Especificación Técnica de la Solución

MTI-III-ETS

**Mejoras al módulo de Conciliación Bancaria, Corte de caja, Crédito y cobranza, Caja para facturación 3.3**

**Fecha de Publicación**

29/09/2017

Versión 1.0

**Documento Confidencial.**

Derechos Reservados. El contenido y formato de este documento son propiedad de Grupo Metropolitano S. A. de C. V. y es confidencial. Cualquier reproducción del todo o en parte está estrictamente prohibida sin el permiso escrito de Grupo Metropolitano, S. A. de C. V.

Este documento está sujeto a cambios. Comentarios, correcciones o preguntas deberán ser dirigidas al autor.

# Control de Cambios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Descripción** | **Autor** | **Autoriza** | **Fecha** |
| 1.0 | Versión Inicial | Iván Trujillo López itrujillo@softgrade.mx |  | 29/09/2016 |

# Índice

[Control de Cambios 2](#_Toc459043827)

[Índice 3](#_Toc459043828)

[Propósito del Documento 4](#_Toc459043829)

[1. Identificación 4](#_Toc459043830)

[2. Descripción del Requerimiento 4](#_Toc459043831)

[3. Análisis Técnico para la Solución 4](#_Toc459043832)

[4. Diseño Técnico de la Solución 5](#_Toc459043833)

[4.1. Acciones Técnicas 5](#_Toc459043834)

[Especificaciones técnicas. 5](#_Toc459043835)

[Cambios en Base de Datos 5](#_Toc459043836)

[4.2. Consideraciones y Dependencias 5](#_Toc459043837)

[5. Plan de Actividades 6](#_Toc459043838)

[6. Anexos 6](#_Toc459043839)

[6.1. Tabla de Referencia de Posibles Entregables 6](#_Toc459043840)

[6.2. Anexo 1 6](#_Toc459043841)

# Propósito del Documento

El objetivo de este documento es especificar las entradas, salidas y procedimientos de los que ha de consistir la funcionalidad u objeto o componente a desarrollar o modificar, con el fin de satisfacer el requerimiento del usuario.

# Identificación

|  |  |
| --- | --- |
| **N° de PEP o N° Remedy** | No aplica |
| **Tipo de Requerimiento** | *Extensión de funcionalidad.* |
| **Sistema** | Conciliación bancaria |
| **Módulo** | Conciliación |
| **Proceso** | Generación de múltiples TransBans. |
| **Descripción** | La nueva funcionalidad debe:   * Con base a los movimientos conciliados, es decir la interrelación existente entre un registro en estado de cuenta (Externo) y un documento pendiente de pago debe ser capaz de registrar ni uno, uno o múltiples TransBan (Folios de Transferencia Bancara) * Con la inclusión de esta capacidad el proceso que actualmente existe para edificios administrados (Metropoli) no debe ser afectado. * El sistema debe contar con un parámetro de tipo entero que permita establecer la cantidad de documentos relacionados a una TransBan. |

|  |  |  |
| --- | --- | --- |
| **Requerido por** |  | **Fecha**: |
| **Aprobado por** |  | **Fecha**: |

# Descripción del Requerimiento

Incorporar al sistema de conciliaciones bancarias de Grupo metropolitano los elementos necesarios para:

1. Ser capaz de generar ni uno, uno o múltiples movimientos de tipo *TransBan* asociados a un folio de conciliación bancaria.
2. Crear un parámetro que gobernará la cantidad máxima de documentos que pueden estar relacionados a una *TransBan*.
3. El parámetro deberá tener asignado un valor numérico de tipo entero, de no ser el caso la situación debe ser manejada por el sistema y reportada mediante mensaje a usuario y registro de log.
4. El valor propuesto para el parámetro de máximo de documentos es de 500, este mismo es enunciativo, más no limitativo.
5. El sistema deberá relacionar como máximo el número de documentos indicado por el parámetro del punto dos.
6. Si el máximo de documentos vinculados a una TransBan es excedido el sistema deberá sin asistencia del usuario generar una o más folios de TransBan según el escenario.
7. El código fuente de la solución se encuentra disponible en [**https://github.com/ivantrujillo79/conciliacionbancaria**](https://github.com/ivantrujillo79/conciliacionbancaria)
8. Utilice Github Desktop o Sourcetree para conectarse al repositorio
9. El código fuente a editar debe ser **sin excepción** tomado de la rama “development”

