Govern de les Illes Balears
Vicepresidència i Conselleria d'Innovació, Recerca i Turisme
Direcció General de Desenvolupament Tecnològic

una manera de hacer europa

Fondo Europeo de Desarrollo Regional

Unió Europea

Análisis de plugins requeridos en SISTRA2

Mayo 2018

Servicios de Administración Electrónica en el Govern de les Illes Balears

Lot 2 (Servicios electrónicos para la ciudadanía)

Oficina Técnica de Dirección de Proyecto

Control de versiones del documento

Control de Cambios			
Data	**Autor**	**Versión**	**Cambios**
01/06/2018	Indra	v1.0	Análisis de plugins requeridos para la particularización de SISTRA2 por organismo
20/06/2018	Indra	v1.1	Incorporación de cambios surgidos en la reunión de revisión de plugins del 05/06/2018
06/07/2018	Indra	v1.2	Revisión de propiedades del plugin de catálogo de procedimientos y servicios

Revisado por		
Nombre	**Data**	**Área, departamento o empresa**

Aprobado por		
Nombre	**Data**	**Área, departamento o empresa**

Lista de distribución		
Nombre	**Área, departamento o empresa**	**Correo electrónico**

Índex

1. Objeto del documento

En el presente documento se analizan los siguientes aspectos:

- Mecanismo de integración de componentes externos con SISTRA2, diferenciando la estrategia de integración relativa a integración con componentes que únicamente requieren comunicación directa entre aplicaciones y una integración que además requiera redirección del navegador.
- Funcionalidades requeridas en SISTRA2 y cuál ha de ser su equivalencia en plugins, indicando que plugins ya existen y cuáles se deben implementar.
- Diseño de las interfaces de los nuevos plugins a implementar, teniendo en cuenta los detalles de la implementación específica en el contexto de la CAIB.

2. Integración con sistemas externos

La comunicación con sistemas externos por parte de la aplicación SISTRA2 generalizará mediante la utilización de plugins.

Básicamente se podrán distinguir 2 tipos de plugins:

- Plugins que implican exclusivamente comunicación directa entre aplicaciones (p.ej.: SISTRA2 y Regweb3), es decir, se invoca a servicios y se obtienen unos resultados.

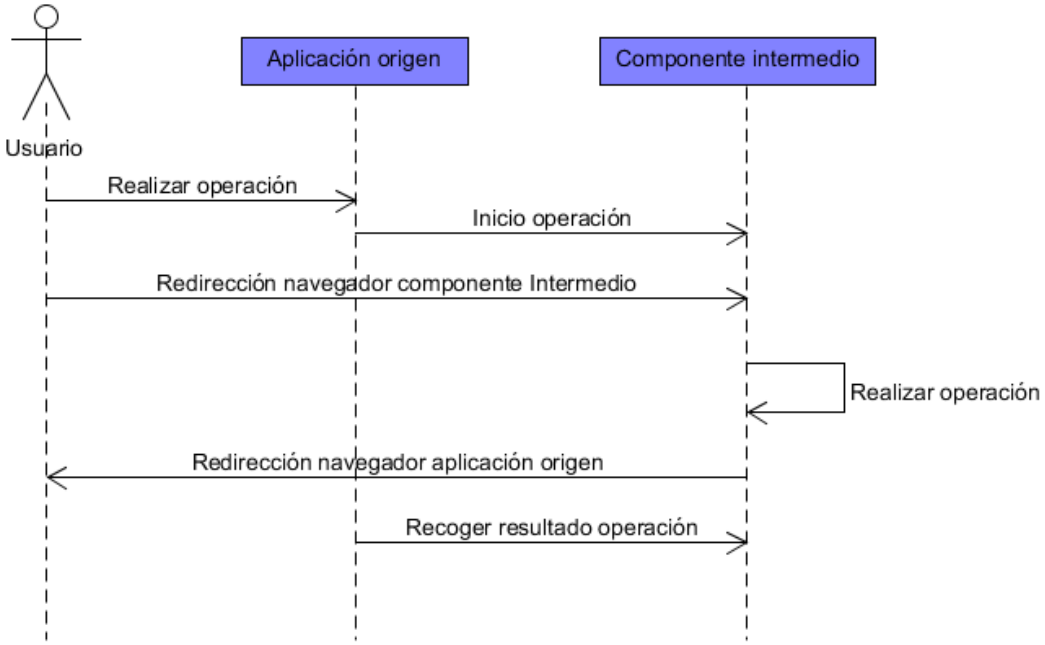
- Plugins que implican comunicación entre aplicaciones y además redirección de navegador a nivel de usuario (p.ej.: SISTRA2 y Cl@ve), es decir, además de invocar servicios entre aplicaciones existe un "salto" del navegador del usuario al plugin para realizar una operación y al finalizar dicha operación un "retorno" a la aplicación origen.

2.1. Plugins que implican redirección de navegador

Por motivos de reutilización y centralización de componentes, para evitar acoplamiento entre aplicaciones y permitir el versionado de APIs, se propone el uso de componentes horizontales basados en servicios.

Estos componentes serán instalables en el servidor (EARs diferenciados) y proveerán un API basado en servicios versionables (SOAP, REST, …). En cualquier caso, se puede ofrecer un cliente de referencia de estas APIs a las aplicaciones consumidoras, pero si la aplicación consumidora quiere evitar dependencias podrá consumir directamente esta API construyéndose su propio cliente.

El mecanismo propuesto para la integración con este tipo de componentes es la redirección y recuperación de datos en base a un solo (OTP). A continuación se detalla el flujo del mecanismo:



1. La aplicación origen, SISTRA2 en el caso que nos ocupa, invoca a, a través del plugin correspondiente, el servicio de componente externo para iniciar la operación. En esta invocación se pasan los parámetros necesarios para la operación y como uno de los parámetros necesarios se le deberá indicar cuál es la URL de *callback* a la aplicación origen tras finalizar la operación en el componente externo. Como resultado de la invocación del servicio se obtendrá la URL del componente externo a la que se debe redirigir el navegador para realizar la operación.

2. Redirección del navegador de la aplicación origen a componente externo, que es dónde se realizará la operación.

3. Se realiza el proceso requerido en el componente externo.

4. Una vez finalizada la operación, se redirige el navegador a la URL de *callback* que la aplicación origen le indicó en paso 1. Esta redirección se realiza pasando un token para que la aplicación origen pueda recoger el resultado (por seguridad esta redirección debería ser por POST y HTTPS).

