



Sistema: **Simple BPM**

Manual de Usuario

Creación:	16/6/2019	Autor:	Danis Chaviano
Versión:	1.1.0	Plantilla:	SS-GENERAL-TE-30-ES
Código:	AG-EVSIMPLE-MU-01-ES	Página:	1 de 29

 *Estamos comprometidos con el medioambiente - Antes de imprimir este documento considere si es necesario.*

Índice de contenido

1	Introducción	5
2	Propósito	5
3	Audiencia	5
4	Objetos de Negocio	6
4.1	OBN	6
4.1.1	Descripción	6
4.1.2	Atributos	6
4.2	Consultas de acceso (CA)	7
4.3	Acciones de tipo OBN	7
4.4	Variables de tipo OBN	8
4.5	Integración con componentes	8
4.6	Componente Tabla de Datos	9
4.7	Componente Select OBN	10
4.8	Formulario para Tabla de Datos	10
5	Auditoría	11
5.1	Auditoría del sistema	11
6	Versionado de procesos	12
6.1	Versiones de Procesos	12
6.2	Importar procesos	13
6.3	Editar procesos	13
7	Componente Agenda Múltiple	14
7.1	Configuración Backend	14
7.1.1	Validación de Agenda Múltiple	15
7.2	Uso del componente Agenda Múltiple	16
8	Acción Buscar Usuario	16
9	Catálogo de servicios REST	17
9.1	Configuración de Servicios REST mediante el catálogo:	17

9.2	Configuración de las Operaciones de los servicios REST:	17
9.3	Configuración de respuestas en la operaciones	18
10	Pasarela de pagos ITC versión 4.....	19
10.1	Configuración de Pasarela de pagos.....	19
10.2	Configuración de parámetros para tiempo de espera	20
10.3	Componente Pagos Suscripción	21
10.4	Componente Pagos.....	22
11	Borrado lógico de trámites	22
12	Componente preguntas captcha	24
12.1	Funcionamiento en un trámite.	25
13	Componente Google recaptcha (invisible).	25
13.1	Crear una cuenta en google recaptcha.....	25
13.2	Editar datos de la cuenta en el manager y en el backend.	27
13.3	Utilizar el componente en el modelado.	28
13.4	Ejemplo en un formulario.....	28
14	Historial de cambios	30

1 Introducción

En este documento se describe la información necesaria para utilizar las nuevas herramientas brindadas por el sistema Simple BPM: **Objetos de Negocio, Auditoría, Versionado de procesos, Catálogo de servicios REST, Pasarela de pago ITC versión 4.0, Componente SAE múltiple y Borrado lógico de trámites.**

2 Propósito

Este documento tiene por objetivo explicar el funcionamiento de las nuevas herramientas brindadas por el sistema Simple BPM.

3 Audiencia

Se encuentra dirigido a los funcionarios de Agestic y a los usuarios que utilicen el sistema.

4 Objetos de Negocio

4.1 OBN

Objetos de Negocio (OBN) en SIMPLE es una funcionalidad que permite almacenar y hacer uso de datos entre diferentes procesos, permitiendo tener almacenado de forma ajena a los trámites información que consideremos relevante.

Esta funcionalidad está implementada en SIMPLE a partir de la versión 2.1.

4.1.1 Descripción

La creación o importación/exportación de un OBN se realiza mediante la opción en la barra de tareas **Objeto de Negocios**.

Al momento de crear un OBN, los datos solicitados de forma obligatoria serán Identificador, Descripción y al menos un atributo. El Identificador es un valor único, que no podrá ser repetido entre distintos OBN.

El proceso de importación y exportación incluye toda la estructura de un OBN (atributos y consultas de acceso) pero no incluye los registros existentes.

4.1.2 Atributos

Los atributos se definen dentro de un OBN y existen varios tipos:

- **Cadena de caracteres:** Acepta cualquier tipo de caracteres y se le define un máximo de caracteres permitidos.
- **Lógico:** Permite los valores 1 o 0.
- **Caracter:** Almacena un único carácter.
- **Entero:** Permite únicamente números enteros.
- **Real:** Permite únicamente números reales.
- **Fecha:** Permite únicamente una fecha con formato dd/mm/aaaa.
- **Hora:** Permite únicamente una hora con formato hh:mm
- **OBN:** Es un atributo que hace referencia a otro OBN previamente creado.

Todos los atributos pueden ser marcados como **múltiples**, esto quiere decir que tendrán más de un posible valor para cada registro dentro de la OBN. Un atributo seleccionado como múltiple no podrá ser clave lógica del OBN.

Los atributos marcados como **clave lógica** determinarán que su contenido no puede ser repetido entre los registros de la OBN. En caso de existir más de un atributo como clave lógica, aplicará el mismo concepto solo que lo que no se podrá repetir es el conjunto de datos de todos los atributos.

Todos los atributos pueden ser marcados como clave lógica a excepción del atributo de tipo OBN.

4.2 Consultas de acceso (CA)

Las consultas de acceso (CA) de un OBN son utilizadas para obtener datos filtrando por sus atributos.

Existen 2 tipos de CA, las de tipo **Contador** y las de tipo **Obtener**.

Las de tipo **Contador**, una vez ejecutada su consulta, nos retornará únicamente un número

correspondiente a la cantidad de registros devueltos por la consulta.

Las consultas de tipo **Obtener** retornarán los datos almacenados en la OBN que coincidan con la consulta ejecutada.

Las CA que se permiten ingresar son de tipo básicas. No se puede mediante una consulta de un OBN1 obtener datos del OBN2.

4.3 Acciones de tipo OBN

Las acciones de tipo OBN sirven para hacer interactuar un OBN con un proceso creado en SIMPLE.

Existen varios tipos de acciones OBN:

- **Iniciar OBN:** Esta acción permite inicializar un OBN con el fin de insertar un nuevo registro. Esta acción no traerá ningún dato almacenado del OBN, solo servirá para crear una variable donde almacenar datos.
- **Guardar OBN:** Esta acción permite ingresarle como valor una variable de tipo OBN para dar de alta un nuevo registro en un OBN. Esta acción no genera variables, únicamente crea nuevos registros dentro de un OBN. Las variables asociadas a esta acción deben ser generadas previamente mediante la acción **Iniciar OBN** o **Consultar**.
- **Obtener todo:** Esta acción sirve para obtener todos los registros existentes dentro de un OBN.
- **Consultar:** La acción consultar hace referencia a una CA de un OBN, pudiendo ser cualquier tipo (CA de tipo Obtener o Consultar) y permitiendo determinar para el caso de la CA de tipo Obtener si se quiere recibir un único valor o una colección de datos (todos los registros que coincidan con la consulta ejecutada).

