**(Obrigatório)**

|  |  |
| --- | --- |
| Especificação de Requisitos | |
| Projeto/Versão: M\_SER\_MI001/ 12.1.8 | Requisito/Módulo: PCREQ-6629/ TSS |
| Sub-Requisito/Función: | Tarefa/Chamado: PDR\_SER\_MI001-176 |
| País: México | Data Especificação: 07/08/2015 |

|  |  |  |
| --- | --- | --- |
| Rotinas Envolvidas | | |
| **Rotina** | **Tipo de Operación** | **Opción de Menu** |
|  | Incluir |  |

|  |  |
| --- | --- |
| Estratégia de Desenvolvimento e Liberação | |
| Produto | *TSS* |
| Release que está sendo desenvolvido | *12.1.8* |
| Possui Réplica? | ( )Sim (X)Não |
| Qual a versão? |  |

## Objetivo

Implementar la integración de TSS con los PACs correspondientes para el Timbrado Electrónico de Recibos de Nómina para México.

## Definición de la Regla de Negocio

La Integración con los PACSserá realizada a través del TSS mediante la parametrización de los métodos dependiento del PAC utilizado para timbrar.

Para la integración entre el TSS y Protheus, se dispoinibiliza el siguiente Web service con los métodos necesarios para la integración:

* **Web Service: PRUEBASWSORDENPRODUCCION**
  + Métodos
* **PRUEBASORDENPRODUCCION**: Método para registro de las entidades y sus datos para transmitir al PAC.

A integración del TSS con el PAC será realizada forma assíncrona. Siendo:

**Processo assíncrono:**

Emisión de documentos: TSS recibirá la petición, guardará estas en la BD e indicará al ERP rque se han recebido los datos de la petición de timbrado, estas peticiones serán processadas por el TSS a través del Job “wfRecNomCFDI” para después poder ser consultadas por el ERP a través del método MonitorDoc.

**Definición de los métodos**

* **PRUEBASRETORDENP**: A través de este método Protheus deberá realizar la configuración para las entidades registradas en el TSS. El principal objetivo será guardar la información necesaria para realizar las conexiones con el o los PAC’s que se requiera para el timbrado de los recibos electrónicos.

Los datos serán guardados en la tabla SPED001M.

**Parámetros de entrada:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parâmetro | Tipo | Tam. | Ocorrência | Pai | Observação |
| C2\_PRODUTO | STRING | - | 1-1 | - | Producto |
| C2\_QUANT | STRING | - | 1-1 | - | Cantidad |
| C2\_DATPRI | STRING | - | 1-1 | - | La feha se introduce sin ninguna mascara |
| C2\_DATPRI | STRING | - | 1-1 | - | La feha se introduce sin ninguna mascara |
| C2\_EMISSAO | STRING | - | 1-1 | - | La feha se introduce sin ninguna mascara |
| CONSIDSUBSTRUC | STRING | - | 1-1 | - | Valores aceptados S=SI, N=NO |
| PRUEBASRETORDENP | STRING | - | 1-1 | - | VARIABLE QUE CONTIENE EL METODO DE RESPUESTA. |

**Exemplo:**

**Petición**

Principio del formulario

<?xml version="1.0" encoding="utf-8"?"

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<C2\_PRODUTO>**STRING**</C2\_PRODUTO>

<C2\_QUANT>**STRING**</C2\_QUANT>

<C2\_DATPRI>**STRING**</C2\_DATPRI>

<C2\_DATPRF>**STRING**</C2\_DATPRF>

<C2\_EMISSAO>**STRING**</C2\_EMISSAO>

<CONSIDSUBSTRUCT>**STRING**</CONSIDSUBSTRUCT>

</soap:Body>

</soap:Envelope>

Final del formulario

**Procesamiento de datos**

Deberá validar los parámetros de Entrada siguientes:

**Entidad,**

Con el el parámetro Entidad, buscará en la tabla SPED001M, si no encuentra el registro, no deberá seguir el proceso y enviar :

<body>

<cfgEmpresaResult>

<result>.F.</result>

<mensagem>No existe Entidad relacionada al ID, favor de verificar.</mensagem>

</cfgEmpresaResult>

</body>

PAC,

Deberá validar que el parâmetro PAC se encuentre dentro de los valores : PAC >0 AND PAC < 12