5. La aplicación origen recoge el resultado de la operación a partir del token invocando al componente externo.

Con todo lo descrito la funcionalidad está aislada en un API simple basada en servicios y el intercambio de información, tanto en la llamada como en la respuesta, se realiza de forma segura, de aplicación a aplicación, a través de este API.

La propuesta tiene también como objetivo que, en la medida de lo posible, estos componentes externos, que ofrecen una funcionalidad horizontal, puedan ser reusables por diferentes

aplicaciones. Estos podrían ser los casos de Cl@ve autenticación, pasarela de pagos, Firmaweb u otros.

3. Definición de plugins

3.1. Diseño interfaz de plugins

Respecto al interfaz de los plugins se seguirán las pautas de diseño marcadas del proyecto PluginsIB.
El proyecto PluginsIB, disponible en el repositorio GITHUB, es un proyecto impulsado por la OTAE basado en un modelo de APIs-Plugins que da soporte a diversas funcionalidades que pretende su reutilización en diferentes aplicaciones.
Actualmente en el proyecto PluginsIB ya existen plugins que dan cobertura a diversas funcionalidades. En este documento se analizan las funcionalidades requeridas por SISTRA2 y se estudia si ya existe cobertura por algún plugin existente en PluginsIB. En caso de requerir una funcionalidad no soportada en PluginsIB se definirá un nuevo plugin siguiendo las pautas de diseño de PluginsIB (posteriormente se decidirá por parte de la OTAE si este nuevo plugin puede ser reusable para otras aplicaciones y puede pasar a formar parte del proyecto PluginsIB o en caso contrario será parte del proyecto de SISTRA2).

3.2. Funcionalidades requeridas en SISTRA2

En este apartado se presentan las necesidades de plugins detectados respecto a la funcionalidad requerida en SISTRA2 y se estudia si existe una implementación de los mismos en el proyecto PluginsIB o será necesario un nuevo plugin.

Funcionalidad requerida	Plugin requerido
• **Autenticación**	SISTRA2 ha de permitir el acceso no autenticado y el acceso autenticado mediante Cl@ve. En el ámbito del proyecto PluginsIB no existe un plugin de este tipo, será requerido el desarrollo del mismo. Actualmente existe un componente desarrollado por la Autoritat Portuaria de Baleares (APB) que centraliza la comunicación con Cl@ve autenticación y ofrece esta funcionalidad de autenticación basada en servicios SOAP. Se propone realizar una evolución de este componente para soportar multientidad e implementar nuevas mejoras (logout, api REST,…).

	Por otro lado, será necesaria también la definición de un nuevo plugin de autenticación y una implementación del plugin que siga el mecanismo de integración con componentes que requieren redirección del navegador del usuario.
• **Firma en cliente con sellado de tiempo**	SISTRA2 requiere la firma en cliente con sellado de firmas (XAdES, CAdES y PAdES). La firma en cliente se delegará en un componente externo de forma que se redirigirá el navegador hacia este componente externo, se realizará la firma y se retornará de nuevo a SISTRA2. Actualmente se está desarrollando por parte de la Fundación BIT el API de Firma Simple que implementará esta funcionalidad. Actualmente no existe un plugin específico en el proyecto PluginsIB, por lo que será necesario la definición de la interfaz de un nuevo plugin y una implementación del mismo basada en el API de Firma Simple. Este plugin seguirá el mecanismo definido para la integración con componentes que requieren redirección del navegador del usuario.
• **Firma en servidor**	Según los requerimientos de SISTRA2 no está prevista la necesidad de esta funcionalidad. En caso de requerirse debería incorporarse esta funcionalidad en la implementación del plugin de firma, que a su vez haría uso de los servicios correspondientes del API de Firma Simple.
• **Validación de firma (servidor) y recuperación de datos del firmante.**	SISTRA2 requiere estas funcionalidades para validar los firmantes. En el proyecto PluginsIB existe un plugin *plugins-validatesignature* que se utilizará en este caso. Se propone usar la implementación "afirmacxf" para no tener que gestionar las dependencias con Integr@.
• **Sellado de tiempo en servidor**	Dado que se prevé que el sellado de tiempo se realizará en la firma en cliente, no se contempla que en SISTRA2 se requiera sellado de tiempo en servidor.

• **Integración con el catálogo de procedimientos**	SISTRA2 requiere conectar con el catálogo de procedimientos de una entidad. En análisis del proyecto PluginsIB se ha detectado que no existe ningún plugin de catálogo de procedimientos. Por tanto, se debería crear un plugin de este tipo. Además, a nivel de CAIB se tendría que crear una implementación para ROLSAC.
• **Integración con registro telemático**	SISTRA2 requiere conectar con el registro de una entidad. En análisis del proyecto PluginsIB se ha detectado que no existe ningún plugin de registro, por lo que será necesaria su creación. Para su definición se basará en el API actual que ofrece REGWEB3 (dispone de los campos especificados en SICRES3 más una serie de campos específicos de REGWEB3). Por otro lado, se tendrá que crear una implementación del plugin para REGWEB3, que es la solución de registro corporativa de la CAIB.
• **Integración con sistemas de pagos**	SISTRA2 delegará en un componente externo la realización de un pago de forma que se redirigirá el navegador hacia este componente externo, se realizará el pago y se retornará de nuevo a SISTRA2. Además, existe como requisito que este componente externo sea reusable dentro de la CAIB para otras aplicaciones distintas a SISTRA2. Actualmente existe un plugin de pagos desarrollado bajo el proyecto PluginsIB para pagos a través de Redsys. Tras analizar el plugin se ha detectado que no es válido para SISTRA2: no se gestionan las redirecciones según el mecanismo definido en este documento de diseño y tiene un gran acoplamiento con *HttpServletRequest/HttpServletResponse* que hacen que no se pueda incluir como componente en la capa de servicio, gestiona estados de sesiones de pagos en memoria, no permite varias plataformas de pago (ATIB, TPV,..), gestión de pagos presenciales, etc. Se propone por tanto la creación de un componente horizontal de pasarela de pagos basado en un API de servicios REST y que siga el mecanismo de integración con componentes que requieren redirección del navegador del usuario. Posteriormente se definiría un plugin con una interfaz basada en los