Estas acciones se ejecutan de igual manera que el resto de las acciones de SIMPLE aplicando la lógica necesaria en el proceso.

4.4 Variables de tipo OBN

Todas las acciones de tipo OBN generan variables, que a diferencia de las ya existentes en SIMPLE, se utilizan de la forma **\$variable**.

Sin ser la acción de tipo Guardar OBN que no genera una variable sino que sirve para modificar o dar de alta un nuevo registro en un OBN, el resto de las acciones generan una variable que dependiendo su uso se pueden utilizar los recursos del OBN de la manera **\$variable.atributo**.

Para el caso de un atributo de tipo OBN, se puede utilizar la variable de la forma \$variable.atrOBN.atributo únicamente en modo de visualización.

Para el caso de un atributo de tipo OBN multiple, se puede usar la combinación **\$variable.atrOBN.atributoOBNLista**. Concatenando la palabra **Lista** seguida del nombre del atributo e iniciando con mayúscula para obtener la lista de ids de los obn asociados en esa relación.

4.5 Integración con componentes

Las variables OBN puede ser utilizadas en todos los componentes que permiten valor por defecto. Su comportamiento es exactamente el mismo que las variables comunes de SIMPLE. Otro de los usos que se les puede dar es para asociar un atributo de una variable OBN a un componente o directamente la variable devuelta por una acción a un componente.

Los componentes que permiten asociar variables OBN utilizando el formato **\$obn.atributo** son:

- **Textbox, Textarea, Select, Radio, Fecha, Tabla.**

Asociar un componente SIMPLE a un atributo de una variable OBN sirve para que al momento de querer insertar un registro en el OBN se carguen los atributos del mismo en base a lo ingresado en los componentes. Esta inserción se realiza mediante la acción **Guardar OBN**.

Si al momento de guardar un registro en un OBN donde las claves lógicas enviadas ya existen, el registro se actualizará siempre y cuando la variable se haya originado a partir de una acción **Consultar**. Si la variable OBN se creó a partir de una acción de **Iniciar OBN** se desplegará un mensaje de error de que ya existe un registro en el OBN con esos datos.

No se permite modificar datos de un OBN configurada como atributo de otro OBN. Estos atributos solo se podrán visualizar configurándolos como valor por defecto de la manera **\$obn.atrOBN.atributo**.

4.6 Componente Tabla de Datos

El componente **Tabla de Datos** es de uso específico para OBN. Este componente se puede asociar a una variable OBN o un atributo de tipo OBN. Existen 3 tipos de Tabla de Datos:

- **Asociar:** Las tablas de este tipo se deben asociar a atributos de tipo OBN de la forma **\$variable.atributo** obtenidas de la acción **Consultar** y sirven para asociar una o mas OBN (dependiendo de si es múltiple o no) a una instancia de la OBN ya creada. Las tablas de este tipo no me permiten crear nuevos registros en la OBN configurada como atributo.
- **Visualización:** Las tablas de datos de este tipo sirven únicamente para visualizar el resultado de una acción de tipo **Consultar** o **Obtener todo**. Permite asociar tanto variables como atributos de una OBN.
- **ABM:** La tabla de tipo ABM aparte de permitir la funcionalidad de la tabla **Asociar**, también permite dar de alta nuevos registros en un OBN, borrarlos o modificarlos. A una tabla ABM se le puede asociar tanto variables como atributos OBN. A las tablas de datos de este tipo se le tienen que asociar un formulario de tipo **Formulario para Tabla de Datos** donde se ingresarán los datos del nuevo registro o modificarán los de uno existente.

A la Tabla de Datos se le configura la cantidad de filas por páginas. Si la cantidad de registros a mostrar excede el límite de filas configurado la tabla realizará un paginado de los resultados.

Las columnas que se configuran en la Tabla de Datos tienen que corresponder con los atributos del OBN asociado a la tabla. Para cada columna se permite ingresar una etiqueta y un atributo. La etiqueta será lo que se mostrará como nombre de la columna y el atributo deberá tener **exactamente el mismo nombre** que tiene el atributo definido en el OBN. De esta forma podremos configurar que atributos del OBN queremos visualizar en la tabla.

No se permite visualizar atributos de tipo OBN en una tabla de datos. Para visualizar el

contenido de atributos de tipo OBN se deberá hacer mediante la **integración con los componentes** de SIMPLE de la forma **\$obn.atrOBN.atributo**.

4.7 Componente Select OBN

El componente **Select OBN** es de uso específico para OBN. Este componente se puede asociar a una variable OBN o un atributo de tipo OBN. Existen 3 tipos de Tabla de Datos:

Configuración:

- **OBN:** OBN de tipo OBN Múltiple, puede ser atributos de OBN o una variable obn múltiple Ejemplo. **\$getPersona.obnEmpresa.obnOficinas**.
- **Etiqueta OBN:** Atributo del obn definido en el campo **OBN** para mostrar como etiqueta en las opciones del select.
- **Valor OBN:** Atributo del obn definido en el campo **OBN** para definir como valor de las opciones del select.

4.8 Formulario para Tabla de Datos

Los formularios para tabla de datos son accedido únicamente mediante el componente **Tabla de Datos** de tipo **ABM**.

El nombre de los componentes ingresados en dicho formulario deben coincidir con el nombre del atributo del OBN en el cual se están alterando sus registros. Estos componente no crearán variables SIMPLE para ser utilizadas en el proceso, únicamente registrarán los cambios dentro del OBN. Si en el formulario se ingresaron componentes con nombres que no coinciden con los atributos, el contenido ingresado será descartado.

En el componente **Tabla de datos** solo se puede utilizar el de tipo **Asociar**.

5 Auditoría

5.1 Auditoría del sistema

La Auditoría del Sistema se realiza de forma automática a partir de la versión 2.0 de Simple.

Para el acceso a la consulta de la Auditoría el usuario debe tener el acceso de **Seguimiento > Auditoría** o ser usuario **Admin** del sistema.

Es accesible desde el Backend en el menú **Seguimiento > Auditoría**. Se desglosa por las diferentes funciones que son auditadas, por ejemplo: Proceso, mediante el cual se puede ver el registro de todos los cambios del proceso en el que se observan las Tareas, Formularios, Campos, Acciones ,etc.