# Análisis Técnico para la Solución

De acuerdo al análisis técnico efectuado para cumplir con el requerimiento solicitado se ejecutarán las siguientes acciones:

**Creación de parámetro**

El sistema cuenta con un par de tablas para la creación y asignación de valor de parámetros, estas serán utilizadas para el objeto de este punto.

**Ventana emergente de verificación carga de conciliación manual**

Debido a que el proceso implica la carga de un archivo externo, éste deberá ser validado y a través de un formulario el usuario será informado de las incidencias o de los registros que han sido cargados a un área de staging.

**Creación de tabla relación para *TransBans* y conciliaciones**

Con la finalidad de que el sistema vincule a un determinado folio de conciliación múltiples *TransBan* se hace necesaria la creación de una tabla que permita vincular ambas entidades por medio de la “copia” de los campos que conforman la llave primaria de sus correspondientes tablas.

**Modificación del proceso de conciliación**

Una vez que la parametrización ha sido establecida y los controles de carga de datos se han ejecutado, el proceso de conciliación toma los diferentes insumos y es a partir de ellos deberá vincular dos cosas:

1. La relación entre un folio de conciliación y los folios de *TransBan*
2. La relación existente entre un folio de *TransBan* y los documentos conciliados
3. En presencia de *TransBan* con una cantidad de documentos relacionados superior al parámetro el proceso hará una segmentación de ésta en múltiples *TransBan*.

**Transaccionalidad del proceso**

El proceso de asiento de *TransBan* implica la afectación de diversas tablas principales del sistema E.R.P. SIGAMET, la afectación potencial a la consistencia de información en el mismo es un aspecto que debe evitarse, por lo mismo es que el procedimiento debe impedir dicha posibilidad a través de la implementación de transacciones breves; esto último con el fin de impedir la creación de cuellos de botella en otros procesos del negocio.

# Diseño Técnico de la Solución

## Acciones Técnicas

# Especificaciones técnicas.

## Creación de parámetro

El sistema E.R.P. SIGAMET -en lo siguiente SIGAMET- se compone de diversos módulos, de lo anterior la existencia de un catálogo que contiene los mismos, el nombre de la tabla es “Modulo”, por otra parte, cada módulo del sistema cuenta con sus parámetros, lo anterior es posible gracias a la tabla llamada “Parametro”. Ambas tablas mencionadas se esquematizan a continuación.



Fig. 1 Tablas de parametrización

El módulo de conciliaciones bancarias corresponde al id 30 y es éste el que debe ser utilizado para relacionar con el nuevo parámetro que será etiquetado como “**MaxDoctosTransBan**”.

Con el fin de obtener el valor actual de MaxDoctosTransban se deberá utilizar el método “**ValorParametro**” de la clase “SeguridadCB.Public.Parametros” que se ilustra a continuación:



Fig. 2 Clase *Parametros*

La clase *Parametros* se documenta a continuación:

# Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **lista** ArrayList  Private |  | *Predeterminado:* new ArrayList() |

Tabla 1 Atributos

# Operaciones

| **Método** | **Notas** | **Parámetros** |
| --- | --- | --- |
| **Parametros()**  Internal |  | **DataTable** [in]listaParametros |
| **ValorParametro()** string  Public |  | **short** [in]modulo  **string** [in]nombreParametro |

Tabla 2 Operaciones

La implementación bajo ninguna circunstancia usará el procedimiento almacenado **spObtieneParametroPorModulo** de forma directa para recuperar el valor de MaxDoctosTransBan o cualquier otro parámetro que en funcionalidades futuras se requiera, esto no aplica a funcionalidades ya implementadas.