De lo contrario retornar:

<body>

<cfgEmpresaResult>

<result>.F.</result>

<mensagem>Identificador de PAC inválido</mensagem>

</cfgEmpresaResult>

</body>

Si se encontró el registro de la entidad (SPED001M->ID\_ENT == ::Entidad)

cXML += “<Account>”

cXML += “<idEnt>” + ::Entidad +”</idEnt>”

cXML += “<RFC>” +::RFC + “</RFC>”

cXML += ”<certificate Password=’” + ::senha + ”’>” + ::certificado

+“</certificado>”

cXML += “<UserPAC>” + ::UserPac + ”</UserPAC>”

cXML += “<PassPAC>” + ::Pass

PAC + “</PassPAC>”

cXML += “</Account>”

SPED001M-> XMLCFG := cXML

Y retornar :

<body>

<cfgEmpresaResult>

<result>.T.</result>

<mensagem>Registro exitóso</mensagem>

</cfgEmpresaResult>

</body>

Si no se encuentra la entidade, retornar

Deberá validar que el parâmetro PAC se encuentre dentro de los valores : PAC >0 AND PAC < 12

De lo contrario retornar:

<body>

<cfgEmpresaResult>

<result>.F.</result>

<mensagem>Identificador de PAC inválido</mensagem>

</cfgEmpresaResult>

</body>

Generar um nuevo registro en la Tabla SPED000L creando como a continuación:

SPED000L-> ID\_ENT := ::Entidad

SPED000L-> PARAMETRO := “MV\_IDPAC”

SPED000L-> CONTEUDO := ::PAC

Cualquier otro error

<body>

<cfgEmpresaResult>

<result>.F.</result>

<mensagem>Ocurrió un error inesperado. Configuración no establecida</mensagem>

</cfgEmpresaResult>

</body>

**Parámetros de salida:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parámetro | Tipo | Tam. | Ocorrência | Pai | Observação |
| cfgEmpresaResult | Tag | - | 1-1 | - | Tag Raiz do retorno |
| Result | Booelan | 005 | 1-1 | admEmpresaResult | Verdadero si se realizo el registro de la empresa. |
|  | String | 255 | 1-1 | admEmpresaResult | Mensage de retorno |

**Exemplo:**

**Retorno:**

<body>

<cfgEmpresaResult>

<result>Boolean</result>

<mensagem>String</mensagem>

</cfgEmpresaResult>

</body>

* cfgAmb: A través del ERP se debe configurar el ambiente a utilizar para realizar las peticiones al PAC del envío y timbrado de documentos.
* SendDoc : El objetivo de este método es recibir las peticiones de timbrado o cancelación de los documentos para ser enviados posteriormente al PAC.

**Parâmetros de entrada:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parámetro | Tipo | Tam. | Ocorrência | Pai | Observação |
| SendDoc | Tag | - | 1-1 | - | Tag raiz do método |
| Token | String | 05 | 1-1 | SendDoc | Token de utilización del TSS |
| entidad | String | 06 | 1-1 | SendDoc | Id de la entidad |
| Documentos | Tag | - | 1-1 | SendDoc | Documentos para procesamiento. |
| loteSig | Tag | - | 1-1 | Documentos | Lote de documentos |
| docSig | Tag | - | 1-N | loteSig | Documentos |
| Id | String | 44 | 1-1 | docSig | Id de identificación del documento en TSS. |
| modelo | String | 02 | 1-1 | docSig | Modelo del documento:  RT = Recibo Nomina a Timbrar  RC = Recibo Nómina a Cancelar |
| Xml | Base64Binary | 400k | 1-1 | docSig | Xml del recibo de nómina |

**Exemplo:**

**Petición:**

<body>

<SendDoc>

<Token>String</Token>

<entidad>String</entidad>

<Documentos>

<loteSig>

<docSig>

<id>String</id>

<modelo>String</modelo>

<xml>base64Binary</xml>

</docSig>

<docSig>

<id>String</id>

<modelo>String</modelo>

<xml>base64BInary</xml>

</docSig>

<loteSig>

</Documentos>

</SendDoc>

</body>

Donde el XML enviado debe estar deacuerdo a los requerimentos especificados por el SAT. Ejemplo de envío

**Procesamiento de Datos**

1. Al recibir una petición del método SendDoc, TSS valida los parámetros de entrada de la siguiente manera:

Deberá verificar que el la Entidad exista en la tabla de registro de entidades SPED001L->ID\_ENT == :: Entidad, de los contrario deberá retornar un SoapFault indicando que la entidad no existe.