	servicios del componente anterior y una implementación de dicho plugin para este componente de pasarela de pago.
• **Dominios remotos**	Esta funcionalidad se debería implementar a través de un plugin de resolución de dominios remotos. Dado que no existe ningún plugin de esta tipología en PluginsIB se tendría que crear uno, incorporando también una implementación basada en la definición de un contrato de servicio REST.
• **Generación de códigos de barra para documentos**	Para la implementación de esta funcionalidad se puede utilizar el plugin *plugins-barcode* del proyecto PluginsIB. Esta funcionalidad podría utilizarse para la generación de códigos QR a insertar en los documentos generados por SISTRA2. Uno de estos documentos podría ser la copia de solicitud que el ciudadano deberá presentar, en caso de preregistro, de manera presencial y con la correspondiente firma. La inserción del código QR favorecería la recuperación de datos del formulario electrónico por parte del backoffice.
• **Conversión de documentos**	La conversión de documentos se puede implementar a través del plugin *plugins-documentconverter* del proyecto PluginsIB. De esta manera se dará soporte a la conversión de documentos, por ejemplo, en el caso de conversión en PDF de manera previa a la firma de un documento. Para CAIB se utilizará la implementación del plugin basada en el consumo de servicios OpenOffice, que según la Fundació Bit permite la conversión PDF/A, que es una funcionalidad requerida.
• **Gestor de formularios externos**	Esta funcionalidad se deberá implementar a través de un plugin de formularios externos. Dado que no existe ningún plugin de esta tipología en PluginsIB se tendría que crear uno, incorporando también una implementación base. Esta funcionalidad no se incluirá en la versión inicial de SISTRA2 y su análisis detallado se realizará más adelante.

• **Integración con representación.**	Esta funcionalidad se debería implementar a través de un plugin de representación. Dado que no existe ningún plugin de esta tipología en PluginsIB se tendría que crear uno, incorporando también una implementación para REA y Habilit@. Esta funcionalidad no se incluirá en la versión inicial de SISTRA2 y su análisis detallado se realizará más adelante.

4. Diseño de nuevos plugins

Los *plugins* requeridos identificados en el punto anterior deberán definir una interfaz que marque las acciones necesarias para la integración con SISTRA2. Seguidamente se presentan las interfaces de comunicaciones para cada uno de los *plugins*.

4.1. Plugin de catálogo de procedimientos y servicios

Este plugin tiene como objeto la recuperación de información asociada a un trámite del catálogo de procedimientos y servicios corporativo del organismo. Esta información será utilizada tanto por el módulo SISTRAMIT para la autoconfiguración de un trámite telemático, así como por el módulo SISTRAGES para la recuperación de información correspondiente a qué procedimientos utilizan un determinado trámite telemático.

Método	Descripción	Parámetros	Resultado
obtenerProcedimientos	Consulta procedimientos del catálogo que hace uso de un trámite telemático	**codTramiteSistra**: identificador del trámite telemático en SISTRA2 **versionSistra:** versión del trámite telemático de SISTRA2.	**Procedimientos**: Lista de procedimientos que utilizan el trámite telemático.
obtenerInfoTramite	Consulta la información del trámite en el catálogo de procedimientos para su autoconfiguración	**codTramiteCatalogo**: identificador del trámite en el catálogo de procedimientos	**Tramite**: Datos del trámite recuperados del catálogo de procedimientos

Como estructura de datos para modelizar un procedimiento se utiliza Procedimiento, que contiene los siguientes campos:

Campo	Descripción
identificador	Código del procedimiento en el catálogo de procedimientos
codigoSia	Código SIA del procedimiento
titulo	Título del procedimiento
organoResponsable	Código DIR3 del órgano instructor del procedimiento administrativo

Como estructura de datos para modelizar un trámite se utiliza Tramite, que contiene los siguientes campos:

Campo	Descripción
identificador	Identificador trámite en el catálogo de procedimientos
tituloTramite	Título del trámite en el catálogo de procedimientos
procedimientoCP	Procedimiento asociado al trámite (clase Procedimiento)
emailSoporte	Correo electrónico de soporte para incidencias funcionales del trámite
vigente	Vigencia del trámite en el catálogo de procedimientos (S/N)
plazo	Plazo (fecha de inicio y fin) del trámite, en caso de haberlo.
urlInfoProc	URL de presentación de información del procedimiento administrativo
organoDestino	Código DIR3 del órgano destino del trámite
docAPresentar	Listado de documentos a presentar en el trámite. El objeto Documento tendrá los atributos siguientes: - Tipo (solicitud/anexo) - Título - Obligatoriedad (S/N) - URL Plantilla
Tasas	Listado de tasas a satisfacer en el trámite. El objeto Tasa tendrá los atributos siguientes: - Modelo - Código de Tasa - Concepto - Título

La implementación CAIB del plugin de catálogo de procedimiento se realizará sobre ROLSAC. Esta implementación estará basada en el consumo de los servicios proporcionados por el API REST de ROLSAC.

A continuación, se indican las llamadas a los distintos servicios del API REST que se deberán efectuar en la implementación de cada método del plugin:

Método	Servicios a invocar
obtenerProcedimientos	**Servicio tramites** => Se tendrá que invocar el este servicio incluyendo un filtro de trámites por codigoTramiteTelematico y versionTramiteTelematico para la recuperación de los códigos de procedimiento. **Servicio procedimientos** => Se tendrá que invocar este servicio pasando el parámetro código de procedimiento para cada uno de los códigos de procedimientos recuperados del servicio anterior. Con este servicio recuperaremos toda la información relativa al procedimiento.
obtenerInfoTramite	**Servicio tramites (/tramites/{codigo}) =>** Se invocará este servicio pasando como parámetro el código del trámite en el catálogo de procedimientos y se obtendrá la información relativa al trámite y el código de procedimiento relacionado. **Servicio procedimientos (/procedimientos/{codigo}) =>** A partir del código de procedimiento obtenido del servicio tramites, se utilizará el servicio procedimientos para la obtención de los datos del procedimiento. **Servicio documentos_tramites (/documentos_tramites) =>** A partir del código de trámite del catálogo de procedimientos se invocará al servicio de documentos para obtener todos los documentos asociados al trámite.

En las tablas siguientes se muestra la equivalencia, en caso de haberla, entre los campos existentes en ROLSAC y los atributos de las estructuras de datos Tramite y Procedimiento. Algunos de estos

atributos se requieren con el objeto de poder crear trámites en SISTRA2 a partir de los datos del trámite en el catálogo de procedimientos.