## Ventana emergente de verificación carga de conciliación manual

Como parte del requerimiento la carga de archivo es necesaria una verificación de calidad de los datos, de lo anterior debe verificarse lo siguiente:

1. Conformidad del contenido del archivo con el *layout* esperado
2. No existencia de líneas en blanco
3. Existencia y contenido del encabezado.
4. Extensión del archivo (XLS y XLSX)

En caso de que por lo menos alguna de las situaciones mencionadas se suscite el sistema deberá reportar al usuario mediante los siguientes mensajes de error.

|  |  |
| --- | --- |
| **Validación** | **Mensaje mostrado al usuario** |
| 1 | “El formato del contenido del archivo que desea cargar no cumple la especificación \n Se esperan las columnas: Documento, Cuenta y Monto” |
| 2 | “No es posible cargar el archivo ya que contiene líneas en blanco, posición: {Número de línea}” |
| 3 | “El archivo no cuenta con encabezado, por favor verifique” |
| 4 | “El archivo a cargar debe ser de formato Excel, con extensión de archivo XLS o XLSX” |

Tabla 3 Mensajes de error durante el proceso de carga

La ventana emergente de carga del archivo deberá cumplir con la siguiente apariencia:



Fig. 3 Ventana de verificación de carga

Se utilizará el control ModalPopUpExtender que ofrece asp.net el ID del control será: “**mpeCargaArchivoConciliacionManual**” para presentar y ocultar el formulario descrito en la figura número tres.

El formulario descrito en la figura tres deberá ser llamado desde la página /conciliación/DetalleConciliacion.aspx a través de un botón y una opción de menú contextual en el grid de documentos internos.

A su vez se usará un control DataGridView ofrecido por asp.net con el ID “**dgvDetalleConciliacionManual**” con el objeto de presentar detalle al usuario de los registros (documentos) que pudieron ser conciliados a través de la carga del archivo.

Una vez que el archivo elegido por el usuario se ha cargado sin incidentes el sistema deberá mostrar el siguiente mensaje de confirmación:



Fig. 4 Mensaje de confirmación de carga

Donde:

**{Número de coincidencias}**: Corresponde al valor asignado a un contador que contiene el total de registros que pudieron relacionarse con pedidos con saldo del cliente (Internos) que se obtiene a través de la propiedad “**registrosCociliados**” de la clase “**ValidadorConciliacionManual**”.

**{Número de contrato}**: Es el número del contrato asociado al pedido indicado por el usuario en el archivo importado. Existen dos casos:

1. Si el cliente buscado es de tipo padre, el mensaje deberá ser como se indica en la figura cinco.
2. Si el cliente es hijo, deberá mostrarse como se indica en la figura cuatro.



Fig. 5 Mensaje para cliente padre

# Validación de ingestión de archivos

Con el fin de garantizar que la información cargada al sistema de conciliación bancaria es consistente y que no introducirá errores en el proceso es preciso ejecutar las validaciones indicadas en la tabla número tres. Para cobrar el punto mencionado será necesaria la creación de un paquete “*Validacion*” en la librería de clases en el proyecto Conciliación.RunTime de la solución.

Las interfaces y clases a implementar para el paquete *Validacion* son las siguientes:

## Validacion

Tipo: Paquete

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Conciliacion.Migracion.Runtime

Detalle: *Creado el 05/10/2017*. *Última modificación el 05/10/2017*

**Validacion** - *(diagrama Lógico )*

Creado por: Desarollo\_Transforma el 05/10/2017

Última modificación: 05/10/2017

Versión: 1.0.



Fig. 6 Librería de clases *Validacion*

### CuentaBancariaExisteException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  CuentaBancariaExisteException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### DetalleValidacion

Tipo: **Clase**

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Estructura en la que se contiene el detalle del resultado de una validación particular siendo exitosa o fallida.

