



## **Integración BackOffice**

Código: SISTRA-INTBOF



**GOVERN DE LES  
ILLES BALEARS**

## INTEGRACIÓN BACKOFFICE

## HOJA DE CONTROL DE DOCUMENTO

## DOCUMENTO/ARCHIVO

Título: Integración BackOffice	Nombre Archivo/s: SISTRA-INTEGRACION-003.doc
Código: SISTRA-INTBOF	Soporte lógico: Word
Fecha: 08/02/2010	
Versión: 3	

## REGISTRO DE CAMBIOS

Versión	Páginas	Motivo del cambio
1	24	Creación del Documento
2	23	Añadir interfaz webservice
3	24	Simplificación webservice registro telemático

## DISTRIBUCIÓN DEL DOCUMENTO

Nombre	Personal

## CONTROL DEL DOCUMENTO

PREPARADO	REVISADO/ APROBADO	ACEPTADO	ACEPTADO
Rafael Sanz	José Vicente Juan Pérez	Jeroni Navarrete	Bernat Alberti

Cumplimentar con el nombre, la firma y la fecha

Sólo para clientes

---

**INTEGRACIÓN BACKOFFICE**

---

<b>INDICE</b>
---------------

<b>1. INTRODUCCIÓN .....</b>	<b>4</b>
<b>2. ASISTENTE TRAMITACIÓN (SISTRA Y FORMS).....</b>	<b>5</b>
2.1. DOMINIOS.....	5
2.2. TRÁMITES TIPO CONSULTA .....	7
<b>3. REPOSITORIO DE DOCUMENTOS SEGURO (REDOSE).....</b>	<b>9</b>
<b>4. BANDEJA TELEMÁTICA (BANTEL) .....</b>	<b>13</b>
<b>5. ZONA PERSONAL (ZONAPER).....</b>	<b>15</b>
5.1. EXPEDIENTES .....	16
5.2. EVENTOS .....	16
5.2.1. Documentos .....	17
<b>6. REGISTRO TELEMÁTICO (REGTEL) .....</b>	<b>18</b>
6.1. GENERACIÓN DE REGISTROS DE ENTRADA.....	20
6.2. GENERACIÓN DE REGISTROS DE SALIDA (NOTIFICACIONES TELEMÁTICAS).....	20
<b>7. MOVILIDAD (MOBTRATEL) .....</b>	<b>22</b>
<b>8. ANEXO VII: GESTIÓN DE TRANSACCIONES .....</b>	<b>24</b>

## INTEGRACIÓN BACKOFFICE

---

### 1. Introducción

El objeto de este documento es definir la integración de los BackOffices con la Plataforma de Tramitación Telemática.

Esta integración afecta a los siguientes puntos:

- integración con el Sistema de Tramitación (resolución de Dominios y Consultas)
- integración con la Bandeja Telemática (Procesamiento de entradas)
- integración con la Zona Personal (Mis expedientes)
- integración con el Repositorio de Documentos Seguro (RDS)
- integración con el Registro Telemático (Notificaciones y Registros de entrada)
- integración con el módulo de Movilidad (Envío de mensajes Email y SMS)

El acceso por parte de los BackOffices a la plataforma (y viceversa) puede realizarse mediante acceso por EJB y acceso por Web Service.

A la hora de diseñar una solución con acceso por EJB hay que tener en cuenta la gestión de la transaccionalidad que dependerá de si el módulo de integración esta instalado en el mismo servidor dónde esta instalada la plataforma o bien esta instalada en otro servidor (ver Anexo Transacciones).

En cuanto al acceso por Web Service para cada módulo se definirá un WSDL para:

- servicios ofrecidos por el módulo
- servicios invocados por el módulo

La plataforma soporta el versionado de web services para permitir la gestión del cambio de especificaciones en el interfaz, definiendo para cada versión de interfaz un WSDL.

La interfaz de acceso por Web Service emula a la interfaz de acceso por EJB de modo que la ayuda (javadoc, etc.) es aplicable a la interfaz de Web Service.

## INTEGRACIÓN BACKOFFICE

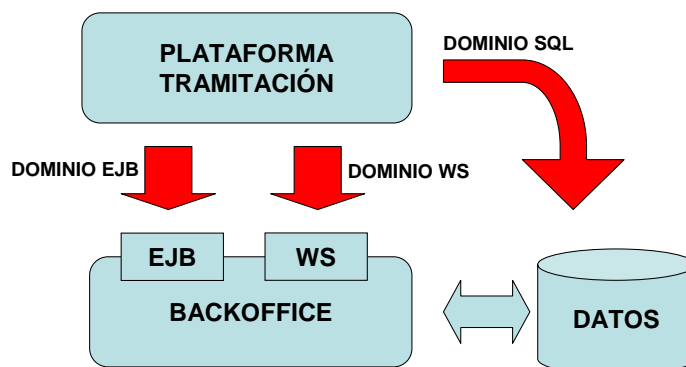
### 2. Asistente tramitación (SISTRA y FORMS)

#### 2.1. Dominios

Un Dominio es una lista de valores que normalmente provienen de una consulta de BBDD. Hay casos en los que es necesario en el proceso de tramitación telemática (para completar un formulario o realizar una validación) acceder a dominios que deba resolverlos la aplicación de BackOffice ya que que es el BackOffice quién puede acceder a esos datos.