Procedimiento:

Campo Procedimiento	Campo ROLSAC
codigoCatalogo	Código de procedimiento
codigoSia	Código SIA del procedimiento
titulo	Título del procedimiento
organoResponsable	Órgano instructor del procedimiento

Tramite:

Campo Tramite	Campo ROLSAC
codigoSiaProc	Código SIA del procedimiento
tituloTramite	Título del trámite
emailSoporte	Actualmente existe, a nivel de datos de contacto del procedimiento, el email del responsable. Se ha de revisar si se puede utilizar este campo.
vigente	Vigencia del trámite en el catálogo de procedimientos (S/N)
plazo	Fecha de inicio y fecha de cierre del trámite
urlInfoProc	URL de presentación de información del procedimiento administrativo
organoDestino	Codigo DIR3 del primer órgano jerárquicamente superior al órgano instructor que disponga de dicho dato.
docAPresentar	- Tipo (solicitud/anexo) => Se ha de revisar la correspondencia con "Modelos" y "Documentos relacionados con el trámite" actualmente existentes. - Título => Título del documento - Obligatoriedad (S/N) => **No tiene correspondencia en ROLSAC** - URL Plantilla => **No tiene correspondencia en ROLSAC**
Tasas	- Modelo => **No tiene correspondencia en ROLSAC** - Código de Tasa => codificación de tasa - Concepto => **No tiene correspondencia en ROLSAC**

	- Título => descripción de la tasa

4.2. Plugin de registro

Este plugin tiene como objeto la recuperación de información de registro (oficinas registrales, libros, asuntos y órganos destino) y la realización de apuntes registrales sobre la solución de registro del organismo.
Aunque sería deseable que la interfaz se basara estrictamente en los datos definidos por la norma SICRES3, para su definición se basará en el api actual que ofrece REGWEB3, que dispone de los campos especificados en Sicres3 más una serie de campos específicos de REGWEB3.
Este plugin contemplará, a petición de la CAIB, funcionalidades no requeridas por SISTRA2, como pueden ser el registro de salida y otras funcionalidades que se detecten en tiempo de desarrollo de otras aplicaciones involucradas en el esquema de Administración Electrónica CAIB.

Método	Descripción	Parámetros	Resultado
obtenerOficinasRegistro	Obtiene el listado de oficinas de registro de una entidad	**codigoEntidad**: código DIR3 de la entidad a consultar**.**	**Oficinas**: Lista de oficinas de registro.
obtenerLibrosOficina	Obtiene el listado de libros a los que la oficina da servicio	**codigoEntidad**: código DIR3 de la entidad a consultar**.** **codigoOficina:** código DIR3 de la oficina a consultar	**Libros**: Listado de libros a que da servicio la oficina.
obtenerTiposAsunto	Obtiene el listado de tipos de asunto de una entidad	**codigoEntidad**: código DIR3 de la entidad a consultar**.**	**TiposAsunto:** Listado de tipos de asunto de la entidad
registroEntrada	Realiza apunte registral de entrada	**codigoEntidad**: código DIR3 de la entidad a consultar**.** **asientoRegistral:** clase que modelizará los datos requeridos para el registro. (incluidos anexos)	**ResultadoRegistro**: Indica el resultado de registrar un asiento: número de registro y fecha.
registroSalida	Realiza apunte registral de salida	**codigoEntidad**: código DIR3 de la entidad a consultar**.**	**ResultadoRegistro:** Indica el resultado de registrar un asiento:

		asientoRegistral: clase que modelizará los datos requeridos para el registro de salida. (incluidos anexos)	número de registro y fecha.
obtenerJustificanteRegistro	Recupera el justificante de un registro de entrada	**codigoEntidad**: código DIR3 de la entidad a consultar**.** **numeroRegistro:** número de registro de entrada	**JustificanteRegistro:** Array de bytes con el justificante en formato PDF.

4.3. Plugin de dominios remotos

La gestión de consultas remotas a un *backoffice* se realizará a través del plugin de dominios remotos y como resultado se podrá obtener un resultado de conjunto de valores y/o ficheros.

La securización de los dominios se establecerá a partir de las propiedades del plugin a nivel de entidad:

- Se establecerá un usuario/password a nivel general para la entidad (se identificaría de forma general a SISTRA2 con ese usuario/password frente a las aplicaciones a las que invoca).
- Se podrá establecer también un usuario/password específicamente para cada dominio (se identificaría a SISTRA2 con un usuario/password específico según la aplicación a la que invoca).

Método	**Descripción**	**Parámetros**	**Resultado**
resolverDominio	Consulta datos de un dominio contra un backoffice	**identificadorDominio**: código del dominio definido en SISTRA2. **urlEndpoint:** URL del endpoint del Webservice publicado por el Backoffice para la resolución del dominio. **parametrosDominio:** listado de tuplas código parámetro/ valor	**resultadoDominio**: Objeto de respuesta que encapsula los resultados del dominio, incluyendo la descripción del error si la hubiera.

		parámetro. Será posible pasar archivos como parámetro con previa conversión a Base64.	

Como estructura de datos para modelizar la respuesta de un dominio se utiliza resultadoDominio, que contiene los siguientes campos:

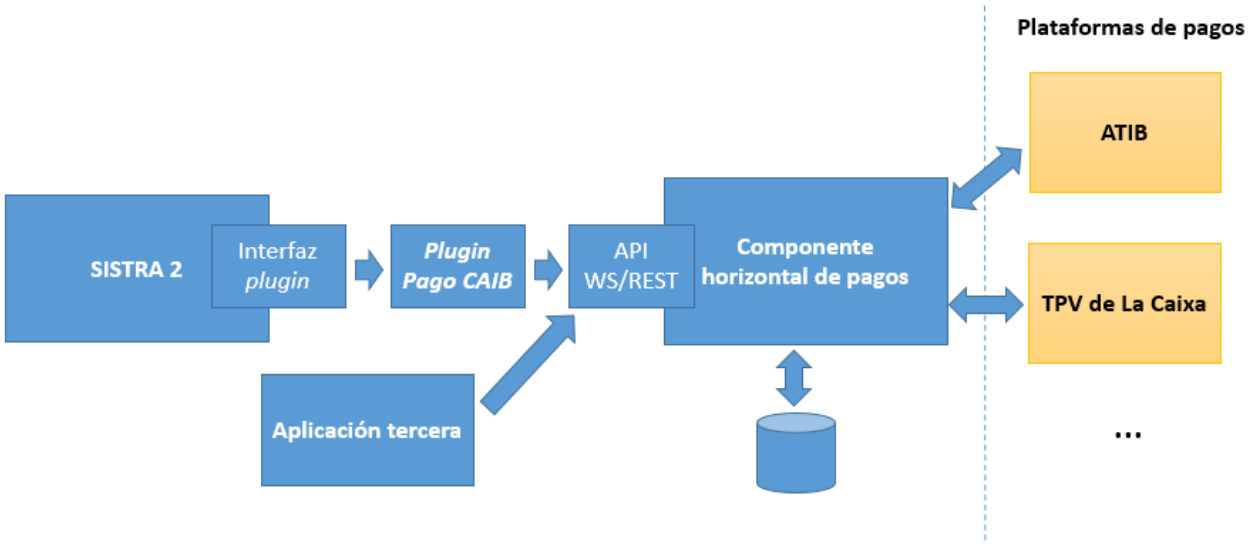
Campo	Descripción
error	Indica si ha habido error o no (S/N)
codigoError	Código de error devuelto por el sistema remoto, en caso de haberlo.
descError	Descripción de error devuelto por el sistema remoto, en caso de haberlo.
registros	Listado de registros con el resultado de la consulta del dominio (formato filas/columnas)
archivos	Listado de ficheros retornados. Cada fichero tendrá asociado un identificador de fichero y el nombre del mismo, con extensión.