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **VerificacionValida** bool  Public | Indica mediante true o false si el proceso concluyó exitosamente | *Predeterminado:* |
| **CodigoError** int  Public | En caso de error cada validador devolverá un número diferente (consultar cada clase) de cero, si el proceso es exitoso devuelve cero | *Predeterminado:* |
| **Mensaje** string  Public | Mensaje detallado que regresa el proceso de validación específico, tanto si se trata de un error o un éxito se espera un mensaje que detalle la situación | *Predeterminado:* |

### DocumentoReferenciaExisteException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  DocumentoReferenciaExisteException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### EncabezadoExisteException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  EncabezadoExisteException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### Exception

Tipo: **Clase**

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase base Exception del .net Framework

Referencia:

https://msdn.microsoft.com/en-us/library/system.exception(v=vs.110).aspx

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  LineaVaciaException | Public  Exception |  |
| Generalización  Origen -> Destino | Public  LayoutException | Public  Exception |  |
| Generalización  Origen -> Destino | Public  FormatoExcelException | Public  Exception |  |
| Generalización  Origen -> Destino | Public  EncabezadoExisteException | Public  Exception |  |
| Generalización  Origen -> Destino | Public  MontoValidoException | Public  Exception |  |
| Generalización  Origen -> Destino | Public  DocumentoReferenciaExisteException | Public  Exception |  |
| Generalización  Origen -> Destino | Public  CuentaBancariaExisteException | Public  Exception |  |

### FormatoExcelException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  FormatoExcelException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### LayoutException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  LayoutException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### LineaVaciaException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  LineaVaciaException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### MontoValidoException

Tipo: **Clase** Exception

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase que hereda de Exception

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Generalización  Origen -> Destino | Public  MontoValidoException | Public  Exception |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **ResultadoValidacion** DetalleValidacion  Public | Instancia de la clase DetalleValidacion con el detalle del error | *Predeterminado:* |

### ValidadorCyC

Tipo: **Clase**

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase concreta que se encarga de la validación del archivo Excel de conciliación de ejecutivo de crédito y cobranza

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Dependencia  Origen -> Destino | Public  ValidadorCyC | Public  IValidadorExcel |  |

Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **RutaArchivo** string  Public | Ruta del archivo a validar | *Predeterminado:* |
| **NombreArchivo** string  Public | Nombre del archivo a validar una vez que se encuentre en el servidor | *Predeterminado:* |
| **TipoMime** string  Public | Se pasa el tipo MIME del archivo para validación de tipo de archivo | *Predeterminado:* |

Operaciones

| **Metodo** | **Notas** | **Parámetros** |
| --- | --- | --- |
| **new()** ValidadorCyC  Public | Constructor por defecto |  |
| **new()** ValidadorCyC  Public | Constructor sobrecargado, requiere ruta y nombre de archivo a validar | string [in] RutaArchivo  string [in] NombreArchivo  string [in] TipoMIME |
| **ValidacionCompleta()** List<DetalleValidacion>  Public | Único método público para iniciar el proceso de validación. |  |
| **ValidaCuentaBancaria()** DetalleValidacion  Private | Método que verifica que todos los registros en el archivo corresponden a la cuenta bancaria asociada al folio de conciliación |  |
| **ValidaDocumentoReferencia()** DetalleValidacion  Private | Método que verifica la existencia del DocumentoReferencia proporcionado en el renglón |  |
| **ValidaMonto()** DetalleValidacion  Private | Verifica que el monto en el renglón:  -No esté en blanco  -No sea Nulo  -Sea numérico (dos puntos decimales)  -Sea mayor a cero |  |
| **ValidaEncabezado()** DetalleValidacion  Private | Valida mediante Hash MD5 que la concatenación sin espacios ni caracteres de finalización de los tres encabezados del archivo correspondan al hash MD5 de la cadena siguiente:  DocumentoCuentaMonto |  |
| **ValidaFormatoExcel()** DetalleValidacion  Private | string contentType = FileUpload1.PostedFile.ContentType  if (contentType == "application/vnd.ms-excel" )  {  }  else  {  LanzarError  }  Lista de tipos MIME  https://developer.mozilla.org/es/docs/Web/HTTP/Basics\_of\_HTTP/MIME\_types/Lista\_completa\_de\_tipos\_MIME |  |
| **ValidaLayout()** DetalleValidacion  Private | Verificar que las tres columnas y todas las filas contienen datos, que no hay más que tres columnas con datos y que se trata de sólo tres columnas |  |
| **ValidaLineaVacia()** DetalleValidacion  Private | Busca la existencia de líneas del archivo Excel en blanco y **REPORTA EL ID DE LAS LÍNEAS** |  |

### IValidadorExcel

Tipo: **Interfaz**

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: Validacion Palabras claves:

Detalle: *Creado el* *05/10/2017*. *Última modificación el 05/10/2017*.