Cuenta con un sistema de filtros para acotar las búsquedas por **Fecha de último cambio** y/o **Operación realizada** (Alta, Modificación o Baja).

También se realiza auditoría sobre: **Catálogo de servicios, Catálogo de bloques, Pasarela de pagos, Reportes por proceso, Usuarios frontend, Usuarios backend y Grupos de usuarios**. De cada uno de estos se puede revisar los cambios de cada una de sus sub-categoría como por ejemplo: **Catálogo de Servicios**, mediante este se puede acceder a las **Operaciones** de cada servicio.

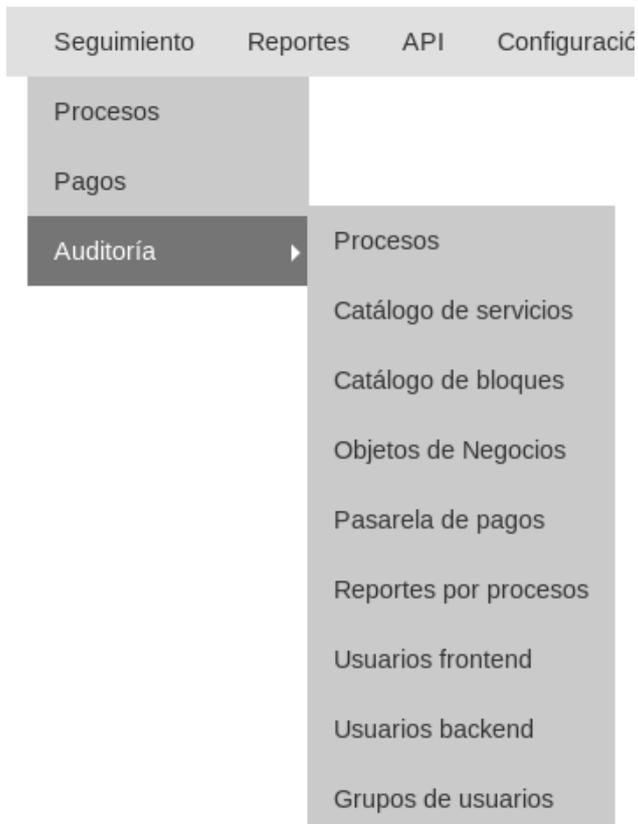


Imagen 1: Menú de Auditoría.

6 Versionado de procesos

6.1 Versiones de Procesos

El modelador de procesos podrá editar una versión de proceso o generar una nueva a partir de la versión publicada. Los procesos se clasificarán en tres estados (Publicado, Archivado, Borrador). En el listado siempre aparecerá el proceso en estado publicado de no existir una versión en este estado se mostrará la versión en estado Borrador.

En el listado de procesos se podrá publicar una nueva versión que se encuentre como estado Borrador. Al publicar una versión, deberá seleccionar si aplicar el cambio para los nuevos trámites o aplicarlo para todos los trámites incluyendo los trámites en cursos y los terminados. Para esta segunda opción se debe tener presente que si en el modelado existen grandes cambios puede que los trámites viejos no sean compatibles teniendo campos nuevos o campos que y tareas que ya no existan en esta nueva versión.

En el listado de trámites sólo se muestra la versión del proceso publicada, teniendo en cuenta que solo puede existir una versión publica de un proceso.

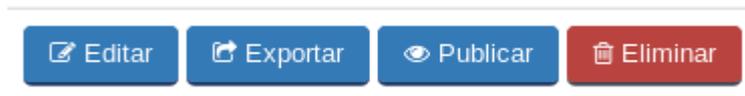


Imagen 2: Acciones

sobre un proceso.

6.2 Importar procesos

En el listado de proceso existen 4 operaciones a realizar :

- **Nuevo:** Crea un nuevo proceso el cual queda como versión 1 y en estado borrador.
- **Importar:** Importa un proceso y queda en estado borrador y versión 1.
- **Importar versión:** Se crea una nueva versión de un proceso seleccionado. Tener en cuenta que esta nueva versión podrá ser asociada a la hora de publicarla con los trámites viejos por lo que se debe estar seguro que es una versión del proceso seleccionado.
- **Clonar:** Brinda la opción de clonar una versión seleccionada de un determinado proceso, creando un proceso idéntico pero como versión 1 y en estado borrador.

Listado de Procesos

Listado de Procesos



Imagen 3: Menú de opciones del Listado de proceso.

6.3 Editar procesos

En la edición del proceso se puede acceder a las distintas versiones del mismo y publicar la versión que se seleccione u ocultar la versión publicada.

Cuando una versión es publicada, la versión actual pasa a estado Archivado. Al entrar a la edición de una versión no publicada cambia su estado a borrador.

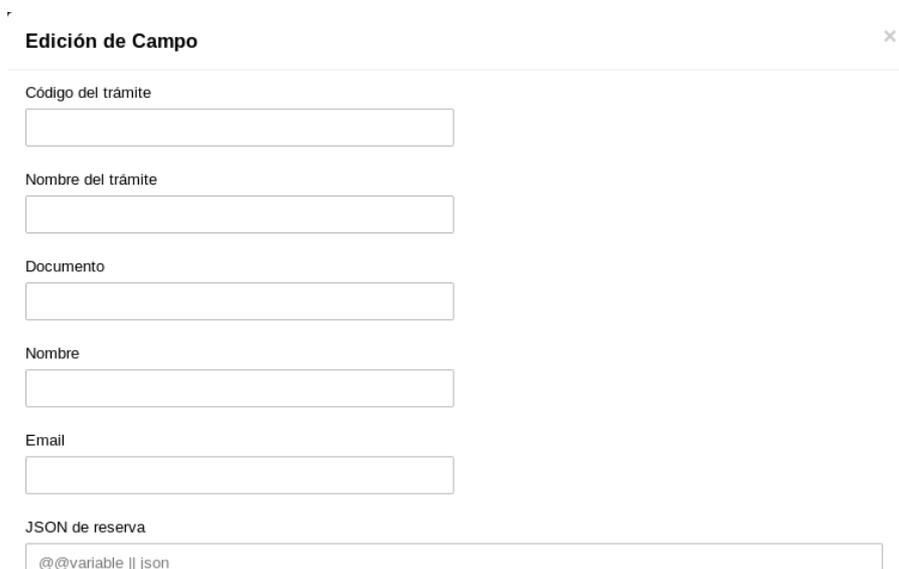
7 Componente Agenda Múltiple

Componente similar a "**Agenda embebida**" que permitirá realizar ABM de múltiples reservas utilizando WS de reservas múltiples de SAE. El componente esperará recibir los datos de las personas en formato JSON. Los datos pueden ser asociados a una variable con el contenido en formato JSON, esta puede ser una de un componente **tabla**.