Validará que el parámetro igual a “RT” o “RC”, de lo contrario enviará SOAPFault indicando Modelo no válido..

1. Una vez realizadas las validaciones anteriores, grabar cada uno de los documentos en la BD, como a continuación

Si Modelo es RT

Verificar si existe el registro en la tabla SPED050L en dónde SPED050L->ID\_ENT+NFE\_ID = ::ENTIDAD+::DOCSIG:ID

Si no existe generar un nuevo registro en esta tabla como a continuación:

RecLock("SPED050L",.T.)

SPED050L->ID\_ENT := ::Entidad

SPED050L->NFE\_ID := ::docsig:id

SPED050L->MODELO := ::docsig:Modelo

SPED050L->DATE\_NFE := Date()

SPED050L->TIME\_NFE := Time()

SPED050L->AMBIENTE := Obtener de la SPED000L con la Entidad + “MV\_AMBIENT”

SPED050L->MODALIDADE:= 1

SPED050L->STATUS := 1

SPED050L->XML\_ERP := ::docSig:Xml SPED050L->ERRCODE := 0

SPED050L->ERRMSG := ""

SPED050L->(MsUnlock())

Si existe el registro, verificar, si el campo SPED050L->STATUS es 1,4 o 5 deberá actualizarse el registro de la sig manera:

RecLock("SPED050L",.F.)

SPED050L->ID\_ENT := ::Entidad

SPED050L->NFE\_ID := ::docsig:id

SPED050L->MODELO := ::docsig:Modelo

SPED050L->DATE\_NFE := Date()

SPED050L->TIME\_NFE := Time()

SPED050L->AMBIENTE := Obtener de la SPED000L con la Entidad + “MV\_AMBIENT”

SPED050L->MODALIDADE:= 1

SPED050L->STATUS := 1

SPED050L->XML\_ERP := ::docSig:Xml SPED050L->ERRCODE := 0

SPED050L->ERRMSG := ""

SPED050L->(MsUnlock())

Si SPED050L->STATUS = 6 y SPED050L->AMBIENTE = 2 y Ambiente actual es 1, deberá modificar el registro como se describió en el paso anterior.

Si SPED050L->STATUS = 6 y SPED050L->AMBIENTE=1 , el registro no podrá modificarse y se deberá retornar lo siguiente:

<body>

<SendDocResult>

<loteSigResult>

<docSigResult>

<resultDoc>.F.</resultDoc>

<erro>

<codigo>04</codigo>

<descricao>El document ya fue timbrado en ambiente Productivo. Solo procede cancelación

</descricao>

</erro>

<id>String</id>

</docSigResult>

</SendDocResult>

</body>

Si Modelo es RC

Verificar que exista registro en SPED050L en dónde SPED050L->ID\_ENT = ::Entidad, SPED050L->NFE\_ID=::DocSig:Id , SPED050L->STATUS = ‘6’ , Y SPED050L->MODELO = “RT ”, Es decir que el recibo a cancelar esté previamente timbrado.

Si el registro existe, actualizar lo siguiente:

RecLock("SPED050L",.F.)

SPED050L->ID\_ENT := ::Entidad

SPED050L->NFE\_ID := ::docsig:id

SPED050L->MODELO := ::docsig:Modelo

SPED050L->DATE\_NFE := Date()

SPED050L->TIME\_NFE := Time()

SPED050L->AMBIENTE := Obtener de la SPED000L con la Entidad + “MV\_AMBIENT”

SPED050L->MODALIDADE:= 1

SPED050L->STATUS := 7

SPED050L->XML\_ERP := ::docSig:Xml SPED050L->ERRCODE := 0

SPED050L->ERRMSG := ""

SPED050L->(MsUnlock())

Si existe pero, SPED050L->STATUS <> 6 deberá de retornar:

<body>

<SendDocResult>

<loteSigResult>

<docSigResult>

<resultDoc>.F.</resultDoc>

<erro>

<codigo>05</codigo>

<descricao>La cancelación procede solo sobre recibos previamente timbrados.

