



PROVINCIA DE SANTA FE  
CFI  
INFORME FINAL  
MARZO 2018

TECSO COOPERATIVA DE TRABAJO LTDA.

ADAPTACIÓN SIAT-GPL PARA MMYCC – PARTE 1  
INFORME FINAL

## ÍNDICE GENERAL

1. INTRODUCCIÓN .....	5
2. RESPECTO DE LA ETAPA 1 .....	7
2.1 PLAN DE TRABAJO .....	7
2.2 DOCUMENTOS GENERADOS .....	8
2.3 REQUERIMIENTOS ADICIONALES .....	10
3. RESPECTO DE LA ETAPA 2 .....	13
3.1 PLAN DE TRABAJO .....	14
3.2 RECOMENDACIONES Y DEFINICIONES PARA REQ. MULTIAGENCIA .....	16
3.3 DESARROLLO DE REQUERIMIENTOS .....	19
3.3.1 ADAPTACIÓN PARA BASE DE DATOS ORACLE .....	19
3.3.2 INTEGRACIÓN CON CAS .....	19
3.3.3 INTEGRACIÓN CON BASE DE PERSONAS DE LA PROVINCIA .....	20
3.3.4 MAVENIZACIÓN DEL PROYECTO .....	20
3.3.5 ADAPTACIÓN SWE .....	20
3.3.6 ACTIVIDAD EXTRA .....	20
3.3.7 ADAPTACIÓN PARA MÚLTIPLES MUNICIPIOS Y COMUNAS.....	22
3.3.8 PRUEBAS AUTOMÁTICAS.....	34
3.3.9 ARTEFACTOS ENTREGADOS.....	36
4. CONCLUSIONES.....	38

## ÍNDICE DE IMÁGENES

<b>IMAGEN 1:</b> ARQUITECTURA EN CAPAS MVC .....	16
<b>IMAGEN 2:</b> ESQUEMA DE DESPLIEGUE .....	17
<b>IMAGEN 3:</b> CAPTURA DE PANTALLA MENÚ.....	21
<b>IMAGEN 4:</b> CAPTURA DE PANTALLA ADMINISTRACIÓN AGENCIA.....	27

## ÍNDICE DE ANEXOS

<b>ANEXO I: DOCUMENTO PARA PREPARACIÓN DE ENTORNOS .....</b>	<b>44</b>
<b>ANEXO II: ACTIVIDADES PARA LA IMPLEMENTACIÓN EN UN MUNICIPIO .....</b>	<b>55</b>

## 1. INTRODUCCIÓN

En el marco del proyecto denominado “ADAPTACIÓN DE SIAT-GPL PARA SU UTILIZACIÓN EN LOS MUNICIPIOS Y COMUNAS DE SANTA FE A TRAVÉS DE LA PLATAFORMA TECNOLÓGICA PROVINCIAL - PARTE 1” se presenta a continuación el informe final.

El objetivo de este informe es presentar el resultado del trabajo correspondiente al proyecto mencionado que incluye las dos primeras etapas de un proyecto más general.

El objetivo del proyecto general es contar con una versión de SIAT-GPL adaptada a las necesidades de los Municipios y Comunas (SIAT-GPL-MMyCC), que pueda ser hosteada en la infraestructura tecnológica de la Provincia y que quede a disposición de todo aquel Municipio o Comuna que lo requiera.

Las etapas incluidas en el proyecto y foco del trabajo realizado del cual se presenta el informe corresponde a:

### **Etapas 1: Relevamiento funcional y tecnológico en 2 Municipios y/o Comunas**

El objetivo de esta etapa es la indagación sobre las necesidades específicas de Municipios y Comunas de distintos tamaños y características. Se propone trabajar sobre 2 que sean representativas.

Actividades de la Fase 1:

- Relevamiento de procesos tributarios en gral.
- Relevamiento de sistemas existentes
- Lectura de ordenanzas municipales
- Generación de Informe de situación
- Generación de Informe con adaptaciones mínimas necesarias
- Generación de Informe con esfuerzo de implementación de SIAT en ambos MMyCC

## **Etapas 2: Desarrollo de SIAT-GPL-MMyCC – Parte 1**

En esta etapa se contempla el desarrollo, testing, y/o integraciones con sistemas dentro de la órbita de la plataforma tecnológica provincial.

Actividades de la Fase 2:

- F1: Adaptación para múltiples Municipios y Comunas: el sistema deberá adaptarse para que soporte un esquema multi-agencia sobre una misma instancia
- Adaptación para Base de Datos Oracle – símil API-SIAT-GPL
- Integración con CAS – símil API-SIAT-GPL
- Adaptación SWE - símil API-SIAT-GPL
- Integración con Base de Personas de la pcvia. – símil API-SIAT-GPL
- Mavenización del proyecto - símil API-SIAT-GPL

## 2. RESPECTO A LA ETAPA 1

El objetivo de esta etapa es la indagación sobre las necesidades específicas de Municipios y Comunas de distintos tamaños y características. Se propone trabajar sobre 2 que sean representativas. Como resultado de esta etapa, se obtendrá un informe conteniendo: la situación actual, adaptaciones mínimas necesarias y un análisis del esfuerzo que demandaría implementar SIAT.

En el plan de trabajo inicial se incluyeron a los siguientes municipios de la provincia de Santa Fe:

- Sunchales
- San José de la Esquina

Durante el proyecto, el Municipio de San José de la Esquina no continuó con el interés de participar del proyecto y fue reemplazado por un nuevo municipio:

- Villa Gobernador Galvez

### **2.1 Plan de trabajo etapa 1**

Al comenzar las actividades se planteó el siguiente plan de trabajo:

Se llevarán a cabo varias reuniones presenciales en el Municipio de Sunchales, y en la Comuna de San José de la Esquina respectivamente. En cada reunión se presentarán las distintas funcionalidades que posee SIAT-GPL de manera ordenada, resaltando la interacción con sistemas externos ( la que habrá con la base de datos de personas de la pvcia., las que haya con los sistemas existentes del municipio [RAFAM, PROGRAM, Catastro, Comercio si estos fuesen los casos]), y dirigidas a potenciales usuarios claves del sistema que serán determinados por los municipios.

Por cada municipio y/o comuna, en función de la lectura de las Ordenanzas vigentes, y lo trabajado en las reuniones de relevamiento; se generará un Documento de Configuración de Recurso Tributario (uno p/TGI, otro p/Drel), un Documento de Código de Barras (estructura manejada por SIAT), y un Documento de Visión (situación actual, adaptaciones mínimas necesarias, estimación de esfuerzo de implementación de SIAT-GPL).

Se generará un Documento de Visión (situación actual, adaptaciones mínimas necesarias, estimación de esfuerzo de implementación de SIAT-GPL) resumiendo las situaciones de

los municipios relevados a ser compartido con personal de la pvcia.

Respecto del seguimiento, se llevarán a cabo 2 (dos) reuniones con personal de la provincia mediante las cuales se pueda comprobar el avance, y certificar la finalización de la Etapa 1 en función de los entregables presentados.

(Nota: Con mayor detalle de los alcance de cada reunión en el documento correspondiente entregado previamente)

## **2.2 Documentos generados**

Se llevaron a cabo las actividades planificadas, a las cuales fue necesario sumar y replanificar nuevas reuniones debido al cambio de uno de los Municipios seleccionados durante el transcurso de las actividades. Esto demandó un esfuerzo adicional y un retraso en los tiempos generales del proyecto.

Incluso luego de comenzar la etapa 2 se realizaron algunas reuniones adicionales en Villa Gobernador Galvez para completar algunos temas por falta de disponibilidad previa por parte del Municipio para llevar a cabo las actividades necesarias.

Como resultado de las actividades de la primer etapa se generaron y entregaron los siguientes documentos:

### **Minutas de reunión:**

- Sunchales:
  - MMYCC-MR-Reunion-20170824-SUN-1.pdf
  - MMYCC-MR-Reunion-20170825-SUN-2.pdf
  - MMYCC-MR-Reunion-20170904-SUN-3.pdf
  - MMYCC-MR-Reunion-20170905-SUN-4.pdf
  - MMYCC-MR-Reunion-20171002-SUN-5.pdf
  - MMYCC-MR-Reunion-20171003-SUN-6.pdf
  - MMYCC-MR-Reunion-20171009-SUN-7.pdf
- San José de la Esquina:
  - MMYCC-MR-Reunion-20170828-SJE-3.pdf
  - MMYCC-MR-Reunion-20170829-SJE-4.pdf
  - MMYCC-MR-Reunion1-SJE.pdf
  - MMYCC-MR-Reunion2-SJE.pdf

- Villa Gobernador Galvez:
  - MMYCC-MR-Reunion-20170921-VGG-1.pdf
  - MMYCC-MR-Reunion-20170925-VGG-2.pdf
  - MMYCC-MR-Reunion-20170928-VGG-3.pdf
  - MMYCC-MR-Reunion-20171005-VGG-4.pdf
  - MMYCC-MR-Reunion-20171121-VGG-5.pdf
  - MMYCC-MR-Reunion-20171205-VGG-6.pdf

#### **Documentos de Configuración:**

- Sunchales:
  - MSU-SIAT-DREI-Configuración.pdf
  - MSU-SIAT-TGI-Configuración.pdf
  - MSU-SIAT-Códigos-de-Barra.pdf
  - MSU-SIAT-Migracion.pdf
- Villa Gobernador Galvez:
  - VGG-SIAT-DREI-Configuración.pdf
  - VGG-SIAT-TGI-Configuración.pdf
  - VGG-SIAT-Códigos-de-Barra.pdf
  - VGG-SIAT-Migracion.pdf

#### **Documentos de Visión:**

- MMYCC-DV-Sunchales.pdf
- MMYCC-DV-VillaGobernadorGalvez.pdf
- MMYCC-DV-Resumen.pdf
- SIAT-MMyCC-Anexos.pdf

En los documentos finales se presenta la visión general del Municipio, actividades necesarias para la implementación del sistema y una configuración inicial para los tributos principales. A su vez se presentan algunos requerimientos de modificaciones adicionales a ser previstos en la versión modificada del producto.

## 2.3 Requerimientos Adicionales

Del trabajo realizado en la etapa de relevamiento se tomaron algunos requerimientos especiales para los municipios testigos que podrían requerir modificaciones o nuevas funciones a agregar al producto antes de su implementación.

A continuación se presentan los requerimientos adicionales relevados:

### Requerimientos para Sunchales

Módulo			
ID	Descripción Breve	Prioridad	ID Req. Asociados
Definición y Administración			
R001	Configurar sellados.	M	
Gestión y Administración de Deuda			
R002	Implementación de la declaración jurada anual en el recurso DRel	A	
R003	Adaptar SIAT para que no se puedan generar convenios si el contribuyente posee otro convenio vencido	A	
R004	Adaptar SIAT para que un convenio se active al abonar la primer cuota	A	
R005	Adaptar SIAT para que las multas se expresan en UCM (Unidad Codificación Municipal)	A	
Padrones			
R006	Revisar compatibilidad de base de datos de personas de la provincia de Santa Fe para obtener el número de CUIT y Documentos de sus contribuyentes	A	
Balance			
R007	Generar los archivos de sincronización para utilizarlos en RAFAM	A	
R008	Implementar en Punta de Caja funcionalidad para cobrar con cheques de terceros	M	

R009	Implementar en Punta de Caja funcionalidad para cobrar con Tarjetas de crédito	M	
R010	Implementar en Punta de Caja funcionalidad para cobrar con Tarjetas de débito	M	
R011	Gestión de cheques de terceros	M	
R012	Gestión de cupones tarjetas de crédito	M	
R013	Gestión de cupones tarjetas de débito	M	
R014	Implementar Nuevo Código de Barra	M	

### Requerimientos para Villa Gobernador Gálvez

Módulo			
ID	Descripción Breve	Prioridad	ID Req. Asociados
Definición y Administración			
Gestión y Administración de Deuda			
R001	Adaptar SIAT para que no se puedan generar convenios si el contribuyente posee otro convenio vencido	A	
R002	Adaptar SIAT para que los convenios o planes de pago se calculen con interés directo.	A	
Padrones			
R003	Analizar e implementar compatibilidad de base de datos de personas de la provincia de Santa Fe para obtener el número de CUIT y Documentos de sus contribuyentes	A	
Balance			
R004	Generar los archivos de sincronización para utilizarlos en sistemas propio de tesorería	A	
R005	Implementar en Punta de Caja funcionalidad para cobrar con cheques de terceros	M	



R006	Gestión de cheques de terceros	M	
------	--------------------------------	---	--

**Nota:** Para mayor detalle sobre los requerimientos o los temas relevados se cuenta con la documentación mencionada en el punto anterior.

### 3. RESPECTO A LA ETAPA 2

El objetivo de esta etapa es el desarrollo, testing, y/o integraciones con sistemas dentro de la órbita de la plataforma tecnológica provincial dentro del alcance estipulado para la “parte 1” descrito a continuación.

1. Adecuación a infraestructura tecnológica de la provincia:
  1. Adaptación para Base de Datos Oracle – símil API-SIAT-GPL
  2. Integración con CAS – símil API-SIAT-GPL
  3. Adaptación SWE - símil API-SIAT-GPL
  4. Integración con Base de Personas de la pcvia. – símil API-SIAT-GPL
  5. Mavenización del proyecto - símil API-SIAT-GPL
2. Adaptación para múltiples Municipios y Comunas: el sistema deberá adaptarse para que soporte un esquema multi-agencia

Para las pruebas de integración con la infraestructura de la provincia se contó con una conexión punto a punto a la red provincial y también conexiones vpn, mediante las cuales se accedieron a los servicios de CAS y de la Base de Datos de Personas y Domicilios de la Provincia.

Respecto a quizá uno de los principales puntos de adaptación “Adaptación para múltiples Municipios y Comunas”, llamado en los documentos de trabajo como “Adaptación Multi-agencia”, se realizó una evaluación tecnológica y de infraestructura considerando las nuevas tecnologías y soluciones más innovadoras, así como algunos puntos tomados en el relevamiento de los Municipios. Con esto se presentó un documento con dos alternativas, recomendando por nuestra parte fuertemente una de ellas.

Finalmente la definición tomada desde la provincia, por parte del personal técnico consultado, fue respecto a la otra opción de las planteadas. Vale aclarar que esta implementación también es válida y por dicha razón fue presentada como alternativa, aunque consideramos importante marcar el trabajo y recomendación realizada. Quedando este trabajo a disponibilidad de la Provincia para considerar en una nueva etapa u otros proyectos similares.

