

Configuración de Declaración Jurada para Tributos Autoliquidables

Si se está configurando un nuevo tributo (tasa/derecho u contribución) de tipo autoliquidable, o sea que requiere de una presentación de declaración jurada por parte del contribuyente para determinar la obligación tributaria, se deberá configurar y preparar el módulo que permita registrar dicha declaración.

Esta preparación requiere de varias actividades:

1. Configuración de características principales del Recurso
2. Configuración de características para Declaración Jurada del Recurso (Conceptos a Declarar, Alícuotas por actividad, Mínimos generales y especiales, Alicuotas adicionales, y otros parámetros)
3. Programa de cálculo y determinación de importe para la declaración jurada

A continuación se presenta una guía/explicación general con algunos ejemplos para ayudar a preparar esta configuración. Se tomará para esta sección como ejemplo la configuración para una Tasa/Derecho/Contribución sobre comercio, pero servirá de todas formas para otros tributos. En algunos casos se comentarán ejemplos adecuados para otros posibles tributos.

Configuración de características principales del Recurso

Desde la opción de configuración de recursos:

Administración del Sistema -> Recursos -> Mantenedor de Recursos

En la configuración general del recurso, al dar de alta el mismo en las características principales se configuran los primeros parámetros para habilitar el módulo

Características Principales		
Categoría:	Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad	
Código:	TISHPYS	
Descripción:	Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad	
Es Autoliquidable:	Si	Indica si el recurso es Autoliquidable o no
Fecha Desde:	01/01/1980	Indica la fecha en que se adoptó el último estado para la bandera esAutoliquidable
Permite Declaración Jurada:	Si	
Permite Declarar Periodos en Convenio:	Si	Indica si permite realizar declaración jurada sobre periodos en convenio
Es Fiscalizable:	Si	Indica si el recurso es Fiscalizable o no
Fecha Desde:	01/01/1980	Indica la fecha en que se adoptó el último estado para la bandera esFiscalizable
Fecha de Comienzo de cobro con Interés Diario:		
Fecha Alta:	01/01/1980	
Permite Modificación Fecha Vencimiento Deuda:	Si	Indica si el recurso permite la modificación de la Fecha de Vencimiento de la Deuda

Se debe indicar el recurso como autoliquidable:

Es Autoliquidable: Si

Y marcar que permite declaraciones juarads

Permite Declaración Jurada: Si

También se debe indicar en este punto si se permitirá realizar declaraciones rectificativas sobre periodos tomados por un convenio de pago vigente

Permite Declarar Periodos en Convenio: No

En caso que se permita, se considerará al registro de deuda en convenio como un pago del mismo periodo para que el efecto de procesamiento de la declaración genere deuda por la diferencia en caso de corresponder sin modificar el registro incluído en el convenio de pago.

Configuración de características para Declaración Jurada del Recurso

Para el caso de recursos marcados como autoliquidables se habilita una segunda opción de edición con las parametros y datos que se utilizarán en la Declaración Jurada

Búsqueda de Recurso

Permite buscar y administrar Recurso

Volver

Parámetros de la Búsqueda

Categoría: Todos

Código:

Descripción:

Limpiar

Imprimir

Buscar

Resultado de la Búsqueda

					Código	Descripción	Tipo de Categoría	Categoría	Fecha Alta	Fecha Baja	Estado
					MPROV	Multa de Proveedores	Tributario	Multa de Proveedores	01/01/2015		Activo
					TGI	Tasa General Inmobiliaria	Tributario	Tasas de padrón	01/01/1980		Activo
					TISHPYS	Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad	Tributario	Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad	01/01/1980		Activo
					TSS	Tasa por Servicios Sanitarios	Tributario	Tasas de padrón	01/01/1980		Activo

<< < 2 / 2 > >>

Volver

Agregar

Accediendo a esta opción debemos comenzar a configurar varias características y datos. Entre ellos se configurarán los conceptos a declarar con sus parámetros, mínimos generales y especiales, etc.

Lo primero que se deberá configurar es el “Tipo de Conceptos a Declarar”

Tipo de Conceptos a Declarar			
			Descripción
			Actividad
			Agregar

En el caso de los tributos principales sobre comercio como por ejemplo:

- Derecho de Registro e Inspección
- Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad
- Contribución sobre la Actividad Comercial

El tipo de concepto por el cual se declara suele ser por “Rubro Comercial” o “Actividad Comercial”

En otros tributos podría corresponder simplemente a uno genérico “Concepto”, u otros como “Carteles”, etc.

Luego se deberá configurar las “Características para Declaración

Jurada”

Características para Declaración Jurada	
Cálculo para Declaración Jurada: <code>calc_ddjj_tishpys.js</code>	Nombre del archivo javascript para cálculo para declaración jurada
Cód. Atributo Conceptos: <code>Rubro</code>	Indica el atributo que determina los conceptos a declarar en declaración jurada
Tipo de Concepto a Declarar: <code>Actividad</code>	
Clasificación de Deuda Rectificativa: <code>DDJJ rectificativa</code>	
Permite Declarar Conceptos No Habilitados: <code>Si</code>	
Permite Uso de Anticipo: <code>No</code>	Indica si permite el uso de anticipos en declaración jurada
Permite Uso de Saldo a Favor: <code>No</code>	Indica si permite el uso de saldos a favor en declaración jurada
Permite Imputación de Retenciones: <code>Si</code>	Indica si permite imputar retenciones en declaración jurada
Imputar Retenciones Registradas: <code>Si</code>	Indica si sólo permite imputar retenciones registradas en Siat
Permite Imputar Deducciones Adicionales: <code>No</code>	Indica si permite imputar deducciones adicionales por importe en declaración jurada
Permite Declarar Adicionales: <code>Si</code>	Indica si permite declarar adicionales en declaración jurada
Aplica Multa en Declaración Jurada: <code>Si</code>	
Tipo de Multa: <code>Multa por infracción a los deberes formales - Persona Física (DDJJ)</code>	
Visualizar Determinación de Deuda: <code>Si</code>	Indica si se visualiza el paso de determinación/simulación de deuda en declaración jurada
Permite Declaración Jurada de Oficio: <code>Si</code>	
Clasificación de Deuda de Oficio: <code>Ajuste Verificado</code>	
Tipo de Multa para Declaración de Oficio: <code>Multa por infracción a los deberes formales - Persona Física (DDJJ)</code>	

En esta sección se deberá indicar el nombre del archivo con el programa “javascript” para el cálculo de la ddjj.

Cálculo para Declaración Jurada: `calc_ddjj.js`

El archivo debe encontrarse en la siguiente ubicación del fileshared de SIAT:
“fs/privado/grs/siat”

El próximo parámetro indica el código de algún atributo de cuenta u objeto imponible del cual se pueda obtener el o los códigos de los conceptos por los cuales se declarará:

Cód. Atributo Conceptos: `Rubro`

En el caso de Comercio generalmente se configura un atributo de código “Rubro” o “Actividad” asociado al Objeto Imponible “Comercio” u “Empresa” con y configurado para permitir multiples valores.

Luego cada Cuenta contributiva conocerá los “Rubros” o “Actividades” sobre la cual está habilitada y esta información la utilizará el módulo de DDJJ para presentar los conceptos a declarar.

En el tipo de concepto a Declarar seleccionamos el registro creado en el paso anterior.

En el parámetro “Clasificación de Deuda Rectificativa” se debe seleccionar una de las clasificaciones de deuda configuradas en el recurso y que no estén marcadas como original. Esta clasificación es la que se utilizará para la deuda a emitir por diferencias en caso de declaraciones juradas rectificativas. Algunos ejemplos utilizados son: “DDJJ rectificativa”, “Diferencia”, “Adicional”.

Si no se cuenta con información lo suficientemente confiable para restringir la DDJJ a la declaración de conceptos previamente habilitados en la cuenta o el objeto imponible (ej: comercio), se puede habilitar para que el contribuyente pueda agregar sobre que conceptos va a realizar la declaración. Para esto se deberá setear el parámetro “Permite Declarar Conceptos No Habilitados” en Sí.

En algunas implementaciones se permite a los contribuyentes realizar anticipos por periodos futuros. Estos anticipos se realizan a través de otra opción del sistema todavía no incorporada. De todas formas si existen las estructuras de datos y consultas de posibles anticipos. Es por esto que se prevee en el módulo la posibilidad de su utilización en las DDJJ. Al activar el parámetro “Permite Uso de Anticipo”, se habilita una nueva sección en la DDJJ que busca y permite imputar como deducciones anticipos registrados.