Componente disponible a partir de la versión 2.0-r1 de la herramienta.

7.1 Configuración Backend

Para configurar la agenda múltiple además de los parámetros del componente **Agenda Embebida** se deben definir nuevos parámetros que se describen a continuación:



Edición de Campo

Código del trámite

Nombre del trámite

Documento

Nombre

Email

JSON de reserva

- **Código del trámite** (Requerido): Código del trámite al cual estarán asociadas todas las reservas múltiples del bloque.
- **Nombre del trámite** (Opcional): Nombre del trámite que se envía a trazabilidad.
- **Documento** (Requerido): Número de documento de la persona que realizará las

reservas múltiples.

- **Nombre** (Requerido): Nombre y apellido de la persona que realizará las reservas múltiples.
- **Email** (Requerido): Dirección de correo electrónico de la persona que realizará las reservas múltiples.
- **JSON de reserva** (Requerido): Dato recibido mediante una variable de simple (variable donde se almacene un JSON, puede ser un componente **tabla**) o un JSON, donde se especifiquen los datos de las personas que se van a agendar.
- **Datos de SAE** (Requerido): Una tabla donde es ingresando el nombre de los campos de SAE con el formato agrupación.variable asociándolo a la posición donde se encontrará en el JSON que va a recibir de SIMPLE (**JSON de reserva**).

Datos de SAE

Agrupación	Variable	Posición JSON	Visible
datos_personales	TipoDocumento	0	<input checked="" type="checkbox"/>
datos_personales	NroDocumento	1	<input checked="" type="checkbox"/>
datos_personales	Mail	2	<input type="checkbox"/>

Imagen 5: Ejemplo de configuración de Datos de SAE.

7.1.1 Validación de Agenda Múltiple

El componente Agenda Múltiple incluye nuevas validaciones, las cuales se describen a continuación:

- **validar_agenda_mult_sae_total**: Valida que todas las reservas estén realizadas.
- **validar_agenda_mult_sae**: Valida que se realizó al menos una reserva.

7.2 Uso del componente Agenda Múltiple

Al mostrar la Agenda, en caso de no haber configurado un recurso en específico se muestran los recursos disponibles para reservas múltiples, al seleccionar un recurso aparece el listado con los datos de las personas a agendar y se procede a realizar la reserva de todas las personas en la lista antes de confirmarla.

AGENDA MÚLTIPLE

Los campos indicados con

Seleccionar ubicación: Recurso_ultimo Recurso_2



Dirección: aoissmfn
Teléfono: 25130000

Seleccionar	Persona	Fecha	Hora	Acción
<input checked="" type="radio"/>	CI 6334161633			
<input type="radio"/>	CI 345345345			
<input type="radio"/>	P J719472			

Imagen 6: Agenda Múltiple

8 Acción Buscar Usuario

Dado la terna país-tipodocumento-nrodocumento busca dentro de los usuarios frontend (ciudadanos o funcionarios) el registro asociado a dicha terna.

El resultado está formado con la siguiente estructura:

En caso de encontrarlo devuelve un array con los datos

ej. ["1", "uy-ci-12345678", "Nombre Apellido"]

En caso de no ser encontrado un array indicando que no fue recuperado

ej. ["0","uy-ci-12345678","Usuario no encontrado"]

9 Catálogo de servicios REST

9.1 Configuración de Servicios REST mediante el catálogo:

Endpoint*: Ruta base del servicio. Ej. <https://test03.simple.pge.red.uy/simple/backend/api>

Método*: Métodos permitidos GET o POST



PDI SOAP REST

Activo

Nombre*

Endpoint*

Método* POST GET

Tiempo de espera de conexión y respuesta (segundos)

Imagen 7: Configuración de un servicio REST

9.2 Configuración de las Operaciones de los servicios REST:

Código*: Este campo tiene un valor autogenerado y no se puede editar. Es usado por simple para identificar la operación en el servicio.

Nombre real de la operación*: Parte del PATH del REST que identifica la operación del servicio.

Ej. /tramites

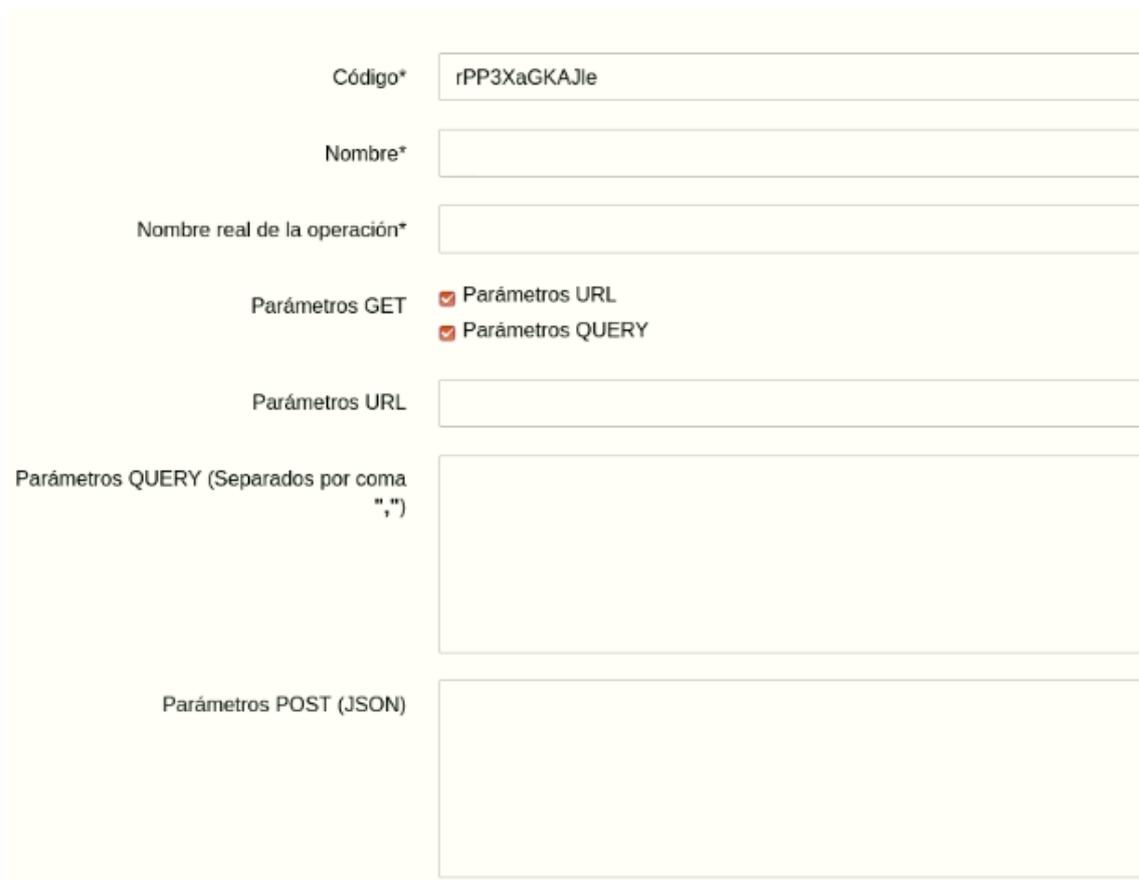
Parámetros GET: Opciones de envío de parámetros por la URL. Se definen en dos