4.4. Plugin de pagos

SISTRA2 delega en un componente externo la realización de un pago de forma que se redirige el navegador hacia este componente externo, se realiza el pago y se retorna de nuevo a SISTRA2 al finalizar el pago. Este componente debe soportar diferentes plataformas de pago y según permita la plataforma de pago realizar pagos electrónicos o presenciales. Se encarga de la gestión del proceso de pago con las diferentes plataformas, almacenando en BBDD los estados del pago para poder tener una auditoría de los mismos y se comunicará a través de una API de servicios. Este componente horizontal de pagos puede ser usado por otras aplicaciones diferentes a SISTRA2.

Desde SISTRA2 se utilizará este componente mediante un plugin. Para ello se definirá una nueva interfaz de plugin de pagos compatible con el funcionamiento comentando y se realizará una implementación de este plugin para el componente horizontal de pagos.

En el diagrama siguiente se muestra la solución planteada.



Este componente incorporará para el caso CAIB la integración con la pasarela de pagos ATIB y el TPV de La Caixa, aunque este último no estará disponible para la versión inicial de SISTRA2. El componente horizontal de pagos tendrá la siguiente arquitectura:

- **Negocio**: lógica de negocio de integración con las diferentes pasarelas de pago.
- **Frontal**: capa web del componente que gestionará la redirección hacia las plataformas de pago.
- **Backoffice**: módulo de administración del gestor de pagos que permita verificar estado de los pagos y en caso necesario realizar operaciones como la confirmación pagos. Esto puede ser de gran utilidad para la gestión de casos en las que falle la confirmación del pago por parte de la pasarela de pago correspondiente como es el caso del funcionamiento del TPV de La Caixa (Redsys) donde por parte de La Caixa sólo se realiza un intento de confirmación de pago y no hay mecanismo automático con La Caixa para consultar el estado del pago.

- **Webservices**: capa de servicios web (REST o SOAP) para integración con la aplicación que requiera realizar el pago.
- **Procesos**: procesos de *background* (purgado de tickets, etc.)

Por otro lado, puede ser necesario la generación de un justificante de pagos estándar por parte del componente, ya que puede haber plataformas de pagos que no permitan la recuperación del justificante para su presentación al interesado en la aplicación desde la que se ha realizado el pago (p.ej.: TPV). Este justificante debe incluir valores identificativos del pago como son el importe de la tasa, el tipo de pago, la pasarela utilizada, el localizador del pago, etc. Desde el punto de vista de SISTRA2 este justificante de pago puede ser obligatorio en el caso de preregistro ya que, si se realiza el pago electrónico en un preregistro, el justificante de pago es un documento obligatorio a presentar presencialmente y podría pasar en el caso del pago por TPV que el ciudadano no hubiese descargado el justificante en la aplicación del TPV. Para las pasarelas que sí que permitan recuperar el justificante generado por la pasarela (p.e. ATIB) se usará el justificante propio de la pasarela.

Respecto al plugin de pagos, a continuación, se muestra el interfaz que deberá implementar:

Método	Descripción	Parámetros	Resultado
iniciarSesionPago	Permite iniciar una sesión de pagos	**datosPago**: pasarela de pago, tipo de pago permitido, modelo, tasa, importe, fecha devengo, datos sujeto pasivo y fecha límite de inicio de pago (opcional). **parametrosPasarela:** parámetros adicionales específicos según pasarela de pagos (p. ej.: para TPV órgano emisor). **urlCallBack:** URL de *callback* de la aplicación invocante a la que el gestor de pagos deberá invocar una vez finalizada la realización del pago. En esta URL de *callback* se pasará como parámetro por POST el campo ticket de sesión de pago generado como resultado de la llamada.	**datosInicioSesion**: Ticket de sesión de pagos y URL a la que se deberá redirigir el navegador del usuario para ir al gestor de pagos.

reanudarSesionPago	Permite reanudar una sesión de pagos ya iniciada	**ticketSesionPago**: identificador de la sesión de pagos necesario para el retorno del gestor de pagos hacia la aplicación invocante.	**urlGestorPagos**: URL a la que se deberá redirigir el navegador del usuario para ir al gestor de pagos.
comprobarEstadoSesionPago	Permite comprobar el estado de una sesión de pagos	**ticketSesionPago**: identificador de la sesión de pagos.	**estadoPago:** datos relativos al estado del pago (detalle de los datos especificado abajo)
calcularImporteTasa	Permite calcular el importe de una tasa	**identificadorTasa:** código identificador de la tasa. **paramConsulta:** lista de tuplas código/valor para especificar parámetros. Dependerá de la pasarela de pagos correspondiente.	**importeTasa:** importe en céntimos de la tasa.
obtenerJustificantePago	Permite, para pagos realizados, la obtención del documento justificante del mismo	**ticketSesionPago**: identificador de la sesión de pagos. **LocalizadorNrc**: identificador del pago en la pasarela de pagos correspondiente.	**Justificante:** array de bytes con el contenido del documento justificante.