En SIAT, desde diferentes módulos, se generan o pueden generar saldos a favor del contribuyente. Si se marca el tipo de saldo (según su origen) para permitir su uso en DDJJ se puede habilitar su utilización para deducir durante la presentación seteando en ‘Sí’ la opción “Permite Uso de Saldo a Favor”.

Para habilitar la declaración de retenciones para su deducción en la DDJJ se debe habilitar la opción “Permite Imputación de Retenciones”. Luego si se habilita la opción siguiente “Imputar Retenciones Registradas”, se restringe posibilidad de indicar libremente el monto y certificado de retención y sólo se podrán seleccionar retenciones previamente registradas en el sistema.

En caso de necesitar contar con un campo de libre carga para que el contribuyente indique un monto a deducir se puede habilitar la opción “Permite Imputar Deducciones Adicionales”.

En algunos casos luego de aplicar el cálculo básico de determinación por bases, o unidades se aplican alícuotas adicionales por ejemplo para:

- Por Sillas y Mesas en espacio público
- Por Publicidad
- Fondo de Promoción ...

- etc

En caso de ser necesario algún adicional se deberá calcular y considerar en el programa de cálculo de la DDJJ. De todas formas se deberá habilitar la sección de la DDJJ seteando en 'Sí' la opción "Permite Declarar Adicionales"

De forma similar en caso de aplicar alguna multa en forma automática, siempre desde el programa de cálculo, se deberá habilitar la opción "Aplica Multa en Declaración Jurada" para que se visualice la sección correspondiente en la DDJJ. En el campo "Tipo de Multa" se puede seleccionar un tipo de multa previamente registrado para contar de forma más directa con el mismo en el programa de cálculo. Este punto es opcional ya que podría registrarse una multa sin tener el tipo configurado en el recurso, esto dependerá de como se desarrolle el programa de cálculo.

La presentación de la DDJJ es una opción que lleva varios pasos. El primer paso para declarar bases, unidades y/o valores de atributos. Un segundo paso donde con los importe determinados se pueden declarar distintas deducciones (retenciones, pagos del mismo periodo, etc). Existe un tercer paso opcional donde se visualiza el efecto que provocará el procesamiento de la DDJJ sobre la deuda. Esto es si se modifica el registro de deuda de clasificación original o se emite nuevos registro por valores de diferencia. Aunque este último paso en algunos casos puede ser de interés, desde el punto de vista del contribuyente puede generar cierta confusión o no ser relevante. Es por esto que se permite ocultar el mismo desde la configuración del recurso a través del parámetro "Visualizar Determinación de Deuda".

Finalmente se puede configurar si para el recurso se habilitará la modalidad/tipo de declaración para "Determinación de Oficio" por parte de usuarios del municipio. Con dicha opción y los permisos correspondientes del módulo de seguridad se podrá acceder a través de un nuevo botón a dicho tipo de declaración. Se deberá indicar la clasificación de deuda a utilizar, ya que en dicho modo siempre se emite deuda sin modificar la original o por diferencia emitida por declaración del contribuyente. Por ejemplo: "Ajuste Fiscal", "Ajuste Verificado", "Determinación Oficio", etc.

Adicionalmente se podrá seleccionar un tipo de multa que podrá accederse desde el programa de cálculo para aplicar en caso que corresponda.































Configuración de Conceptos a Declarar y sus parámetros

En la declaración jurada el contribuyente declara bases, montos o unidades sobre conceptos en función del tributo.

En el caso de Comercio normalmente estos conceptos corresponden a

“Rubros o Actividades Comerciales”.

Debemos entonces a continuación cargar y configurar el nomenclador de conceptos por los cuales se declara con sus parámetros.

Conceptos a Declarar En DDJJ						
			Fecha Desde	Fecha Hasta	Código Concepto	Descripción
			01/01/1990		523590	Venta al por menor de artículos para el hogar n.c.p.
			01/01/1990		602230	Servicio de transporte escolar
			01/01/1990		662000	Administración de fondos de jubilaciones y pensiones (A.F.J.P.)
			01/02/1990		602220	Servicios de transporte automotor de pasajeros mediante taxis y remises; alquiler de autos con chofer
			01/02/1990	31/05/2012	921910	Servicios de salones de baile, discotecas y similares
			01/03/1990		921120	Distribución de filmes y videocintas
			01/01/1991		924910	Servicios de esparcimiento relacionados con juegos de azar y apuestas
			01/01/1991		999045	Tombola, prode,lotería quini 6 hipodromos
			01/03/1991		522120	Venta al por menor de productos de almacén y dietética
			01/01/1992		702000	Servicios inmobiliarios realizados a cambio de una retribución o por contrata

Agregar

El código es importante considerando que el atributo configurado arriba deberá contener valores registrados en este nomenclador si se desea que se comience la presentación con los conceptos precargados.

Administración de Recurso

Volver

Datos del Recurso

Tipo de Categoría: Tributario

Código: TISHPYS

Tipo Objeto Imponible: Comercio/Industria

Categoría: Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Descripción: Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Conceptos del Recurso a Incluir en DDJJ


Tipo de Concepto:


Seleccionar...

Código:

Descripción:

Observación:

Fecha Desde: 

Fecha Hasta: 

Volver

Agregar

Para cada concepto se deben completar los parámetros vigentes:

Conceptos del Recurso a Incluir en DDJJ

Tipo de Concepto: Actividad

Código: 522120

Descripción: Venta al por menor de productos de almacén y dietética

Observación:

Fecha Desde: 01/03/1991 Fecha Hasta:

Modificar

Parámetros del Concepto

			Fecha Desde	Fecha Hasta	Alícuota	Mínimo	Monto Fijo	Unidad de Medida
			01/03/1991		0.009	0	0	

Agregar

Mínimos por Rango

No se encontraron registros para la Búsqueda

Agregar

En estos puntos muchos de los campos no son requeridos y dependerán de la programación del cálculo de la DDJJ. Estos campos permiten tener los parámetros registrados con vigencias y luego acceder a ellos en el programa del cálculo.

Conceptos del Recurso a Incluir en DDJJ

Código: 522120

Descripción: Venta al por menor de productos de almacén y dietética

Observación:

Fecha Desde: 01/03/1991 Fecha Hasta:

Parámetros de Concepto a Declarar en DDJJ

(*) Fecha Desde:

Fecha Hasta:

Alícuota:

Mínimo:

Monto Fijo:

(*) Declara Base Imponible:

(*) Declara Cantidad de Unidades:

Unidad de Medida:

Mínimo Por Unidad:

Monto Fijo Por Unidad:

Mínimo por Rango de Unidad:

Código de Atributo para Rango:

Mínimo para Cantidad de Unidades:

Agregar

Se pueden cargar valores para alícuotas, mínimos especiales o monto fijo según corresponda.

Es requerido informar si para el concepto se debe solicitar la declaración de

Base Imponible y/o Cantidad de Unidades. Normalmente la mayoría de los casos declaran base imponible y no declaran unidades.

Las unidades pueden utilizarse en casos como por ejemplo para indicar cantidad de plazas de estacionamiento, habitaciones en hoteles, maquinas, kw, etc; en conceptos donde el cálculo requiera esta información tanto para la determinación o del monto a pagar o de algún mínimo especial a determinar.

En el caso de declarar unidades se debe seleccionar un tipo de unidad de la lista (previamente registrada) desde el detalle correspondiente que se presentará a continuación.

Se brinda la posibilidad de llevar en la configuración información adicional como ser un: si debe calcular un “Mínimo Por Unidad”, un valor mínimo requerido para la cantidad de unidades, si debe calcular “Monto Fijo Por Unidad”, e incluso la posibilidad de indicar si el mínimo especial se debe obtener de una tabla de rangos en función del valor de unidad o por valor de un atributo.

Para registrar y administrar mínimos por rango para un concepto

Mínimos por Rango para Concepto a Declarar

Volver

Conceptos del Recurso a Incluir en DDJJ

Código: 522120

Descripción: Venta al por menor de productos de almacén y dietética

Observación:

Fecha Desde: 01/03/1991

Fecha Hasta:

Mínimos por Rango para Concepto a Declarar

Valor de Referencia Desde:

Valor de Referencia Hasta :

Valor:

Fecha Desde:

Fecha Hasta:

Volver

Agregar

Como se mencionó más arriba estos valores no tiene efecto inmediato a excepción de los requeridos. El resto dependerá de su utilización en el programa de cálculo y se brindan como variables públicas adicionales que en caso de ser utilizadas luego podrán gestionarse sus valores con vigentes desde esta configuración y sin volver a modificar el programa.

