



**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

# Diseño Técnico SISTRA2

Diciembre 2017

**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
21/12/2017	Indra	v1.0	Diseño Técnico Sistra2 v0
08/01/2018	Indra	v1.1	Errata capa negocio SISTRAHELP
12/02/2018	Indra	v1.2	Integración plugins externos
19/02/2018	Indra	v1.3	Resolución mínima pantallas
19/06/2018	Indra	v1.4	Incorporación de API REST de SISTRAGES e información relativa a roles y permisos
25/06/2018	Indra	v1.5	Incorporación de API REST de SISTRAMIT
06/07/2018	Indra	V1.6	Redefinición permisos área
17/07/2018	Indra	V1.7	Revisión dudas API REST y añadir sección de contextos por módulo.
18/07/2018	Indra	V1.8	Se indican particularidades a nivel de BBDD. Se indica expresamente los detalles de promoción de dominios.
25/07/2018	Indra	V1.9	Se indican detalles para la generación automática del schema.
25/07/2018	Indra	V1.10	Se modifican roles para API.
30/07/2018	Indra	V1.11	Se corrige errata en API Interna STG.

Revisado por		
Nombre	Data	Àrea, departamento o empresa

Aprobado por		
Nombre	Data	Àrea, departamento o empresa

Lista de distribución		
Nombre	Àrea, departamento o empresa	Correo electrónico

## Índex

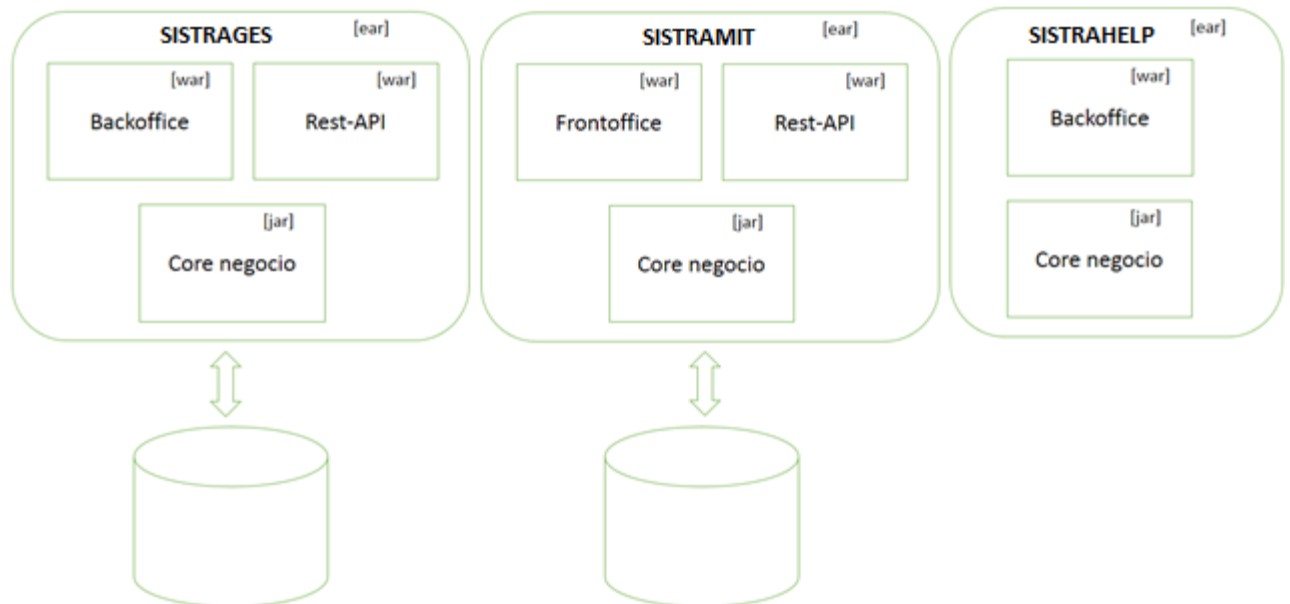
<b>Control de versiones del documento .....</b>	<b>2</b>
<b>1. Introducción .....</b>	<b>5</b>
<b>2. Arquitectura tecnológica.....</b>	<b>6</b>
<b>3. Integración con plugins externos.....</b>	<b>8</b>
<b>4. Esquemas de BBDD.....</b>	<b>9</b>
<b>5. Arquetipo de la aplicación .....</b>	<b>10</b>
<b>6. APIs integración .....</b>	<b>12</b>
6.1. API SISTRAGES.....	12
6.2. API SISTRAMIT .....	14
<b>7. Roles .....</b>	<b>16</b>
<b>8. Promoción trámites y dominios .....</b>	<b>18</b>
8.1. Exportación versión trámite .....	18
8.2. Importación versión trámite.....	18
8.3. Exportación dominio.....	20
8.4. Importación dominio .....	20
<b>9. Contextos aplicación .....</b>	<b>21</b>
<b>10. Infraestructura de soporte.....</b>	<b>21</b>