Como estructura de datos para modelizar la respuesta del método comprobarEstadoSesionPago se utiliza estadoPago, que contiene los siguientes campos:

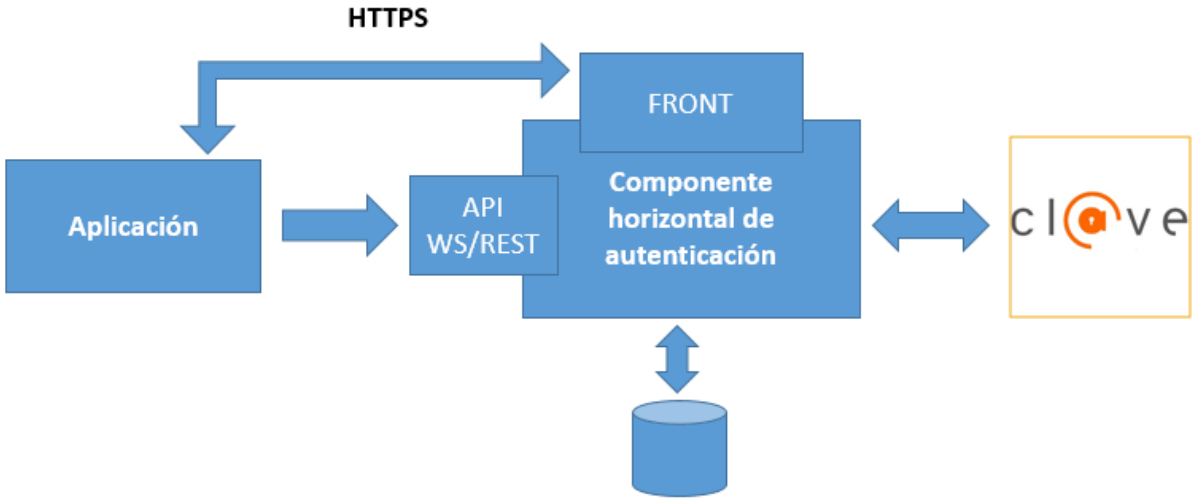
Campo	Descripción
estado	Estado del pago (No existe/Pago en curso/Pago pendiente confirmar/Pago confirmado/Excedido tiempo de pago)
descripcionEstado	Mensaje descriptivo del estado del pago
tipoPago	Para un pago realizado indica el tipo de pago (tarjeta, cargo en cuenta o presencial)

identificadorPago	Identificador del pago en la pasarela de pagos (p. ej.: NRC)
fechaPago	Fecha en la que se ha realizado el pago

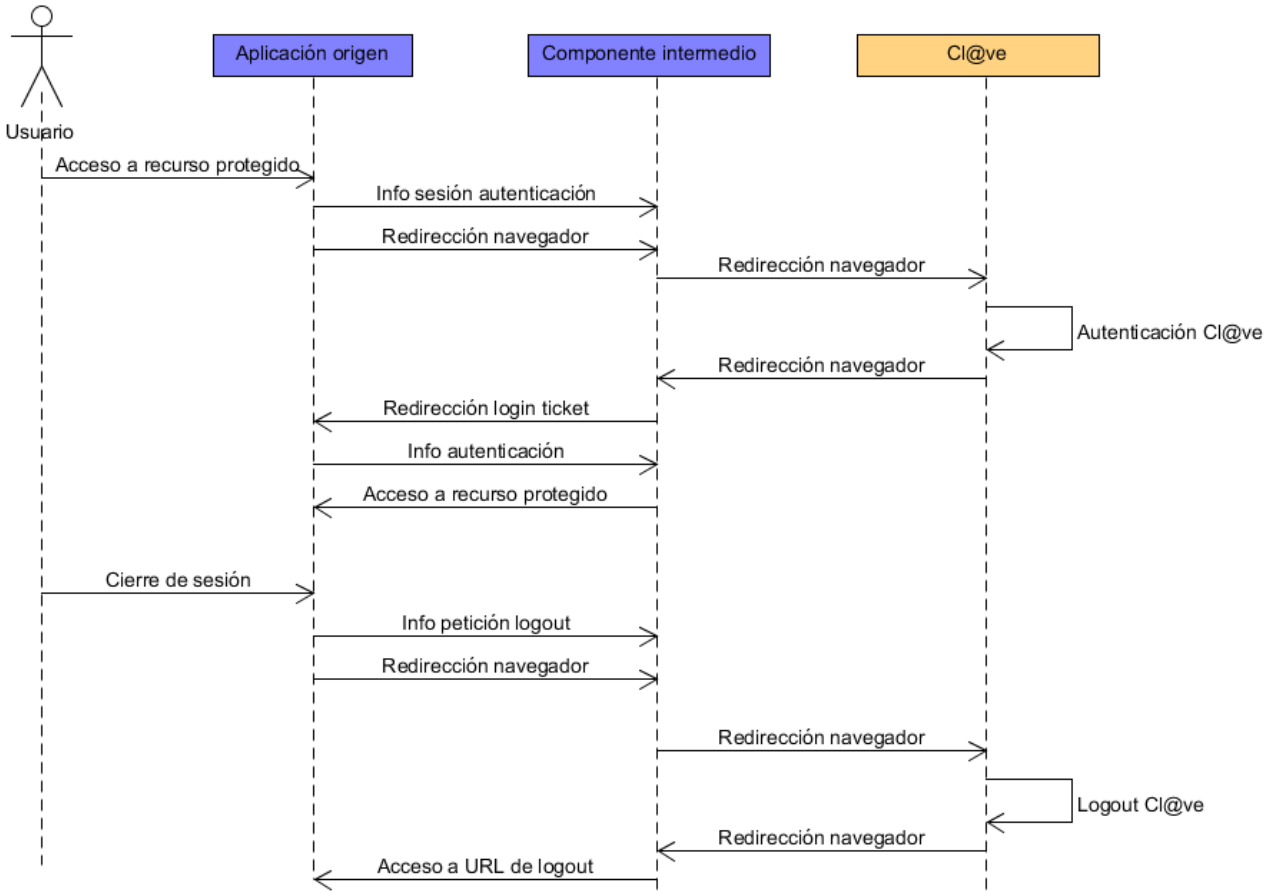
4.5. Plugin de Autentación

Actualmente existe un componente desarrollado por la Autoritat Portuaria de Baleares (APB) que centraliza la comunicación con Cl@ve autenticación y ofrece esta funcionalidad de autenticación basada en servicios SOAP.
Se propone realizar una evolución de este componente a uno propio CAIB para soportar multientidad, permitir nuevos tipos de autenticación (inicialmente anónimo y Cl@ve), eliminar el modo de funcionamiento acoplado a BBDD de SISTRA1 e implementar nuevas mejoras (*logout*, API REST,…).
Este componente intermedio incluirá la gestión de la página de login, de manera que se muestre la misma página en las diferentes aplicaciones corporativas que hagan uso del componente. De esta manera, el componente recogerá las peticiones de identificación por parte de las aplicaciones y les trasladará la información de autenticación mediante un ticket de acceso.
En el componente intermedio se gestionarán todos los parámetros de conexión con Cl@ve para las diferentes entidades (spId, providerName, spSector, spApplication, certificados). Para los casos en los que las aplicaciones peticionarias del servicio de identificación únicamente requieran la autenticación mediante Cl@ve, se estudiaría la posibilidad de configurar el componente intermedio para obviar la página de login y redireccionar automáticamente hacia Cl@ve.