Configuración de Tipo de Unidades

En caso de necesitar permitir declarar cantidad de unidades para algunos de los conceptos, antes de completar la configuración del mismo se deberán registrars los tipos de unidades disponibles

Unidad de Medida				
			Nomenclatura	Descripción
			canchas	cantidad de canchas
			habitaciones	cantidad de habitaciones
			sillas	cantidad de sillas
			puestos	cantidad de puestos
			vehiculos	cantidad de vehículos
			dias	dias
			M3	M3 comercializado
			aparatos	cantidad de aparatos
			maquinas	cantidad de maquinas
			valor	valor por dia
				Agregar

Estas unidades son las que se disponibilizan en la configuración de parámetros comentada anteriormente. El valor “Nomenclatura” es el que se observará al declarar las cantidades.

Administración de Recurso

[Volver](#)

Datos del Recurso

Tipo de Categoría: Tributario

Categoría: Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Código: TISHPYS

Descripción: Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Unidad de Medida

(*) Nomenclatura:

(*) Descripción:

Código Afip:


[Volver](#)

[Agregar](#)

El parámetro “Código Afip” sólo es necesario y se utiliza en caso de sincronismo de DDJJ con AFIP. Para la guía que estamos siguiendo este valor puede dejarse vacío.

Configuración de Tipo de Alícuota y Alícuotas Adicionales

Si en la Declaración Jurada se aplicarán alícuotas adicionales una vez determinado el importe, se deberá completar la configuración que se presenta a continuación.

Tipo de Alícuota				
			Código	Descripción
			FPCT	Fondo Prom. Com. y Tur.
				Agregar

Debe registrarse el tipo de alícuota adicional. Esta información se utilizará para visualizar su descripción en caso de aplicarse en el programa de cálculo.

Administración de Recurso

[Volver](#)

Datos del Recurso

Tipo de Categoría: Tributario

Categoría: Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Código: TISHPYS

Descripción: Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Tipo de Alícuota

(*) Código:

(*) Descripción:

Concepto del Recurso: <Ninguno>




[Volver](#)

[Agregar](#)

En algunos casos se requiere que el importe determinado por las alícuotas adicionales se guarde como un concepto particular de la deuda a determinar (nos referimos a los conceptos de emisión, no a los conceptos a declarar que configuramos unos pasos antes).

Nota: Por ejemplo: Un adicional por un “Fondo de Promoción” y cuya configuración del recurso consta de dos conceptos de emisión: Tasa, Fondo. En este podría indicarse que el tipo de alícuota configurado para la determinación del fondo se debe impactar en el concepto de emisión “Fondo”.

Luego podrá registrarse el o los valores de alícuota con su vigencia.

Alícuotas Adicionales						
			Alícuota	Tipo de Alícuota	Fecha Desde	Fecha Hasta
			20.0%	Fondo Prom. Com. y Tur.	01/01/2013	31/12/2030
						<div>Agregar</div>

Estos valores estarán disponibles para ser utilizados en el programa de cálculo.

Volver

Datos del Recurso

Tipo de Categoría: Tributario

Categoría: Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Código: TISHPYS

Descripción: Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Alícuota Adicional

(*)Tipo de Alícuota: Seleccionar...

(*) Alícuota:

(*) Fecha Desde:

(*)Fecha Hasta:

Observación:

Volver

Agregar

Configuración de Mínimos Generales del Recurso

Para guardar y mantener datos de mínimos generales (que no dependan de los conceptos a declarar) se puede utilizar la siguiente tabla

Mínimos del Recurso							
			Fecha Desde	Fecha Hasta	Valor Desde	Valor Hasta	Mínimo
			01/01/2013	31/12/2015	0	0	150
			01/01/2016	31/12/3000	0	0	180
							Agregar

En estos casos se puede guardar valores de mínimo con su vigencia que podrán obtenerse desde el programa de cálculo de la DDJJ.

Volver

Datos del Recurso

Tipo de Categoría: Tributario

Categoría: Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Código: TISHPYS

Descripción: Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Datos de los mínimos a declarar

(*) Mínimo:

(*) Valor Desde:

(*) Fecha Desde:

(*) Valor Hasta:

(*)Fecha Hasta:

Volver

Agregar

Los campos “Valor Desde” y “Valor Hasta”, permiten guardar valores de referencia para consultar a partir de otro dato.

Por ejemplo podría consultarse la cantidad de personal registrada en algún atributo de la cuenta o el objeto imponible, o solicitarse su declaración en la misma presentación de la DDJJ. Luego en el programa de cálculo utilizar dicho valor para obtener el mínimo que corresponde según su rango.

En otros casos podría determinarse en función de los ingresos anuales, de ciertos topes respecto a las bases que declara u otras situaciones que se consideren en el programa de cálculo.

Un ejemplo de carga con valores de referencia:

Mínimos del Recurso							
			Fecha Desde	Fecha Hasta	Valor Desde	Valor Hasta	Mínimo
			01/01/2001	31/12/2005	1501	999999999	36
			01/01/2001	31/12/2005	0	1500	18
			01/01/2006	29/02/2008	0	0	30
			01/01/2006	29/02/2008	1	1	45
			01/01/2006	29/02/2008	2	2	70
			01/01/2006	29/02/2008	3	999999999	90
			01/03/2008	31/08/2010	0	0	50
			01/03/2008	31/08/2010	1	1	90
			01/03/2008	31/08/2010	2	2	140
			01/03/2008	31/08/2010	3	999999999	180
			01/09/2010	31/05/2012	0	0	70
			01/09/2010	31/05/2012	1	1	125
			01/09/2010	31/05/2012	2	2	195

Configuración de Atributos a utilizar en Declaración Jurada

En el programa de cálculo a veces es necesario consultar datos de información del objeto imponible o cuenta. Por ejemplo para Comercios: la categoría de un hotel, o la cantidad de personal, la zona de ubicación, etc.

Otras veces se necesita solicitar al contribuyente que durante la presentación complete otros datos informativos.

Para estos casos se extiende el módulo a través de los atributos de SIAT. En esta sección se pueden indicar atributos, previamente configurados, para que su valor pueda ser accedido desde el programa de cálculo y/o solicitado en la presentación.

Atributo a utilizar en Declaración Jurada						
			Atributo	A declarar	Fecha Desde	Fecha Hasta
			Cantidad de Personal	Si	01/01/1990	
			Notificada	Si	01/01/1990	
			Calcula multa por Omisión automáticamente	Si	01/01/1990	
			Multa Omisión deber formal (%)	Si	01/01/1990	
			Multa Defraudación fiscal (%)	Si	01/01/1990	
			Tipo de Presentación	Si	01/01/1990	
			Tipo de Monto	Si	01/01/1990	
			Ingresos devengados año calendario anterior	Si	01/09/2008	30/06/2013
			Derecho de Ocupación de la vía pública: Ocupación de la vía pública, subsuelo y espacio aéreo	Si	01/09/2014	
			Derecho de Publicidad: Publicidad o propaganda efectuada en la vía pública (carteles, afiches, altavoces, etc)	Si	01/09/2014	
						<button>Agregar</button>

Indicando la opción “A declarar” en Sí, el atributo se agregará a la DDJJ para su carga por parte del contribuyente. Mediante el parámetro “Sólo en Determinación de Oficio”, se indica que el atributo no deberá solicitarse en la DDJJ normal sino únicamente en la modalidad de Determinación de Oficio (de uso interno y para fiscalización).

Luego se puede indicar si el dato es requerido o se permite dejar sin valor.

Administración de Recurso

Volver

Datos del Recurso

Tipo de Categoría: Tributario

Categoría: Tasa Por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Código: TISHPYS

Descripción: Tasa por Inspección Sanitaria, Higiene, Profilaxis y Seguridad

Atributo de DDJJ

(*) Atributo: Buscar Atributo

(*) A declarar:

(*) Sólo en Determinación de Oficio:

(*) Es requerido:

(*) Fecha Desde:

Fecha Hasta:

Volver

Agregar

En caso de necesitar registrar nuevos atributos ver la configuración correspondiente en [Documentación Módulo Administración](#)

Programa de cálculo y determinación de importe para la declaración jurada

Una vez definidos los parámetros y configuración necesitamos desarrollar el programa de cálculo que utilizará la Declaración Jurada para determinar la obligación tributaria.