El Sistema de Tramitación puede invocar de manera remota a dominios ubicados en los BackOffices. Esta invocación se puede realizar de tres formas:

- acceso directo JDBC: requiere conectividad directa entre el Sistema de Tramitación y la BBDD destino.
- acceso mediante EJB: el BackOffice debe implementar un EJB que será invocado por el Sistema de Tramitación para resolver el dominio.
- acceso mediante Web Service: el BackOffice debe implementar un Web Service que será invocado por el Sistema de Tramitación para resolver el dominio



Para identificar a un dominio se le asignará un identificador (máximo 10 caracteres). La asignación de estos identificadores deberá seguir el esquema de nomenclatura definido en el anexo de Definición de identificadores.

El acceso a un dominio puede ser:

- Acceso directo JDBC

En este tipo de acceso se deberá indicar el path JNDI de un Datasource accesible desde el servidor de aplicaciones dónde este instalado el Sistema de Tramitación.

Además hay que indicar la SQL de consulta. Esta SQL puede ir parametrizada (los parámetros serán de tipo String).

En este tipo de acceso el BackOffice no debe implementar ningún elemento.

## INTEGRACIÓN BACKOFFICE

---

Ej: SQL para obtener las localidades de una provincia.

```
SELECT LOC_CODIGO CODIGO, LOC_DESC LOCALIDAD FROM LOCALIDADES WHERE  
LOC_PROV = ?
```

- Acceso mediante EJB

El BackOffice debe implementar un EJB que debe definirse para que utilice las siguientes interfaces *home* y *remote* :

- home: es.caib.sistra.persistence.intf.DominioEJBHome
- remote: es.caib.sistra.persistence.intf.DominioEJB

Estas interfaces definen un único método para realizar la consulta:

```
public ValoresDominio obtenerDominio(String id, List parametros)
```

id: Identificador del dominio a recuperar

parámetros: Lista de parámetros (se tomarán como String)

Para la implementación del EJB se ofrece una librería con las interfaces y clases necesarias: **sistra-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

- Acceso mediante Web Service

El BackOffice deberá implementar el Web Service especificado en el WSDL en la carpeta del proyecto **moduls\modul-sistra\moduls\webservicesClient\wsdl**.

La interfaz de acceso por Web Service emula a la interfaz de acceso por EJB de modo que la ayuda especificada en el acceso por EJB es aplicable a la interfaz de Web Service.

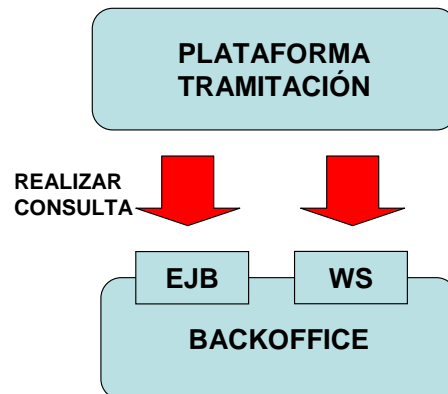
## INTEGRACIÓN BACKOFFICE

### 2.2. Trámites tipo consulta

Existe una variedad de trámite cuya finalidad es obtener una información del BackOffice y presentarla por pantalla (Ej.: resultados de notas, consultas de certificados, etc.).

Estos trámites consisten en uno o más formularios que recogen los datos para realizar la consulta y posteriormente realizan esta consulta contra el BackOffice. Como resultado se obtienen uno o más documentos de respuesta.

Estos trámites no registran ni depositan información en la Bandeja Telemática.



El acceso al backoffice puede ser:

- Acceso por EJB:

El BackOffice debe implementar un EJB que debe definirse para que utilice las siguientes interfaces *home* y *remote* :

- home: es.caib.sistra.persistence.intf.ConsultaEJBHome
- remote: es.caib.sistra.persistence.intf.ConsultaEJB

Estas interfaces definen un único método para realizar la consulta:

```
public DocumentoConsulta[] realizarConsulta(FormularioConsulta forms[])
```

Como parámetro obtenemos una lista de FormularioConsulta que contienen los formularios del trámite y como respuesta hay que devolver uno o más DocumentoConsulta .

Los documentos devueltos pueden ser de los siguientes tipos:

- Archivo: Devuelve un archivo binario
- RDS: Devuelve una referencia RDS de un documento (almacenado en el RDS)
- XML: Devuelve un XML y la referencia a una plantilla visualizadora del RDS.
- URL: Devuelve una URL (la página de respuesta mostrará esta URL)

Si no se devuelve ningún documento la plataforma indicará que no se han obtenido resultados para los datos introducidos.

Para la implementación del EJB se ofrece una librería con las interfaces y clases necesarias: **sistra-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

## INTEGRACIÓN BACKOFFICE

---

- Acceso mediante Web Service

El BackOffice deberá implementar el Web Service especificado en el WSDL en la carpeta del proyecto ***moduls\modul-sistra\moduls\webservicesClient\wsdl***.