El diagrama de componentes en el que se basa la solución es el siguiente:



A nivel de funcionamiento, el diagrama de secuencia en el que se basará la funcionalidad del componente es el que se presenta a continuación:



1. El usuario accede a un recurso protegido de la aplicación, y en ese momento se invoca al servicio web del componente intermedio para pasar la información de la sesión de autenticación:
    - urlCallbackLogin: URL a la que se debe redirigir tras finalizar el proceso de autenticación. A esta URL se le pasará por POST un parámetro con un ticket para poder obtener la información de autenticación.
    - métodos: métodos de autenticación soportados (anónimo, aFirma, AEAT y SS)
    - idioma: idioma

    El componente intermedio genera sesión de autenticación y almacena en una tabla la información pasada en estos parámetros. Como respuesta al webservice se indica la url a la que se debe redirigir el navegador, en la que se mostrará la página de login.

3. La aplicación redirige el navegador a la URL obtenida como resultado de la invocación al webservice.
4. En función del método de autenticación seleccionado por el usuario, el componente intermedio llevará a cabo la operativa correspondiente. En el caso de Cl@ve, el componente

prepara la petición de autenticación a Cl@ve y la firma. En esta petición se indica la URL de retorno que debe utilizar Cl@ve para retornar al componente intermedio. Se redirige el navegador del usuario a Cl@ve pasando por POST la petición de autenticación. En caso de acceso anónimo, se obviarán los pasos 5 y 6 expuestos a continuación y se pasará directamente el 7.

5. Cl@ve procede a autenticar al usuario según el método seleccionado.
6. Cl@ve redirige el navegador a la URL de retorno pasando por POST un parámetro con la respuesta firmada, que contiene los datos de la autenticación. El componente intermedio procesa la respuesta de Cl@ve y con los datos de la autenticación se actualiza en la tabla de BBDD los datos de la autenticación generando un ticket.
7. El componente intermedio redirige a la url de callback de login pasando el ticket.
8. La aplicación accede al webservice del componente intermedio para obtener la información de autenticación a partir del ticket. Una vez usado el ticket, este se borra de la tabla para evitar que pueda ser usado otra vez. Se procede con la autenticación y se valida el usuario en la aplicación.
9. Una vez validado el usuario, se permite el acceso al recurso.

Por tanto, este componente intermedio se compone de:

- *webservices* para integrarse con las aplicaciones (solicitud sesión autenticación y obtención de los datos del usuario autenticado)
- *frontal web* para redirección entre la aplicación y el sistema de autenticación. Este frontal incorpora la pantalla de login en la que el usuario deberá seleccionar el método de autenticación.

Los servicios que implementará el componente son que se describen seguidamente:

Método	iniciarSesion	
Descripción	Genera sesión de autenticación y almacena en una tabla la información pasada en estos parámetros. Como respuesta se indica la URL a la que se debe redirigir el navegador.	
Parámetros de Entrada		
	entidad	Código DIR3 de la entidad para la que se realiza la autenticación
	urlCallbackLogin	URL a la que se debe redirigir tras finalizar el proceso de autenticación. A esta URL se le pasará por POST un parámetro con un ticket para poder obtener la información de autenticación.
	metodosAutenticacion	Método de autenticación a utilizar separados por ";". Los valores soportados son: anónimo, aFirma, AEAT, SS. p.ej.: "anonimo;aFirma;AEAT;SS;"

<table>
<tr><td></td><td>qaa</td><td>Opcional. Indica el nivel de seguridad aplicado. Los valores soportados: 2, 3, 4. Por defecto será el valor 3.
<table>
<tr><th>Nivel calidad</th><th>Sistema de identificación</th><th>Nivel ENS</th><th>Proveedor de servicios de identificación y autenticación</th><th>Posibles ejemplos de uso</th></tr>
<tr><td>Nivel 4</td><td>• DNIe
• Otros certificados reconocidos en soporte Hardware</td><td>ALTO</td><td>@firma</td><td>Acceso a datos de salud</td></tr>
<tr><td>Nivel 3</td><td>• Certificados electrónicos SW reconocidos
• Claves concertadas de la Seguridad Social combinadas con mensaje SMS</td><td>MEDIO/ ALTO</td><td>@firma
GISS</td><td>Acceso a expedientes con información personal con nivel de protección medio</td></tr>
<tr><td>Nivel 2</td><td>• PIN24H
• Claves concertadas de la Seguridad Social sin SMS</td><td>BAJO</td><td>AEAT
GISS</td><td>Acceso a expedientes con información personal con nivel de protección bajo</td></tr>
</table></td></tr>
<tr><td></td><td>idioma</td><td>Idioma (es, ca,…).</td></tr>
<tr><td></td><td>forzarAutenticacion</td><td>En caso que el sistema de autenticación utilizado implemente SSO, este parámetro permite indicar si se fuerza la autenticación (true) o si por el contrario se intenta aplicar el SSO.</td></tr>
<tr><td colspan="3">Parámetros de Salida</td></tr>
<tr><td rowspan="2">respuesta</td><td colspan="2">Resultado del servicio</td></tr>
<tr><td>urlRedireccion</td><td>URL a la que se debe redirigir el navegador con el identificador de sesión como parámetro GET</td></tr>
</table>

<table>
<tr><td>Método</td><td colspan="2">obtenerDatosTicket</td></tr>
<tr><td>Descripción</td><td colspan="2">Obtiene la información de autenticación a partir del ticket.</td></tr>
<tr><td colspan="3">Parámetros de Entrada</td></tr>
<tr><td></td><td>ticket</td><td>Identificador único del ticket.</td></tr>
<tr><td colspan="3">Parámetros de Salida</td></tr>
<tr><td rowspan="6">respuesta</td><td colspan="2">Resultado del servicio</td></tr>
<tr><td>metodoAutenticacion</td><td>Método de autenticación solicitado en la operación “iniciarSesion”.</td></tr>
<tr><td>nif</td><td>NIF.</td></tr>
<tr><td>nombre</td><td>Nombre.</td></tr>
<tr><td>apellidos</td><td>Concatenación de apellido1 y apellido2.</td></tr>
<tr><td>apellido1</td><td>Primer apellido.</td></tr>
</table>

	apellido2	Segundo apellido.