El programa debe ubicarse en la carpeta del fileshared de SIAT:

“fs/privado/grs/siat”

El archivo debe tener el mismo nombre que el configurado en el recurso en el parámetro “Cálculo para Declaración Jurada”.

Imágen Ejemplo de previsualización de editor de texto con un programa de cálculo de DDJJ



```
/**
 * Cálculo de Declaración Jurada
 *
 *
 * Parametros de entrada:
 * - fechaVencimiento: Fecha Vencimiento de la original
 * - decJur: estructura de declaracion jurada inicializada y con parametros de base imponible y unidades cargados
 *
 * Parametro de salida:
 * - decJur: estructura de declaracion jurada con valores calculados
 */
Grs.load("<siat/siat.js>")

main = function(decJurAPI){
    var decJur = decJurAPI.decJur;
    var listCueExe = decJurAPI.listCueExe;

    //obtenemos el valor de utrim vigente correspondiente al periodo
    var utrim = decJurAPI.getValorUtrim();

    // Se inicializan las observaciones de la DecJur
    decJur.setObservaciones("");

    // Determinamos el concepto principal como el de mayor base imponible
    var mayorBase = 0;
    var decJurDetPrincipal;
    for (var i = 0; i < decJur.getListDecJurDet().size(); i++) {
        var decJurDet = decJur.getListDecJurDet().get(i);
        if(decJurDet.getBase() > mayorBase){
            mayorBase = decJurDet.getBase();
            decJurDetPrincipal = decJurDet;
        }
    }

    // Marcamos el concepto principal
    if(mayorBase > 0){
        decJurDetPrincipal.marcarComoPrincipal();
    }

    var subTotal = 0;
    var minimoGeneral = 0;
    var cantidadPersonal = decJurAPI.mapAtributos.CantPer

    // Obtener minimo general a partir de cantidad de personal declarado
    var fechaVencimiento = new Date(decJurAPI.fechaVencimiento.getTime());
    var fechaLimite = new Date(2006,02,01)
    var recMinDec;
    if(fechaVencimiento >= fechaLimite){
        // Si el periodo declarado es posterior al 2006 se toma minimo general por cantidad de personal
        recMinDec = decJurAPI.getRecMinDecVigente(cantidadPersonal);
    }else{
        // Determina mayor base imponible
    }
}
```

El lenguaje en el cual se debe programar es Javascript y la estructura básica del código debe ser como se presenta a continuación:

```
/**
 * Cálculo de Declaración Jurada
 *
 *
 * Parametros de entrada:
 * - fechaVencimiento: Fecha Vencimiento de la original
 * - decJur: estructura de declaracion jurada inicializada y con paramet
 *
 * Parametro de salida:
 * - decJur: estructura de declaracion jurada con valores calculados
 */

Grs.load("<siat/siat.js>")
```

```

main = function(decJurAPI){

    var decJur = decJurAPI.decJur;
    var listCueExe = decJurAPI.listCueExe;

    // ..... lógica de cálculo de ddjj ....

    return decJurAPI;
}

```

La función a implementar recibe como parámetro un objeto que funciona como una API o facilitador.

De este parámetro se obtiene la estructura precargada de la Declaración Jurada (value objects). Además se brindan algunos métodos útiles para buscar otros datos de configuración o formatear tipo de datos.

A continuación se presenta la estructura del objeto que llega como parámetro y los métodos de los que dispone

```

public class DecJurJSAPI {

    public Date fechaVencimiento;
    public DecJurVO decJur;
    public Map<String, String> mapAtributos = new HashMap<String,String>();
    public List<CueExe> listCueExe = new ArrayList<CueExe>();

    public DecJurJSAPI();

    /**
     * Devuelve el RecMinDec vigente para el valor y fecha pasados como parametros
     *
     * @param valor
     * @param fecha
     */
    public RecMinDec getRecMinDecVigente(String valorStr);

    /**
     * Compara las dos fechas sin tener en cuenta la hora.
     * Retorna -1 si dateToValidate es Menor que dateToCompare
     * Retorna 0 si dateToValidate es Igual que dateToCompare
     * Retorna 1 si dateToValidate es Mayor que dateToCompare
     */
    public int dateCompare(Date dateToValidate, Date dateToCompare);

    /**
     * Valida si la fecha de presentacion es posterior a la fecha de vencimiento
     *
     * Devuelve 'true' en caso de estar fuera de termino.
     *
     * @return
     */
    public boolean presentaDecJurFueraDeTermino();

    /**
     * Obtiene la fecha de inicio del periodo tomando la información de periodo
     *
     * @return
     */
    public Date getFechaInicioPeriodo();
}

```



```

/**
 * Devuelve la fecha correspondiente al anio, mes y dia
 * En caso de error devuelve null
 * Enero es el 0
 * @param anio
 * @param mes
 * @param dia
 * @return Date
 */
public Date getDate(int anio, int mes, int dia);

/**
 * Convierte el String a Double. Si no puede devuelve null.
 *
 * @param strValor
 * @return
 */
public static Double getDouble(String strValor);

/**
 * Imprime el texto en el log del servidor
 *
 * @param strJavaScript
 */
public void logDebug(String strJavaScript);

/**
 * Función que devuelve el valor de UTRIM del período
 * Por defecto retorna el valor 1.
 * @return Double
 */
public Double getValorUtrim();
}

```

En la variable pública “fechaVencimiento” se cuenta con la fecha de vencimiento del periodo a declarar.

En la variable pública “mapAtributos” se cuenta con un mapa de string en donde vendran precargados los valores de atributos configurados en el recurso para usar en la declaración jurada. La clave para buscar en el mapa es el código del atributo sobre el cual se quiere consultar su valor.

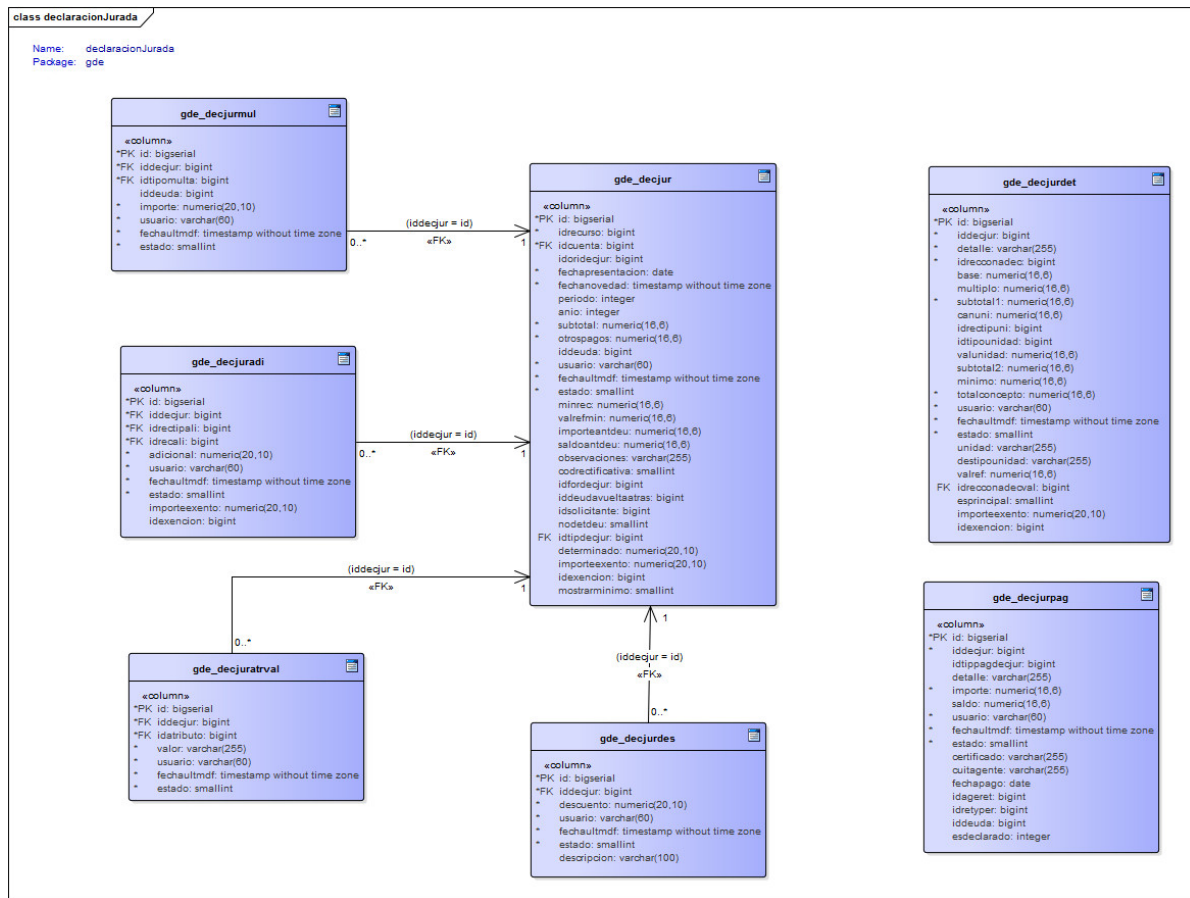
Ejemplo:

Donde “CantPer” es el código del atributo definido para “Cantidad de Personal” (sólo es un ejemplo ya que depende de la configuración de los atributos en el módulo de administración del sistema)

La variable pública “decJur” es una instancia del value objet con la estructura precargada de la Declaración Jurada. En estos casos para acceder a sus datos se realiza a través de métodos “get” y “set”.