Clase abstracta que funciona como fábrica de clases concretas que implementan determinada verificación de layout sobre archivos de formato Excel

*Conecciones*

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| Dependencia  Origen -> Destino | Public  ValidadorCyC | Public  IValidadorExcel |  |

El proceso a ejecutar por el flujo de carga se describe en el diagrama de secuencia de la figura siete:



Fig. 7 Proceso de carga y validación de archivo de documentos internos a conciliar por cyc.

La secuencia de pasos es la siguiente:

***Conecciones***

| **Conector** | **Origen** | **Destino** | **Notas** |
| --- | --- | --- | --- |
| **Secuencia**  new(string, string, string)  Origen -> Destino | Public  btnCargarArchivo | Public  objValidador |  |
| **Secuencia**  ValidacionCompleta()  Origen -> Destino | Public  btnCargarArchivo | Public  objValidador |  |
| **Secuencia**  ValidaFormatoExcel()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  ValidaEncabezado()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  ValidaLayout()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  ValidaLineaVacia()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  ValidaCuentaBancaria()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  ValidaDocumentoReferencia()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  ValidaMonto()  Origen -> Destino | Public  objValidador | Public  objValidador |  |
| **Secuencia**  Origen -> Destino | Public  objValidador | Public  DetalleConciliacion.aspx |  |

La prueba de concepto realizada debe ser integrada al sistema a través de un Web User Control, lo anterior se hará en la carpeta /ControlesUsuario/CargaManualExcelCyC (la carpeta no existe, crearla y cuidar de que se incluya en el control de versiones en el archivo del proyecto web). El control será llamado **“wucCargaManualExcelCyC”**.

Crear la clase en el proyecto *SitioConciliacion* bajo las siguientes especificaciones:



Fig. 8 Definición de Web User Control

### wucCargaManualExcelCyC

Tipo: **Clase**

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: SitioConciliacion

Detalle: *Creado el* *08/10/2017*. *Última modificación el 08/10/2017*.

## Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **TotalRegistrosCargados** int  Public | Propiedad que regresa el total de registros cargados en sistema, si existe un error debe asignarle un valor de cero.  ésta propiedad sólo tiene getter, no tiene sentido que componentes externos modifiquen su valor. | *Predeterminado:* |
| **RegistrosCargados** int  Public | Lista de elementos de la clase PedidoExcelCyc con el contenido leído por el componente. | *Predeterminado:* |
| **DetalleProcesoDeCarga** List<DetalleValidacion>  Public | Regresa el detalle del proceso de validación y carga, con dicho resultado se despliegan mensajes de error o de éxito según sea el caso. | *Predeterminado:* |

La implementación de llamados a la librería de clases *ValidadorCyC* será la misma que se indicó en el diagrama de secuencia de la figura siete.

Una vez que la clase haya sido implementada instancie en el formulario de conciliaciones uno a varios que corresponde al archivo siguiente:

/Conciliacion/FormasConciliar/UnoAVarios.aspx

El nombre de la instancia del control será **wucCargaExcelCyC** tal y como se muestra en el siguiente diagrama:



Fig. 9 Flujo de llamados para carga de archivo Excel de CyC

## 6.1.1 Conciliación por cantidad y referencia

La funcionalidad solicitada se refiere a la implementación de un nuevo tipo de conciliación llamada **“CONCILIACION DE INGRESOS POR COBRANZA A ABONAR PEDIDO”** y en lo particular para el algoritmo de Cantidad y referencia misma que en el catálogo TipoConciliacion tiene el ID 6, el flujo comentado se encuentra implementado en el formulario siguiente:

**FormasConciliar/CantidadYReferenciaConcuerdan.aspx**

Cree el nuevo tipo de conciliación, este deberá ser dado de alta en la tabla “TipoConciliacion” con el siguiente detalle:

|  |  |
| --- | --- |
| Identificador | Descripción |
| 6 | ‘CONCILIACION DE INGRESOS POR COBRANZA A ABONAR PEDIDO’ |

Modifique el procedimiento almacenado ***spCBCargarComboTipoConciliacion*** para que el formulario principal ***“Inicio.aspx”*** presente la nueva opción en el combo titulado “Tipo Conciliación” de acuerdo a lo presentado en el diagrama siguiente.