## 1. Introducción

El objetivo de este documento es definir el diseño técnico de la aplicación de SISTRA2, indicando cuál es su arquitectura tecnológica y su infraestructura de soporte.

## 2. Arquitectura tecnológica

En la arquitectura de la aplicación distinguen 3 módulos diferenciados:



- Gestor de Trámite (SISTRAGES): implementa un backoffice con la consola de administración y de desarrollo de trámites.
- Asistente de Tramitación (SISTRAMIT): implementa un frontoffice con el asistente de tramitación y gestiona la persistencia de los trámites electrónicos en curso hasta que se registren. Se comunica con el SISTRAGES a través de su API para obtener la configuración del sistema y la lógica de los trámites.
- Helpdesk (SISTRAHELP): implementa un backoffice con el Helpdesk que permita resolver las incidencias de tramitación de los ciudadanos.

Por tanto, se genera un EAR para cada módulo: SISTRAGES, SISTRAMIT y SISTRAHELP. De esta forma dependiendo de la carga de trabajo que pueda tener cada módulo, se puede balancear la carga para tener alta disponibilidad. Se prevé que no todos los módulos van a tener la misma carga de trabajo, sobre todo en producción, donde el SISTRAMIT va a tener con diferencia la mayor carga de trabajo, con lo que puede realizar un escalado añadiendo más instancias. Además, la separación en varios EARs permitirá realizar despliegues de los módulos de forma separada.

Las tecnologías usadas en cada módulo son las siguientes:

SISTRAGES	Backoffice	JSF2.2 / PrimeFaces 6.1 Spring 4.3
	Rest-API	Jersey 2.6 Spring 4.3
	Negocio	EJB3 Spring 4.3 JPA2
SISTRAMIT	Frontoffice	Spring MVC 4.3 JSP JS (Zepto, Modernizr, Maskedinput, Haschange, RequireJS, Velocity, Markup,...)
	Rest-API	Jersey 2.6 Spring 4.3
	Negocio	EJB3 Spring 4.3 JPA2
SISTRAHELP	Backoffice	JSF2.2 / PrimeFaces 6.1 Spring 4.3
	Negocio	EJB3 Spring 4.3

Respecto a la resolución mínima de las pantallas de las aplicaciones de backoffice se establece el mínimo de 1280x1024 al considerarse una aplicación de escritorio.

### 3. Integración con plugins externos

La integración con plugins externos se realizará mediante APIs basadas en servicios, preferiblemente REST, de forma que no exista acoplamiento entre SISTRA2 y los plugins externos o componentes que ofrecen dichos servicios.

Básicamente se pueden distinguir 2 tipos de plugins:

- Plugins que implican exclusivamente comunicación entre aplicaciones (p.e. 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.e. 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.

El análisis de estos plugins se realiza en el documento anexo “SISTRA2\_plugins”.



## 4. Esquemas de BBDD

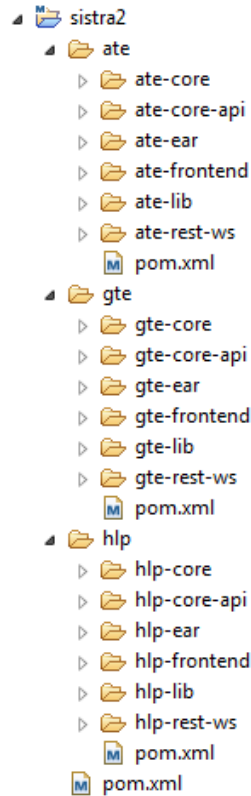
Los módulos que requerirán esquemas de BBDD serán:

- Gestor de Trámites (SISTRAGES): contiene las tablas de configuración del sistema y de la configuración y lógica de los trámites.
- Asistente de Tramitación (SISTRAMIT): contiene las tablas de persistencia de los trámites en curso por los ciudadanos y el registro de eventos de auditoría.