### **3.1 Plan de trabajo etapa 2**

En el plan de trabajo presentado se presentaron las siguientes actividades generales con las cuales se llevó la planificación y ejecución de la etapa:

- Preparación entornos y código
- Adaptación para Base de Datos Oracle
- Integración con CAS
- Integración con Base de Personas de la provincia
- Mavenización del proyecto
- Propuesta y definición para requerimiento multiagencia
- Adaptación SWE
- Presentación 2do informe para certificación
- Adaptación para múltiples Municipios y Comunas (multiagencia)
- Preparación datos de pruebas simulando dos o tres municipios
- Generación de test automáticos
- Pruebas y correcciones puntuales sobre módulos y procesos masivos

Luego a partir de la definición de la Provincia respecto a la alternativa seleccionada para la implementación del requerimiento de adaptación para multi-agencia se agregaron las siguientes actividades llevadas a cabo desde la presentación del 2do informe de avance para la finalización del proyecto:

- Agregar columna idAgencia en todas las tablas de SIAT (aproximadamente 700 tablas)
- Crear agencia genérica 'siat' y usuario especial "superadmin" para administrar agencias
- Modificación de login para guardar información de agencia en el contexto de usuario
- Modificar las clases GenericAbstractDAO y SqlQuery para que en los inserts y update se guarde el idagencia del contexto
- Nueva opción de Administración de Agencias para Municipios y Comuna
- Interceptar los DAOs de hibernate y los del SqlQuery para las consultas 'select' para agregar filtros por idAgencia para cada tabla
- Crear tabla de excepciones para que el interceptor de sql no agregue filtros por idAgencia sobre tablas de datos maestros comunes a todas las agencias.

- Completar tabla de excepción para el interceptor de sql con tablas maestros identificadas.
- Modificar DAOs específicos por inserciones especiales (mover deuda, selección almacenada, otros)
- Modificar el sistema para que al obtener el 'filesharepath' siempre incluya el código de agencia tomado del contexto
- Modificar funcionamiento de cache para contemplar uno por agencia (paso de cache simple a mapas de cache)
- Adaptar Refrescar Cache
- Modificar el loggeo de Log4j para que agregue el código de agencia del contexto
- Modificar Adp (administrador de procesos) para acceso a datos por filtro e información de agencia
- Revisar si es necesario realizar alguna modificación sobre GRS por filtros de Agencia.
- Funcionalidad Grs Multiagencia, modificación en programación de ejecución y funciones para disponer del id de agencia para filtrar en forma manual en los sql de los reportes.
- Test Automáticos:
  - ABM de Parámetro del Sistema
  - Alta de Agencia
  - Convenio de Pago
  - Gestión de Cuenta
  - Login y Logout Con CAS
  - Modificar Domicilio de Envío
  - Modificar el Valor de un Atributo de Objeto Imponible
- Solucionar Defectos encontrados durante las pruebas. Durante el trabajo se encontraron los siguientes puntos ya solucionados:
  - Consulta de Procesos
  - Crear deuda a cuenta X de una agencia
  - Error Domicilio de Envío
  - Error en "Ver" búsqueda de contribuyente.
  - Error en la reimpression/reconfeccion de Deuda
  - Problema en procesos con mismo codproceso para distintas agencias.

### 3.2 Recomendación y definiciones para requerimiento multiagencia

Como se mencionó anteriormente uno de los requerimientos principales del proyecto respondía a la “Adaptación para múltiples Municipios y Comunas: el sistema deberá adaptarse para que soporte un esquema multi-agencia sobre una misma instancia.”

Para este requerimiento se plantearon dos alternativas con sus ventajas y desventajas y la recomendación de Tecso respecto a cual implementar.

Este resumen fue incluido ya en el segundo informe de avance y se vuelve a incluir para dejar expresados las alternativas y recomendación propuesta y la definición finalmente tomada para la cual se prosiguió con el plan de actividades mencionadas en la sección anterior.

A modo de resumen:

Siat posee un diseño en capas como se muestra la *imagen 1*.

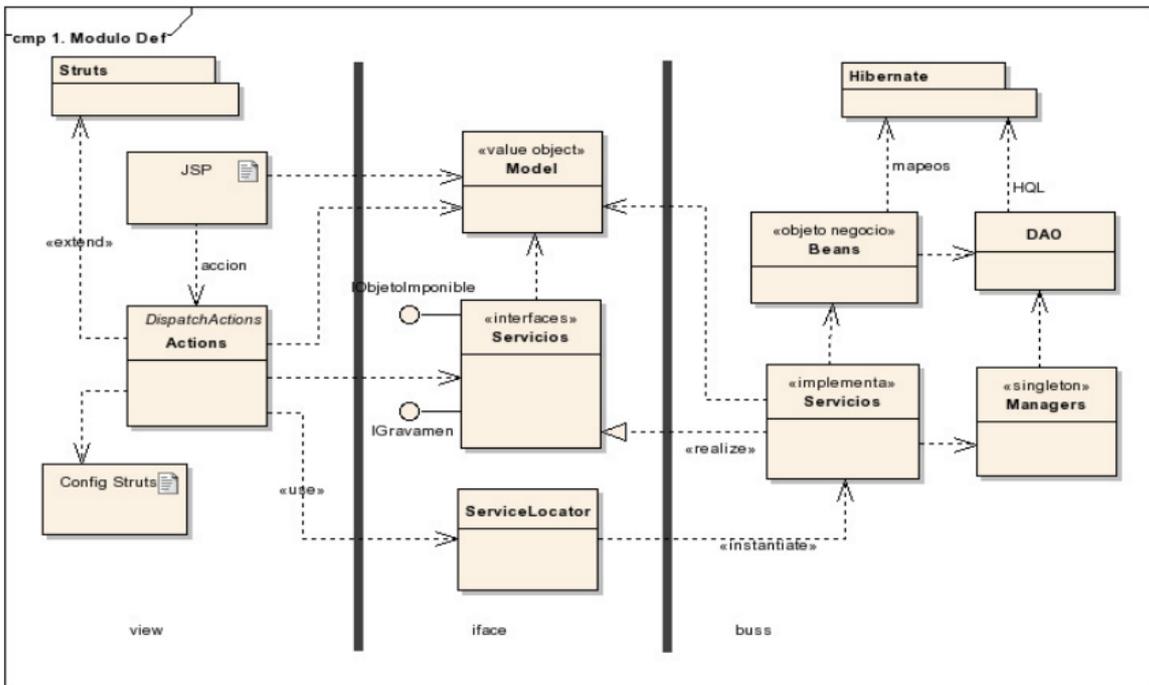
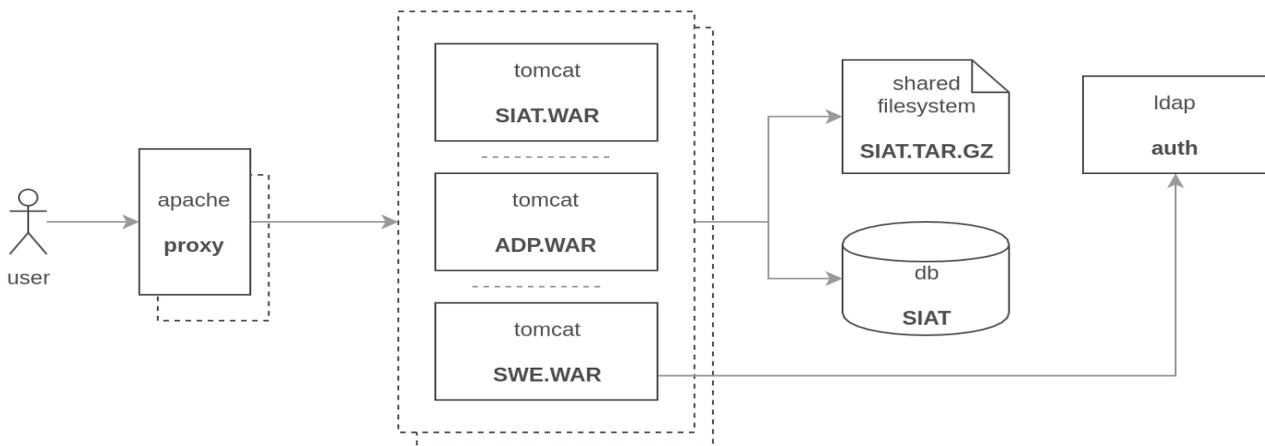


Imagen 1: Arquitectura en Capas MVC.

Por lo general Siat se despliega como se muestra en la *imagen 2*:



*Imagen 2: Esquema de Despliegue.*

En base a esto se presentaron dos alternativas:

- Una solución más bien monolítica, basada en introducir cambios estructurales y funcionales para soportar multi municipios. **(opción seleccionada)**
- Una solución tipo sandbox, basada en introducir una capa de infraestructura que permite desplegar y administrar multi instancias de siat mono municipio en *contenedores* aislados. (opción recomendada)

La versión base y liberada de SIAT-GPL está preparada para trabajar con un único municipio (agencia). Para permitir que una misma implementación de SIAT sirva para la administración tributaria de distintos municipios se debieron realizar varios cambios en el core del sistema. El principal cambio es hacer que el sistema pueda distinguir sobre qué municipio está operando.

Las alternativas presentadas para resolver el requerimiento fueron:

- “*Cortar*” al modelo de datos actual con información del municipio. Esto implica agregar información del municipio a cada tabla del modelo y realizar modificaciones en la arquitectura y la aplicación sobre el acceso de datos.
- Automatizar el instanciado de SIAT-GPL con un esquema de base de datos para cada municipio e implementar algún tipo de entrada común que rutee los request a la instancia que corresponda.

Los pro y contras de cada alternativa y la recomendación realizada por Tecso se encuentran descriptos en documentos mencionado.

### **Alternativa seleccionada**

Se realizaron reuniones con referentes técnicos de la “Secretaría de Tecnologías para la Gestión” convocados por el “Director Provincial de Desarrollo de Software”.

El 9 de Enero luego de analizar la propuesta, la “Secretaría” definió descartar la alternativa 2 propuesta por los siguientes motivos:

*“Habiendo analizado las propuestas de la empresa TECSO en relación al sistema SIAT-GPL Multiagencia para Municipios y Comunas y en cuanto a la segunda opción de utilizar plataformas Docker, si bien está en los planes futuros de la STG evaluar la factibilidad de utilizar alguna tecnología de contenedores, a la fecha este tipo de tecnologías no está en uso en la infraestructura central.*

*Además en la topología que utilizamos actualmente, con esquemas de alta disponibilidad y balanceo de carga de aplicaciones, todo está optimizado para la utilización de maquinas virtuales con aplicaciones en clusters distribuidos, con administración centralizada, donde cada departamento ya tiene sus funciones claramente definidas y la injerencia de cada Municipio y Comuna se encuadra en la administración y utilización del aplicativo.*

*A nivel creación de esquemas de datos, hay que tener en cuenta la cantidad de municipios que se desea implementar, en una de las propuesta de la empresa se crea un esquema de datos por municipio, en el caso de ser elevado este número se complica la administración de los mismos, tareas de backups, etc, por lo que se recomienda la utilización de un solo esquema de datos.”*

Del análisis participaron Raúl Ronchi (Bases de Datos), Juan Moragues (Aplicaciones) y Mónica Cabrera (Procesamiento Distribuido).

### **3.3 Desarrollo de requerimientos**

Se realizaron las adaptaciones y pruebas planificadas las cuales se describen a continuación.

Respecto a las adaptaciones generales sobre la aplicación para funcionar con el motor de base de datos Oracle, modificaciones de organización del proyecto (maven) y de integración con servicios de aplicaciones de la provincia como ser la autenticación (CAS) o la obtención de datos de persona, tal como se describió en el informe de avance anterior se realizaron los siguientes trabajos:

**Nota:** Para las pruebas de interacción con los servicios de la provincia se hizo uso de la conexión punto a punto que contamos en nuestra oficina con la red de la provincia que nos permite probar el acceso por CAS o a los WS que brindan datos de personas.

#### **3.3.1 Adaptación para Base de Datos Oracle – símil API-SIAT-GPL**

Se implementó el nuevo dialecto, y realizaron modificaciones generales necesarias por el manejo de id seriales, búsquedas paginadas, manejo de fecha hasta el día (sin hora y minutos), entre otras.

Se realizaron modificaciones específicas en procesos donde por sql se utilizan tablas temporales para guardar selecciones o resultados intermedios.

Se creó un esquema de base de datos con una configuración inicial demo para las pruebas (un único municipio) del cual se disponibilizó un dump a la provincia con la primer entrega que acompaña el 2do informe de avance.

Sobre este esquema luego se realizaron las modificaciones pertinentes a la alternativa seleccionada para el requerimiento multiagencia y junto a este informe final se disponibiliza un nuevo dump de la base de datos a la provincia con los cambios realizados.

#### **3.3.2 Integración con CAS – símil API-SIAT-GPL**

Se implementó la autenticación a través del CAS de la provincia.

Al ingresar a la aplicación si no se encuentra logeado se redirige al servicio de autenticación indicado. Al volver de la autenticación crea la sesión de usuario.

La configuración del CAS se encuentra en el archivo 'context.xml' del servidor (tomcat) de la aplicación. A continuación agregamos una configuración del mismo para visualizar las url de los servicios contra los que se probó el desarrollo realizado:

```
<Environment description="Server Name" name="cas/serverName" override="false" type="java.lang.String" value="https://mmycctest.santafe.gob.ar:443/siat" />
```

```
<Environment description="CAS Login Url" name="cas/casServerLoginUrl" override="false" type="java.lang.String" value="https://dsso.santafe.gob.ar/service-auth/login" />
```

```
<Environment description="CAS Url Prefix" name="cas/casServerUrlPrefix" override="false" type="java.lang.String" value="https://dsso.santafe.gob.ar/service-auth" />
```

```
<Environment description="CAS Url Exclude" name="cas/casUrlExcludePaths" override="false" type="java.lang.String" value="/jax-ws/*; /ws-rs/**/*; /app/par/*"/>
```

**Nota 1:** La url "<https://mmycctest.santafe.gob.ar:443/siat>" es sólo a modo de ejemplo, y dependerá del entorno en el cual se despliegue la aplicación en la infraestructura de la provincia. En las pruebas se utilizó la correspondiente al entorno de prueba utilizado.

**Nota 2:** Para las pruebas se utilizó el usuario 'usuario1' (con mismo password) .

### **3.3.3 Integración con Base de Personas de la pcvia. – símil API-SIAT-GPL**

Se realizaron las implementaciones de interfaces para consumir datos de persona y ubicación a través del WS brindado por la provincia.

Para el entorno de test se utilizan los siguientes datos configurados en parámetros de la aplicación:

- **WSDL servicio de DB de Personas:**  
<https://dapp.santafe.gob.ar/personas/ws/persona?wsdl>
- **Usuario del WS Persona:** consultasiatdev
- **Contraseña del WS Persona:** consultasiatdev

### **3.3.4 Mavenización del proyecto - símil API-SIAT-GPL**

Se reorganizó el proyecto SIAT y SWE para utilizar Maven para su compilación y empaquetado. Se reestructuro en distintos subproyectos para cumplir con buenas practicas y estándares de gestión de dependencias en Maven.

### **3.3.5 Adaptación SWE - símil API-SIAT-GPL**

Se realizaron las modificaciones a la aplicación de seguridad web del sistema para autenticar a través de CAS según se indica en el punto anterior y se realizó la reorganización del proyecto utilizando maven.