Ver también: Para mayor información consulta estructura en Javadoc
ar.gov.rosario.siat.gde.iface.model.DecJurVO

Se agrega como información de referencia el modelo de datos que está representado en las estructura VO disponibles



Un ejemplo del uso de la estructura contenida en el DecJurVO para recorrer los conceptos a declarar podría ser:

```

var decJur = decJurAPI.decJur;

for (var i = 0; i < decJur.getListDecJurDet().size(); i++) {
    var decJurDet = decJur.getListDecJurDet().get(i);
    decJurDet.setTotalConcepto(decJurDet.getBase() * decJurDet.getRecConADecVal(
        subTotal += decJurDet.getTotalConcepto();

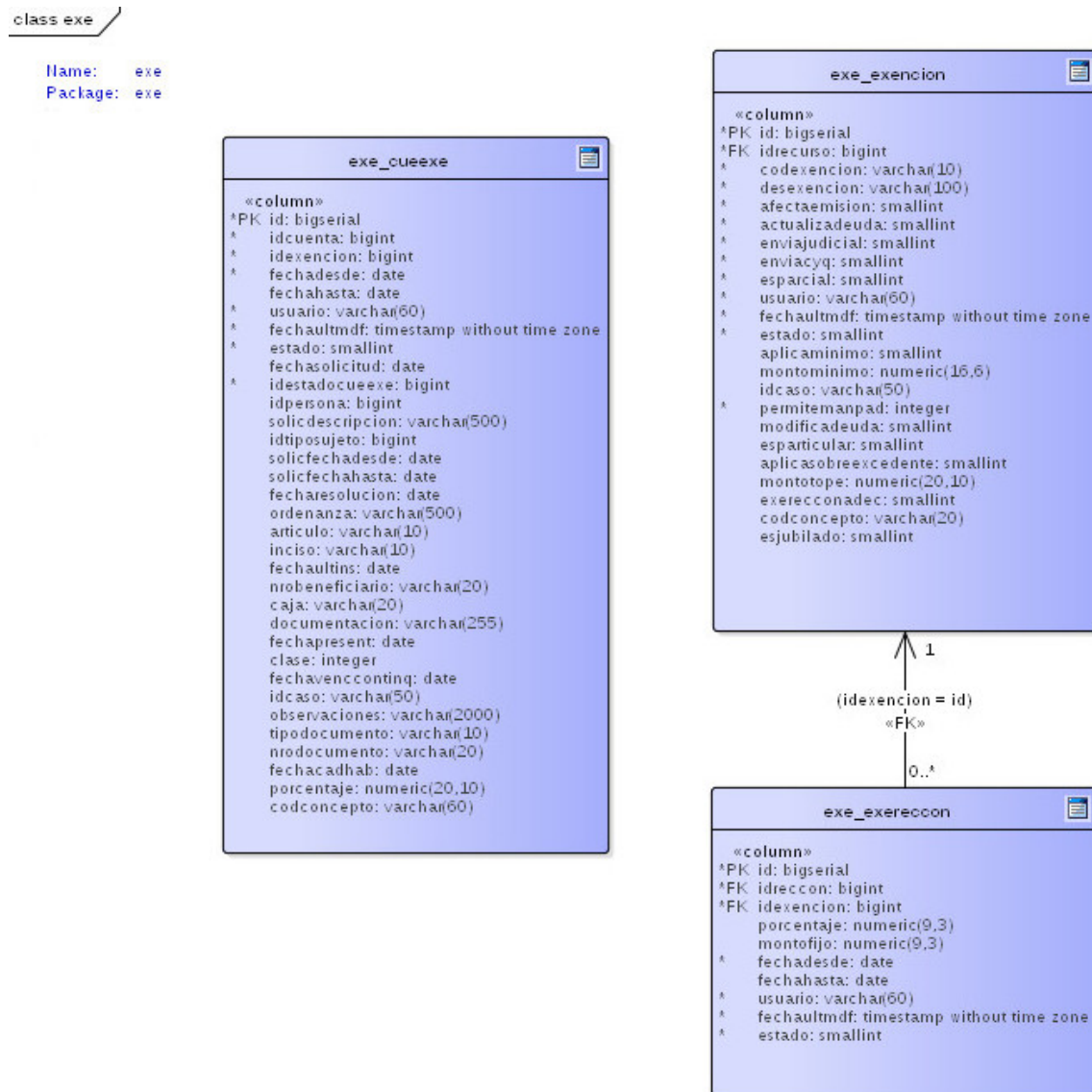
    var minimo = decJurDet.getRecConADecVal().getMinimo();
    if (decJurDet.getRecConADecVal().getMulMinPorUni().getEsSI()) {
        minimo = decJurDet.getRecConADecVal().getMinimo() * decJurDet.getCanl
    }

    if (minimoGeneral < minimo) {
        minimoGeneral = minimo;
    }
}
  
```

La variable pública “listCueExe” contiene una lista de las Exenciones vigentes para la cuenta. Corresponde a una lista de objetos de tipo “value objet” CueExeVO que se describe a continuación:

Ver también: Para mayor información consulta estructura en Javadoc
ar.gov.rosario.siat.exe.iface.model.CueExeVO

Se agrega como información de referencia el modelo de datos que está representado en las estructura VO disponibles



Un ejemplo del uso de la estructura contenida en el CueExeVO para recorrer las exenciones vigentes en la cuenta podría ser:

```
var listCueExe = decJurAPI.listCueExe;

// Analisis de Exencion por Actividad
var montoExento = 0;
for (var j = 0; j < listCueExe.size(); j++) {
    var cueExe = listCueExe.get(j);
    if (cueExe.getExencion().getExeRecConADec() != null && cueExe.getExencion().getExeRecConADec().getExeRecConADec() != null) {
```

```

// Si la exencion aplica para la actividad a declarar
if (cueExe.getCodConcepto() == decJurDet.getRecConADec().getCodConcepto()) {
    var porcentajeExento = 0;
    // Exencion Particular (con porcentaje definido al otorgar la exencion)
    if (cueExe.getExencion().getEsParticular() != null && cueExe.getPorcentaje() != 0) {
        porcentajeExento = cueExe.getPorcentaje()/100.0;
    } else {
        // Exencion No Particular
        if (cueExe.getExencion().getListExeRecCon() != null) {
            porcentajeExento = cueExe.getExencion().getListExeRecCon().getPorcentaje();
        }
    }

    var sobreExcedente = 0;
    if (cueExe.getExencion().getAplicaSobreExcedente() != null && cueExe.getExencion().getMontoTope() != 0) {
        sobreExcedente = cueExe.getExencion().getMontoTope();
    }

    montoExento = (decJurDet.getTotalConcepto() - sobreExcedente);
    if (montoExento < 0) montoExento = 0;

    // Cargamos observacion
    if (montoExento > 0) {
        decJurDet.setImporteExento(montoExento);
        decJurDet.setIdExencion(cueExe.getExencion().getId());
        var observaciones = decJur.getObservaciones();
        observaciones += " Actividad con Exención "+cueExe.getActividad();
        decJur.setObservaciones(observaciones);
    }
    break; // Espera solo una exencion por actividad
}
}
}
decJurDet.setTotalConcepto(decJurDet.getTotalConcepto()-montoExento);