El módulo de Helpdesk (SISTRAHELP) en principio no requiere esquema de BBDD propio, ya que obtiene los datos necesarios a través del API de otros módulos (eventos auditoría del SISTRAMIT, estado de un pago a través del gestor de pagos correspondiente, etc.)

## 5. Arquetipo de la aplicación

El arquetipo de la aplicación está construido con Maven y dispone de un pom raíz que genera los 3 módulos de la aplicación (SISTRAGES, SISTRAMIT y SISTRAHELP).



Para cada uno de ellos se distingue la siguiente estructura de módulos:

- xxx-core-api: contiene el api interno de la capa negocio de la aplicación, es decir, el interfaz entre la capa web (frontal web, servicios web rest/soap, etc.) y la capa de negocio. Básicamente se definen 3 packages:
  - services: interfaz de servicios ofrecidos
  - model: clases de modelo que se usan en los services para el intercambio de información
  - exception: lista de errores que puede generar la capa de negocio
- xxx-core: contiene la implementación de la capa de negocio. Distinguimos los siguientes packages:
  - ejb: capa EJB3 de implementa el facade a los services de negocio (implementa los interfaces de xxx-core-api). Estos servicios invocarían a los services definidos en Spring.
  - services: services de negocio en Spring (implementa los interfaces de xxx-core-api).
    - Distingue los siguientes packages

- component: componentes reusables dentro de la capa de negocio (services internos a la capa de negocio).
- repository: capa de acceso a la BBDD mediante JPA
  - interceptor: capa AOP que centraliza la auditoría y generación de logs
- xxx-frontend: interfaz web visual de la aplicación. Depende del módulo se emplea JSF/Primefaces (SISTRAGES y SISTRHELP) o Spring MVC/JSP (SISTRAMIT).
- xxx-rest-ws: interfaz web con servicios web tipo rest con Jersey. Se usará Swagger para la documentación del API. Si hiciese falta p.e. definir una capa SOAP con Apache CXF, se puede definir otro módulo (xxx-soap-ws). Dependiendo del uso de este api de servicios, si se va a consumir de forma externa a SISTRA2 habrá que definir versionado de estos servicios.
- xxx-ear / xxx-lib: módulos auxiliares que permiten el empaquetamiento de los ears/wars.

## 6. APIs integració

A nivel general se distinguen 2 tipos de APIs: una interna entre los módulos de SISTRA2 y otra para comunicarse con componentes externos. Ambas APIs estarán securizadas mediante un role de acceso específico (ver apartado de roles).

El API interna no requerirá versionado, ya que todos los módulos de SISTRA2 se evolucionarán dentro de una misma versión, por lo que no tiene sentido.

El API externa en cambio sí que tiene que disponer de versionado, para que los componentes externos que lo utilicen puedan ir evolucionando de una forma ordenada conforme se produzcan cambios en este API.

### 6.1. API SISTRAGES

#### API Interna

Usada por resto de módulos de SISTRA2 para recuperar opciones de configuración y definición. Será utilizada por SISTRAMIT (STT) para recuperar la configuración y definición de los trámites. Será usada por SISTRHELP para recuperar la definición de entidad (permisos y roles).