La interfaz de acceso por Web Service emula a la interfaz de acceso por EJB de modo que la ayuda especificada en el acceso por EJB es aplicable a la interfaz de Web Service.



## INTEGRACIÓN BACKOFFICE

### 3. Repositorio de Documentos Seguro (REDOSE)

El Repositorio de Documentos Seguro es un contenedor de documentos en formato electrónico que gestiona:

- Registro de modelos y versiones de documentos: para insertar un documento en el RDS hay que indicar de que modelo/versión es el documento.
- Integridad: permite controlar la integridad del documento mediante funciones de hash
- Control de acceso a los mismos: para poder acceder a un documento es necesario disponer de una clave de acceso. Además se auditan las operaciones sobre los documentos.
- Firma digital: un documento puede estar firmado digitalmente. Se permite asociar una firma digital a un documento (verificando que la firma es válida).
- Renderización en PDF de documentos xml: permite definir que un determinado modelo de documento es estructurado (xml) y permite definir plantillas de visualización en PDF para esos documentos.

El interfaz del RDS es el siguiente:

RdsFacadeEJB	
+ insertarDocumento (DocumentoRDS documento)	: ReferenciaRDS
+ actualizarDocumento (DocumentoRDS documento)	: void
+ actualizarFichero (ReferenciaRDS ref, byte datos[])	: void
+ asociarFirmaDocumento (ReferenciaRDS refRds, Signature firma)	: void
+ crearUso (UsoRDS usoRDS)	: void
+ consultarDocumento (ReferenciaRDS refRds)	: DocumentoRDS
+ consultarDocumento (ReferenciaRDS refRds, boolean recuperarFichero)	: DocumentoRDS
+ consultarDocumentoFormateado (ReferenciaRDS refRds, String idioma)	: DocumentoRDS
+ consultarDocumentoFormateado (ReferenciaRDS refRds, String tipoPlantilla, String idioma)	: DocumentoRDS
+ formatearDocumento (DocumentoRDS documentoRDS, String modelo, int version, String tipoPlantilla, String idioma)	: DocumentoRDS
+ eliminarDocumento (ReferenciaRDS refRds)	: void
+ eliminarUso (UsoRDS usoRDS)	: void
+ eliminarUsos (String tipoUsoId, String referencia)	: void
+ listarUsos (ReferenciaRDS refRDS)	: List

Una aplicación de BackOffice podría utilizar directamente el RDS en estos supuestos:

- Generación de registros telemáticos de Entrada / Salida: insertar los documentos anexos referenciados en el asiento registral.
- Generación de avisos de expedientes en la Zona Personal: un aviso puede tener asociado documentos.
- Trámites de tipo Consulta: un trámite de tipo consulta puede devolver documentos del RDS.
- Procesamiento entradas Bandeja Telemática: en caso de que al procesar las entradas de la Bandeja Telemática se quieran guardar la referencia RDS de los documentos para poder acceder posteriormente.

En estos casos anteriormente descritos la aplicación de BackOffice puede insertar un documento en el RDS y obtendrá una **ReferenciaRDS** compuesta por un código y una clave de acceso para acceder al mismo.

ReferenciaRDS	
- codigo	: long
- clave	: String

## INTEGRACIÓN BACKOFFICE

Un documento en el RDS queda modelizado por la siguiente clase:

DocumentoRDS	
- referenciaRDS	: ReferenciaRDS
- modelo	: String
- version	: int
- titulo	: String
- estructurado	: boolean
- nif	: String
- usuarioSeycon	: String
- unidadAdministrativa	: long
- nombreFichero	: String
- extensionFichero	: String
- datosFichero	: byte[]
- firmas	: Signature[]
- codigoUbicacion	: String
- fechaRDS	: Date
- hashFichero	: String
- plantilla	: String

<b>referenciaRDS</b>	Referencia RDS del documento
<b>modelo/version</b>	Modelo/versión del documento
<b>titulo</b>	Título
<b>estructurado</b>	Indica si el documento es estructurado (XML). Este dato es calculado por el RDS en función del modelo (no es un dato modificable, solo es de consulta).
<b>nif</b>	Nif de la persona a la que pertenece el documento en el RDS
<b>usuarioSeycon</b>	Identificador usuario de la persona a la que pertenece el documento en el RDS (en caso de que el acceso sea autenticado)
<b>unidadAdministrativa</b>	Unidad Administrativa responsable del documento
<b>nombreFichero</b>	Nombre físico del fichero asociado al documento (Ej: documento1.doc)
<b>extensionFichero</b>	Extensión del fichero asociado al documento. Máximo 4 caracteres. (Ej: doc)
<b>datosFichero</b>	Datos del fichero asociado al documento
<b>firmas</b>	Firmas del documento
<b>codigoUbicacion</b>	Código de ubicación dónde se almacenará el documento. Por defecto se utiliza la por defecto del RDS.
<b>fechaRDS</b>	Fecha de introducción en el RDS. Este dato es calculado por el RDS.
<b>hashFichero</b>	Hash del fichero calculado por el RDS. Este dato es calculado por el RDS.
<b>plantilla</b>	Código plantilla a utilizar para visualizar el documento estructurado. Si no se especifica utiliza la plantilla por defecto indicada en la versión del modelo.
<b>urlVerificacion</b>	Si el documento ha sido formateado y lleva barcode de verificación indica la url de verificación (no es un dato modificable, solo es de consulta).