Método	iniciarLogoutSesion	
Descripción	Genera sesión de *logout* y almacena en una tabla la información pasada en estos parámetros. Como respuesta se indica la URL a la que se debe redirigir el navegador. Este *logout* será requerido si el sistema de autenticación empleado implementa SSO, como es el caso de Cl@ve.	
Parámetros de Entrada		
	entidad	Código DIR3 de la entidad para la que se realiza la autenticación
	urlCallBack	Url a la que se debe redirigir tras finalizar el proceso de *logout*.
	idioma	Idioma (es, ca,…).
Parámetros de Salida		
respuesta	Resultado del servicio	
	resultado	Booleano indicando si ha ido bien la operación.
	datos	En caso de que la operación sea correcta se devuelve: • urlRedireccion: URL a la que se debe redirigir el navegador con el identificador de *logout* de sesión como parámetro GET
	error	En caso de que la operación no sea correcta contendrá los campos: • codigoError: código de error • descripcionError: descripción del error

La integración de SISTRA2 con este componente intermedio, siguiendo la filosofía de plugins establecida, se realizará a través de un plugin de login. Dicho plugin se incluirá en el proyecto Plugins-IB y deberá disponer de una implementación CAIB basada en el consumo de los servicios del componente intermedio anteriormente presentado.

Por lo que respecta al plugin, la interfaz que deberá implementar es la siguiente:

Método	**Descripción**	**Parámetros**	**Resultado**

iniciarLoginSesion	Permite generar una sesión de autenticación	**entidad:** código DIR3 de la entidad para la que se realiza la autenticación **idioma:** idioma **metodosAutenticacion:** métodos de autenticación a permitir **qaa:** nivel de seguridad aplicado (opcional) **forzarAutenticacion:** permite indicar si se fuerza la autenticación o se aplica el SSO de la solución de autenticación **urlCallbackLogin:** URL a la que se debe redirigir tras finalizar el proceso de autenticación	**urlRedireccion:** URL a la que se debe redirigir el navegador con el identificador de sesión como parámetro
obtenerDatosTicket	Permite obtener la información de autenticación a partir del ticket de sesión	**ticket:** identificador del ticket de sesión	**infoAutenticacion:** datos de la autenticación (método, NIF, nombre, apellido1, apellido2 y apellidos)
iniciarLogoutSesion	Permite generar una sesión de *logout*	**entidad:** código DIR3 de la entidad para la que se realiza la autenticación **idioma:** idioma **urlCallbackLogin:** URL a la que se debe redirigir tras finalizar el proceso de *logout*	**resultadoLogout**: datos resultantes del proceso de logout, como son el resultado, URL a la que se ha de redirigir el navegador, y error si lo hubiera

4.6. Plugin de Firma en cliente

La funcionalidad de firma electrónica de documentos por parte del ciudadano en el asistente de tramitación (firma cliente) se llevará a cabo a través del plugin de firma. La implementación CAIB de este plugin consumirá los servicios REST del API de Firma Web Simple, que contará con un servidor intermedio que proporcione los diferentes proveedores de firma (AutoFirma, Miniapplet, FiRe u otros) y toda la gestión de la firma (selladores de tiempo, gestión de errores, gestión de usuarios de aplicación y propiedades de firma asociadas). Este servidor intermedio permitirá, a nivel de entidad y/o usuario de aplicación, configuraciones tales como:

- Filtro de Certificados
- Política de Firmas: Identificador de Política (OID), Hash del OID (Base64), Algoritmo del Hash del OID, URL al documento de política de firmas
- Tipos Firma: PADES, XADES, CADES, SMIME, ...
- Algorismo de Firma: SHA1, SHA256, ...
- Modo de Firma: implícita o explícita (attached o detached)
- Proveedores de firma permitidos
- Proveedores de sellado de tiempo
- …

En cuanto al plugin de firma, a continuación, se muestra el interfaz que deberá implementar.

Método	**Descripción**	**Parámetros**	**Resultado**
generarSesionFirma	Permite generar una sesión de firma	**infoSesionFirma:** datos relativos a la sesión de firma, como son el idioma, el código de entidad, el NIF del firmante, nombre de usuario firmante, código SIA del procedimiento,	**idSesionFirma:** identificador de la sesión de firma
anyadirFichero	Permite añadir un fichero para firmar	**ficheroAFirmar:** datos del fichero a añadir para firma	
iniciarSesionFirma	Permite iniciar la operación de firma	**idSesionFirma:** identificador de la sesión de firma **urlCallBack:** URL de *callback* de la aplicación invocante a la que la pasarela de firma deberá invocar una vez finalizada la realización de la firma.	**urlPasarelaFirma**: URL a la que se deberá redirigir el navegador del usuario para ir la pasarela de firma.

		paramAdic: parámetros adicionales según pasarela de firma.	
obtenerEstadoSesionFirma	Permite obtener el estado de la sesión de firma	**idSesionFirma:** identificador de la sesión de firma	**estadoSesionFirma:** estado de la sesión de pago
obtenerFirmaFichero	Permite obtener la firma de un fichero firmado.	**idSesionFirma:** identificador de la sesión de firma **idFicheroFirma**: identificador de fichero a firmar	**ficheroFirmado:** datos del fichero firmado
cerrarSesionFirma	Permite cerrar una sesión de firma	**idSesionFirma:** identificador de la sesión de firma	

Como estructura de datos para modelizar la respuesta de los métodos obtenerEstadoSesionFirma y obtenerFirmaFichero se utilizan estadoSesionFirma y ficheroFirmado respectivamente. Seguidamente se detallan los atributos de cada una de estas estructuras.

estadoSesionFirma:

Campo	Descripción
estado	Código de estado de la sesión de firma
descripcionEstado	Mensaje descriptivo del estado de la sesión de firma
estadoFirmaDocumentos	Listado de estados de la firma de cada fichero. Se devolverá el par identificador de fichero a firmar y estado.

ficheroFirmado:

Campo	Descripción
nombreFichero	Nombre del fichero firmado
mimetypeFichero	MimeType del fichero firmado
firmaFichero	Array de bytes con la firma del documento