Se distinguen los siguientes servicios:

Uri	Servicio	Parámetros	Resultado	Módulo invocante
/rest/interna/configuracionGlobal	Obtiene configuración global	No tiene	Propiedades globales y plugins globales	STT / STH
/rest/interna/entidad/{id}	Obtiene configuración entidad	id: id entidad	Configuración entidad y plugins entidad	STT / STH
/rest/interna/tramite/{idTramite}/{version}/{idioma}	Obtiene definición trámite	idTramite: id trámite version: versión trámite idioma: idioma	Definición del trámite y configuración de los dominios utilizados en el trámite	STT
/rest/interna/entidad/{id}/avisos	Obtiene avisos entidad	id: id entidad	Obtiene avisos activos de la entidad	STT
/rest/interna/dominioFuenteDatos/{idDominio}	Consultar dominio de Fuente de Datos	idDominio: id dominio parametros: json con los parámetros del dominio (pasado por POST)	Resuelve un dominio de Fuente de Datos	STT
/rest/interna/entidad/{id}/permisosHelpdesk	Consulta de los permisos de acceso al Helpdesk para una entidad. Usado desde Helpdesk para consultar permisos de acceso.	id: id entidad	Permisos de acceso al Helpdesk.	STH



## API Externa

Servicios usados desde otras aplicaciones para consultar datos de definición de trámites (p.e. recuperar trámites electrónicos disponibles desde el Catalogo de procedimientos y servicios, para poder establecer un selector).

Uri	Servicio	Parámetros	Resultado
/rest/externa/ tramite	Recupera trámites y versiones asociadas	Filtro: <ul style="list-style-type: none"> <li>• Versiones activas: si filtra sólo versiones activas</li> <li>• Área: descripción área</li> <li>• Trámite: descripción trámite</li> </ul>	Trámites con su lista de versiones asociadas.

## 6.2. API SISTRAMIT

### API Interna

Será utilizada por SISTRAGES para solicitar invalidaciones de datos (configuración, trámites, dominios...) y por SISTRAHELP para recuperar los datos de persistencia y log de errores.

Uri	Servicio	Parámetros	Resultado	Módulo invocante
/rest/interna/invalidacion	Solicita una invalidación de datos para forzar el refresco de caché	Invalidación: <ul style="list-style-type: none"> <li>• Tipo: definición versión tramite (T) / datos dominio (D) / configuración entidad (E) / configuración global (C)</li> <li>• Identificador: según el tipo de invalidación indica el elemento a invalidar (versión trámite, dominio o entidad)</li> </ul>	No retorna nada	STG
/rest/interna/auditoria/evento	Solicita recuperación auditoría eventos. La auditoría de eventos permite recuperar las acciones y errores asociadas a una sesión de tramitación.	Filtro áreas: lista de áreas sobre las que el usuario tiene permiso. Filtro paginación: <ul style="list-style-type: none"> <li>• página: nº página</li> <li>• tamaño: tamaño página</li> </ul> Filtro búsqueda: <ul style="list-style-type: none"> <li>• Id sesión tramitación</li> <li>• DNI</li> <li>• Trámite/Versión</li> <li>• Código procedimiento</li> <li>• Fecha inicio/fin</li> <li>• Tipo evento:               <ul style="list-style-type: none"> <li>○ Inicio de trámite</li> <li>○ Carga desde persistencia</li> <li>○ Pago electrónico</li> <li>○ Registro trámite</li> <li>○ Fin trámite</li> <li>○ Borrado trámite</li> </ul> </li> </ul>	Lista de eventos	STH

		<ul style="list-style-type: none"> <li>○ Error</li> <li>○ Debug</li> </ul>		
/rest/interna/auditoria/tramite	Solicita recuperación sesiones de tramitación. Una sesión de tramitación hace referencia a un trámite iniciado a través del asistente de tramitación (STT).	Filtro áreas: lista de áreas sobre las que el usuario tiene permiso. Filtro paginación: <ul style="list-style-type: none"> <li>• página: nº página</li> <li>• tamaño: tamaño página</li> </ul> Filtro búsqueda: <ul style="list-style-type: none"> <li>• Dato formulario</li> <li>• Trámite/Versión</li> <li>• Código procedimiento</li> <li>• DNI</li> <li>• Fecha inicio/fin</li> </ul>	Lista de sesiones de tramitación	STH
/rest/interna/auditoria/tramite/{idTramitacion}/fichero/{idFichero}	Solicita recuperación de un fichero de una sesión de tramitación. Mediante esta operación se puede consultar el contenido de un formulario, anexo, pago, etc. asociado a una sesión de tramitación.	idTramitacion: Id sesión tramitación idFichero: Id fichero Filtro áreas: lista de áreas sobre las que el usuario tiene permiso.	Contenido fichero	STH
/rest/interna/auditoria/pago	Solicita recuperación lista de pagos. Mediante esta operación se permite consultar la lista de pagos iniciados desde el asistente de tramitación.	Filtro áreas: lista de áreas sobre las que el usuario tiene permiso. Filtro paginación: <ul style="list-style-type: none"> <li>• página: nº página</li> <li>• tamaño: tamaño página</li> </ul> Filtro búsqueda: <ul style="list-style-type: none"> <li>• Tipo acceso (autenticado/no autenticado)</li> <li>• Id sesión tramitación</li> <li>• DNI</li> <li>• Fecha inicio/fin</li> </ul>	Lista de pagos	STH
/rest/interna/auditoria/pago/{id}	Solicita detalle de un pago. Mediante esta operación se permite consultar el estado de un pago iniciado desde un trámite en el asistente de tramitación	Id: id pago Filtro áreas: lista de áreas sobre las que el usuario tiene permiso.	Detalle de un pago: <ul style="list-style-type: none"> <li>• Información asistente</li> <li>• Información pasarela</li> </ul>	STH