### **3.3.6 Actividad extra**

Adicionalmente y aprovechando la experiencia realizada para el proyecto de adaptación API-SIAT, se modificaron los estilos de la aplicación para utilizar las hojas de estilo de la provincia con el fin de minimizar los cambios que puedan ser necesario en un futuro sobre

la aplicación ante un cambio de imagen.

Los links utilizados son :

```

<link rel="stylesheet" href="//www.santafe.gob.ar/assets/standard/css/normalize.css" type="text/css" />
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/standard/css/1140.css" type="text/css" />
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/jquery-ui/jquery-ui-1.10.3/css/gobierno/jquery-ui-1.10.3.custom.css" type="text/css" />
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/bootstrap/bootstrap-3.0.2-dist/tables-bootstrap/css/bootstrap.min.css" type="text/css" />
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/standard/css/menu/menu.css" type="text/css" />
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/standard/css/pagination/html5-pagination.css" type="text/css" />
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/standard/css/fonts.css" type="text/css" >
<link rel="stylesheet" href="//www.santafe.gob.ar/assets/standard/css/styles-1.0.2.css" type="text/css" >
<script src="https://www.santafe.gob.ar/assets/jquery/jquery-1.11.1.min.js"></script>
<script src="https://www.santafe.gob.ar/assets/jquery-ui/jquery-ui-1.10.3/js/jquery-ui.min.js"></script>
<script src="https://www.santafe.gob.ar/assets/jquery-ui/jquery-ui-1.10.3/js/jquery.ui.datepicker-es.js"></script>
<script src="https://www.santafe.gob.ar/assets/jquery/jquery.ui.touch-punch.min.js"></script>
<script src="https://www.santafe.gob.ar/assets/standard/js/match-media.js"></script>
<script src="https://www.santafe.gob.ar/assets/standard/js/interface.js"></script>

```

Adjuntamos una imagen de los menús para mostrar el cambio de estilo en la aplicación:



Imagen 3: Captura de pantalla menú

### **3.3.7 Adaptación para múltiples Municipios y Comunas**

#### **Agregar columna idAgencia en todas las tablas de SIAT**

Como primer paso se modificó el esquema de base de datos agregando en todas las tablas la nueva columna 'idAgencia'. Para realizarlo se armaron los alters a partir de:

```
SELECT 'ALTER TABLE '||table_name||' ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;' FROM user_tables order by table_name;
```

Luego depuramos eliminando el alter para 'schema\_version'. También se elimina el alter para la tabla 'seg\_agencia' que fue creada con la columna.

Modificamos nuevamente las columnas agregadas para que por defecto inserte null. De esta forma fallará la inserción si no se setea el valor.

Realizamos esta acción para poder realizar mejor las pruebas y que en caso que el sistema no inserte el idagencia falle la inserción.

Para esto generamos los alter de esta forma:

```
SELECT 'ALTER TABLE '||table_name||' MODIFY (IDAGENCIA DEFAULT NULL);' FROM user_tables order by table_name;
```

Ejemplos de los sql de alternación de las tablas generados en ambos pasos:

```
ALTER TABLE AFI_ACTLOC ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE AFI_AUDITORIA ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE AFI_CONDICIONTRIBUTARIA ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE AFI_DATOSDOMICILIO ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE AFI_DATOSPAGOCTA ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
....  
ALTER TABLE SEG_OFICINA ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE SEG_ROLOFICINA ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE SEG_TIPNOVUSU ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;  
ALTER TABLE SEG_USUARIOSIAT ADD IDAGENCIA NUMBER(19) default 1 NOT NULL;
```

Y:

```
ALTER TABLE AFI_ACTLOC MODIFY (IDAGENCIA DEFAULT NULL);  
ALTER TABLE AFI_AUDITORIA MODIFY (IDAGENCIA DEFAULT NULL);  
ALTER TABLE AFI_CONDICIONTRIBUTARIA MODIFY (IDAGENCIA DEFAULT NULL);  
ALTER TABLE AFI_DATOSDOMICILIO MODIFY (IDAGENCIA DEFAULT NULL);  
ALTER TABLE AFI_DATOSPAGOCTA MODIFY (IDAGENCIA DEFAULT NULL);  
ALTER TABLE AFI_DECACTLOC MODIFY (IDAGENCIA DEFAULT NULL);  
.....  
ALTER TABLE SWE_TIPOAUTH MODIFY (IDAGENCIA DEFAULT 1);
```

```
ALTER TABLE SWE_USRAPL MODIFY (IDAGENCIA DEFAULT 1);  
ALTER TABLE SWE_USRAPLADMSWE MODIFY (IDAGENCIA DEFAULT 1);  
ALTER TABLE SWE_USRAUTH MODIFY (IDAGENCIA DEFAULT 1);  
ALTER TABLE SWE_USRRROLAPL MODIFY (IDAGENCIA DEFAULT 1);
```

En total fueron modificadas 734 tablas del sistema.

Asociado a este cambio se realizó una modificación en el core de la aplicación incorporando el mapeo de la nueva columna 'idAgencia' en la clase BaseBean.java.

### **Modificación de login para guardar información de agencia en el contexto de usuario**

Se modificó la creación de sesión del usuario para identificar la agencia asignada al mismo y guardarla en el contexto de usuario.

Se identificaron dos tipos de ingreso a SIAT:

1. Ingreso por CAS: Se toma la agencia a través del idagencia de la entidad SEG\_USUARIOSIAT.
2. Anónimo: Se toma la agencia a través del código de agencia pasado como parámetro de la URL externa. Ejemplo: *&agencia=sunchales*. El código de agencia se evaluará contra código de agencia de SEG\_AGENCIA. En caso de no encontrar la agencia establecida en la URL se carga un error recuperable para informar por pantalla.

En el contexto de usuario se guardan dos valores a ser utilizados luego para el filtrado y administración por agencia:

- El "id" de la tabla 'SEG\_AGENCIA' en una variable 'idAgencia'
- El "código" de la tabla 'SEG\_AGENCIA' en una variable 'codAgencia'

### **Modificar las clases GenericAbstractDAO y SqlQuery para que en los inserts y update se guarde el idagencia del contexto**

Se realizaron modificaciones a los objetos de acceso a datos (DAOs) a través de la clase heredadas de la arquitectura 'GenericAbstractDAO.java' y 'DaoStd.java' para que graben el 'idAgencia' que viene seteado en el contexto de usuario 'UserContext' al momento de grabar datos en cualquier tabla del sistema.

Se realizó un test unitario para probar las modificaciones: 'DaoUpdateAgenciaTest.java'

Con este cambio cualquier operación de guardado a través de la arquitectura grabará el valor de idAgencia que corresponda (Por ejemplo al emitir una deuda, o dar de alta una nueva cuenta contributiva).

## Crear agencia genérica 'siat' y usuario especial "superadmin" para administrar agencias

Creamos un usuario del sistema particular para la administración de las agencias.

```
--CREACION USUARIO SUPERADMIN EN SWE
Insert into SWE_USRAPL
(USERNAME, IDAPLICACION, USERID, FECHAALTA, FECHABAJA, USUARIO, FECHAULTMDF, ESTADO, PERMITEWEB, IDAGENCIA)
values ('superuser', 2, 0, SYSDATE, null, 'admin', SYSDATE, 1, 0, 1);

--CREACION DE USUARIO SUPERADMIN EN SIAT
Insert into SEG_USUARIOSIAT
(USUARIOSIAT, IDAREA, IDPROCURADOR, IDINVESTIGADOR, IDINSPECTOR, IDRNP, USUARIO, FECHAULTMDF, ESTADO, IDABO
GADO, IDSUPERVISOR, IDMANDATARIO, FECHAULTLOGIN, NOMBAPELLIDO, IDOFICINA, ESAPI, FECHADESDEINHAB, FECHAHAS
TAINHAB, MOTIVOINHAB, CUITUSUARIO, IDPERSONA, EMAIL, IDAGENCIA)
values ('superuser', 1, null, null, null, null, 'admin', sysdate, 1, null, 1);
```

Este usuario pertenecerá a una agencia genérica "SIAT" de id=1 que siempre deberá existir en el sistema (antes que cualquier otra).

```
INSERT INTO seg_agencia (ID, CODIGO, DESCRIPCION, USUARIO, FECHAULTMDF, ESTADO, IDAGENCIA) VALUES
(1, 'SIAT', 'AGENCIA SIAT', 'admin', sysdate, 1, 1);
```

Luego creamos un rol de acceso "SUPERADMIN" con permisos sobre acciones de administración para:

- Acciones del mantenedor de agencia.
- Acciones del mantenedor de parámetros.
- Acciones del mantenedor de procesos.

```
--CREACION ROL SWE DE SUPERADMIN
Insert into SWE_ROLAPL (IDAPLICACION, CODIGO, DESCRIPCION, USUARIO, FECHAULTMDF, ESTADO, PERMITEWEB, IDAGENCIA) values (2, 'superadmin', 'Super
Administrador', 'superadmin', SYSDATE, 1, 1, 1);

--ASIGNACION DE USUARIO SUPERUSER CON ROL SUPERADMIN EN SWE
Insert into SWE_USRRAPL (IDROLAPL, IDUSRAPL, USUARIO, FECHAULTMDF, ESTADO, IDAGENCIA) values (
(
select ID
from SWE_ROLAPL
where CODIGO = 'superadmin'
),
(
SELECT ID
FROM SWE_USRAPL
WHERE USERNAME = 'superuser' and IDAPLICACION = 2
),
'admin',
SYSDATE,
1,
1);

--ASIGNAMOS ACCIONES DE PARAMETROS, PROCESOS Y MANTENEDOR DE AGENCIA AL ROL SUPERADMIN
```





```
IF :new.id IS null THEN
  SELECT SEG_AGENCIA_ID_SEQ.nextval INTO :new.id FROM DUAL;
END if;
END;
/
```

```
GRANT SELECT on SEG_AGENCIA_ID_SEQ to ROL_MMYCC_SIAT;
GRANT SELECT, INSERT, UPDATE, DELETE on SEG_AGENCIA to ROL_MMYCC_SIAT;
```

Durante el alta se debe indicar un nombre de usuario administrador para el municipio/comuna.

Durante el alta se crea el usuario en SWE (Sistema de Seguridad Web Extendida de SIAT) con el rol 'admin' asignado y el usuario SIAT en la tabla 'seg\_usuariosiat' con el idagencia seteado.

Además de crear la agencia se realizan las siguientes tareas adicionales:

- Creación de la estructura del fileshare para la agencia. A partir de un directorio base (ej: fs/siat) se copiará a un nuevo directorio con nombre igual al código de agencia que se haya configurado (ej: fs/sunchales)
- Se realiza copia/instanciación de parámetros (def\_parametro). Se insertan nuevos registros de parámetros siat seteando el idagencia correspondiente. Se toman como base los parámetros registrados para la agencia siat.
- Se realiza una copia/instanciación de los procesos (pro\_proceso). Se insertan nuevos registros de procesos seteando el idagencia correspondiente. Se toman los procesos configurados para la agencia SIAT. Se modifica el directorio input agregando el código de agencia en el path.

La nueva opción de menú es: "Administración del Sistema → Agencias → Administrar Agencias para Municipios y Comunas."

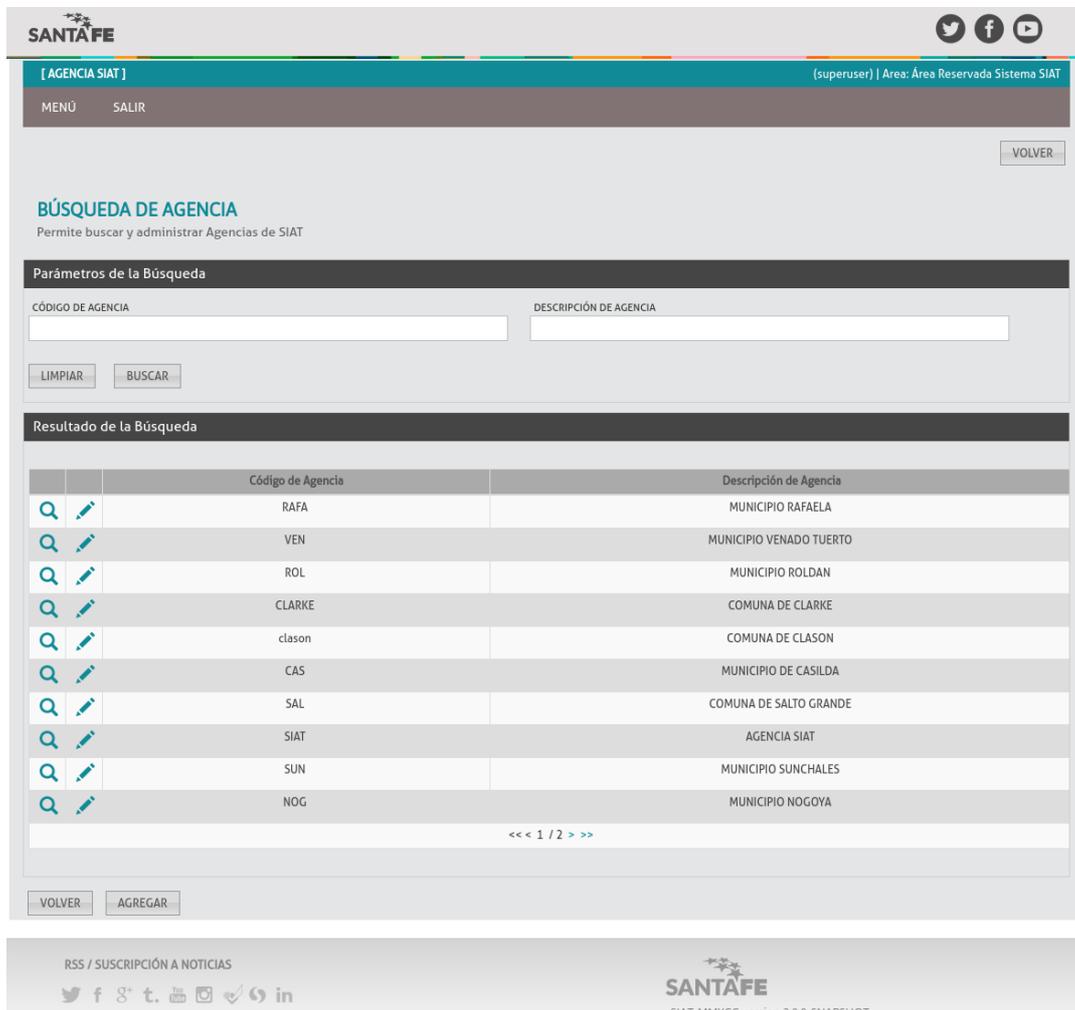


Imagen 4: Captura de pantalla administración agencia

## Interceptar los DAOs de hibernate y los del SqlQuery para las consultas 'select' para agregar filtros por idAgencia para cada tabla

Se creo un componente encargado de agregar a los 'where' de una query pasada como parámetro (SqlQueryInspector). Se parsea el query string en una serie de objetos anidados utilizando la herramienta JSQLParser, una librería de Github.

En el caso de las queries realizadas por Hibernate, la llamada al nuevo componente se hace desde 'HibernateQueryInterceptor' una especie de filtro que extiende de 'org.hibernate.EmptyInterceptor'.