Parámetros URL: Parámetros que forman parte del patch del servicio. Ej.

/@@idTramite, Estos pueden ser valores o variables de simple.

Parámetros QUERY: Los parámetros que se envían como datos en la url luego del carácter "?", Estos tienen que estar separados por coma ",". Ej. token=\$token, datos=@@datos. Los parámetros permiten el uso de variables de SIMPLE.

Parámetros POST: En caso del servicio ser configurado como método de envío POST. Los datos enviados por este método serán en formato JSON y permiten el uso de variables de SIMPLE.



The image shows a configuration form for REST services. It includes the following fields and options:

- Código***: Input field containing "rPP3XaGKAJle".
- Nombre***: Empty input field.
- Nombre real de la operación***: Empty input field.
- Parámetros GET**: Two checked checkboxes for "Parámetros URL" and "Parámetros QUERY".
- Parámetros URL**: Empty input field.
- Parámetros QUERY (Separados por coma ",")**: Empty input field.
- Parámetros POST (JSON)**: Empty input field.

Imagen 8: Configuración de las operaciones en los servicios REST

9.3 Configuración de respuestas en la operaciones

Las respuestas de los servicios deben ser tipo JSON. El JSON de respuesta será tratado como un XML permitiendo configurar las respuestas de igual forma que los servicios de tipo SOAP o PDI.

Ej.

JSON (Respuesta del servicio)	XML
<pre> { "arrayColores": [{ "nombreColor": "rojo", "valorHexadec": "#f00" }, { "nombreColor": "verde", "valorHexadec": "#0f0" }, { "nombreColor": "azul", "valorHexadec": "#00f" }] } </pre>	<pre> <root> <arrayColores> <element> <nombreColor>rojo</nombreColor> <valorHexadec>#f00</valorHexadec> </element> <element> <nombreColor>verde</nombreColor> <valorHexadec>#0f0</valorHexadec> </element> <element> <nombreColor>azul</nombreColor> <valorHexadec>#00f</valorHexadec> </element> </arrayColores> </root> </pre>

10 Pasarela de pagos ITC versión 4

La pasarela de pago cuenta con dos flujos de trabajo dividido en dos componentes de pagos.

- Pagos
- Pagos Suscripción

10.1 Configuración de Pasarela de pagos

Para crear la pasarela de pagos ITC en el backend de simple, menú Pasarela de Pagos → Nuevo → Método → Antel Pagos v4.0

Inicio
Modelador de Procesos
Catálogo de Servicios
Catálogo de Bloques
Objetos de Negocios
Pasarela de Pagos

Pasarela de Pagos / Pasarela

Pasarela

Datos generales

Activa

Nombre

Método

Datos del método

Código de comercio

Consumidor final

Vencimiento (AAAA/MM/DD HH:mm)

10.2 Configuración de parámetros para tiempo de espera

Parámetro	Valor	Descripción
espera_estado_automatica_pasarelav4	5000	Tiempo de espera para consultas de estado de pago.
espera_estado_manual_pasarelav4	5000	Tiempo de espera para consultas de estado de pago cuando el usuario da clic en "Consultar Estado" del pago.
espera_suscripcion	5000	Intervalo de espera para consultar

		nuevamente los medios de pago.
--	--	--------------------------------

* 1000 = un segundo

10.3 Componente Pagos Suscripción

El componente de Pagos Suscripción debe utilizarse para generar trámites de usuarios autenticados.

El componente Pagos Suscripción debe asociarse a una acción de pasarela tipo Antel v4.0. El componente genera un botón que dirige al usuario a la pasarela de pagos Antel para suscribir los medios de pagos que desee y luego ser redireccionado al sistema donde mostrará una lista su lista de medios de pagos suscritos y que sean válidos para el trámite ejecutado.

En esta ventana el usuario debe seleccionar el medio de pago con el que desea pagar el trámite.

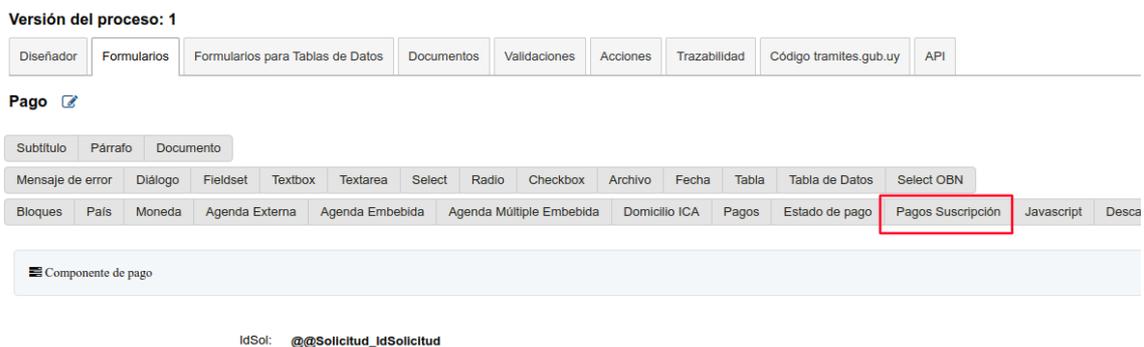


Imagen 9: Componente Pagos Suscripción. Edición del modelado de trámites.