**Campos en rojo: obligatorios al dar de alta un documento**

**Campos en negro: opcionales al dar de alta un documento**

**Campos en azul: campos de sólo consulta un documento**

## INTEGRACIÓN BACKOFFICE

Un documento en el RDS deberá tener asociado uno o más **UsoRDS**. Un **UsoRDS** indica el uso de este documento en el sistema. Si un documento no tiene asociado ningún uso quiere decir que ese documento no se utiliza en el sistema y puede ser borrado. Los usos son generados automáticamente por la plataforma.

UsoRDS	
- tipoUso	: String
- referenciaRDS	: ReferenciaRDS
- referencia	: String

<b>tipoUso</b>	Indica que uso se hace del documento TRA: Documento de un trámite en persistencia RTE: Documento asociado a un registro de entrada RTS: Documento asociado a un registro de salida BTE: Documento asociado a Bandeja PRE: Documento asociado a un preregistro EXP: Documentos asociados a un expediente de gestión
<b>referenciaRDS</b>	Referencia al documento
<b>referencia</b>	Referencia de la aplicación al uso (depende de tipoUso). Por ejemplo, si un documento tiene un uso RTE esta referencia indicará el número de registro de entrada, sin un documento tiene un uso EXP esta referencia indicará el código de expediente

**Campos en rojo: obligatorios al dar de alta**

Campos en negro: opcionales al dar de alta

Campos en azul: campos de sólo consulta

Estos usos son generados automáticamente por la plataforma, p.e. en la creación de un registro salida (notificación) el BackOffice debe insertar todos los documentos en el RDS e invocar al registro telemático para crear la notificación. En el proceso de crear la notificación la plataforma asociará los usos correspondientes a los documentos.

El acceso al repositorio de documentos puede realizarse:

- Acceso por EJB:

La plataforma implementa un EJB que permite a los BackOffices interactuar con el RDS: `jndi = es.caib.redose.persistence.RdsFacade`.

Las clases necesarias para acceder a este módulo se encuentran en el jar: **redose-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

- Acceso por web service:

La plataforma ofrece un webservice que permitirá la inserción y consulta de documentos.

El BackOffice deberá implementar un cliente para este Web Service especificado en el WSDL en la carpeta del proyecto **moduls\modul-redose\moduls\webservices\wsdl**.

La url del Web Service es:

## INTEGRACIÓN BACKOFFICE

---

`http(s)://host:puerto/redosews/services/{version}/BackofficeFacade`

dónde {version} sera: v1, v2, etc. según el versionado del Web Service

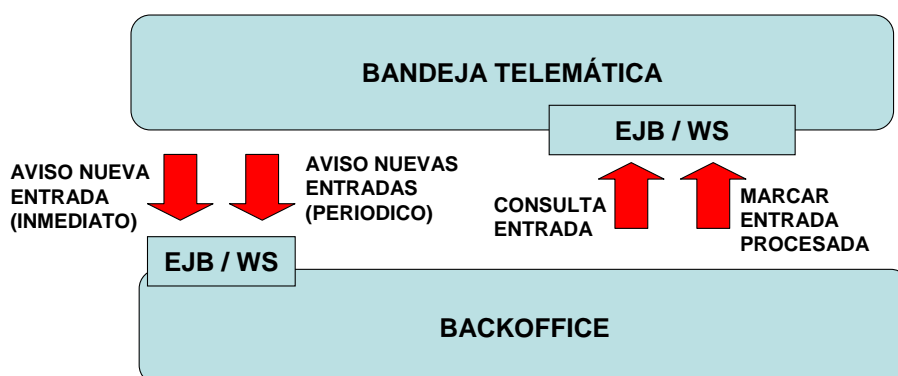
Para la serialización de las firmas en el Web Service se serializa en bytes mediante el plugin de firma (método `parseFirmaToBytes`) y se indica el formato de la firma y el contenido de la misma en bytes.

## INTEGRACIÓN BACKOFFICE

### 4. Bandeja Telemática (BANTEL)

Como resultado de la Tramitación Telemática se deposita en la Bandeja Telemática de Entrada los formularios y documentos aportados por el ciudadano. El BackOffice deberá acceder a la Bandeja Telemática de Entrada para recoger las nuevas entradas para procesarlas.

A continuación vemos las diferentes llamadas entre el BackOffice y la Bandeja Telemática:



- Aviso nuevas entradas: en la Bandeja Telemática se puede configurar (por trámite) que la Bandeja avise al BackOffice de las nuevas entradas que se van produciendo (entradas marcadas en la Bandeja como no procesadas). Hay dos tipos de avisos:
  - o Inmediato: nada más producirse la entrada se avisa al BackOffice indicando el número de entrada.
  - o Periódico: cada cierto período de tiempo se avisa al BackOffice de las nuevas entradas producidas.

Para recoger estos avisos se deberá implementar un EJB o Webservice en el BackOffice.