Se realizó el siguiente test unitario para trabajar sobre el interceptor

'HibernateQueryInterceptorTest'.

Las clases principales para la intercepción son:

SqlQueryInspector.java  
HibernateQueryInterceptor.java  
HibernateQueryInterceptorTest.java

Para recorrer los distintos componentes del sql se implementó un patrón "Visitor" para las expresiones (ExpressionVisitor). De esta forma se evalúa el query general y cada subquery en forma recursiva. Se evalúan subqueries en el select, en el from y el where.

Durante el desarrollo y primeras pruebas encontramos un problema con los joins de tipo left. Ya que este debe traer registro de la tabla de la izquierda si no existe relación con la de la derecha y al agregar los filtros por agencia para estas tablas se fuerza a que no encuentre resultados.

Modificamos el interceptor para no incluir los alias de tablas de join de tipo left para el filtro.

Esto no sería problema porque siempre tiene que haber una tabla a la izquierda que se encuentra filtrada por idAgencia.

También fue necesario excluir en la consideración del from los casos sobre el objeto "dual". Este es utilizado para consultar valores de secuencias que son únicos para todas las agencias.

Modificamos la clase 'SqlQuery.java' para interceptar las consultas que realizamos sin pasar por hibernate.

### **Crear tabla de excepciones para que el interceptor de sql no agregue filtros por idAgencia**

Mientras realizamos pruebas y avanzamos con el análisis de otros casos detectamos un problema en varias tablas de datos maestros.

En muchos casos existen tablas donde los registros son importantes para el funcionamiento del sistema a nivel código. Ejemplo: gde\_viadeuda, gde\_estadodeuda, pro\_tipoproceso, etc. Para estas tablas incluso en el código muchas veces se refiere a ID de los registros. Estas tablas no pueden especializarse por agencia.

Por esto como solución se prepara un mapa para escapar la intervención de sql para estas tablas. O sea que el interceptor no agregue filtro si el nombre de la tabla está en un mapa que programado en el código. En este mapa se incluyen las tablas que por código deben ser única para todas las agencias.

Se creo una clase llamada TablesNoIncludeInterceptor.java que contiene una propiedad que declara el nombre de las tablas que no deben ser analizadas por el interceptor. Es un singleton que se crea bajo la primer demanda.

Se modifiko la lógica del interceptor para que antes de agregar las clausulas de idagencia, verifique que la tabla analizada no esté declarada en la clase.

### Completar tabla de excepción para el interceptor de sql

Se completó el mapa de excepciones de tablas para las cuales el interceptor no debe filtrar por agencia. Las tablas son:

```
"aud_tipoauditoria", "afi_estForDecJur", "bal_canal", "bal_estSalAFav", "bal_tipoorigen", "bal_accion", "bal_estadoArc",  
"bal_estadoCom", "bal_estadoEnvio", "bal_estadoFol", "bal_estadoSugit", "bal_estDetDJ", "bal_estDetPago", "bal_estDiaPar",  
"bal_estEjercicio", "bal_estOtrIngTes", "bal_estSalAFav", "bal_estTranAfip", "bal_tipoArc", "bal_tipoCom", "bal_tipoComDeu",  
"bal_tipoDistrib", "bal_tipoImporte", "bal_tipoOperacion", "bal_tipOriMov", "bal_tipoSellado", "cas_accionExp", "cas_estsolicitud",  
"cdm_estadoObra", "cdm_estPlaCua", "cdm_estPlaCuaDet", "cdm_tipPlaCuaDet", "cyq_estadoProced", "cyq_motivoBaja",  
"def_viadeuda", "def_desimp", "def_tipo", "def_periodoDeuda", "def_genCodGes", "def_categoria", "def_genCue", "def_tipoatributo",  
"def_vencimiento", "def_tipoEmision", "def_tipodeuda", "def_tipCodEmi", "def_tipoAct", "dre_tipoTramiteRS", "ef_estadoActa",  
"ef_estadoOpelInvCon", "ef_estadoOrden", "ef_estadoPlanFis", "ef_estadoOpelInvCon", "ef_origenOrden", "ef_tipoActa", "ef_tipoOrden",  
"emi_estAjuDeu", "esp_estHab", "exe_estadoCueExe", "gde_estadoDeuda", "gde_estConDeu", "gde_tippromas", "gde_criAsiPro",  
"gde_tipoDeudaPlan", "gde_tipoPerFor", "gde_tipoDocApo", "gde_estadoConvenio", "gde_estadoConCuo", "gde_tipoPago",  
"gde_accionLog", "gde_estadoAjuste", "gde_estadoCobranza", "gde_estadoMulta", "gde_estcerdeu", "gde_estPlaEnvDeuPr",  
"gde_estProPreDeuDet", "gde_motAnuDeu", "gde_motExc", "gde_motivoCierre", "gde_oriDecJur", "gde_tipDecJur", "gde_tipImpug",  
"gde_tipoMulta", "gde_tipPagDecJur", "pad_estCue", "pad_tipoDomicilio", "pad_estObjImp", "pad_tipoTitular", "pad_contribuyente",  
"pad_estCueEve", "pad_tipCueEve", "pad_tipoBroche", "pad_tipoRepartidor", "pad2_localidad", "pad2_pais", "pad2_provincia",  
"pro_estadoCorrida", "pro_tipoEjecucion", "pro_tipoProgEjec", "rod_estadoTramiteRA"
```

### Modificar algunos DAOs específicos por inserciones especiales (mover deuda, selección almacenada, otros)

Se modificaron las funciones de mover deuda en DeudaDAO y los utilizadas por los procesos masivos respecto a preparación de Selecciones Almacenadas

Algunos de los métodos modificados: 'moverDeuda' en DeudaDAO, métodos 'copiarDeuda...' en DeudaAdminDAO y DeudaJudicialDAO, método 'moverDeudaAdmAJud' de ProcesoMasivoDAO (para el envío masivo a judicial)

Las modificaciones son sobre sql específicos de inserción donde se debió agregar la columna 'idAgencia' a partir del agregado de la misma a todas las tablas del sistema.

### **Modificar el sistema para que al obtener el 'filesharepath' siempre incluya el código de agencia tomado del contexto**

Se realizaron modificaciones al sistema para que al obtener/armar el 'filesharepath' siempre incluya el código de agencia tomado del contexto como parte del mismo.

La carpeta donde se ubican los archivos compartidos (fileshare) para cada agencia contiene en el path el código de la misma.

Ejemplo:

```
../mmycc/fs/siat/  
../mmycc/fs/sunchales/  
../mmycc/fs/vgobgalves/  
../mmycc/fs/santateresa/  
../mmycc/fs/pujato/
```

A partir de cada código de agencia se encuentran las carpetas: privado y publico con toda la información requerida.

La carpeta 'siat' contiene la versión básica a partir de la cual se preparará al crear una nueva agencia la nueva estructura.

```
../mmycc/fs/siat/privado  
../mmycc/fs/siat/publico
```

El path se toma a partir de la clase 'SiatParam.java' pero toma su valor de una parámetro seteado en el contexto:

URL nuevo fileshare (context.xml):

```
<Environment description="Ubicación del FS" name="/siat/conf/FileSharePath" override="false"  
type="java.lang.String" value="/siat/mmycc/fs"/>
```

Se agregaron dos métodos en la clase 'SiatParam.java'

1. getFileSharePathAllAgenciasActivas → Para obtener la URL de fileshare de cada una de las agencias registradas y activas en siat.

2. `getFileSharePathAgenciaSiat` → Para obtener la URL del fileshare de la agencia por defecto (Agencia → siat)
3. Se creó la propiedad de clase `"public static final String CODAGENCIASIAT = "siat";"` para hacer referencia al código de la agencia de siat.
4. `GetRootFileShare()` → Retorna URL directorio root del fileshare, desde donde parten los fileshare de las agencias

Además fue necesario modificar el método `'initializeSiat()'` de la clase `'DefConfiguracionServiceHbmlImpl.java'`. Al iniciar los módulos del sistema debemos buscar a todas las agencias activas y realizar la funcionalidad de configuración de la instancia para cada una de las agencias obtenidas.

Se creó la clase `'FileshareManager.java'` que contiene los métodos necesarios para la creación y eliminación del fileshare de la agencia.

Se creó una nueva excepción (`FileshareException.java`) que se lanzan en los procesos tanto de creación como eliminación del fileshare.

Se creó una nueva clase `CopyDir.java` que se colocó en DEMODA y permite la copia de directorios de forma recursiva de un directorio origen a un directorio destino.

Se creó el archivo `ManagerFileshareTest.java` para registrar pruebas UNITARIAS.

### **Modificar funcionamiento de cache para contemplar uno por agencia (paso de cache simple a mapas de cache)**

Se analizaron los datos almacenados en Redis y pudimos observar que se guardan:

1. Localidades, calles y provincias.
2. Permisos SWE (específicos).

Estos datos no deben ser replicados para cada agencia, por lo tanto no fue necesario realizar modificaciones sobre este tipo de cache.

Sobre los mapas en memoria de la instancia, la mayoría de las clases son clases singleton, algunos instanciados cuando se inicia el sistema y otros son instanciados "bajo demanda".

Modificamos los métodos que se encuentran en `initializeSiat()` de la clase

DefConfiguracionServiceHbmImpl ya que se utiliza para levantar SIAT. Buscamos los datos de cada una de las agencias registradas en el sistema y la cargamos en mapas (parámetros, frases, etc):

1. Parámetros: Cargamos un mapa de <String, SiatParam> que contiene un mapa que almacena los parámetros de cada una de las agencias registradas en el sistema.
2. SiatHibernateUtil -> Buscamos el parámetro HibernateDialect desde el valor de la agencia SIAT.
3. Log del metodo initModules() de initializeSiat() fue colocado buscando el parámetro de la agencia SAT.
4. Fileshare -> Se valida si ciertas carpetas de cada fileshare de agencia esta creada, en caso de no estarlo las crea.
5. Frases -> El singleton de Frases se trasformo en un mapa de singleton <String, Frase> y cada uno de las posiciones del mapa tendrá almacenado los datos particulares de cada agencia.
6. Parámetro "forcePoint" seteado desde el valor del parámetro de la agencia SIAT.
7. Parámetro "CACHE\_HABILITADO" seteado desde el valor del parámetro de la agencia SIAT. Usado para saber donde se recuperan los datos de provincias, localidades y calles (son únicas para todas las agencias)
8. Se ejecuta el método siatPeep.start() una vez para cada una de las agencias registradas en el sistema.
9. Parámetros TIMEOUT\_ANONIMO, TIMEOUT\_AUTENTICADO obtenidos desde el valor del parámetro de la agencia SIAT.

### **Modificar el logeo de Log4j para que agregue el código de agencia del contexto**

Realizamos modificaciones la log de la aplicación en el servidor para agregar el código de agencia del contexto (error, info y debug)

De esta forma cuando se encuentre en uso se podrá filtrar luego el log de la instancia de aplicación a partir del código de agencia para analizar por municipio cuando fuere necesario.

Se muestra en el log "(codigoagencia)" donde "codigoagencia" será el código de la agencia registrado en la tabla SEG\_AGENCIA.

Se muestran los logs de tipo DEBUG, INFO y ERROR.

Se modificaron los siguientes archivos para lograr el funcionamiento indicado:

```
siat-app/src/main/resources/dev/log4j.properties
siat-server/src/main/java/server/ar/gov/rosario/siat/sys/AppMonitorFilter.java
siat-server/src/main/java/server/ar/gov/rosario/siat/sys/SiatFilter.java
```

De las pruebas realizadas copiamos a continuación un breve ejemplo de los logs generados. En mismo server ingresamos con dos usuarios de distinta agencia y se mostró de forma correcta el log junto con el código de la agencia correspondiente:

```
2018-01-24 17:16:22,864 DEBUG http-nio-8090-exec-1 superclason (clason) [LetRequest:701] - callMethod.arg = null
2018-01-24 17:16:22,864 DEBUG http-nio-8090-exec-1 superclason (clason) [LetRequest:712] - invoking method: public
coop.tecsa.demoda.sys.Search ar.gov.rosario.siat.emp.form.FormLetStd.search(coop.tecsa.demoda.sys.Search) throws
java.lang.Exception
2018-01-24 17:16:22,865 DEBUG http-nio-8090-exec-1 superclason (clason) [LetRequest:582] - GetModel: for mrk:null
uri:/siat/app/emp/form/search find model!
2018-01-24 17:16:22,868 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'idAccionModulo' value:'11992'
2018-01-24 17:16:22,869 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'method' value:'select'
2018-01-24 17:16:22,869 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'idItemMenuNivel3' value:'1363'
2018-01-24 17:16:22,869 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'back' value:'/siat/app/menu'
2018-01-24 17:16:22,869 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'idItemMenuNivel2' value:'1362'
2018-01-24 17:16:22,870 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'idItemMenuNivel1' value:'1359'
2018-01-24 17:16:22,870 DEBUG http-nio-8090-exec-1 superclason (clason) [Util:214] - copying key:'url' value:'/app/emp/form/search'
2018-01-24 17:16:22,870 DEBUG http-nio-8090-exec-1 superclason (clason) [LetRequest:701] - callMethod.arg =
coop.tecsa.demoda.sys.Search@3bd58152
2018-01-24 17:16:22,871 DEBUG http-nio-8090-exec-1 superclason (clason) [LetRequest:712] - invoking method: public
coop.tecsa.demoda.sys.Search ar.gov.rosario.siat.emp.form.FormLetStd.search(coop.tecsa.demoda.sys.Search) throws
java.lang.Exception
2018-01-24 17:16:22,871 DEBUG http-nio-8090-exec-1 superclason (clason) [LetRequest:547] - setModel: refreshUri:
/siat/app/emp/form/-204c5b5a/refresh
2018-01-24 17:16:25,898 INFO http-nio-8090-exec-9 admTot (pujato) [SiatRedisUtil:26] - java:comp/env/redis/server-
172.16.17.53:6379
2018-01-24 17:16:25,899 INFO http-nio-8090-exec-9 admTot (pujato) [Task:92] - newtask: thread http-nio-8090-exec-9.
2018-01-24 17:16:25,900 DEBUG http-nio-8090-exec-9 admTot (pujato) [SiatMenuAction:111] - select(): enter
2018-01-24 17:16:25,900 DEBUG http-nio-8090-exec-9 admTot (pujato) [DemodaUtil:104] - populateVO: enter
-----
2018-01-24 17:16:25,900 DEBUG http-nio-8090-exec-9 admTot (pujato) [DemodaUtil:105] - populateVO: populating class
ar.gov.rosario.siat.seg.iface.model.MenuAdapter
2018-01-24 17:16:25,900 DEBUG http-nio-8090-exec-9 admTot (pujato) [DemodaUtil:115] - populateVO: request: method -> select
2018-01-24 17:16:25,900 DEBUG http-nio-8090-exec-9 admTot (pujato) [DemodaUtil:145] - populateVO: propiedad NO encontrada:
```

## Modificar Adp (administrador de procesos) para acceso a datos por filtro e información de agencia

Se realizaron modificaciones sobre el core de Adp (administrador de procesos) en lo que respecta al manejo de acceso a datos para considerar la agencia (idAgencia), agregando como filtro o completando según el tipo de sql.