### API Externa

Será utilizada por componentes externos (Carpeta Ciudadana, Cuadro de Mando...) para obtener datos de tramitación.

Uri	Servicio	Parámetros	Resultado
/rest/externa/tramite	Solicita recuperación trámites en persistencia	Filtro búsqueda: <ul style="list-style-type: none"> <li>• DNI</li> <li>• Fecha inicio/fin</li> </ul>	Lista de trámite en persistencia para un ciudadano.
/rest/externa/evento	Solicita recuperación eventos para ser procesados por el cuadro de mando	Filtro búsqueda: <ul style="list-style-type: none"> <li>• Fecha inicio/fin</li> </ul>	Lista de eventos.

## 7. Roles

Existen 2 niveles de control de seguridad:

- Nivel “grueso”: roles genéricos de la aplicación por los que están protegidos los EJBs.
- Nivel “fino”: roles que se pueden dar de alta en la aplicación que permiten especificar la seguridad por roles definidos dinámicamente. La gestión de estos roles dinámicos ya depende de cómo se quiera administrarlos.

A nivel general se distinguen los siguientes roles en la plataforma:

- STG\_SUP: Rol de superadministrador para administrar la configuración global de la plataforma.
- STG\_ADM: Rol de administrador de entidad, tendrá acceso tanto a la configuración de la entidad como al resto de elementos de una entidad. Para ser administrador de una entidad además de este rol necesitará el rol específico asignado en la configuración de una entidad para el administrador.
- STG\_DES: Rol de desarrollador de entidad. Para ser efectivo, además de este rol se deberán aplicar mediante la aplicación permisos sobre las áreas que se deseen (bien mediante un rol o directamente al usuario). Los permisos a aplicar sobre las áreas son:
  - Administrador de área:
    - Entorno Desarrollo:
      - Trámites: mantenimiento completo
      - Versiones de trámite: mantenimiento completo
      - Dominios/Fuentes datos: mantenimiento completo de dominios/fuentes datos de área
      - Promoción versiones trámite: permitido (ver detalles en punto Promoción)
    - Entorno Preproducción / Producción:
      - Trámites: consulta
      - Versiones de trámite: consulta y mantenimiento de control de acceso
      - Dominios/Fuentes datos: Modificación dominios/fuente datos de área. Consulta dominios/fuente datos de entidad.
      - Promoción de trámite: permitido (ver detalles en punto Promoción)
  - Desarrollador de área:
    - Entorno Desarrollo:
      - Trámites: mantenimiento completo
      - Versiones de trámite: mantenimiento completo



- Dominios/Fuentes datos: mantenimiento completo de dominios/fuentes datos de área
- Promoción versiones trámite: permitido (ver detalles en punto Promoción)
- Entorno Preproducción / Producción:
  - Trámites: consulta
  - Versiones de trámite: consulta
  - Dominios/Fuentes datos: consulta
  - Promoción versiones trámite: no permitido
- Consulta: tiene acceso de consulta a los elementos del área
- Helpdesk: puede resolver incidencias con el módulo de Helpdesk sobre trámites de esa área.
- STG\_API\_INT: Rol de acceso al API interno Rest de SISTRAGES (es un rol para asignar a aplicaciones).
- STG\_API\_EXT: Rol de acceso al API externo Rest de SISTRAGES (es un rol para asignar a aplicaciones).
- STT\_API\_INT: Rol de acceso al API interno Rest de SISTRAMIT (es un rol para asignar a aplicaciones).
- STT\_API\_EXT: Rol de acceso al API externo Rest de SISTRAMIT (es un rol para asignar a aplicaciones).