Fig. Flujo de carga de la pantalla principal del sistema

Modifique el código fuente localizado en el archivo CantidadYReferenciaConcuerdan.aspx.cs para remover las asignaciones de un valor fijo de 2 al miembro ***tipoConciliacion*** esta mala práctica deberá corregirse ya que actualmente está introduciendo errores en el proceso de conciliación. El valor que debe asignarse a la variable en todos casos proviene del query string llamado ***TipoConciliacion***. Adicionalmente modifique el alcance de la variable *tipoConciliacion* para que pase de *Public* a *Private*, para esta actividad válgase del siguiente diagrama.



Proceso de carga para Cantidad y Referencia concuerdan

## 6.1.2.3 Inclusión de un control de filtrado de clientes y resalte de facturas

Para los depósitos de los que no se tenga algún número de referencia de cliente, el usuario deberá buscar los pedidos de algún cliente en particular y a su vez debe ser posible resaltar en el grid de internos una factura determinada.

Para lo anterior se construirá un web user control, la clase se llamará “**wucBuscaClientesFacturas”** y será creada en la carpeta **“/ControlesUsuario/BuscadorClienteFactura/”**.

La definición y descripción del control es la que se describe a continuación.

### wucBuscaClientesFacturas

Tipo: **Clase**

Estado: . *Versión* 1.0. *Fase* 1.0.

Paquete: SitioConciliacion

Detalle: *Creado el* *09/10/2017*. *Última modificación el 09/10/2017*.

## Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **NumeroClienteFiltrar** string  Private | Número del cliente o de referencia a ser filtrado | *Predeterminado:* |
| **NumeroFacturaResaltar** string  Private | Número de factura que será resaltado en el GridRelacionado | *Predeterminado:* |
| **GridRelacionado** GridView  Public | Tipo GridView | *Predeterminado:* |

## Operaciones

| **Metodo** | **Notas** | **Parámetros** |
| --- | --- | --- |
| **FiltraCliente()** void  Private | Método privado en el que durante su ejecución los registros asignados en el datasource del grid asignado por medio de la propiedad **"GridRelacionado",** serán filtrados para sólo mostrar los pedidos relacionados con el cliente provisto por la propiedad **"NumeroClienteFiltrar".** | string [in] Cliente  GridView [in] GridRelacionado |
| **ResaltaFactura()** void  Private | Método que resaltará la columna Factura (efecto de marca textos) en el grid provisto por la propiedad **"GridRelacionado"** para todos los registros que cuenten con la cadena provista por el usuario a través de la propiedad **"NumeroFacturaResaltar"**. | string [in] Factura  GridView [in] GridView |

Una vez que el archivo Excel de conciliación manual ha sido leído, su contenido será cargado en una tabla de paso “**PartidasManuales**”

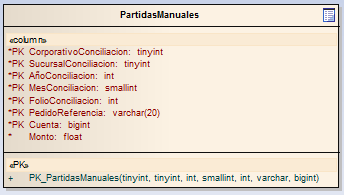


Figura 8. Tabla de staging de PartidasManuales

La tabla PartidasManuales se documenta a continuación:

# Atributos

| **Atributo** | **Notas** | **Restricciones y etiquetas** |
| --- | --- | --- |
| **CorporativoConciliacion** tinyint  Public    «column» | Referencia al catalogo Corporativo | *Predeterminado:* |
| **SucursalConciliacion** tinyint  Public    «column» | Referencia al catalogo Sucursal | *Predeterminado:* |
| **AñoConciliacion** int  Public    «column» | Año de la creación de la conciliación | *Predeterminado:* |
| **MesConciliacion** smallint  Public    «column» | Mes de la creación de la conciliación | *Predeterminado:* |
| **FolioConciliacion** int  Public    «column» | Consecutivo por mes, año, sucursal de la conciliación | *Predeterminado:* |
| **PedidoReferencia** varchar  Public    «column» | Pedido o Factura manual que se cubre con el pago, compuesta por AñoPedido + Célula + Pedido | *Predeterminado:* |
| **Cuenta** bigint  Public    «column» | Número de cuenta en la que se recibió el pago | *Predeterminado:* |
| **Monto** float  Public    «column» | Monto que se cubre (remanente) del pedido | *Predeterminado:* |

Tabla 4. PartidasManuales

# Operaciones

| **Metodo** | **Notas** | **Parámetros** |
| --- | --- | --- |
| **PK\_PartidasManuales()**  Public |  | **tinyint** [in]CorporativoConciliacion  **tinyint** [in]SucursalConciliacion  **int** [in]AñoConciliacion  **smallint** [in]MesConciliacion  **int** [in]FolioConciliacion  **varchar** [in]PedidoReferencia  **bigint** [in]Cuenta |

Tabla 5. Llave primaria de PartidasManuales

IF EXISTS (SELECT \* FROM dbo.sysobjects WHERE id = object\_id('PartidasManuales') AND OBJECTPROPERTY(id, 'IsUserTable') = 1)

DROP TABLE PartidasManuales;

CREATE TABLE PartidasManuales ( CorporativoConciliacion tinyint NOT NULL,

SucursalConciliacion tinyint NOT NULL,

AñoConciliacion int NOT NULL,

MesConciliacion smallint NOT NULL,

FolioConciliacion int NOT NULL,

PedidoReferencia varchar(20) NOT NULL,

Cuenta bigint NOT NULL,

Monto float NOT NULL);

ALTER TABLE PartidasManuales ADD CONSTRAINT PK\_PartidasManuales

PRIMARY KEY CLUSTERED (CorporativoConciliacion, SucursalConciliacion, AñoConciliacion, MesConciliacion, FolioConciliacion, PedidoReferencia, Cuenta);

## Creación de tabla relación para TransBans y conciliaciones



Fig. 9 Tabla MovimientoCajaConciliacion y catálogo de status

El script de generación de las tablas MovimientoCajaConciliacion y StatusMovimientoCajaConciliacion es el siguiente:

IF EXISTS (SELECT \* FROM dbo.sysobjects WHERE id = object\_id('MovimientoCajaConciliacion') AND OBJECTPROPERTY(id, 'IsUserTable') = 1)

DROP TABLE MovimientoCajaConciliacion;

CREATE TABLE MovimientoCajaConciliacion ( Caja tinyint NOT NULL,

FOperacion datetime NOT NULL,

Consecutivo tinyint NOT NULL,

Folio int NOT NULL,

CorporativoConciliacion tinyint NOT NULL,

SucursalConciliacion tinyint NOT NULL,

AñoConciliacion int NOT NULL,

MesConciliacion smallint NOT NULL,

FolioConciliacion int NOT NULL,

Status tinyint NOT NULL);

ALTER TABLE MovimientoCajaConciliacion ADD CONSTRAINT PK\_MovimientoCajaConciliacion

PRIMARY KEY CLUSTERED (Caja, FOperacion , Consecutivo, Folio, CorporativoConciliacion, SucursalConciliacion, AñoConciliacion, MesConciliacion, FolioConciliacion);

IF EXISTS (SELECT \* FROM dbo.sysobjects WHERE id = object\_id('StatusMovimientoCajaConciliacion') AND OBJECTPROPERTY(id, 'IsUserTable') = 1)

DROP TABLE StatusMovimientoCajaConciliacion;

CREATE TABLE StatusMovimientoCajaConciliacion ( Status tinyint NOT NULL,

Descripcion varchar(50) NOT NULL);

ALTER TABLE StatusMovimientoCajaConciliacion ADD CONSTRAINT PK\_StatusMovimientoCajaConciliacion

PRIMARY KEY CLUSTERED (Status);

## Modificación del proceso de conciliación

Una vez que el usuario ha conciliado de forma automática y aún existan movimientos que no fueron relacionados se ejecutará el proceso de conciliación manual, para que esto ocurra es necesario que el archivo Excel de conciliación manual haya sido cargado al sistema, por lo anterior se requiere una verificación en la que se consulte a la tabla MovimientoCajaConciliacion así como su correspondiente status para que todo documento que no se encuentre en ella pueda ser procesado conforme el flujo de conciliación y en caso contrario el documento no sea procesado (insertado en MovimientoCajaConciliacion).