Cambios realizados en adp.

1. AdpProcess → agregamos la propiedad idagencia.
2. AdpDao → método loadProcessFromRs() --> seteamos el valor de la propiedad idagencia con el valor que llega de la base idagencia.
3. AdpDao → método createRun() → Agregamos el campo idagencia para realizar el insert.

Archivos modificados:

```
src/coop/tecsso/adpcore/AdpProcess.java
src/coop/tecsso/adpcore/AdpRun.java
src/coop/tecsso/adpcore/engine/AdpDao.java
src/coop/tecsso/adpcore/engine/AdpParameter.java
```

Durante la pruebas en uno de los usos de adp (para ejecución de grs) para una agencia determinada, ADP no pudo identificar la agencia correcta que estaba ejecutando el proceso porque se realizaba la gestión por el código de proceso que ahora se repite para las distintas agencias. AdpCore utiliza el código de proceso para obtener el registro en pro\_proceso y a partir de ese proceso obtiene la agencia (idagencia). Con este idagencia se registran las tablas de corrida, paso corrida, log corrida, file corrida etc.

Se completó entonces las modificaciones reemplazando la instanciación de las corridas de ADP para en lugar de usar el código de proceso, se utilice otro método que usa el id del proceso para el que se genera la corrida.

Se realizaron con éxito pruebas en GRS y los distintos tipos de proceso ADP.

### **3.3.8 Pruebas automáticas**

Se realizaron test automáticos que se encuentran incluidos en el código de la aplicación entregado con los cuales se realizaron pruebas de funcionalidades críticas o testigos mientras se realizaban las modificaciones al código.

Entre ellos se pueden destacar:

- ABM de Parámetro del Sistema:  
***/siat-app/src/test/resources/features/intranet/mantenedores/parametros.feature***  
donde se automatizaron los siguientes escenarios:
  - Esquema del escenario: Consultar valor de parámetro de una agencia
  - Esquema del escenario: Registrar nuevo parámetro de agencia
  - Esquema del escenario: Validar campos requeridos en alta de parámetro de una agencia

- Esquema del escenario: Modificar valor de parámetro de una agencia
- Esquema del escenario: Eliminar parámetro de una agencia
- **Alta de Agencia**  
Cucumber Feature: 'alta-agencia.feature'  
JUnit Steps: Se resolvió con steps definidos en 'CommonSteps.java' y en 'AltaAgenciaSteps.java'
- **Convenio de Pago**  
***/siat-app/src/test/resources/features/intranet/convenios/impresion-convenio.feature***  
donde se automatizaron los siguientes escenarios:
  - Esquema del escenario: Imprimir formalización de un convenio de una agencia
  - Esquema del escenario: Buscar un convenio inexistente para una cuenta de una agencia***/siat-app/src/test/resources/features/intranet/convenios/impresion-cuota-convenio.feature***  
donde se automatizaron los siguientes escenarios:
  - Esquema del escenario: Imprimir cuota de un convenio de una agencia
  - Esquema del escenario: Impresión sin seleccionar una cuota de convenio***/siat-app/src/test/resources/features/intranet/convenios/formalizacion\_anulacion\_convenio.feature***  
donde se automatizaron los siguientes escenarios:
  - Esquema del escenario: Seleccionar una deuda y formalizar un convenio de pago para una agencia
  - Esquema del escenario: Seleccionar un convenio de pago de una agencia y anularlo***/siat-app/src/test/resources/features/intranet/convenios/formalizacion\_saldocaducidad\_convenio.feature***  
donde se automatizaron los siguientes escenarios:
  - Esquema del escenario: Seleccionar una deuda y formalizar un convenio de pago para una agencia
  - Esquema del escenario: Generar un saldo a favor de una cuenta de una agencia
  - Esquema del escenario: Imputar saldo a favor a una cuota de convenio de una agencia
  - Esquema del escenario: Seleccionar un convenio de pago de una agencia y aplicar saldo por caducidad
  - Esquema del escenario: Actualizar el valor de la deuda al inicial
- **Gestión de Cuenta**  
***/siat-app/src/test/resources/features/intranet/gestion-deuda/reimpresion-deuda.feature***  
donde se automatizaron los siguientes escenarios:
  - Esquema del escenario: Re-imprimir una deuda vencida de una cuenta de una agencia
  - Esquema del escenario: Re-imprimir una deuda NO vencida de una cuenta de una agencia
  - Esquema del escenario: Buscar una cuenta inexistente para la agencia
- **Login y Logout Con CAS**  
***/siat-app/src/test/resources/features/intranet/login/login-siat-cas.feature***  
***/siat-app/src/test/resources/features/intranet/login/logout-siat-cas.feature***  
Los escenarios que incluyen son los siguientes:
  - Esquema del escenario: Ingreso al login con usuario habilitado y cerramos la sesión
  - Esquema del escenario: Ingreso al login con usuario inhabilitado
  - Esquema del escenario: Cerrar sesión de forma correcta
- **Modificar Domicilio de Envío**  
Cucumber Feature: 'modif-domicilio-envio.feature'

- JUnit Steps: Se resolvió con steps definidos en 'CommonSteps.java'
- **Modificar el Valor de un Atributo de Objeto Imponible**  
*/siat-app/src/test/resources/features/intranet/objeto-imponible/update-valor-objimp.feature*  
 donde se automatizaron los siguientes escenarios:
  - Como Usuario
    - Quiero ingresar al objeto imponible de una cuenta de una agencia
    - Para modificar el valor de uno de sus atributos.
  - Esquema del escenario: Modificar valor de atributo de objeto imponible (sin dominio)
  - Esquema del escenario: Modificar valor de atributo de objeto imponible (con dominio)

Dentro de los escenarios se validan caminos básicos y caminos alternativos.

En la documentación adicional entregada se adjuntan como imagen algunos de los resultados obtenidos al ejecutar las pruebas automáticas.

### 3.3.9 Artefactos entregados

Con el segundo informe de avance (primero de la etapa 2) se disponibilizó a la provincia los siguientes artefactos:

Artefacto	Herramienta	Ubicación
Dump de base de datos	Drive	<b>siat-mmycc-r1-dump.tar.gz</b>
Informe de Avance 2da Certificación	GIT / Drive	../01.gestionProyecto/informesAvance/MmyCC-SIAT-IA-Informe-2da-certificacion.pdf (también se compartieron por drive en carpeta)
Alternativas para implementación de requerimiento multiagencia.	GIT / Drive	../03.analisisDiseno/arquitectura/propuestaTecnicaSIATMmyCC.pdf (también se compartieron por drive en carpeta)
Documentos de Arquitectura SIAT	GIT / Drive	../03.analisisDiseno/arquitectura/TECSO-SIAT-DAEspecificacionArquitectura-AnexoII-Estandares.pdf ../03.analisisDiseno/arquitectura/TECSO-SIAT-DAEspecificacionArquitectura-AnexoIII-HowTo.pdf ../03.analisisDiseno/arquitectura/TECSO-SIAT-DAEspecificacionArquitectura-secuencias.pdf ../03.analisisDiseno/arquitectura/TECSO-SIAT-DAEspecificacionArquitectura-SWE.pdf ../03.analisisDiseno/arquitectura/TECSO-SIAT-DAEspecificacionArquitectura.pdf ../03.analisisDiseno/arquitectura/TECSO-SIAT-DAEspecificacionArquitecturaAnexoI-ADP.pdf ../03.analisisDiseno/arquitectura/TECSO-SIAT-EspecificacionModeladoDatosAnexoIV.pdf (también se compartieron por drive en carpeta)
Documento para	GIT / Drive	../06.configuracion/ambientes/siat-mmycc-entornos.pdf (también se compartieron por drive en carpeta)

preparación entornos		
Documento de Planificación para Etapa 2	GIT / Drive	../01.gestionProyecto/planCronograma/siatmmycc-cfi-adaptacion-etapa2-20171124.pdf (también se compartieron por drive en carpeta)
Código de la aplicación	GIT	<a href="https://gitlab.tecsco.coop/tecsco/siat-mmycc-santafe.git">https://gitlab.tecsco.coop/tecsco/siat-mmycc-santafe.git</a> <a href="https://gitlab.tecsco.coop/tecsco/swe-mmycc-santafe.git">https://gitlab.tecsco.coop/tecsco/swe-mmycc-santafe.git</a> (por email se envió usuario de acceso generado)
Paquetes war de la versión generada para SIAT, ADP y SWE.	Drive	mmyccsiat-1.0.0_RC1.tar.gz (contiene: <i>siat.war</i> , <i>adpsiat.war</i> , <i>swe.war</i> ) siat-files.tar.gz (contiene versión del fileshare para despliegue)

A los cuales para la entrega final se le agregan:

Artefacto	Herramienta	Ubicación
Dump de base de datos	Drive	<b>siat-mmycc-r2-dump.tar.gz</b> Link de archivo compartido: <a href="https://drive.google.com/open?id=1ytBwyQNzHbPj31KoEw4I-OOW36J9xFGN">https://drive.google.com/open?id=1ytBwyQNzHbPj31KoEw4I-OOW36J9xFGN</a>
Código de la aplicación	GIT	<a href="https://gitlab.tecsco.coop/tecsco/siat-mmycc-santafe.git">https://gitlab.tecsco.coop/tecsco/siat-mmycc-santafe.git</a> <a href="https://gitlab.tecsco.coop/tecsco/swe-mmycc-santafe.git">https://gitlab.tecsco.coop/tecsco/swe-mmycc-santafe.git</a> (por email se envió usuario de acceso generado)
Paquetes war de la versión generada para SIAT, ADP y SWE.	Drive	mmyccsiat-2.0.0_RC1.tar.gz (contiene: <i>siat.war</i> , <i>adpsiat.war</i> , <i>swe.war</i> ) siat-files.tar.gz (contiene versión del fileshare para despliegue)

## 4. CONCLUSIONES

Se realizaron todas las actividades involucradas en ambas etapas del proyecto actual dentro del marco general por el cual se podrá luego continuar con próximas adaptaciones, nuevos módulos y las tareas de implementación en los municipios seleccionados.

Al trabajo de relevamiento realizado sobre los Municipios seleccionados en el proyecto, sumamos para el análisis de conclusiones nuestra experiencia en trabajos de implementación en varios Municipios de diferentes tamaños en 3 provincias de nuestro país.

SIAT-GPL se encuentra implementado en los Municipios de Rosario, Ciudad de Santa Fe, Reconquista (Santa Fe), Paraná (Entre Ríos), Nogoyá (Entre Ríos), Villaguay (Entre Ríos), Almirante Brown (Buenos Aires). Además hemos participado de trabajos en otros Municipios que por diversas razones (principalmente por cambios políticos y discontinuación del proyecto) no llegaron a implementar el sistema como por ejemplo: Rio Tercero (Córdoba) y Morón (Buenos Aires). Así mismo hemos realizados trabajos de evaluación inicial de factibilidad en otros Municipios de las Provincias de Santa Fe, Córdoba, Entre Ríos y Buenos Aires; quienes iniciaron contactos para evaluar la implementación. Entre estos últimos podemos mencionar: Ciudad de Córdoba, La Matanza, Escobar, Gualeguaychú, Colón, Arroyo Seco, entre otros.

Como se mencionó en el ante proyecto volvemos a destacar que como un resumen de estas experiencias, podemos considerar que una implementación de SIAT-GPL en un Municipio mediano, constituye un proyecto muy complejo que demanda de unos 16 meses de trabajo full-time de un equipo conformado por tres profesionales más un equipo aportado por el Municipio donde intervienen funcionarios que conocen las particularidades de la administración y funcionarios que conocen los sistemas informáticos existentes.

Vale destacar que las tareas y actividades que demandan la mayor cantidad de tiempo, no están vinculadas a tareas técnicas como podría ser la programación de un determinado módulo o aspectos de configuración, sino las que vienen dadas por:

- El cambio organizacional que debe realizar el Municipio para la implementación del Sistema
- Los cambios en materia de ordenanzas, especialmente en relación al cálculo de intereses, planes de pagos vigentes, periodicidad de la emisión de deuda, exenciones, condiciones de caducidad, etc.

- La Migración de Datos: Depende de la situación de cada Municipio, pero en general, existen muchísimas particularidades e inconsistencias en los datos, que se producen como consecuencia de las múltiples ordenanzas y sus implementaciones a lo largo de los años.
- El tratamiento de los “cientos de casos particulares” que generalmente existen en la Administración Tributaria Municipal.

Como estrategia de implementación, se utiliza un enfoque incremental, donde se inicia por los recursos más importantes (en términos de recaudación) como Tasa General Inmobiliaria, DREI (Derecho Registro e Inspección) y Contribución por mejoras. Y luego, se continúa con otras tasas, derechos y contribuciones por los general enmarcados como rentas varias o gravámenes especiales.

En Municipios de menor tamaño, por un lado pueden simplificarse algunas actividades de migración y/o de implementación inicial de rentas que muchas veces no se encuentran sistematizadas.

Por otra parte estos Municipios cuentan con menor disponibilidad de personal para llevar a cabo las tareas de revisión, validación y definición muchos puntos importantes del proyecto.

Es por esto que sin importar el tamaño del Municipio, uno de los factores determinante del éxito de la implementación, es contar con un responsable del proyecto por parte del Municipio que cuente con la posibilidad cierta de tomar decisiones o articular las mismas en el tiempo planificado así como velar por el seguimiento general del trabajo del resto del personal del municipio.

En las conversaciones previas al armado del proyecto se mencionaba apuntar inicialmente a Municipios considerados de 2da categoría y/o Comunas.

La provincia está organizada en cinco regiones, formalmente está conformada por 19 departamentos, y estos a su vez se integran por distritos categorizados en Municipios y Comunas. Formada por 51 Municipios y las 312 Comunas que integran el territorio provincial.

Son Municipalidades todos aquellos centros urbanos que tengan una población mayor a 10 mil habitantes. A su vez se dividen en dos categorías: de primera aquellas que tengan más de 200 mil habitantes (actualmente sólo cumplen con esta condición las ciudades de Santa Fe y Rosario); y de segunda los municipios restantes.

Los Municipios inicialmente seleccionados, según información del censo del 2010, cuentan con:

- Sunchales: 21,304 habitantes
- San José de la Esquina : 7,072 habitantes

El Municipio de “Villa Gobernador Gálvez” que finalmente reemplazó en la selección a “San José de la Esquina” cuenta con 80,769 habitantes.