- Consulta entrada: una vez recibido el aviso el BackOffice puede consultar los datos de la entrada a través de un EJB o Webservice que se encuentra en la Bandeja Telemática. Estos datos contienen los formularios y documentos enviados por el ciudadano.
- Marcar entrada procesada: una vez procesada la entrada por el BackOffice se debe marcar en la Bandeja Telemática la entrada como **procesada** ya que la próxima vez que se avise de las nuevas entradas se volverá a incluir esta entrada. Si el BackOffice no puede procesar la entrada debido a un error interno (p.e. error al insertar en las tablas de BackOffice la información, etc.) deberá marcar la entrada como **procesada con error** ya que sino se reintentaría cada vez procesarla. En este caso es tarea del BackOffice controlar este tipo de errores e indicar a sus usuarios de este hecho. A través del visor de la Bandeja Telemática podrán acceder a la información de la misma si quieren procesarla manualmente.

Una cuestión importante a tener en cuenta es que es posible que una misma entrada se avise varias veces, ya que mientras siga estando como pendiente de procesar se irá notificando al backoffice. Para evitar incidencias es conveniente que el backoffice implemente un control que evite que una entrada se procese dos veces. Este control puede ser establecer como clave alternativa el número de entrada en la tabla dónde se almacenen los datos de la entrada procesada (aseguraría que no se procesase dos veces).

## INTEGRACIÓN BACKOFFICE

---

El acceso a la bandeja telemática puede realizarse:

- Acceso por EJB:

Por un lado el BackOffice debe implementar un EJB si desea recibir avisos que debe definirse para que utilice las siguientes interfaces *home* y *remote* :

- home: es.caib.bantel.persistence.intf.BteConectorFacadeHome
- remote: es.caib.bantel.persistence.intf.BteConectorFacade

Estas interfaces definen un único método para realizar el aviso:

```
public void avisoEntradas(String [] entradas)
```

Por otra parte la plataforma implementa un EJB que permite a los BackOffices recoger las entradas: jndi = es.caib.bantel.persistence.BteFacade.

Las clases necesarias para acceder a este módulo se encuentran en el jar: **bantel-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

- Acceso por web service:

Del mismo modo que en el acceso por EJB la plataforma define el WSDL del Web Service que deben implementar los BackOffices para recibir avisos en la carpeta: **\moduls\modul-bantel\moduls\webservicesClient\wsdl**

Para que los BackOffices puedan acceder mediante Web Service para recoger las entradas la plataforma implementa un Web Service que sigue el WSDL definido en la carpeta: **\moduls\modul-bantel\moduls\webservices\wsdl**

La url del Web Service es:

http(s)://host:puerto/bantelws/services/{version}/BackofficeFacade

dónde {version} sera: v1, v2, etc. según el versionado del Web Service

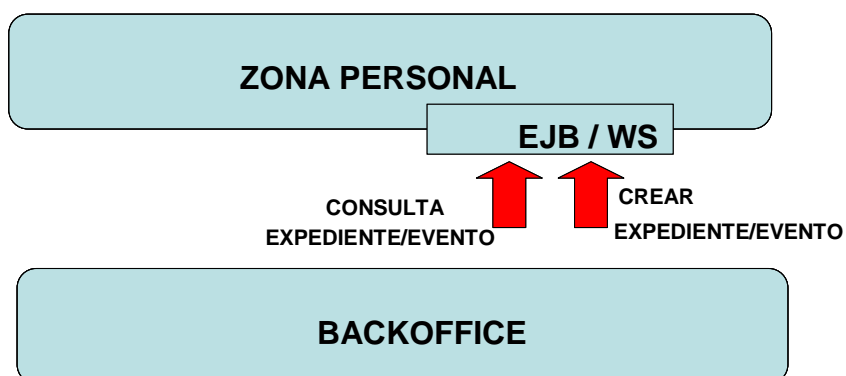
Hay una diferencia de funcionamiento en el acceso por Web Service referente al acceso de las entradas respecto al funcionamiento mediante acceso por EJB. Esta diferencia radica en una mejora de la seguridad en el acceso a las entradas ya que se requiere el uso de una clave de acceso que es suministrada en el proceso de aviso de entradas.

## INTEGRACIÓN BACKOFFICE

### 5. Zona personal (ZONAPER)

Se provee a las aplicaciones de BackOffice un interfaz para crear expedientes en la zona personal de un ciudadano. Se podrán asociar distintos avisos, llamados eventos, a un expediente, y a su vez cada uno de estos avisos podrá tener documentos asociados para que el ciudadano consulte cualquier aspecto de sus tramitaciones.

A continuación vemos las llamadas que las aplicaciones de BackOffice podrían realizar para interactuar con la zona de expedientes de un ciudadano en la zona personal:



El acceso a la zona personal puede realizarse:

- Acceso por EJB:

La plataforma implementa un EJB que permite a los BackOffices interactuar con la zona personal: `jndi = es.caib.zonaper.persistence.PadBackOfficeFacade`.

Las clases necesarias para acceder a este módulo se encuentran en el jar: **zonaper-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

- Acceso por web service:

La plataforma ofrece un webservice que permitirá la inserción de expedientes y avisos.

