  // Obtenemos el código del método
                  String methodCode = "";
```

```
ParamMethodApplicationList paramMethodApplicationList = new ParamMethodApplicationList();
            paramMethodApplicationList.setApplicationAlias(appId);
            paramMethodApplicationList.setApplicationId(appId);

ResultMethodApplicationList resultMethodApplicationList =
infoServicesPAUWS.methodApplicationList(paramMethodApplicationList);
es.aragon.pau.core.ws.data.Method[] methods =
resultMethodApplicationList.getMethods();
                  for (es.aragon.pau.core.ws.data.Method m : methods) {
                        if (m.getName().equals(methodName)) {
                              methodCode = m.getMethodCode();
                              break;
                        }
                  }

                  // Solicitamos autorización
ParamAuthorizeApplication paramAuthorizeApplication = new ParamAuthorizeApplication();
                  paramAuthorizeApplication.setApplicationId(appId);
                  paramAuthorizeApplication.setApplicationAliasA(remoteApplicationId);
                  paramAuthorizeApplication.setApplicationAliasB(appId);
                  paramAuthorizeApplication.setIp(ip);
                  paramAuthorizeApplication.setMethodCode(methodCode);
                  resultAuthorizeApplication =
infoServicesPAUWS.authorizeApplication(paramAuthorizeApplication);
                  LOG.info("Servicio PFI-SCE: Aplicación Llamante" +
remoteApplicationId + " IP:" + ip);
            } catch (Exception e) {
                  throw new
InternalCoreException(SCECoreConstants.ERROR_APPLICATION_AUTHORIZATION, e);
            }
            // Si tiene errores pongo el mensaje de autorización
            if (resultAuthorizeApplication.isErrors()) {
                  throw new InternalCoreException(
                              SCECoreConstants.ERROR_APPLICATION_AUTHORIZATION,
            "La IP " + ip + " no está autorizada");
            }
      LOG.info("END authorizeRequest " + remoteApplicationId + "-" + appId + "-" + ip);
      }
      private void updateStateByPFI(String csv, String ccsvState)
```


EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

```
                    throws InternalCoreException {
            LOG.info("BEGIN updateStateByPFI " + csv);
            if (!DocumentumConstants.Estado.BORRADOR.equals(ccsvState)
                        && !DocumentumConstants.Estado.ORIGINAL.equals(ccsvState)) {
      // No ha cambiado ni a borrador ni a firmado (ORIGINAL según Portafirmas)
      throw new InternalCoreException(SCECoreConstants.NOT_VALID_VALUE, INVALID_STATE);
            }
            documentManager.updateDocumentByExternalApps(csv, ccsvState);
            LOG.info("END updateStateByPFI " + csv);
      }
      ...
      }
```

8.1.3 Actualizar quorum y fecha límite de preparación

Tras crear un documento (para mas de un firmante) desde una aplicación externa, podremos también establecer un quorum y una fecha límite de firma para ese documento.

Para ello, la aplicación en cuestión, invocará este servicio, informando del CSV del documento a actualizar, además de los datos sobre fecha y quorum.

Un ejemplo de utilización de este método desde una aplicación externa podría ser este:

Primero encapsulamos los datos que debemos pasar en el parámetro de entrada del tipo *ParamUpdateQuorumAndLimitDate*

```
ParamUpdateQuorumAndLimitDate paramCheckQuorumAndDate =
new ParamUpdateQuorumAndLimitDate();
      paramCheckQuorumAndDate.setCsv("CSVJ96LLVC3PHAP01PFI");
      paramCheckQuorumAndDate.setQuorum(2);
      paramCheckQuorumAndDate.setQuorumOnly(Boolean.TRUE);
      Calendar cal = Calendar.getInstance();
      cal.setTime(new Date());
                        Integer incFecha=7;
                        cal.add(Calendar.DATE, incFecha);
                        paramCheckQuorumAndDate.setDate(cal.getTime());

//Realizamos la llamada al método

ResultUpdateQuorumAndLimitDate resultCheckQuorumAndDate =
iDocumentIntegrationService.updateQuorumAndLimitDateDocument(paramCheckQuorumAnd
```

```
Date);
```


EDIFICIO PIGNATELLI
PASEO MARÍA AGUSTÍN, 36
50071 ZARAGOZA
TEL.: 976 71 4442
FAX.: 976 71 4185

9 Procedimiento de alta de una aplicación para integrarse con el servicio

Para que una aplicación pueda acceder a lo servicios expuestos es necesario que dicha aplicación esté dada de alta en la plataforma PAEGA. Para ello es necesario que la aplicación indique al equipo de PAEGA que realiza el alta de la siguiente información:

- Aplicación:
  - Código
  - Nombre
  - Versión
  - Organismo propietario o del cual depende la aplicación
  - URL
  - URL de respuesta para que PFI avise de los cambios de estado del documento.
- IPs:
  - Lista de IPs desde las cuales la aplicación accede a los servicios expuestos
- Métodos:
  - Lista de métodos a los cuales la aplicación necesita acceder. Debe de indicarse el nombre del servicio y el método. Para conocer los métodos expuestos consultar la URL donde se expone el servicio: https://*entorno*/pfi_core/services/