De la experiencia del trabajo realizado y reforzada por la experiencia previa de los profesionales en otros Municipios podemos destacar algunos puntos a considerar para las implementaciones :

- Lo Municipios de menor tamaño suelen tener una fuerte necesidad de completar la digitalización de otras áreas de la administración como por ejemplo: Personal y Sueldos, Tesorería y Contable, Gastos y Proveedores, Expedientes.  
La necesidad varia ya que algunas administraciones han logrado avances en algunos de los puntos, pero de todas formas es importante establecer claramente desde el comienzo el alcance del sistema a disponibilizar que tiene como foco la administración de los recursos del municipio y toda la gestión de la deuda, y que aunque con puntos de contacto y/o interacción hacia otros sistemas no necesariamente brinda soluciones concreta sobre dichas otras necesidades.  
Las mismas podrán ser materia de ampliación del sistema o complementación a través de otros sistemas satélites por fuera del alcance del proyecto actual.  
Vale la pena destacar que en el caso de San José de la Esquina se cuenta con la implementación de un sistema privativo enfocado para pequeños municipios que cuenta actualmente con algunos de estos puntos, por lo cual el cambio de sistema necesariamente les requiere dar reemplazo a dichos módulos.
- Los Municipios de mayor tamaño, aún sin llegar a la escala de los de primera categoría, suelen tener mayores expectativas de especialización y adaptación de ciertos procesos a sus procedimientos actuales, y/o a la evolución constante del sistema por nuevas necesidades que se son comunes encontrar en municipios en crecimiento.
- Todos los Municipios, aunque con mayor énfasis los de mayor tamaño, quieren tener acceso y control sobre sus datos. Es por esto que algunas de las preguntas recurrentes en los trabajos fueron sobre:
  - El acceso por parte de personal del municipio a las base de datos del sistemas
  - La situación y disponibilización de los datos si en algún momento se decide salir de la plataforma provincial y pasar a contar con servidores propios y/o por

nuevo cambio de sistema.

- Respecto a la migración de datos de sistemas anteriores, la situación depende principalmente del grado de digitalización de los tributos en el municipio. Aunque en municipios de cierto tamaño suele ser mayor el pedido de migrar todos los datos disponibles. A su vez, el problema con que cuentan es que dependiendo del tamaño del municipio y los sistemas previos, no siempre se dispone del personal apto para la captación y adaptación de los datos en los tiempos deseados. Este punto sólo se remarca para tener en cuenta en las expectativas generadas para los tiempos de implementación ya que es uno de los puntos que mayor trabajo requiere cuando no se logran consensos para simplificar los datos a migrar. Claramente las complejidades aumentan de manera considerable cuando el tamaño del Municipio y en particular de los datos digitalizados es mayor. Es importante que se comprenda que estas actividades requiere de un fuerte trabajo de preparación, control y validación de los datos por parte de personal del municipio acompañando al personal externo dedicado a la implementación.

Del relevamiento realizado existen algunas preocupaciones de los municipios respecto a los datos de personas que serán disponibilizados por la provincia. Es por esto que algunos de los puntos relevados responde a una revisión y evaluación de los mismos para garantizar el funcionamiento esperado por los municipios.

Para finalizar, del trabajo realizado, confirmamos que el alto grado de parametrización y adaptabilidad del sistema permite la implementación en la mayoría de los Municipios de la Provincia con las observaciones realizadas.

Los requerimientos adicionales tomados son ajustes o adaptaciones que podrán ser incorporadas al sistema antes de la implementación en el marco de trabajos de adaptaciones restantes del proyecto general en una próxima etapa..

Se realizaron todas las actividades de adaptación involucradas en esta etapa del desarrollo para las adaptaciones de SIAT-GPL y enmarcadas en el proyecto finalizado.

Finalizada las adaptaciones de esta primer parte de modificaciones sobre SIAT-GPL nos parece importante destacar algunos puntos que fueron presentados para prestar especial atención sobre la alternativa para el requerimientos multiagencia seleccionada y que son importantes tener presentes para las próximas etapas del proyecto:

- Al existir una única base de datos para todos los municipios, no va a ser posible

darle acceso fácilmente a usuarios técnicos de los mismos (administradores de base de datos por ejemplo) directamente a ella, ya que van a poder ver los datos de todos los otros municipios. (En general hemos detectado que es un requisito que siempre surge por parte del Municipio)

- Existe el riesgo que en futuros desarrollos se escapen las intervenciones de sql y algún desarrollador agregue en el negocio un sql (consulta o escritura) que no filtre por municipio. Este punto es importante tenerlo en cuenta para los equipos de desarrollo de la provincia u otros proveedores a los que pueda encargarse dichas tareas.
- En caso de necesitar realizar una operación de update, por parte de personal técnico de la provincia o el municipio, para arreglar algún dato migrado con error (situación muy común en este tipo de implementaciones), se correrá el riesgo de afectar los datos de otros municipios ante su incorrecta aplicación. Esto agrega más responsabilidad sobre quienes deban o puedan administrar la base de datos general. Nuevamente se vuelve a mencionar para remarcar a los analistas de la provincia o proveedores a los que pueda encargarse dichas tareas.

Se aclara de todas formas que la solución seleccionada pudo ser implementada y en la versión final entregada se cuenta con la aplicación que permite soportar múltiples agencias en una misma instancia y utilizando el mismo esquema de base de datos.

Desde la versión resultante se debe ingresar inicialmente al sistema con un usuario de “super-administrador” como el mencionado en este informe y asignado a una agencia general “SIAT”. Desde este acceso se podrá utilizar la nueva opción de administración de agencia para dar de alta un nuevo Municipio o Comuna que comenzará a utilizar el sistema.

Luego ingresando con el usuario “administrador” creado junto con la nueva agencia, se podrán realizar las configuraciones y definición de parámetros específicos necesarias para el Municipio en cuestión como el alta de los diferentes usuarios para la agencia.

En caso de necesitar generar un backup de los datos para brindar al municipio se deberá realizar una bajada de datos de cada tabla filtrando por el idAgencia en cuestión más el idAgencia=1 “SIAT” (para los datos maestros). Este punto fue señalado al momento de definir el tipo de alternativa y aceptado por el personal de infraestructura de la provincia.

Entendemos que para garantizar el éxito de las implementaciones y posterior utilización de la plataforma por parte de los Municipios y Comunas de la Provincia es fundamental



interiorizarse en la solución que se ofrecerá y establecer en forma clara los alcances de la misma tanto respecto a los módulos que se brindan como a las cuestiones técnicas respecto al despliegue en servidores de la provincia y disponibilización de accesos al sistema y a los datos.

# **ANEXO I**

## **DOCUMENTO PARA PREPARACIÓN DE ENTORNOS**

# 1. INTRODUCCIÓN

El objetivo de este documento es detallar el hardware y software necesario para el despliegue y funcionamiento del sistema SIAT para Municipios y Comunas en los entornos de integración, test, y producción.

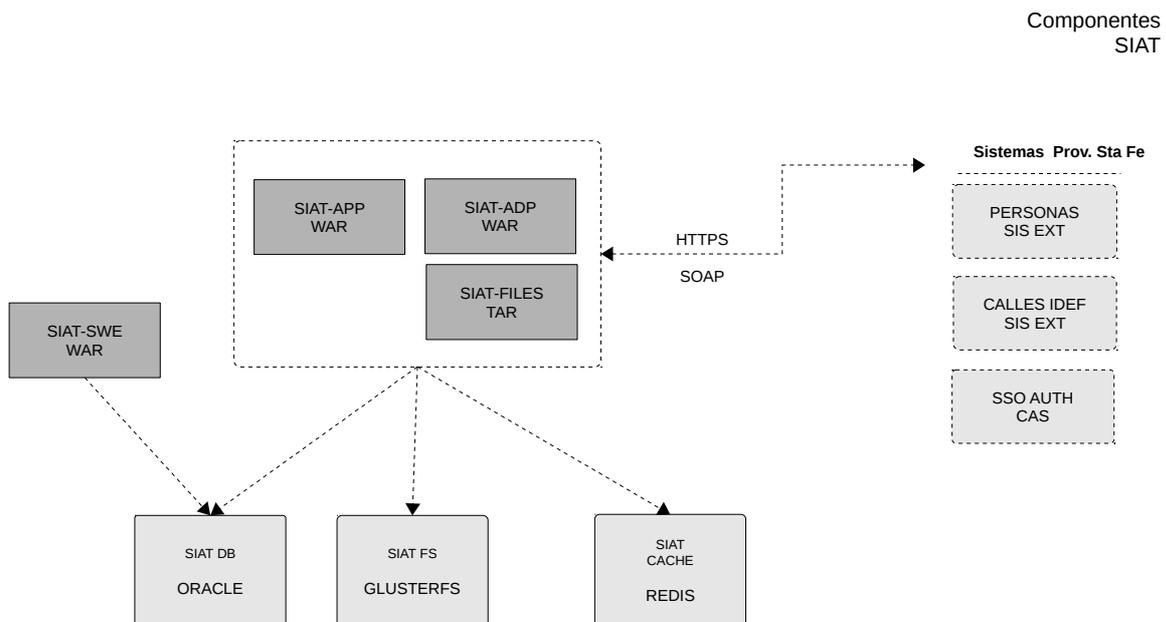
Se aclara que dada la etapa del proyecto sólo se requiere un entorno de test para el personal de la provincia.

# 2. COMPONENTES

En esta sección se enumeran los componentes de SIAT.

Se presenta un diagrama de relación entre ellos y enuncian las versiones de cada uno. Se distinguen en componentes propios de SIAT, componentes de software utilizado por SIAT y se enumeran algunos sistemas externos conectados con SIAT.

## 2.1 Diagrama



## **2.2 Componentes propios**

<b>COMPONENTE</b>	<b>TIPO</b>	<b>Versión</b>	<b>OBSERVACIÓN</b>
SIAT-APP	WAR		
SIAT-ADP	WAR		
SIAT-FILES	TAR		
SIAT-SWE	WAR		

## **2.3 Componentes de Software externo**

<b>COMPONENTE</b>	<b>TIPO</b>	<b>Versión</b>	<b>OBSERVACIÓN</b>
JAVA	OpenJDK	Version 8.0	
APACHE TOMCAT	Contenedor Webapp	Version 8.0	
ORACLE	Base de datos	Version 11g	
GLUSTER FS	File System Remoto		
REDIS	Queues y Memory Cache	Version 2.8	

## **2.4 Sistemas externos**

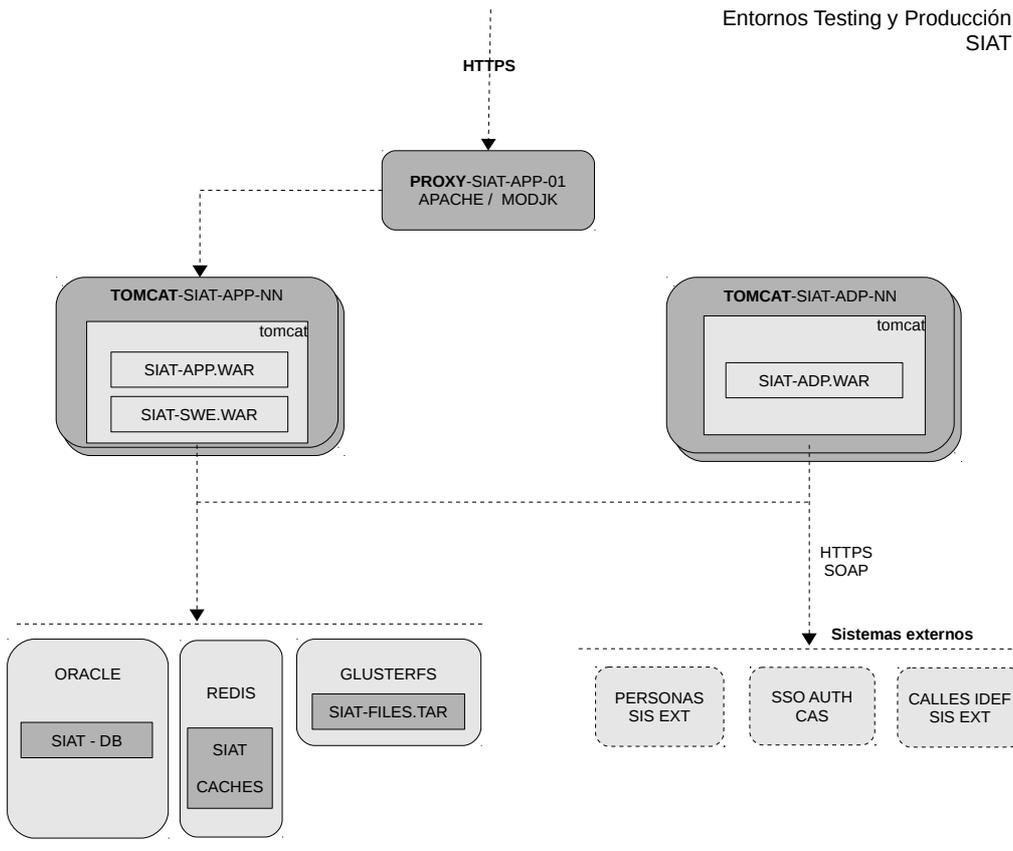
<b>COMPONENTE</b>	<b>TIPO</b>	<b>Versión</b>	<b>Observación</b>
CAS SSO	Sistema Externo		
PERSONAS	Sistema Externo		
CALLES IDEF	Sistema Externo		

### 3. ENTORNO DE TESTING

Este entorno se utiliza por usuarios de la Provincia de Santa Fe, tanto de Área Provincial de Municipios y Comunas de Santa Fe o de la Secretaría de Tecnología de la Provincia, que requieran probar las funcionalidades de SIAT.

Debido a la etapa del proyecto general, y dado que nos encontramos desarrollando la primer parte del mismo, no se especifica el entorno de producción. De todas formas la arquitectura propuesta para testing es similar a la que se utilizará en Producción y sólo varia en el tamaño y número de sus recursos físicos.

#### 3.1 Diagrama Testing



### 3.2 Hardware Testing

VM	NODOS	CPU	Memoria Física	Memoria JVM	HDD	Componentes
TOMCAT-SIAT-APP-*	2	4	8GB	7GB	20GB	SIAT-APP
TOMCAT-SIAT-ADP-*	2	4	8GB	7GB	20GB	SIAT-ADP
GLUSTERFS: /opt/siat/fs					60GB	SIAT-FILES
REDIS-SIAT-*	2	2	4GB			SIAT-CACHE

**NOTA:** los valores sugeridos son en base a valores utilizados en la implementación SIAT en un Municipio de gran tamaño (como Rosario o Santa Fe), aunque el foco son municipios chicos los recursos dependerán de la cantidad de municipios (agencias) que utilicen el sistema.

### 3.3 Servicios por VM PROXY-\* Testing

Servicio	Observación
APACHE 2.x	Frontend de los componentes SIAT-APP
MOD_JK	Balanceo de los componentes SIAT-APP

### 3.4 Servicios por VM TOMCAT-\* Testing

Servicio	Observación
OPENJDK-8	Open Source Java VM
TOMCAT8-STD:8080	Paquete Debian de Apache Tomcat8 estandariza.
GLUSTERFS: /opt/siat/fs	Mount point apuntando a espacio de <b>TESTING</b>

**NOTA:** Esta configuración es estimada y podría modificarse en función de la cantidad de Municipios o Comunas (Agencias) que utilicen el sistema y su tamaño.