El BackOffice deberá implementar un cliente para este Web Service especificado en el WSDL en la carpeta del proyecto **moduls\modul-zonaper\moduls\webservices\wsdl**.

La url del Web Service es:

`http(s)://host:puerto/bantelws/services/{version}/BackofficeFacade`

dónde {version} sera: v1, v2, etc. según el versionado del Web Service

## INTEGRACIÓN BACKOFFICE

---

### 5.1. Expedientes

---

Un expediente es un apartado donde informar al usuario sobre cualquier aspecto de sus tramitaciones. Un expediente estará identificado de forma unívoca por:

- **unidad administrativa** al que pertenece
- **identificador único** de expediente dentro de dicha unidad administrativa
- **clave de acceso** al expediente que permite securizar su acceso, de forma que para cualquier operación sobre el expediente (lectura o modificación) se requiere dicha clave.

Cuando se da de alta un expediente se puede asociar a un trámite telemático mediante el **número de entrada de la BTE** que ha generado dicho trámite. De esta forma quedarán asociados dentro del expediente la solicitud inicial que lo generó.

El expediente puede generarse para un ciudadano identificado (acceso autenticado) o bien puede generarse a partir de un trámite con acceso anónimo.

A la hora de crear un expediente se indicará si se deben generar alertas de tramitación (envíos sms y email) que informen al ciudadano cuando se genere un evento o notificación sobre el expediente.

Estas alertas se deberían ser establecidos en función de los datos ofrecidos por el ciudadano al realizar el trámite inicial.

La generación automática de estas alertas tendrá la siguiente lógica:

- 1) En el propio expediente (generado por el backoffice) se puede indicar si se deben o no generar estas alertas
- 2) Si no se indica nada en el expediente, se hará caso de lo que se indique en la solicitud iniciadora del expediente (es decir, la que se indica con el número de entrada BTE al crear expediente)
- 3) Si no se indica nada en el expediente ni en la solicitud se hará caso de la configuración general de alertas de la zona personal.

### 5.2. Eventos

---

Un expediente puede llevar asociados distintos avisos relacionados con la marcha o estado del expediente. Dichos avisos son denominados "Eventos". Un evento estará identificado de manera única por el **expediente** al que pertenece y la **fecha** en que se produce, y podrá tener además las siguientes propiedades:

1. Título. Título del aviso
2. Texto. Texto descriptivo del aviso.
3. Texto SMS. En caso de que estén activadas las alertas indica el texto a utilizar en la alerta SMS (si no se establece se utilizará un mensaje por defecto).
4. Lista de documentos asociados al aviso.
5. Enlace de consulta. Enlace de consulta a cualquier URL que contiene un recurso asociado al aviso.



## INTEGRACIÓN BACKOFFICE

---

### 5.2.1. Documentos

---

Cada aviso o “evento” puede llevar asociado una lista de documentos. Cada documento se almacenará en el RDS ( Repositorio Seguro de Documentos ).

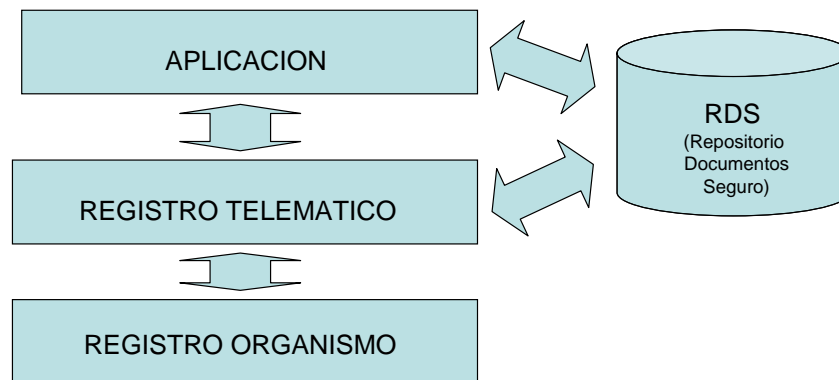
Cuando se crea un evento, se le puede asociar una lista de documentos, de una de las siguientes formas:

1. Asociar un documento ya existente en RDS. Para ello el documento deberá informada, mediante el código y la claveRDS, su referencia en el Repositorio de Documentos Seguro.
2. Asociar un documento nuevo que se creará en RDS. Para crear un documento nuevo en el RDS, se deberían informar las siguientes propiedades:
  - a. Título del documento.
  - b. Nombre del fichero del documento (con extensión)
  - c. Modelo y versión RDS.
  - d. Contenido binario del fichero del documento.