## 8. Promoción trámites y dominios

Para poder controlar la trazabilidad de promoción de trámites entre entornos se permitirá importar versiones de trámite entre estos entornos:

- Sobre Desarrollo desde entornos de Preproducción y Producción, por si se quiere restablecer en Desarrollo una release anterior.
- Sobre Preproducción desde Desarrollo
- Sobre Producción desde Preproducción

En caso de importar una versión de trámite con una release inferior a la que exista en un entorno, se borrarán del historial las entradas de releases superiores.

Además de la promoción de trámites se permitirá la promoción de dominios de forma individual.

### 8.1. Exportación versión trámite

La exportación de una versión de trámite generará un ZIP con toda la información necesaria para importar el trámite en el entorno destino.

En la exportación se tendrán en cuenta los siguientes aspectos:

- Para exportar un trámite desde el entorno de Desarrollo no debe estar bloqueado para cambios.
- Para Áreas y Trámites se exportará Id y Descripción para dar de alta si fuera necesario
- Versión: la propiedad de habilitar *debug* se exportará a deshabilitado
- Dominios: se exportarán toda la configuración del dominio, incluido fuente de datos si es un dominio de este tipo, junto con los valores de la fuente de datos.
- Ficheros: incluir ficheros físicos para plantilla anexos, plantilla paso información y ficheros campo formulario de tipo imagen
- Formularios: plantillas visualización PDF (con ficheros asociados) y formateadores usados. Queda pendiente tratamiento de gestor de formularios externos (de momento se generará error)

### 8.2. Importación versión trámite

Al importar un trámite se mostrará en una pantalla el resumen de la importación con las acciones que se llevarán a cabo sobre cada elemento que forma parte de la versión del trámite y los detalles que se deben revisar sobre cada uno de ellos:

- Se marcará como Error el elemento que se detecte que no se pueda importar
- Se marcará como Warning si sobre algún elemento si hay que realizar una acción y se debe revisar. Una vez revisado, se pasará a Correcto.
- Se marcará a Info si no se debe realizar nada o bien se debe realizar alguna acción automática.

Según el tipo de elemento en los elementos se abrirá un ventana popup con los detalles a revisar, comparando valores actuales del entorno destino en caso de que exista dicho elemento.

Se deberán revisar todos los *Warnings* y pasarlos a revisados para poder importar.

Se tendrán en cuenta los siguientes aspectos:

- Área: si hay que dar el área de alta de forma automática sólo podrá importar la versión un Administrador de Entidad. En caso de existir el área, se generará un *Warning* si no coincide la descripción existente con la que se pretende importar.
- Trámite: en caso de no existir el trámite se crearía automáticamente. En caso de existir se generará un *Warning* si no coincide la descripción para que se revise.
- Versión:
  - Pagos: se indicará si existen pagos que están simulados, para ver si se mantiene o no
  - Registrar: se revisarán los datos de registro (Oficina / Libro)
- Dominio:
  - No se podrá importar en caso de que no existan los dominios globales requeridos en el entorno destino. En este caso para poder importar una versión de trámite previamente el Superadministrador tendrá que crear los dominios globales requeridos.
  - Un Desarrollador o Administrador de área no podrá importar un trámite en caso de que no existan los dominios de entidad requeridos o bien existan pero tengan estructura diferente (p.e. cambio de parámetros de consulta, cambio de estructura de campos de la fuente de datos, etc., es decir, no se considerara que tiene cambios en la estructura si cambia el JNDI, SQL, etc.). Un Superadministrador podrá revisar los dominios de entidad requeridos en caso de que tengan que revisarse.
  - Según el tipo de dominio habrá que revisar lo siguiente:
    - Tipo:
      - BD: Revisar JNDI Datasource / SQL / Parámetros
      - Remoto: Revisar URL / Parámetros
      - Lista fija: Revisar valores
      - Fuente datos: Revisar Fuente datos destino (ID / Estructura / Valores) / SQL / Parámetros
    - Se mostrará un enlace para indicar que otras versiones de trámite usan dicho dominio, para evaluar el impacto si se actualiza un dominio (p.e. su SQL o estructura)

- Formateadores: en el caso de que no existan los formateadores requeridos no se permitirá importar.

