



**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

Estimación SISTRA 2 v0

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 HLP

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



**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

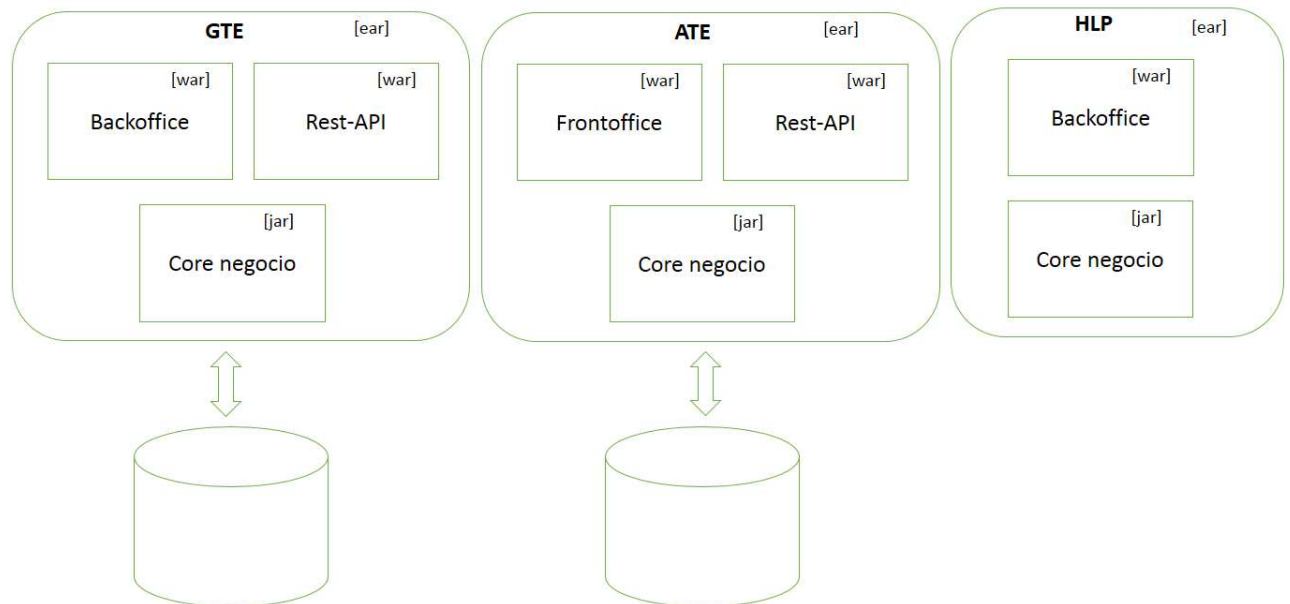
.....	1
Control de versiones del documento	2
1. Introducción	4
2. Arquitectura tecnológica.....	5
3. Esquemas de BBDD.....	7
4. Arquetipo de la aplicación	8
5. Infraestructura de soporte	10

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 (GTE): implementa un backoffice con la consola de administración y de desarrollo de trámites.
- Asistente de Tramitación (ATE): 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 GTE a través de su API para obtener la configuración del sistema y la lógica de los trámites.
- Helpdesk (HLP): 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: GTE, ATE y HLP. 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 ATE 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:

GTE	Backoffice	JSF2.2 / PrimeFaces 6.1 Spring 4.3
	Rest-API	Jersey 2.6 Spring 4.3
	Negocio	EJB3 Spring 4.3 JPA2
ATE	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
HLP	Backoffice	JSF2.2 / PrimeFaces 6.1 Spring 4.3
	Negocio	EJB3 Spring 4.3

3. Esquemas de BBDD

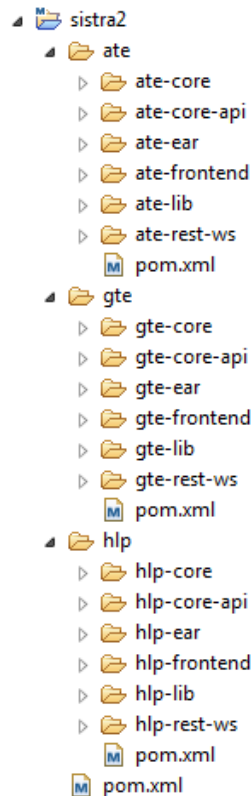
Los módulos que requerirán esquemas de BBDD serán:

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

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

4. 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 (GTE, ATE y HLP).



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 (GTE y HLP) o Spring MVC/JSP (ATE).
- 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.

5. Infraestructura de soporte

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