Imagen 10: Componente Pagos Suscripción. Paso para suscribir los medios de pagos.

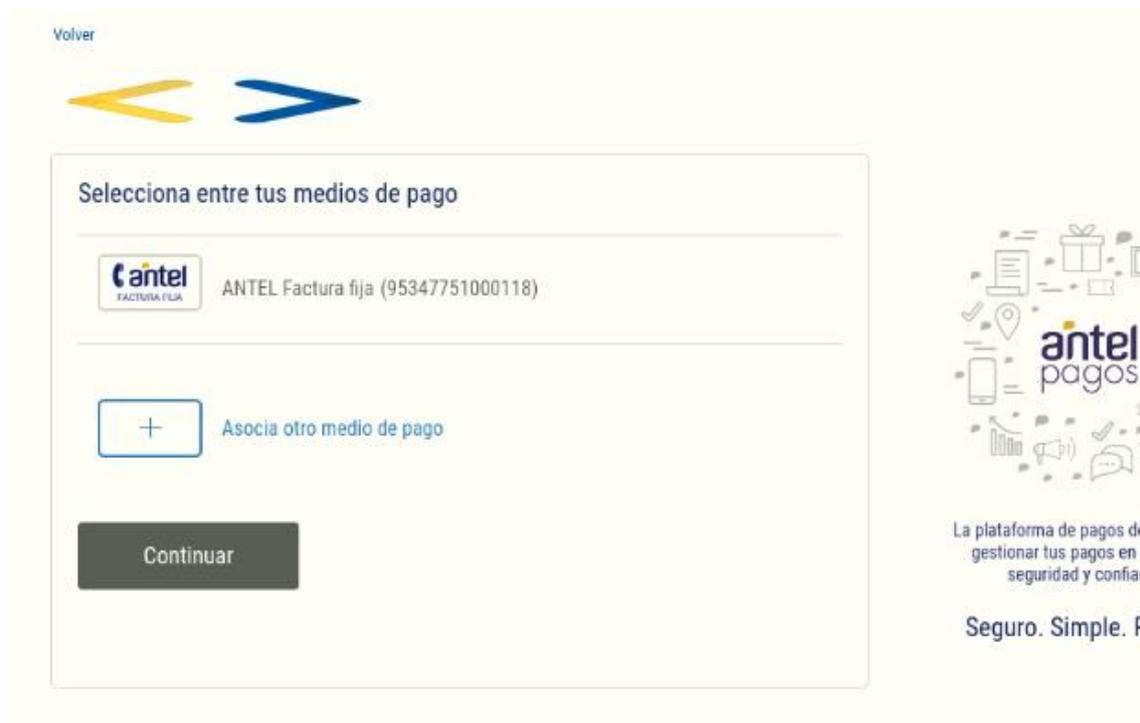


Imagen 11: Ventana de Antel-ePagos para asociar los Medios de Pagos

10.4 Componente Pagos

El componente de Pagos puede ser usado como usuario anónimo en trámites sin autenticación.

Para el modelado del componente Pagos debe asociarse a una acción de pasarela tipo Antel

v4.0. Cuando se inicia un trámite sigue el flujo de la antigua pasarela.

Se verifica si existe un pago, en caso de no existir se procede a realizar el pago en la plataforma Antel. Si existe el pago se verifica su estado y se muestra en pantalla. En caso de ser un pago pendiente se consulta el estado del pago y se muestra el resultado de la notificación enviada por Antel-ePagos para dicha consulta.

11 Borrado lógico de trámites

El borrado lógico de trámites es gestionado desde el backend por usuarios con el rol de seguimiento.

En el menú Seguimiento → Procesos. Se listan todos los procesos. Para cada proceso se muestra el botón **“Ver seguimiento”**. En esta venta es listado el registro de trámites del proceso seleccionado con la opción de **“Eliminar”**.



Id	Estado	Etapa actual	Documento	Fecha de creación	Fecha de Último cambio	Acciones
2163	En curso	Tarea	UY-ci-41143269	20 mar 2018 12:10:02	20 mar 2018 12:10:02	Seguim
2166	En curso	Tarea		20 mar 2018 12:51:27	20 mar 2018 12:51:27	Seguim
2179	En curso	Tarea		20 mar 2018 15:49:53	20 mar 2018 15:49:53	Seguim
2183	En curso	Tarea		20 mar 2018 16:45:26	20 mar 2018 16:45:26	Seguim

Imagen 12: Listado de trámites de un proceso

Cuando los trámites son borrados, estos se pueden recuperar accediendo a la ventana de trámites borrados como se muestra en la imagen Imagen 13: Acceso a la ventana de trámites borrados.