```

A continuación se copia dos ejemplos de programas de cálculo de Declaración Juradas para recursos principales sobre Comercio

Ejemplo 1

```

/**
 * Cálculo de Declaración Jurada
 *
 *
 * Parametros de entrada:
 * - fechaVencimiento: Fecha Vencimiento de la original
 * - decJur: estructura de declaracion jurada inicializada y con parametros de entrada
 *
 * Parametro de salida:
 * - decJur: estructura de declaracion jurada con valores calculados
 */

Grs.load("<siat/siat.js>")

main = function(decJurAPI){

    var decJur = decJurAPI.decJur;
    var listCueExe = decJurAPI.listCueExe;

    // Se inicializan las observaciones de la DecJur

```

```

decJur.setObservaciones("");

// Determinamos el concepto principal como el de mayor base imponible
var mayorBase = 0;
var decJurDetPrincipal;
for (var i = 0; i < decJur.getListDecJurDet().size(); i++) {
    var decJurDet = decJur.getListDecJurDet().get(i);
    decJurDet.marcarComoNoPrincipal(); // Inicializa la marca
    if(decJurDet.getBase() > mayorBase){
        mayorBase = decJurDet.getBase();
        decJurDetPrincipal = decJurDet;
    }
}
// Marcamos el concepto principal
if(mayorBase > 0){
    decJurDetPrincipal.marcarComoPrincipal();
}

var subTotal = 0;
var minimoGeneral = 0;
// Calculo de baseXalicuota y determinacion de minimo a comparar
for (var i = 0; i < decJur.getListDecJurDet().size(); i++) {
    var decJurDet = decJur.getListDecJurDet().get(i);
    decJurDet.setTotalConcepto(decJurDet.getBase() * decJurDet.getAlicuota());

    // Analisis de Exencion por Actividad
    var montoExento = 0;
    for (var j = 0; j < listCueExe.size(); j++) {
        var cueExe = listCueExe.get(j);
        if (cueExe.getExencion().getExeRecConADec() != null) {
            // Si la exencion aplica para la actividad a
            if (cueExe.getCodConcepto() == decJurDet.getTotalConcepto()) {
                var porcentajeExento = 0;
                // Exencion Particular (con porcentaje)
                if (cueExe.getExencion().getEsParticular()) {
                    porcentajeExento = cueExe.getPorcentajeExento();
                } else {
                    // Exencion No Particular
                    if(cueExe.getExencion().getLimaExento()) {
                        porcentajeExento = cueExe.getPorcentajeExento();
                    }
                }
            }

            var sobreExcedente = 0;
            if (cueExe.getExencion().getAplicaSobreExcedente()) {
                sobreExcedente = cueExe.getExentoSobreExcedente();
            }

            montoExento = (decJurDet.getTotalConcepto() * porcentajeExento) + sobreExcedente;
            if (montoExento < 0) montoExento = 0;

            // Cargamos observacion
            if (montoExento > 0) {
                decJurDet.setImporteExento(montoExento);
                decJurDet.setIdExencion(cueExe.getIdExencion());
                var observaciones = decJurDet.getObservaciones();
                observaciones += " Actividad " + cueExe.getActividad() + " Exento: " + montoExento + "  
";
                decJurDet.setObservaciones(observaciones);
            }
            break; // Espera solo una exencion por actividad
        }
    }
}
decJurDet.setTotalConcepto(decJurDet.getTotalConcepto() - montoExento);

```

```
        subTotal += decJurDet.getTotalConcepto();

        var minimo = decJurDet.getRecConADecVal().getMinimo();
        if (decJurDet.getRecConADecVal().getMulMinPorUni().getEsSI()) {
            minimo = decJurDet.getRecConADecVal().getMinimo() * (
        }

        if (minimoGeneral < minimo) {
            minimoGeneral = minimo;
        }
    }

// Comparacion con minimo
if (subTotal>minimoGeneral) {
    decJur.setSubtotal(subTotal);
} else {
    decJur.setSubtotal(minimoGeneral);
}

// Mostrar minimo
decJur.setMinRec(minimoGeneral);
decJur.marcarMostrarMinimo();

// Analisis de Exencion a la cuenta en el periodo
var exentoTotal = false;
var porcentajeExento = 0;
var porcentajeExentoFondo = 0;
var idExencionSobreFondo = null;
for (var j = 0; j < listCueExe.size(); j++) {
    var cueExe = listCueExe.get(j);
    // Solo se tienen en cuenta exenciones geneales (sobre cuentas)
    if (cueExe.getExencion().getExeRecConADec() == null || cueExe.getExeRecConADec().getId() != null) {
        // Exencion Total
        if (cueExe.getExencion().getEsParcial() == null || cueExe.getExencion().getEsParcial() == true) {
            exentoTotal = true;
        } else {
            // Exencion Parcial
            // Exencion Particular (con porcentaje definido)
            if (cueExe.getExencion().getEsParticular() != null) {
                // En este caso se aplica a todos los conceptos
                porcentajeExento = cueExe.getPorcentajeExento();
            } else {
                // Exencion No Particular
                if (cueExe.getExencion().getListExeRecConAli().size() > 0) {
                    porcentajeExento = cueExe.getPorcentajeExento();
                }
            }
        }
    }
}

// Se teea el idExencion sobre la Declaracion Jurada por pagar
decJur.setIdExencion(cueExe.getExencion().getId());

var observaciones = decJur.getObservaciones();
observaciones += " Exención Vigente "+cueExe.getExeRecConADec().getDescripcion()+" ";
decJur.setObservaciones(observaciones);
}
```

```

var montoExentoFondo = 0;
if (porcentajeExentoFondo > 0) {
    montoExentoFondo = decJurAdiGA.getAdicional() * porcentajeExentoFondo;
    if (montoExentoFondo < 0) montoExentoFondo = 0;
}
if (exentoTotal) {
    montoExentoFondo = decJurAdiGA.getAdicional();
}
decJurAdiGA.setImporteExento(montoExentoFondo);
decJurAdiGA.setIdExencion(idExencionSobreFondo);

decJur.getListDecJurAdi().clear();
decJur.getListDecJurAdi().add(decJurAdiGA);

// Aplicacion de Fondo de Obra de Desague
var recAli = Packages.ar.gov.rosario.siat.def.buss.bean.RecAli.getByCodigoAlienacion(decJurAdiGA.getAlienacion());
var decJurAdi = new Packages.ar.gov.rosario.siat.gde.iface.model.DecJurAdi();
decJurAdi.setDecJur(decJur);
decJurAdi.setAdicional(recAli.getAlicuota() * decJur.getSubtotal());
decJurAdi.setRecAli(recAli.toVO(0, false));
decJurAdi.setRecTipAli(recAli.getRecTipAli().toVO(0, false));
// Si existe exencion sobre fondo se aplica
var montoExentoFondo = 0;
if (porcentajeExentoFondo > 0) {
    montoExentoFondo = decJurAdi.getAdicional() * porcentajeExentoFondo;
    if (montoExentoFondo < 0) montoExentoFondo = 0;
}
if (exentoTotal) {
    montoExentoFondo = decJurAdi.getAdicional();
}
decJurAdi.setImporteExento(montoExentoFondo);
decJurAdi.setIdExencion(idExencionSobreFondo);

decJur.getListDecJurAdi().add(decJurAdi);

decJur.setDeterminado(decJur.getSubtotal() + decJurAdiGA.getAdicional());

// Aplicacion de exencion
var montoExento = 0;
if (porcentajeExento > 0) {
    var montoExento = decJur.getDeterminado() * porcentajeExento;
    if (montoExento < 0) montoExento = 0;
}
decJur.setImporteExento(montoExento);

// Caso Exento Total
if (exentoTotal) {
    decJur.setImporteExento(decJur.getDeterminado());
}

// Si la cuenta esta marcada como Parque Industrial la DDJJ no determina deuda
var parqueIndustrial = decJurAPI.mapAtributos.ParqueIndustrial;
if (parqueIndustrial == 1) {
    decJur.marcarNoDeterminaDeuda();
}

return decJurAPI;
}