### 3.5 Valores para entorno testing inicial

En esta etapa del desarrollo y con el fin de comenzar a probar el sistema y contar con la versión entregada en servidores de la provincia podrían utilizarse menos recursos en los equipos.



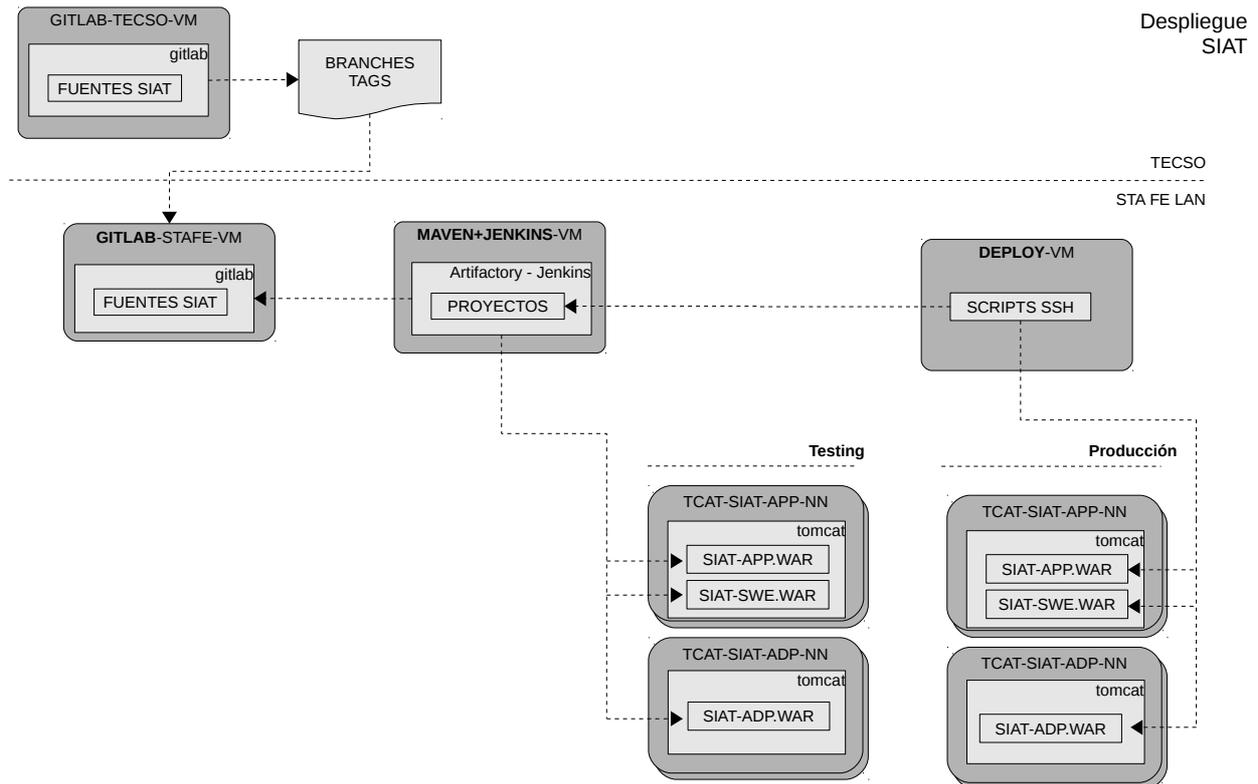
Por ejemplo podría utilizarse una única VM con dos tomcats (**TOMCAT-SIAT-APP**, **TOMCAT-SIAT-ADP**) y el SIAT-FILES en el mismo equipo en lugar de en el GLUSTERFS (lo importante es que todas las instancias -en este caso los tomcat- tengan acceso al mismo).

De igual forma podría utilizarse sólo un nodo de Redis.

La opción del apache para balanceo se utilizaría con más de una instancia. Podría no utilizarse en un primer entorno simple de pruebas. El esquema del diagrama refleja los componentes a utilizar luego en producción y propone un entorno de test con las mismas características principalmente útil una vez que esté más avanzado el proyecto y próximo a utilizarse en producción.

## 4. ESQUEMA DE DESPLIEGUE

### 4.1 Diagrama Despliegues



En el diagrama de despliegue anterior se propone como opción el despliegue utilizando JENKINS. Esto es opcional y puede ser configurado desde la Secretaría de Tecnología en sus entornos si se prefiere.

De otra forma se deberán generar los paquetes (war) con maven como se indica luego (o tomar los enviados por Tecso -por ejemplo para la primer entrega-) y desplegarlos en los tomcats correspondientes.

En el diagrama se menciona también un repositorio de código de la provincia. Para esta entrega se brinda acceso al repositorio git de Tecso con el código. A partir del mismo se puede subir a un proyecto en el repositorio de la provincia. Esta tarea de todas formas se encuentra planificada para la última entrega.

## 5. CONFIGURACIÓN

### 5.1 Entornos Testing y/o Producción

Configuración Apache Tomcat	
Item	Observación
Driver JDBC Oracle	Copiar el driver jdbc de oracle a: Driver: ORACLE DATABASE 11G-Release 2 JDBC DRIVER Archivo: ojdbc6.jar Destino : /var/lib/<INSTANCE_NAME>/lib
Archivo Config server.xml	<b>Ubicación: /var/lib/&lt;INSTANCE_NAME&gt;/conf/server.xml</b>
Datasource SIAT	<pre>&lt;Resource   name="ds/siat"   type="javax.sql.DataSource"   driverClassName="oracle.jdbc.driver.OracleDriver"   url="jdbc:oracle:thin:USER_OWNER@//HOST/SCHEMA"   username="USER_OWNER"   password="USER_OWNER"   validationQuery="select 1 from dual"   maxIdle="2"   maxWaitMillis="5000"   maxTotal="80"   minIdle="1"   initialSize="1"/&gt;</pre>
Datasource SWE	<pre>&lt;Resource   name="ds/swe"   type="javax.sql.DataSource"   driverClassName="oracle.jdbc.driver.OracleDriver"   url="jdbc:oracle:thin:USER_OWNER@//HOST/SCHEMA"   username="USER_OWNER"   password="USER_OWNER"   validationQuery="select 1 from dual"   maxIdle="2"   maxWaitMillis="5000"   maxTotal="80"</pre>

	<pre>minIdle="1" initialSize="1"/&gt;</pre>
Connector	<pre>&lt;Connector ... URIEncoding="UTF-8"/&gt;</pre>
<b>Archivo Config context.xml</b>	<b>Ubicación: /var/lib/&lt;INSTANCE_NAME&gt;/conf/context.xml</b>
Link a Datasources	<pre>&lt;ResourceLink global="ds/siat" name="ds/siat" type="javax.sql.DataSource"/&gt;  &lt;ResourceLink global="ds/swe" name="ds/swe" type="javax.sql.DataSource"/&gt;</pre>
CAS SSO	<pre>&lt;Environment description="Server Name" name="cas/serverName" override="false" type="java.lang.String" value="&lt;siat-app-server&gt;/siat" /&gt;  &lt;Environment description="CAS Login Url" name="cas/casServerLoginUrl" override="false" type="java.lang.String" value="&lt;sso-server&gt;/service-auth/login" /&gt;  &lt;Environment description="CAS Url Prefix" name="cas/casServerUrlPrefix" override="false" type="java.lang.String" value="&lt;sso-server&gt;/service-auth" /&gt;  &lt;Environment description="CAS Url Exclude" name="cas/casUrlExcludePaths" override="false" type="java.lang.String" value="/jax-ws/*;/ws-rs/**/*;/app/par/*"/&gt;</pre>
FileShare Path	<Environment

	<pre>description="Ubicación del FS" name="/siat/conf/FileSharePath" override="false" type="java.lang.String" value="&lt;file-share-path&gt;/&gt;</pre>
Redis	<pre>&lt;Environment description="Redis Server and Port" name="redis/server" type="java.lang.String" value="&lt;redis-server&gt;:&lt;redis-port&gt;/&gt;  &lt;Environment description="Redis Password" name="redis/password" type="java.lang.String" value="&lt;ENTORNO-LONG_PASSWORD&gt;/&gt;</pre>
<b>Configuración Redis</b>	
Archivo Config	Ubicación: /etc/redis/redis.conf
redis.conf	<pre>requirepass ENTORNO-LONG_PASSWORD  maxmemory 3GB</pre>
<b>Configuración SIAT</b>	
Archivo config de Logs	<pre>Ubicación: /var/lib/INSTANCE_NAME/webapps/siat/WEB-INF/classes/log4j.properties IMPORTANTE: Una vez re-desplegado se pierden los cambios ya que el archivo esta en el .WAR que se despliega</pre>

## 6. COMPILACIÓN

<b>Configuración SIAT</b>	
Ubicación	../siat-mmycc-santafe (sincronizado con el tag generado)
Comando	mvn clean mvn package (para testing) mvn package -Ptecsa-siat -Pprod (para producción)
<b>Artefactos Generados</b>	
Siat App War	../siat-mmycc-santafe/siat-app/target/siat.war
Siat Adp War	../siat-mmycc-santafe/siat-adpapp/target/adpsiat.war
Siat Files	../siat-mmycc-santafe/siat-etc/target/siat-files.tar.gz .

<b>Configuración SWE</b>	
Ubicación	../swe-mmycc-santafe (sincronizado con el tag generado)
Comando	mvn clean mvn package (para testing) mvn package -Ptecsa-siat -Pprod (para producción)
<b>Artefactos Generados</b>	
SWE War	../swe-mmycc-santafe/swe-app/target/swe.war

## **ANEXO II**

# **ACTIVIDADES PARA LA IMPLEMENTACION EN UN MUNICIPIO**

# 1. ACTIVIDADES Y ESTIMACIÓN IMPLEMENTACIÓN PARA LOS MUNICIPIOS SELECCIONADOS

## Alcance de la implementación en Sunchales

A continuación se describen las actividades a realizar para obtener cada producto:

### Producto 1 - DRel

Módulo/Actividad	Funcionalidad
<b>Relevamiento</b>	Validación de relevamiento de la tasa
<b>Configuración</b>	Configuración Atributos y Tipo Objeto Imponible
	Configuración Recurso / Vencimientos
	Configuración de Punta de Caja
	Configuración de parámetros del sistema (alícuotas, mínimos, % de excepción, etc )
<b>Migración</b>	Migración de Cuentas y Objetos Imponibles
	Migración de Atributos
	Migración de Obras
	Migración de Titulares
	Migración de Domicilios
	Migración de Calles, Barrios, Localidad
	Migración de Eventos de la cuenta
	Migración de Broches
	Migración de Relaciones de Cuenta
	Migración de Deuda (Administrativa, Judicial, Paga, Impaga, Anulada, Exenta)
	Migración de Pagos de Deuda
	Migración de Convenios de Pago
	Migración de Saldos a Favor
	Migración de Exenciones
Migración de procuradores	
<b>Padrón</b>	Formulario de alta de comercio
	Formulario de modificación de comercio
	Formulario de baja de comercio
	Formulario cambio de domicilio de comercio
<b>Emisión</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección
	Adaptación de boletas de cuotas y recibos de cuotas de convenio

	Cálculo de Emisión y configuración de tablas para emisión
	Adaptar impresión masiva-repartidores
	Adaptación Estructura y Generación de Códigos de Barra
<b>Gestión de deuda</b>	Actualización de Deuda y Cuotas (Recargos resarcitorios)
<b>Gestión Judicial</b>	Adaptación de envío masivo a judicial
	Adaptar planillas y constancias
<b>Notificaciones</b>	Generar notificaciones masivas
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>PDC</b>	Adaptar tickets
<b>Entidades de cobro</b>	Adaptación/configuración de Novedades de Transacciones de Pago para las distintas entidades de cobro habilitadas (pago fácil, banelco y link)
	Adaptación/configuración del proceso de Autorizaciones y Rechazos de Maestro de Débito Automático?
	Adaptación/configuración de procesos de generación de Información para Débito y PAS para las distintas entidades habilitadas
<b>Adaptaciones Generales</b>	Adaptaciones menores de Impresiones y Reportes Generales (títulos, áreas y logos)
<b>Requerimientos adicionales</b>	R001-Configurar sellados.
	R014-Implementar Nuevo Código de Barra
<b>Entornos</b>	Soporte al despliegue en entorno de producción
<b>Seguridad</b>	R015-Crear usuarios y configurar roles
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Soporte a la instalación de Punta de Caja
	Capacitación de Capacitadores
<b>Soporte</b>	Soporte correctivo post producción

## Producto 2 - TGI

Módulo/Actividad	Funcionalidad
<b>Relevamiento</b>	Validación del relevamiento de la tasa
<b>Configuración</b>	Configuración Atributos y Tipo Objeto Imponible
	Configuración Recurso / Vencimientos TGI
	Configuración Recurso / Vencimientos TGI RURAL
	Configuración de Punta de Caja
	Configuración de parámetros del sistema (alícuotas, mínimos, % de excepción, etc )

<b>Migración</b>	Migración de Cuentas y Objetos Imponibles
	Migración de Atributos
	Migración de Titulares
	Migración de Domicilios
	Migración de Calles, Barrios, Localidad
	Migración de Eventos de la cuenta
	Migración de Broches
	Migración de Relaciones de Cuenta
	Migración de Deuda (Administrativa, Judicial, Paga, Impaga, Anulada, Exenta)
	Migración de Pagos de Deuda
	Migración de Convenios de Pago
	Migración de Saldos a Favor
	Migración de Exenciones
	Migración de procuradores
<b>Padrón</b>	Formulario de alta de parcela
	Formulario de modificación de parcela
	Formulario de baja de parcela
	Formulario modificación domicilio de parcela
	Formulario modificación domicilio de envío de boleta
<b>Emisión</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección TGI
	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección TGI RURAL
	Cálculo de Emisión y configuración de tablas para emisión TGI
	Cálculo de Emisión y configuración de tablas para emisión TGI RURAL
	Adaptar impresión masiva-repartidores
<b>Gestión de deuda</b>	Actualización de Deuda y Cuotas (Recargos resarcitorios)
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>PDC</b>	Adaptar tickets
<b>Entidades de cobro</b>	Adaptación/configuración de Novedades de Transacciones de Pago para las distintas entidades de cobro habilitadas (pago fácil, banelco y link)
	Adaptación/configuración del proceso de Autorizaciones y Rechazos de Maestro de Débito Automático?
	Adaptación/configuración de procesos de generación de Información para Débito y PAS para las distintas entidades habilitadas
<b>Adaptaciones Generales</b>	Adaptaciones menores de Impresiones y Reportes Generales (títulos, áreas y logos)

<b>Requerimientos adicionales</b>	R006-Revisar compatibilidad de base de datos de personas de la provincia de Santa Fe para obtener el número de CUIT y Documentos de sus contribuyentes
<b>Entornos</b>	Soporte al despliegue en entorno de producción
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Capacitación de Capacitadores
<b>Soporte</b>	Soporte correctivo post producción

### Producto 3 - CDM

Módulo/Actividad	Funcionalidad
<b>Relevamiento</b>	Validación del relevamiento de la tasa
<b>Configuración</b>	Configuración Recurso / Vencimientos CDM
<b>Migración</b>	Migración de Cuentas
	Migración de Eventos de la cuenta
	Migración de Relaciones de Cuenta
	Migración de Deuda (Administrativa, Judicial, Paga, Impaga, Anulada, Exenta)
	Migración de Pagos de Deuda
	Migración de Convenios de Pago
	Migración de Saldos a Favor
<b>Padrón</b>	Formulario de alta de cuenta
	Formulario de modificación de cuenta
	Formulario de baja de cuenta
<b>Emision</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección
	Cálculo de Emisión y configuración de tablas para emisión
<b>Gestión de deuda</b>	Actualización de Deuda y Cuotas (Recargos resarcitorios)
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>Entornos</b>	Correcciones sobre los requerimientos ya desarrollados
	Soporte al despliegue en entorno de producción
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Capacitación de Capacitadores
<b>Soporte</b>	Soporte correctivo post producción

### Producto 4 - Ingresos Varios (Boletas de cobro)

Módulo/Actividad	Funcionalidad
------------------	---------------

<b>Relevamiento</b>	Validación del relevamiento de la tasa
<b>Configuración</b>	Configuración Recurso / Vencimientos recurso Ingresos Varios
<b>Emision</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión
	Formulario para emisión de boletas de cobro
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Capacitación de Capacitadores
<b>Entornos</b>	Soporte al despliegue en entorno de producción
<b>Soporte</b>	Soporte post producción Ingresos Varios

**Notas:**

- Transferencia Tecnológica, se está considerando 1 sesión de 4hs aprox. por producto.
- Capacitación de Capacitadores, se está considerando 1 sesión de 8hs aprox. por producto.