## INTEGRACIÓN BACKOFFICE

### 6. Registro Telemático (REGTEL)

En la siguiente figura se representan los diferentes elementos que intervienen:



- Aplicación: sistema de gestión que debe interactuar con el registro telemático.
- RDS (Repositorio de Documentos Seguro): contenedor de documentos en formato electrónico.
- Registro Telemático: api que implementa un interfaz al registro de entrada/salida basado en asientos registrales en formato Sicres.
- Registro Organismo: registro del organismo

## INTEGRACIÓN BACKOFFICE

El api de Registro Telemático es el siguiente:

RegistroTelematicoEJB	
+ registroSalida (ReferenciaRDS refAsiento, Map refDocs)	: ResultadoRegistroTelematico
+ registroEntrada (ReferenciaRDS refAsiento, Map refDocs)	: ResultadoRegistroTelematico
+ obtenerAcuseRecibo (String numeroRegistro)	: Date

registroSalida	Genera un registro de salida o notificación. Se pasa como parámetro la referencia RDS del asiento registral de salida y en un Map las referencias de los documentos indicados en el asiento (Key: Identificador documento en asiento - Value: ReferenciaRDS)
referenciaRDS	Genera un registro de entrada. Se pasa como parámetro la referencia RDS del asiento registral y en un Map las referencias de los documentos indicados en el asiento (Key: Identificador documento en asiento - Value: ReferenciaRDS)
obtenerAcuseRecibo	Permite obtener la fecha del acuse de recibo de una notificación.

Tanto las funciones de registro de entrada como salida del registro telemático devuelven el número de registro creado y una referencia RDS del justificante de registro (asiento registral).

ResultadoRegistroTelematico	ResultadoRegistro
- resultadoRegistro : ResultadoRegistro	* numeroRegistro : String
- referenciaRDSJustificante : ReferenciaRDS	* fechaRegistro : String

El acceso al registro telemático puede realizarse:

- Acceso por EJB:

La plataforma implementa un EJB que permite a los BackOffices interactuar con el registro telemático: `jndi = es.caib.regtel.persistence.RegistroTelematico`.

Las clases necesarias para acceder a este módulo se encuentran en el jar: **regtel-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

- Acceso por web service:

La plataforma ofrece un webservice que permitirá realizar las operaciones contra el registro telemático.

Este webservice ofrece una interfaz más simplificada que en el acceso por EJB, automatizando la creación de los asientos registrales que se realizarán de forma interna. Esta interfaz simplifica los procesos de generación de registros de entrada y salida (ver punto 6.1 y 6.2) de forma que la creación del asiento y la inserción de los documentos en el RDS las realiza de forma automatizada el web service.

El BackOffice deberá implementar un cliente para este Web Service especificado en el WSDL en la carpeta del proyecto **modules\modul-regtel\modules\webservices\wsdl**.

## INTEGRACIÓN BACKOFFICE

La url del Web Service es:

`http(s)://host:puerto/regtelws/services/{version}/BackofficeFacade`

dónde {version} sera: v1, v2, etc. según el versionado del Web Service

La interfaz de este web service es la siguiente:

registroEntrada (DatosRegistroEntrada)	Permite establecer las propiedades del registro de entrada (ver wsdl) y el webservice se encargará de generar el asiento registral, insertar los documentos y realizar el proceso de registro. Esta operación no permite la firma del registro de entrada.
prepararRegistroEntrada( DatosRegistroEntrada)	Esta operación permitirá generar un registro de entrada firmado. Para ello se establecerán las propiedades del registro de entrada (ver wsdl) y el web service se encargará de generar el asiento registral e insertar los documentos. Se devolverán la referencia del asiento registral generado para que pueda ser firmado.
registroEntradaConFirma( ReferenciaRDSAsientoRegistral, FirmaWS)	Una vez obtenido el asiento registral a partir de la función anterior se podrá generar el asiento registral firmado.
registroSalida( DatosRegistroSalida)	Permite establecer las propiedades del registro de salida (ver wsdl) y el webservice se encargará de generar el asiento registral, insertar los documentos y realizar el proceso de registro.
obtenerAcuseRecibo( numeroRegistro)	Permite obtener la fecha del acuse de recibo de una notificación.

### 6.1. Generación de registros de entrada

Normalmente los registros telemáticos de entrada serán creados por la Plataforma de Tramitación (Sistra) mediante el asistente de tramitación.

No obstante puede haber aplicaciones que necesiten generar registros telemáticos de entrada a través del interfaz que ofrece la plataforma.

A continuación se detallan los pasos a realizar por una aplicación para generar un registro telemático de entrada:

- 1) Insertar los documentos que forman parte del registro de entrada en el RDS
- 2) Generar asiento registral
- 3) Obtener firma digital del asiento registral por el usuario que va a efectuar la entrada
- 4) Insertar asiento registral y su firma en el RDS
- 5) Invocar al api de Registro Telemático pasándole la referencia RDS del asiento y de los documentos insertados

### 6.2. Generación de registros de salida (notificaciones telemáticas)

Una notificación telemática es la forma de notificar a un ciudadano de forma telemática. Una notificación siempre debe generarse dentro de un expediente.

Distinguimos dos tipos de notificaciones:

- notificación con acuse: para que el ciudadano la pueda consultar previamente debe firmar digitalmente un acuse de recibo, lo que da constancia legal de que el ciudadano ha recibido dicha notificación.

## INTEGRACIÓN BACKOFFICE

---

- notificación sin acuse: el ciudadano puede consultarla sin tener que firmar un acuse de recibo.

A continuación se detallan los pasos a realizar por una aplicación para generar un registro telemático de salida:

- 1) Generar documento aviso de notificación (ver Anexo documento de aviso notificación)
- 2) Generar documento oficio de remisión (ver Anexo documento de oficio remisión)
- 3) Generar asiento registral (ver Anexo generación asiento registral)
- 4) Insertar los documentos que forman parte del registro de salida en el RDS
- 5) Invocar al api de Registro Telemático pasándole la referencia RDS del asiento y de los documentos insertados.