```

Ejemplo 2

```

/**
 * Cálculo de Declaración Jurada

```

```

*/
*
* Parametros de entrada:
*   - fechaVencimiento: Fecha Vencimiento de la original
*   - decJur: estructura de declaracion jurada inicializada y con paramet
*
* Parametro de salida:
*
*   - decJur: estructura de declaracion jurada con valores calculados
*/
Grs.load("<siat/siat.js>")

main = function(decJurAPI){

    var decJur = decJurAPI.decJur;
    var listCueExe = decJurAPI.listCueExe;

    // Se inicializan las observaciones de la DecJur
    decJur.setObservaciones("");

    var subTotal = 0;
    var minimoGeneral = 0;
    // TODO ver si lo tomamos de tabla recMinDec
    minimoGeneral = 150;

    // Determinar subtotal por actividad
    var minimoEspecial = 0;
    for (var i = 0; i < decJur.getListDecJurDet().size(); i++) {
        var decJurDet = decJur.getListDecJurDet().get(i);
        decJurDet.setTotalConcepto(decJurDet.getBase() * decJurDet.get

        // Minimo especial
        var minimo = decJurDet.getRecConADecVal().getMinimo();
        if (decJurDet.getRecConADecVal().getMulMinPorUni().getId() != null)
            minimo = decJurDet.getRecConADecVal().getMinimo() * (

    }

    // Caso de Monto Fijo
    if (decJurDet.getRecConADecVal().getMontoFijo() != null && decJurDet.getRecConADecVal().getMontoFijo() > 0)
        decJurDet.setTotalConcepto(decJurDet.getRecConADecVal().getMontoFijo());

    if (minimoEspecial < minimo) {
        minimoEspecial = minimo;
    }

    // Analisis de Exencion por Actividad
    var montoExento = 0;
    for (var j = 0; j < listCueExe.size(); j++) {
        var cueExe = listCueExe.get(j);
        if(cueExe.getExencion().getExeRecConADec() != null && cueExe.getExencion().getExeRecConADec().getMontoExento() > 0)
            // Si la exencion aplica para la actividad a
            if(cueExe.getCodConcepto() == decJurDet.getRecConADecVal().getCodConcepto())
                var porcentajeExento = 0;
                // Exencion Particular (con porcentaje)
                if(cueExe.getExencion().getEsParticular())
                    porcentajeExento = cueExe.getExencion().getPorcentajeExento();
                }else{
                // Exencion No Particular
                if(cueExe.getExencion().getEsGeneral())
                    porcentajeExento = cueExe.getExencion().getPorcentajeExento();
                }
    }
}

```



```

        var sobreExcedente = 0;
        if(cueExe.getExencion().getAplicaSob
            sobreExcedente = cueExe.getE
        }

        montoExento = (decJurDet.getTotalCon
        if(montoExento < 0) montoExento = 0;

        // Cargamos observacion
        if(montoExento > 0){
            decJurDet.setImporteExento(m
            decJurDet.setIdExencion(cueE
            var observaciones = decJur.g
            observaciones += " Actividad
            decJur.setObservaciones(obse
        }
        break; // Espera solo una exencion p
    }
}
}
decJurDet.setTotalConcepto(decJurDet.getTotalConcepto()-monto
    subTotal += decJurDet.getTotalConcepto();
}
// Se compara el mayor entre el mayor minimo especial y el minimo ge
if(minimoEspecial > minimoGeneral)
    minimoGeneral = minimoEspecial;

// Mostrar minimo
decJur.setMinRec(minimoGeneral);
decJur.marcarMostrarMinimo();

// Se compara subtotal con minimo
if (subTotal>minimoGeneral) {
    decJur.setSubtotal(subTotal);
} else {
    decJur.setSubtotal(minimoGeneral);
}

// Analisis de Exencion a la cuenta en el periodo
var exentoTotal = false;
var porcentajeExentoTasa = 0;
var porcentajeExentoFondo = 0;
var idExencionSobreFondo = null;
for (var j = 0; j < listCueExe.size(); j++) {
    var cueExe = listCueExe.get(j);
    // Solo se tienen en cuenta exenciones geneales (sobre cuenta
    if(cueExe.getExencion().getExeRecConADec() == null || cueExe
        // Exencion Total
        if(cueExe.getExencion().getEsParcial() == null || cu
            exentoTotal = true;
        } else {
            // Exencion Parcial
            // Exencion Particular (con porcentaje defin.
            if(cueExe.getExencion().getEsParticular() !=
                // En este caso se aplica a todos lo
                porcentajeExentoTasa = cueExe.getPor
                porcentajeExentoFondo = cueExe.getPo
            }else{
                // Exencion No Particular
                for (var h = 0; h < cueExe.getExenci
                    var exeRecCon = cueExe.getExe
                // Tasa (DER)
                if(exeRecCon.getRecCon() != 1

```

```

                                porcentajeExentoTasa

                                //Fondo de Comercio (Fondo)
                                if(exeRecCon.getRecCon() != null){
                                    porcentajeExentoFondo = exeRecCon.getPorcentajeExentoFondo();
                                    idExencionSobreFondo = exeRecCon.getIdExencionSobreFondo();
                                }
                            }
                        }
                    }
                }
                // Setea el idExencion sobre la Declaracion Jurada por el Exento
                //decJur.setIdExencion(cueExe.getExencion().getId());

                var observaciones = decJur.getObservaciones();
                observaciones += " Exención Vigente " + cueExe.getExento();
                decJur.setObservaciones(observaciones);
            }
        }

        // Aplicacion de exencion sobre la Tasa (se considera antes de calcular el exento)
        var montoExentoTasa = 0;
        decJur.setImporteExento(0);
        if(porcentajeExentoTasa > 0){
            var montoExentoTasa = decJur.getSubtotal() * porcentajeExentoTasa;
            if(montoExentoTasa < 0) montoExentoTasa = 0;
        }
        decJur.setImporteExento(montoExentoTasa);

        // Aplicacion de Fondo de Asistencia Social
        var recAli = Packages.ar.gov.rosario.siat.def.buss.bean.RecAli.getByRecConId(recCon.getId());
        var decJurAdi = new Packages.ar.gov.rosario.siat.gde.iface.model.DecJurAdi();
        decJurAdi.setDecJur(decJur);
        // El adicional se aplica sobre el valor considerando la exencion
        decJurAdi.setAdicional(recAli.getAlicuota() * (decJur.getSubtotal() - decJur.getImporteExento()));
        decJurAdi.setRecAli(recAli.toVO(0, false));
        decJurAdi.setRecTipAli(recAli.getRecTipAli().toVO(0, false));
        // Si existe exencion sobre fondo se aplica
        var montoExentoFondo = 0;
        if(porcentajeExentoFondo > 0){
            montoExentoFondo = decJurAdi.getAdicional() * porcentajeExentoFondo;
            if(montoExentoFondo < 0) montoExentoFondo = 0;
        }
        if(exentoTotal){
            montoExentoFondo = decJurAdi.getAdicional();
        }
        decJurAdi.setImporteExento(montoExentoFondo);
        decJurAdi.setIdExencion(idExencionSobreFondo);

        decJur.getListDecJurAdi().clear();
        decJur.getListDecJurAdi().add(decJurAdi);

        // Aplicar Descuento Buen Contribuyente
        var buenContribuyente = decJurAPI.mapAtributos.BuenContr;
        var montoDescuento = 0;
        decJur.getListDecJurDes().clear();
        if(buenContribuyente == "1"){
            var decJurDes = new Packages.ar.gov.rosario.siat.gde.iface.model.DecJurDes();
            decJurDes.setDecJur(decJur);

            //Si el importe determinado es <= 500, se aplica el 10% al B
            if(decJur.getSubtotal() <= 500.0){
                decJurDes.setDescripcion("Descuento Buen Pagador 10%");
                decJurDes.setDescuento((decJur.getSubtotal() - montoExentoTasa) * 0.1);
            }else{
                //Si el importe es > 500, se aplica un 5% al BP
            }
        }
    }
}

```

```

        decJurDes.setDescripcion("Descuento Buen Pagador 5%")
        decJurDes.setDescuento((decJur.getSubtotal()-montoExento)/100)
    }
    decJur.getListDecJurDes().add(decJurDes);

    montoDescuento = decJurDes.getDescuento();
}

// Calculo monto determinado (sin restar montos exentos a tasa o fondo)
decJur.setDeterminado(decJur.getSubtotal() + decJurAdi.getAdicional());

// Caso Exento Total
if(exentoTotal){
    decJur.setImporteExento(decJur.getDeterminado());
}

// Aplicacion de multa (si corresponde por configuracion del recurso)
if(decJur.getRecurso().getAplMulDecJur().getEsSI() && decJur.getTipDerecho().getEsSI()){
    // Si la presentacion de la DDJJ es posterior al vencimiento
    if(decJurAPI.presentaDecJurFueraDeTermino()){
        // Multa configurada por recurso (Por defecto para por recurso)
        var tipoMultaV0 = decJur.getRecurso().getTipMulDecJur();

        var decJurMul = new Packages.ar.gov.rosario.siat.gde
        decJurMul.setDecJur(decJur);
        decJurMul.setTipoMulta(tipoMultaV0);
        decJurMul.setImporte(tipoMultaV0.getImporte());
        if(decJurMul.getImporte() != null && decJurMul.getImporte() > 0){
            decJur.getListDecJurMul().clear();
            decJur.getListDecJurMul().add(decJurMul);
        }
    }
}

return decJurAPI;
}

```

Y finalmente un último ejemplo de un programa de cálculo que podría utilizarse en la configuración de un recurso autoliquidable para Agentes de Retención donde no se declaran bases imponibles sino que toma los importes de retenciones registradas o importadas.