Imagen 13: Acceso a la ventana de trámites borrados

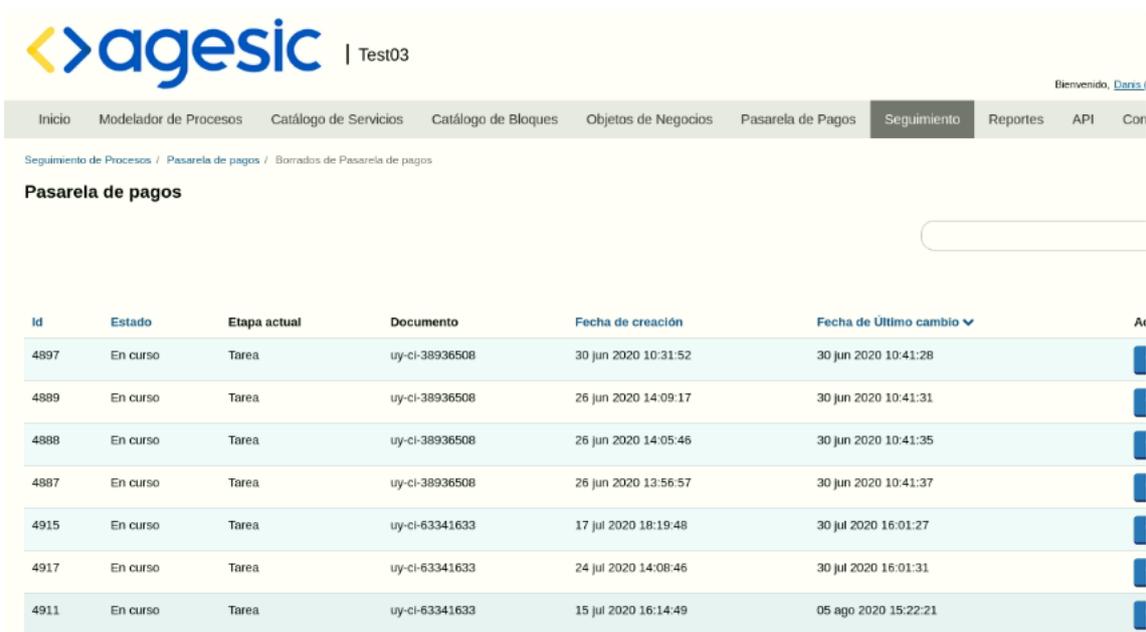


Imagen 14: Lista de trámites borrados con opción Recuperar trámites

12 Componente preguntas captcha

Este componente una vez agregado a un formulario, permite mostrar una pregunta al azar al usuario y capturar una respuesta, para evitar evitar ataques. El modelador deberá ubicar el componente donde lo crea conveniente. Sin embargo dada la utilidad y el objetivo del componente se sugiere ubicarlo al final del formulario. Actualmente no se tiene posibilidad de

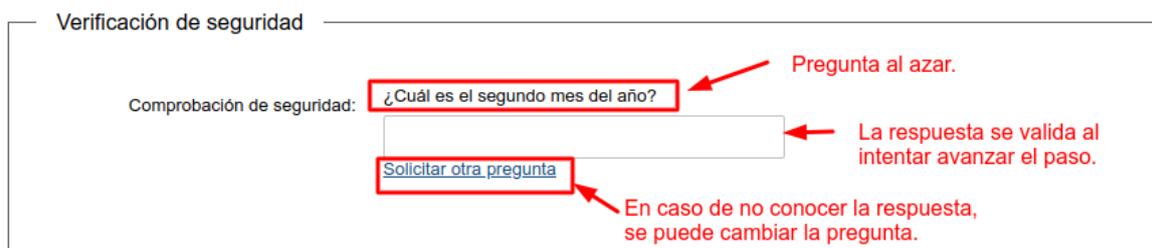
cargar las preguntas desde el backend de simple por lo que la carga de las preguntas deberá realizarse desde la base de datos directamente. A continuación se muestra un prototipo del nuevo componente disponible en la sección de componentes de un formulario.



Imagen 15: Componente de preguntas captcha

12.1 Funcionamiento en un trámite.

A continuación se puede evidenciar la pantalla en un trámite. En la que se pueden diferenciar diferentes secciones. Las preguntas serán cargadas al azar por el componente en tiempo de ejecución y el usuario podrá solicitar cambiarla si así lo desea mediante la opción “Solicitar otra pregunta”. En caso que el componente se encuentre en pantalla será requerido el ingreso de la respuesta a la pregunta para poder continuar con el paso del formulario.



Siguiente >

13 Componente Google recaptcha (invisible).

El componente Google recaptcha (invisible) permite validar un formulario mediante la API de google para detectar posibles ataques de un BOT, dicha verificación se ejecuta al momento de intentar avanzar el formulario.

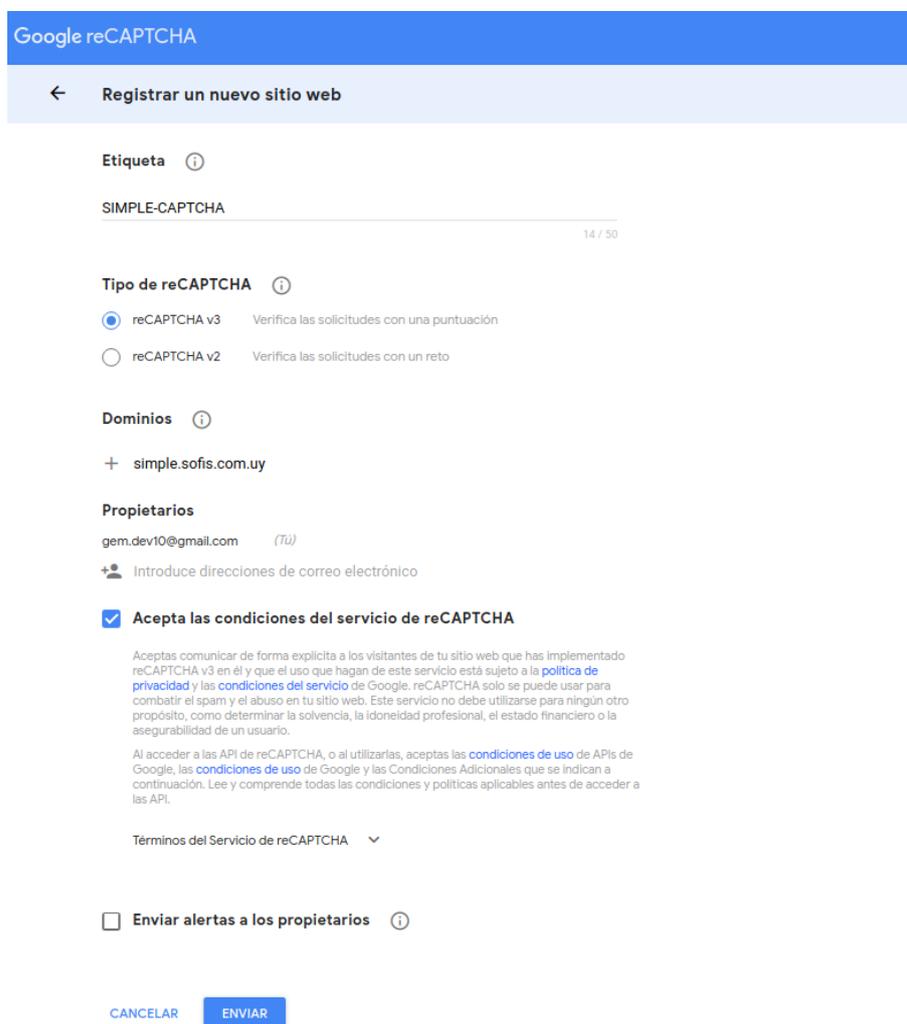
13.1 Crear una cuenta en google recaptcha.

Para obtener los datos Captcha key y Captcha secret, hay que registrar una cuenta en Google recaptcha, los pasos son los siguientes:

- Ir al sitio <https://www.google.com/recaptcha/admin/>

** Se debe iniciar sesión con una cuenta de google

- Aparecerá el siguiente formulario para configurar el google recaptcha.