### 8.3. Exportación dominio

La exportación de un dominio generará un ZIP con toda la información necesaria para importar en el entorno destino.

En la exportación de un dominio se exportará toda la configuración del dominio, incluido fuente de datos si es un dominio de este tipo, junto con los valores de la fuente de datos.

### 8.4. Importación dominio

Al importar un dominio se mostrará en una pantalla el resumen de la importación con los detalles que se deben revisar.

Se tendrán en cuenta los siguientes aspectos:

- Un dominio global deberá ser importado por el Superadministrador, un dominio de entidad deberá ser importado por el Administrador de entidad y un dominio de área deberá ser importado por el Administrador de entidad o un Administrador de área (Desarrollador con permiso de Administrador de área sobre dicha área).
- Según el tipo de dominio habrá que revisar lo siguiente:
  - Tipo:
    - BD: Revisar JNDI Datasource / SQL / Parámetros
    - Remoto: Revisar URL / Parámetros
    - Lista fija: Revisar valores
    - Fuente datos: Revisar Fuente datos destino (ID / Estructura / Valores) / SQL / Parámetros
  - Se mostrará un enlace para indicar que otras versiones de trámite usan dicho dominio, para evaluar el impacto si se actualiza un dominio (p.e. su SQL o estructura)

## 9. Contextos aplicació

Se distinguen los siguientes contextos para cada módulo:

Módulo	Contexto	Descripción
SISTRAGES	/sistragesback	Consola centralizada de configuración y definición de trámites.
	/sistrages/api/rest/interna	API Rest interna que ofrece servicios a ser consumidos por resto de módulos de SISTRA2 (SISTRAMIT y SISTRAHLP).
	/sistrages/api/rest/externa	API Rest pública que ofrece servicios a aplicaciones terceras.
SISTRAMIT	/sistramitfront	Asistente de tramitación
	/sistrages/api/rest/interna	API Rest interna que ofrece servicios a ser consumidos por resto de módulos de SISTRA2 (SISTRAGES y SISTRAHLP).
	/sistrages/api/rest/externa	API Rest pública que ofrece servicios a aplicaciones terceras.
SISTRAHLP	/sistrahlp	Herramienta de Helpdesk para soporte a incidencias.

## 10. Infraestructura de soporte

La aplicación tendrá como infraestructura de soporte JBoss EAP 7 (Wildfly 10) y JDK 8.

A nivel de BBDD se suministrarán los scripts para Oracle (probados con Oracle 12 y con las particularidades a nivel físico establecidas por los estándares de la CAIB) pero al usar JPA podrán utilizarse otros gestores de BBDD como Postgresql.

Para la generación de scripts de forma automática a partir del modelo JPA se generarán a través de la tarea Maven (los scripts se generarán en el directorio: */target/generated-schema*). La configuración de la BD a utilizar se encuentra en el pom principal con las propiedades:

```
schema.generate.databaseProductName  
schema.generate.databaseVersion
```

Estas propiedades hacen referencia a las propiedades de Hibernate para la configuración de la BD

([https://docs.jboss.org/hibernate/orm/5.1/javadocs/org/hibernate/cfg/AvailableSettings.html#HBM2DDL\\_DB\\_NAME](https://docs.jboss.org/hibernate/orm/5.1/javadocs/org/hibernate/cfg/AvailableSettings.html#HBM2DDL_DB_NAME))



Para Oracle se puede usar esta configuración:

```
<schema.generate.databaseProductName>Oracle</schema.generate.databaseProductName>  
<schema.generate.databaseVersion>12</schema.generate.databaseVersion>
```

Para Postgresql se puede usar esta configuración:

```
<schema.generate.databaseProductName>PostgreSQL</schema.generate.databaseProductNa  
me>  
<schema.generate.databaseVersion>9</schema.generate.databaseVersion>
```