Ejemplo 3

```

/**
 * Cálculo de Declaración Jurada para agentes de Retención
 *
 *
 * Parametros de entrada:
 * - fechaVencimiento: Fecha Vencimiento de la original
 * - decJur: estructura de declaracion jurada inicializada y con parametros de entrada
 *
 * Parametro de salida:
 * - decJur: estructura de declaracion jurada con valores calculados
 */
Grs.load("<siat/siat.js>")
Grs.load("<siat/const.js>")

```

```

main = function(decJurAPI){
    var Const = Siat.Const;
    var Sql = Grs.sql(Const.DsSiat);

    var decJur = decJurAPI.decJur;

    // Se inicializan las observaciones de la DecJur
    decJur.setObservaciones("");

    var subTotal = 0;

    // Determinar subtotal por concepto declarado
    var cantAct = 0;
    var totaldeterminado = 0;
    for (var i = 0; i < decJur.getListDecJurDet().size(); i++) {
        cantAct++;
        var decJurDet = decJur.getListDecJurDet().get(i);

        // Obtengo las retenciones registradas en el periodo
        sqlImporteRetenciones = "select sum(importeSujetoARetencion) as total,
        var idCuenta = decJur.cuenta.id;
        var fechaInicioPeriodo = decJurAPI.getDate(decJur.getAnio(), decJur.getMes(), decJur.getDia());
        // La funcion getDate toma el mes comenzando en cero, por lo que se le suma 1
        var fechaInicioProximoPeriodo = decJurAPI.getDate(decJur.getAnio(), decJur.getMes() + 1, decJur.getDia());

        // Se consultan las retenciones registradas con fecha de retencion entre el periodo
        sqlRespuesta = Sql.list(sqlImporteRetenciones , [idCuenta, fechaInicioPeriodo, fechaInicioProximoPeriodo]);

        sqlRespuesta = sqlRespuesta == null ? [] : sqlRespuesta.toArray();
        var totalConcepto = 0;
        var totalSujetoARetencion = 0;
        var cantidad = 0;
        for(var i = 0; i < sqlRespuesta.length; i++) {
            var rta = sqlRespuesta[i];
            importe = rta.total;
            totalConcepto = importe;
            totalSujetoARetencion = rta.sujetoaretencion;
            cantidad = rta.cantidad;
            break;
        }
        if(totalConcepto == null) totalConcepto = 0;
        if(totalSujetoARetencion == null) totalSujetoARetencion = 0;
        if(cantidad == null) cantidad = 0;

        totaldeterminado = totalConcepto*1;
        decJurDet.setBase(totalSujetoARetencion*1);
        decJurDet.setCanUni(cantidad*1);
        //decJurDet.getRecConADecVal().setDecCanUni(new Packages.com.rosariosiat.def.buss.bean.RecConADecVal());

        //Se fija si el agente tiene una alícuota especial para la fecha de retencion
        var alicuota = decJurDet.getRecConADecVal().getAlicuota();
        var atrAlicuota = Packages.ar.gov.rosario.siat.def.buss.bean.RecConADecVal().getAtrAlicuota();
        var alicuotaCuenta = null;
        if (atrAlicuota != null){
            var recAtrCueV = null;
            var recAtrCue = Packages.ar.gov.rosario.siat.def.buss.bean.RecConADecVal().getRecAtrCue();
            if(recAtrCue != null)
                recAtrCueV = Packages.ar.gov.rosario.siat.pad.buss.bean.RecAtrCueV();
            if(recAtrCueV != null){
                //trato de tomar el valor
                try{
                    alicuotaCuenta = parseFloat(recAtrCueV.getAtrAlicuota());
                }catch(e){
                }
            }
        }
    }
}

```

```

    }
    }
    if (alicuotaCuenta != null){
        alicuota = alicuotaCuenta;
    }

    decJurDet.setTotalConcepto(decJurDet.getBase() * alicuota);

    subTotal += decJurDet.getTotalConcepto();
}
decJur.setSubtotal(subTotal);

// Se agrega el valor de Fondo de Promocion total retenido (sólo info
var recAli = Packages.ar.gov.rosario.siat.def.buss.bean.RecAli.getBy...
var decJurAdi = new Packages.ar.gov.rosario.siat.gde.iface.model.Dec...
decJurAdi.setDecJur(decJur);
decJurAdi.setAdicional(recAli.getAlicuota() * decJur.getSubtotal());
decJurAdi.setRecAli(recAli.toVO(0,false));
decJurAdi.setRecTipAli(recAli.getRecTipAli().toVO(0,false));
decJur.getListDecJurAdi().clear();
decJur.getListDecJurAdi().add(decJurAdi);

// Como total se toma el obtenido de las retenciones registradas
decJur.setDeterminado(totaldeterminado*1);
decJur.setImporteExento(0);

// Aplicacion de multa (si corresponde por configuracion del recurso
if(decJur.getRecurso().getAplMulDecJur().getEsSI() && decJur.getTipDe...
    // Si la presentacion de la DDJJ es posterior al vencimiento
    if(decJurAPI.presentaDecJurFueraDeTermino()){
        // Multa configurada por recurso (Por defecto para p...
        var tipoMultaVO = decJur.getRecurso().getTipMulDecJu...

        if(tipoMultaVO != null && tipoMultaVO.getId() > 0){
            var decJurMul = new Packages.ar.gov.rosario.siat.gde.iface.model.DecJurMul();
            decJurMul.setDecJur(decJur);
            decJurMul.setTipoMulta(tipoMultaVO);
            // Se busca importe con vigencia a la fecha de...
            var tipoMultaImporte = null;
            if(tipoMultaVO != null)
                tipoMultaImporte = Packages.ar.gov.rosario.siat.gde.iface.model.TipoMultaImporte();
            decJurMul.setImporte(0);
            if(tipoMultaImporte != null)
                decJurMul.setImporte(tipoMultaImporte.getImporte());

            // En caso que se encuentre previamente notificado
            var notificado = decJurAPI.mapAtributos.Notificado();
            if(notificado == "1"){
                decJurMul.setImporte(decJurMul.getImporte());
            }

            if(decJurMul.getImporte() != null && decJurMul.getImporte().getImporte() != 0){
                decJur.getListDecJurMul().clear();
                decJur.getListDecJurMul().add(decJurMul);
            }
        }
    }
}

return decJurAPI;
}

```

En este último ejemplo aprovechamos para mencionar más posibilidades

que se disponen para la generación del programa de cálculo como realizar una consulta por SQL

```
// Obtengo las retenciones registradas en el periodo
sqlImporteRetenciones = "select sum(importeSujetoARetencion) as sujetoaretencion, :
var idCuenta = decJur.cuenta.id;
var fechaInicioPeriodo = decJurAPI.getDate(decJur.getAnio(),decJur.getPeriodo()-1, 1
// La funcion getDate toma el mes comenzando en cero, por lo tanto al no restar se e
var fechaInicioProximoPeriodo = decJurAPI.getDate(decJur.getAnio(),decJur.getPeriodo

// Se consultan las retenciones registradas con fecha de retencion dentro del periodo
sqlRespuesta = Sql.list(sqlImporteRetenciones ,[idCuenta, fechaInicioPeriodo, fechaI
```

O acceder a clases de SIAT a través de sus paquetes:

```
var tipoMultaImporte = Packages.ar.gov.rosario.siat.gde.buss.bean.TipoMultaImporte.g
```

Una vez realizados todos estos pasos sólo resta ingresar a una cuenta contributiva del recurso, seleccionar una deuda e ingresar a la opción de Declaración Jurada para verificar nuestra configuración.