**Estimación preliminar de la implementación en Sunchales**

Producto	Estimación (hs)
1- Derecho de Registro e Inspección	610
2 - Tasa General de Inmuebles	730
3 - Contribución de Mejoras	360
4 - Ingresos Varios	140

**Notas:**

- No se incluyen en la estimación las horas correspondientes al traslado del personal al Municipio

## Fuera de Alcance para la implementación en Sunchales

- Implementación del sistema del municipio que gestiona el pago de multas que nombramos en el documento como: “STA - Sistema de Pago de Multas”

## Supuestos para la implementación en Sunchales

- La base de personas va a ser la de SIAT-MmyCC (la de la provincia de santa fe)
- La interacción con sistema externo RAFAM se realizará por medio de archivos
- El empadronamiento de los comercios e industrias se realizará mediante los formularios de SIAT-MMyCC.

## Adaptaciones necesarias para SIAT-MMyCC detectadas en Sunchales

De las actividades mencionadas anteriormente es necesario adaptar SIAT-GPL para cumplir con las siguientes necesidades:

<b>Requerimientos adicionales</b>	R002-Implementación de la declaración jurada anual en el recurso DReI
	R003-Adaptar SIAT para que no se puedan generar convenios si el contribuyente posee otro convenio vencido
	R004-Adaptar SIAT para que un convenio se active al abonar la primer cuota
	R005-Adaptar SIAT para que las multas se expresan en UCM (Unidad Codificación Municipal)
	R007-Generar los archivos de sincronización para utilizarlos en RAFAM
	R011-Gestión de cheques de terceros
	R012-Gestión de cupones tarjetas de crédito
	R013-Gestión de cupones tarjetas de débito

## Alcance de la implementación en Villa Gobernador Gálvez

A continuación se describen las actividades a realizar para obtener cada producto:

## Producto 1 - DRel

Módulo/Actividad	Funcionalidad
<b>Relevamiento</b>	Validación de relevamiento de la tasa
<b>Configuración</b>	Configuración Atributos y Tipo Objeto Imponible
	Configuración Recurso / Vencimientos
	Configuración de Punta de Caja
	Configuración de parámetros del sistema (alícuotas, mínimos, % de excepción, etc )
<b>Migración</b>	Migración de Cuentas y Objetos Imponibles
	Migración de Atributos
	Migración de Titulares
	Migración de Domicilios
	Migración de Calles, Barrios, Localidad
	Migración de Eventos de la cuenta
	Migración de Broches
	Migración de Relaciones de Cuenta
	Migración de Deuda (Administrativa, Judicial, Paga, Impaga, Anulada, Exenta)
	Migración de Pagos de Deuda
	Migración de Convenios de Pago
	Migración de Saldos a Favor
	Migración de Exenciones
Migración de procuradores	
<b>Padrón</b>	Formulario de alta de comercio
	Formulario de modificación de comercio
	Formulario de baja de comercio
	Formulario cambio de domicilio de comercio
<b>Emision</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección
	Adaptación de boletas de cuotas y recibos de cuotas de convenio
	Cálculo de Emisión y configuración de tablas para emisión
	Adaptación Estructura y Generación de Códigos de Barra
<b>Gestión de deuda</b>	Actualización de Deuda y Cuotas (Recargos resarcitorios)
<b>Gestión Judicial</b>	Adaptación de envío masivo a judicial
	Adaptar planillas y constancias
<b>Notificaciones</b>	Generar notificaciones masivas
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>PDC</b>	Adaptar tickets

<b>Entidades de cobro</b>	Adaptación/configuración de Novedades de Transacciones de Pago para las distintas entidades de cobro habilitadas (pago fácil, banelco y link)
	Adaptación/configuración del proceso de Autorizaciones y Rechazos de Maestro de Débito Automático
	Adaptación/configuración de procesos de generación de Información para Débito y PAS para las distintas entidades habilitadas
<b>Adaptaciones Generales</b>	Adaptaciones menores de Impresiones y Reportes Generales (títulos, áreas y logos)
<b>Requerimientos adicionales</b>	R004-Generar los archivos de sincronización para utilizarlos en sistemas propio de tesorería
<b>Entornos</b>	Soporte al despliegue en entorno de producción
<b>Seguridad</b>	R007-Crear usuarios y configurar roles
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Soporte a la instalación de Punta de Caja
	Capacitación de Capacitadores
<b>Soporte</b>	Soporte correctivo post producción

## Producto 2 - TGI

Módulo/Actividad	Funcionalidad
<b>Relevamiento</b>	Validación del relevamiento de la tasa
<b>Configuración</b>	Configuración Atributos y Tipo Objeto Imponible
	Configuración Recurso / Vencimientos TGI
	Configuración Recurso / Vencimientos TGI RURAL
	Configuración de Punta de Caja
	Configuración de parámetros del sistema (alícuotas, mínimos, % de excepción, etc )
<b>Migración</b>	Migración de Cuentas y Objetos Imponibles
	Migración de Atributos
	Migración de Titulares
	Migración de Domicilios
	Migración de Calles, Barrios, Localidad
	Migración de Eventos de la cuenta
	Migración de Broches
	Migración de Relaciones de Cuenta
	Migración de Deuda (Administrativa, Judicial, Paga, Impaga, Anulada, Exenta)
	Migración de Pagos de Deuda

	Migración de Convenios de Pago
	Migración de Saldos a Favor
	Migración de Exenciones
	Migración de procuradores
<b>Padrón</b>	Formulario de alta de parcela
	Formulario de modificación de parcela
	Formulario de baja de parcela
	Formulario modificación domicilio de parcela
	Formulario modificación domicilio de envío de boleta
<b>Emisión</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección TGI
	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección TGI RURAL
	Cálculo de Emisión y configuración de tablas para emisión TGI
	Cálculo de Emisión y configuración de tablas para emisión TGI RURAL
	Adaptar impresión masiva-repartidores
<b>Gestión de deuda</b>	Actualización de Deuda y Cuotas (Recargos resarcitorios)
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>PDC</b>	Adaptar tickets
<b>Entidades de cobro</b>	Adaptación/configuración de Novedades de Transacciones de Pago para las distintas entidades de cobro habilitadas (pago fácil, banelco y link)
	Adaptación/configuración del proceso de Autorizaciones y Rechazos de Maestro de Débito Automático?
	Adaptación/configuración de procesos de generación de Información para Débito y PAS para las distintas entidades habilitadas
<b>Adaptaciones Generales</b>	Adaptaciones menores de Impresiones y Reportes Generales (títulos, áreas y logos)
<b>Requerimientos adicionales</b>	R003-Revisar compatibilidad de base de datos de personas de la provincia de Santa Fe para obtener el número de CUIT y Documentos de sus contribuyentes
<b>Entornos</b>	Soporte al despliegue en entorno de producción
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Capacitación de Capacitadores
<b>Soporte</b>	Soporte correctivo post producción

### Producto 3 - TCC (Tasa Conservación de cementerio)

Las tareas para realizar la tasa es una planificación tentativa ya que originalmente no se encuentra en el

relevamiento original, sin embargo se detecta como necesaria su implementación

Módulo/Actividad	Funcionalidad
<b>Relevamiento</b>	Validación del relevamiento de la tasa
<b>Configuración</b>	Configuración Recurso / Vencimientos TCC
<b>Migración</b>	Migración de Atributos
	Migración de Titulares
	Migración de Domicilios
	Migración de Calles, Barrios, Localidad
	Migración de Eventos de la cuenta
	Migración de Broches
	Migración de Relaciones de Cuenta
	Migración de Cuentas
	Migración de Eventos de la cuenta
	Migración de Relaciones de Cuenta
	Migración de Deuda (Administrativa, Judicial, Paga, Impaga, Anulada, Exenta)
	Migración de Pagos de Deuda
	Migración de Convenios de Pago
	Migración de Saldos a Favor
<b>Padrón</b>	Formulario de alta de cuenta
	Formulario de modificación de cuenta
	Formulario de baja de cuenta
<b>Emisión</b>	Adaptación de boletas de deuda, recibos deuda para reimpresión/reconfección
	Cálculo de Emisión y configuración de tablas para emisión
	Adaptar impresión masiva-repartidores
<b>Gestión de deuda</b>	Actualización de Deuda y Cuotas (Recargos resarcitorios)
<b>Balance</b>	Configurar partidas y distribuidores de partidas
<b>Seguridad</b>	Crear usuarios y roles
<b>Entornos</b>	Correcciones sobre los requerimientos ya desarrollados
	Soporte al despliegue en entorno de producción
<b>Actividades para producción</b>	Correcciones sobre los requerimientos ya desarrollados
	Capacitación de Capacitadores
<b>Soporte</b>	Soporte correctivo post producción

## Producto 4 - Ingresos Varios (Boletas de cobro)

Módulo/Actividad	Funcionalidad
Relevamiento	Validación del relevamiento de la tasa
Configuración	Configuración Recurso / Vencimientos recurso Ingresos Varios
	Configurar Recurso Derecho Ocupación Vía Pública
Emision	Adaptación de boletas de deuda, recibos deuda para reimpresión
	Formulario para emisión de boletas de cobro
Balance	Configurar partidas y distribuidores de partidas
Seguridad	Crear usuarios y roles
Actividades para producción	Correcciones sobre los requerimientos ya desarrollados
	Capacitación de Capacitadores
Entornos	Soporte al despliegue en entorno de producción
Soporte	Soporte post producción Ingresos Varios

### Notas:

- Transferencia Tecnológica, se está considerando 1 sesión de 4hs aprox. por producto.
- Capacitación de Capacitadores, se está considerando 1 sesión de 8hs aprox. por producto.

## Estimación preliminar de la implementación en Villa Gobernador Gálvez

Producto	Estimación (hs)
1- Derecho de Registro e Inspección	620
2 - Tasa General de Inmuebles	700
3 - Tasa Conservacion Cementerio	650
4 - Ingresos Varios	150

### Notas:

- No se incluyen en la estimación las horas correspondientes al traslado del personal al Municipio

## Fuera de Alcance para la implementación en Villa Gobernador Galvez

- Quedarán excluidos de la implementación de los sistemas del municipios que llamamos:
  - STA - Tesorería (parte del sistema estaría fuera de SIAT)
  - STA - Contaduría
  - STA - Compras
  - STA - Sueldos

## Supuestos para la implementación en Villa Gobernador Gálvez

- La base de personas va a ser la de SIAT-MmyC (db de personas de la provincia de santa fe)
- La interacción con sistema externo "Tesorería" se realizará por medio de archivos.
- El empadronamiento de los comercios e industrias se realizará mediante los formularios de SIAT-MMyCC.

## Adaptaciones necesarias para SIAT-MMyCC detectadas en VGG

De las actividades mencionadas anteriormente es necesario adaptar SIAT-GPL para cumplir con las siguientes necesidades:

<b>Requerimientos adicionales</b>	R001-Adaptar SIAT para que no se puedan generar convenios si el contribuyente posee otro convenio vencido
	R002-Adaptar SIAT para que los convenios o planes de pago se calculen con interés directo.
	R004-Generar los archivos de sincronización para utilizarlos en sistemas propio de tesorería
	R006-Gestión de cheques de terceros

## Adaptaciones mínimas necesarias para la implementación en Sunchales y Villa Gobernador Gálvez.

Módulo/Actividad	Funcionalidad
<b>Punta de Caja</b>	Implementar cobro con Cheques de Terceros
	Implementar cobro con Tarjeta de Débito
	Implementar cobro con Tarjeta de Crédito
<b>Gestión deuda</b>	Implementación de la declaración jurada anual en el recurso DRel
	Adaptar SIAT para que no se puedan generar convenios si el contribuyente posee otro convenio vencido
	Adaptar SIAT para que un convenio se active al abonar la primer cuota
	Adaptar SIAT para que las multas se expresan en UCM (Unidad Codificación Municipal)
	Adaptar SIAT para que los convenios o planes de pago contengan la opción de seleccionar el tipo de interés a aplicar, entre (Frances y Directo).
<b>Balance</b>	Generar los archivos de sincronización para utilizarlos en RAFAM
	Gestión de cheques de terceros
	Gestión de cupones tarjetas de crédito
	Gestión de cupones tarjetas de débito

### Cuestiones relacionadas a la utilización de la base de personas de la pvcia.

En lo que respecta a la utilización de la base de datos de personas de la provincia de Santa Fe y según lo relevado en Sunchales y VGG surge la problemática en el recurso TGI donde las cuentas tienen una antigüedad importante y en el momento en que se dieron de alta los sistemas no exigían que se ingrese el número de documento o el número de CUIT del contribuyente. Como resultado de esta situación muchas cuentas solo poseen la descripción del titular y en muchas casos no respetando un patrón de ingreso (se pueden encontrar "J. Gonzalez", "Gonzalez Jorge", "Martinez M. L", etc) y sin el número de documento, haciendo que el proceso de unificación de titulares de cuentas con la persona es muy complejo de llevar adelante.

Respecto al recurso DRel encontramos que por lo general las cuentas poseen cargadas el número de CUIT del contribuyente, encontramos factible la utilización de la base de



datos de personas de la provincia ya que es posible la asociación del CUIT de los titulares con el CUIT de la persona en la base de datos de la provincia.

**ACLARACIÓN:** Las estimaciones realizadas podrían ser modificadas por cambios en las próximas etapas del proyectos o las condiciones de los municipios al momento de la implementación.