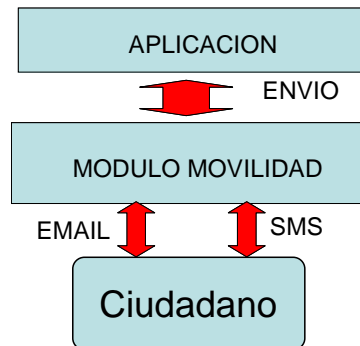
Para simplificar este proceso se proporciona un helper que realiza estos pasos:  
`es.caib.regtel.persistence.util.RegistroSalidaHelper`

Para consultar si una notificación ha sido entregada el api de registro dispone de una función para obtener el acuse de recibo.

## INTEGRACIÓN BACKOFFICE

### 7. Movilidad (MOBTRATEL)

Este módulo dará a la Plataforma la funcionalidad para realizar envíos de mensajes email y/o SMS. Para los envíos de SMS utilizará el plugin de envíos SMS del organismo.



Cualquier módulo de la Plataforma o aplicaciones de gestión podrá remotamente solicitar al módulo de movilidad la realización de un Envío. Un envío estará compuesto por n mensajes email y/o por n mensajes SMS.

El módulo de movilidad permite definir diferentes cuentas emisoras de mensajes de forma que cuando se realice el un envío a este módulo se deberá indicar con que cuenta emisora debe realizarla (p.e. permite tener diferentes remitentes de email, etc.)

Se distinguen dos tipos de envíos:

- programado: se da una fecha de programación de inicio a partir de la cual se realizará el envío
- inmediato: se procederá a realizar el envío una vez el módulo de movilidad reciba la petición.

A los envíos se les podrá establecer una fecha de caducidad de forma que si se producen errores en el envío se seguirá intentando su envío siempre que no se sobrepase dicha fecha de caducidad.

## INTEGRACIÓN BACKOFFICE

---

El api del módulo de movilidad es el siguiente:

MobTraTelFacadeEJB
+ envioMensaje (MensajeEnvio mensaje) : String

El acceso al módulo de movilidad puede realizarse:

- Acceso por EJB:

La plataforma implementa un EJB que permite a los BackOffices interactuar con el registro telemático: jndi = es.caib.mobtratel.persistence.MobTraTelFacade.

Las clases necesarias para acceder a este módulo se encuentran en el jar: **mobtratel-client.jar**. Para consultar la especificación de las funciones y tipos puede consultar el javadoc asociado.

- Acceso por web service:

La plataforma ofrece un webservice que permitirá realizar las operaciones contra el módulo de movilidad.

El BackOffice deberá implementar un cliente para este Web Service especificado en el WSDL en la carpeta del proyecto **modules\modules-mobtratel\modules\webservices\wsdl**.

La url del Web Service es:

http(s)://host:puerto/mobtratelws/services/{version}/BackofficeFacade

dónde {version} sera: v1, v2, etc. según el versionado del Web Service

## 8. ANEXO VII: GESTIÓN DE TRANSACCIONES

El diseño del módulo de integración basado en acceso por EJB para el procesamiento de las entradas tiene que tener en cuenta si esta desplegado en el mismo servidor de la plataforma o por el contrario esta desplegado en diferente servidor. Esto es debido a la gestión transaccional ya que las versiones actuales de Jboss 3.2.8 no permiten transacciones distribuidas entre diferentes servidores (en Jboss 5 se incluye un gestor de transacciones distribuidas).

Por tanto vamos a especificar los criterios de diseño a seguir. Desde el punto de vista transaccional podemos distinguir dos casos:

- Caso 1: Las operaciones del módulo de integración y de las operaciones de la plataforma se incluyen dentro de la misma transacción.
- Caso 2: Las operaciones del módulo de integración y de las operaciones de la plataforma se incluyen en transacciones distintas.

El caso 1 sólo será factible cuando se cumplan estas dos condiciones:

- el módulo de integración este instalado en el mismo servidor de la plataforma
- se configure el módulo de integración para que participe en la misma transacción que la plataforma (configuración adecuada de las transacciones de los ejbs, uso de un datasource XA, etc.)

El caso 1 **sólo** será utilizado en casos muy concretos y cuya funcionalidad así lo requiere, dando preferencia a las implementaciones con el caso 2.

Así el módulo de integración deberá tener en cuenta en el caso 2 que las operaciones con la plataforma se ejecutan en transacciones distintas por tanto deberá organizar su código y su gestión de transacciones bajo esta perspectiva. Por ejemplo, si un proceso de integración debe realizar las siguientes tareas:

- actualizar su BBDD
- generar expediente/aviso/notificación
- marcar entrada como procesada

Deberá tener en cuenta que cada una de estas operaciones se ejecuta con una transacción distinta y por tanto debe preveer su ejecución en 3 fases distintas, de forma que si falla una fase se intente en el próximo aviso de la bandeja a partir de la fase en la que se ha quedado pendiente.