The image shows the Google reCAPTCHA registration interface. At the top, there is a blue header with the text "Google reCAPTCHA". Below this is a light blue navigation bar with a back arrow and the text "Registrar un nuevo sitio web". The main form area contains several sections: "Etiqueta" with a text input field containing "SIMPLE-CAPTCHA" and a character count "14 / 50"; "Tipo de reCAPTCHA" with two radio button options: "reCAPTCHA v3" (selected) and "reCAPTCHA v2"; "Dominios" with a plus sign and the domain "simple.sofis.com.uy"; "Propietarios" with an email address "gem.dev10@gmail.com" and a plus sign to add more; a checked checkbox for "Acepta las condiciones del servicio de reCAPTCHA" with a detailed text block below it; and a checkbox for "Enviar alertas a los propietarios". At the bottom, there are two buttons: "CANCELAR" and "ENVIAR".

Imagen 16:

Formulario de registro de cuenta captcha

- Etiqueta: Un nombre para identificar el sitio (ya que puede crear N cuentas de recaptcha)

- Tipo de reCAPTCHA: Seleccionar la versión 3, la versión 2 puede utilizarse pero es menos precisa y eventualmente lanza verificación por imágenes, lo cual puede ser molesto para el usuario.
- Dominios: Agregar el dominio del sitio web en el que se va a utilizar, por ejemplo si la url es <https://simple.sofis.com.uy/simple> únicamente ingresar el nombre del dominio, es decir, simple.sofis.com.uy
- Aceptar términos y condiciones, luego clic en Guardar.
- Obtener los datos google captcha key y google captcha secret, copiando los valores de las siguientes casillas.

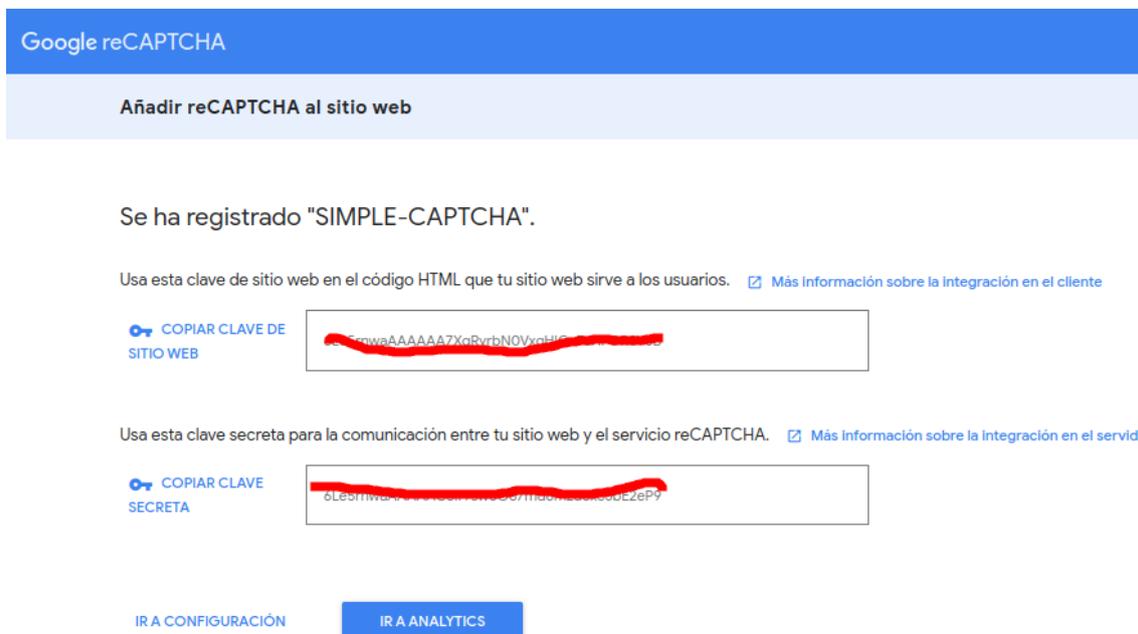


Imagen 17: Datos a utilizar para la cuenta en simple

13.2 Editar datos de la cuenta en el manager y en el backend.

Luego de registrar la cuenta en Google recaptcha y obtener la clave de sitio web y la clave secreta, estas deben registrarse en la cuenta que desea usar el componente de Google captcha.

La configuración puede hacerse editando la cuenta, tanto en la gestión en el manager como en el backend, a continuación un ejemplo de como hacerlo en el backend (los pasos son exactamente iguales en el manager).

- Ir al menú Configuración → Cuenta
- En los datos de la cuenta se muestra un nuevo fieldset con la leyenda “Google recaptcha invisible”, contiene los siguientes campos (opcionales):
 - Captcha Key: Contiene el valor CLAVE DE SITIO WEB obtenido en el punto 13.1
 - Captcha Secret: Contiene el valor de CLAVE SECRETA obtenida en el punto 13.1
 - Hay un botón **Verificar recaptcha**, sirve para validar la comunicación con el API de google y verificar que la clave secreta esté correctamente configurada. Muestra una leyenda en con fondo verde para indicar que está correcto, sí el fondo es rojo, indica mediante un texto que hay error.



Imagen 18: Ejemplo de registro de captcha en una cuenta en SIMPLE

13.3 Utilizar el componente en el modelado.

En el modelado de procesos de SIMPLE, se agregó un nuevo campo para los formularios, llamado Google recaptcha (invisible), la configuración es básica, consiste únicamente en asignar una etiqueta y un nombre al campo.

Versión del proceso: 1



Captcha 



Imagen 19: Componente google recaptcha (invisible)

13.4 Ejemplo en un formulario.

Para los formularios que tienen el componente, en la parte inferior derecha de la pantalla, aparecerá la insignia de que cuenta con verificación de Google recaptcha invisible.



Imagen 20: Ejemplo de ubicación de la insignia

14 Historial de cambios

Versión	Fecha	Autor	Detalle
1.0.0	18/06/19	Danis Chaviano	Primera versión del documento
1.1.0	05/08/2020	Danis Chaviano	Actualización de la versión 3.0-r1 de la herramienta SIMPLE
1.2.0	04/03/2021	Gilberto Mendoza	Se agrega componente de preguntas captcha, disponible en la versión 3.0-r7.
1.3.0	12/03/2021	Gilberto Mendoza	Se agrega componente de Google recaptcha invisible, disponible desde la versión 3.0-r8.