</descricao>

</erro>

<id>String</id>

</docSigResult>

</SendDocResult>

</body>

Códigos de error para el método SendDoc:

|  |  |
| --- | --- |
| Codigo | descrição |
| 01 | Erro de Schema |
| 02 | Entidad inválida o no registrada. |
| 03 | Modelo Inválido. |
| 04 | El documento ya fue timbrado en ambiente Productivo. Solo procede cancelación. |
| 05 | La cancelación procede solo sobre recibos previamente timbrados. |
| 06 | Falla en la grabación del registro. |

**Parámetros de salida:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parâmetro | Tipo | Tam. | Ocorrência | Pai | Observação |
| SendDocResult | Tag | - | 1-1 | - | Retorno del proceso. |
| loteSigResult | tag | - | 1-1 | SendDocResult | Documentos processados. |
| docSigResult | tag | - | 1-N | loteSigResult | Datos del Retorno |
| erro | Tag | - | 0-1 | docSigResult | Estrutura del Error |
| codigo | Numeric | 05 | 1-1 | erro | Código del rechazo |
| descricao | String | 100 | 1-1 | erro | Descripción del Error |
| id | String | 44 | 1-1 | docSigResult | Id del documento |
| resultDoc | Boolean | 05 | 1-1 | docSigResult | Resultado de la recepción del documento |

**Exemplo:**

**Retorno:**

<body>

<remessaDocResult>

<loteSigResult>

<docSigResult>

<resultDoc>boolean</resultDoc>

<erro>

<codigo>Numeric</codigo>

<descricao>String</descricao>

</erro>

<id>String</id>

</docSigResult>

<docSigResult>

<resultDoc>boolean</resultDoc>

<id>String</id>

</docSigResult>

</loteSigResult>

</remessaDocResult>

</body>

* **MonitorDoc**

A través de este método el ERPs podrá realizar consulta de los documentos electrónicos enviados al TSS. El método se define de la siguiente manera:

**Parámetros de entrada:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parâmetro | Tipo | Tam. | Ocorrência | Pai | Observação |
| monitorDoc | Tag | - | 1-1 | - | Retorno do método |
| Token | String | 05 | 1-1 | monitorDoc | Token para uso del TSS |
| entidad | String | 06 | 1-1 | monitorDoc | Id de la entidade registrada en TSS |
| modelo | String | 02 | 0-1 | monitorDoc | Modelo del documento electrónico a consultar  RT = Recibo Timbrar  RC = Recibo Cancelar  Si se envía vacío, retornará ambos modelos de documentos. |
| monitorSig | tag | - | 1-1 | monitorDoc | Tag raiz do monitor de documentos. |
| idInicial | String | 44 | 1-1 | monitorSig | Id del documento inicial a ser consultado. |
| IdFinal | tag | 44 | 1-1 | monitorSig | Id final del documento a ser consultado |

**Exemplo:**

**Requisição:**

<body>

<monitorDoc>

<Token>String</Token>

<entidad>String</entidad>

<modelo>String</modelo>

<monitorSig>

<idInicial>String</idInicial>

<idFinal>String</idFinal>

</monitorSig>

</monitorDoc>

</body>

**Parámetros de salida:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parâmetro | Tipo | Tam. | Ocorrência | Pai | Observação | |
| monitorDocResult | tag | - | 1-1 | - | Retorno do monitor. |

**Exemplo:**

**Retorno:**

<body>

<monitorDocResult>

<DocSigRet>

<AMBIENTE>INTEGER</AMBIENTE>

<AUTORIZACAO>STRING</AUTORIZACAO>

<CODIGO>INTEGER</CODIGO>

<DTAUTORIZACAO>DATE</DTAUTORIZACAO>

<HRAUTORIZACAO>STRING</HRAUTORIZACAO>

<ID>STRING</ID>

<CODIGO>INTEGER</CODIGO>

<DESCRICAO>STRING</DESCRICAO>

<XML>BASE64BINARY</XML>

</DocSigRet>

</monitorDocResult>

</body>

**Tabelas Utilizadas**

* SPED000L – cadastro de parâmetros.
* SPED001L – Cadastro de Empresas localizado
* SPED001M – Cadastro complementar de empresas localizado
* SPED050L – Registro de documentos a timbrar
* SPED052L – Histórico de procesamientos de los documentos