El proceso debe ser capaz de considerar clientes padre e hijo (o sucursales) así como movimientos de tipo pedido o factura manual.

## Transaccionalidad del proceso

Durante el flujo de aplicación de pagos, intervienen cinco tablas clave de SIGAMET, estas se enlistan a continuación:

1. Cobro
2. CobroPedido
3. MovimientoCajaCobro
4. Movimiento Caja
5. Pedido

Es fundamental considerar que durante la aplicación del pago es posible que el proceso falle y de lugar a inconsistencias de datos, por lo anterior es fundamental



## Consideraciones y Dependencias

* Configuración de ambiente de desarrollo
* Configuración de ambiente de aseguramiento de la calidad (QA)
* Configuración de ambiente de UAT
* Actualización de base de datos hasta la versión más reciente proporcionada por Transforma
* Pendientes los casos de prueba y criterios de aceptación

# Plan de Actividades

|  |  |  |
| --- | --- | --- |
| Actividad | Duración | Nombre de recurso |
| **Tiempo total** | **días** |  |
| **Documentación** | **días** |  |
| MTI-III-ETP (Especificación Técnicas de Programa) | 24 hrs | Iván Trujillo |
| MTI-III-ETS (Especificación Técnica de la Solución) | 24 hrs | Iván Trujillo |
| Configuración de ambiente de desarrollo |  |  |
| Instalación de sistema operativo | 2 hrs | Ricardo Rojas |
| Instalación de SQL Server | 2 hrs | Ricardo Rojas |
| Proporcionar respaldos | 2 hrs | Iván Trujillo |
| Restaurar bases de datos Sigamet y SigametFinanciero | 4 hrs | Ricardo Rojas |
| Ejecución de script de actualización | 0 hrs | Ricardo Rojas |
| Configurar servidor vinculado | .5 hrs | Ricardo Rojas |
| Instalar .net framework Ver. 4 | .5 hrs | Ricardo Rojas |
| Instalar *runtime* Crystal Reports | .5 hrs | Ricardo Rojas |
| **Configuración de ambiente de UAT** |  |  |
| Clonado de máquina virtual | 2 hrs | Rubén Morera |
| Instalación de sistema operativo | 2 hrs | Rubén Morera |
| Instalación de SQL Server | 2 hrs | Rubén Morera |
| Proporcionar respaldos | 2 hrs | Rubén Morera |
| Restaurar bases de datos Sigamet y SigametFinanciero | 4 hrs | Rubén Morera |
| Ejecución de script de actualización | 0 hrs | Rubén Morera |
| Configurar servidor vinculado | .5 hrs | Rubén Morera |
| Instalar .net framework Ver. 4 | .5 hrs | Rubén Morera |
| Instalar *runtime* Crystal Reports | .5 hrs | Rubén Morera |
| **Desarrollo** |  |  |
| **Creación de parámetro** |  |  |
| Ejecución de script para creación de parámetro y verificación de recuperación en .net | .75 hrs | Rubén Morera |
| **Ventana emergente de verificación carga de conciliación manual** |  |  |
| Búsqueda de documentos referencia por cliente padre e hijo | 3 hrs | Ricardo Rojas |
| Filtrado por factura | 1 hrs | Ricardo Rojas |
| Creación de formulario (layout) e integración con formulario principal | 16 hrs | Rubén Morera |
| Creación de clase de validación de carga | 16 hrs | Jorge |
| Función de carga de archivo de documentos | 4 hrs | Rubén Morera |
| Rutina de conciliación (vinculación del documento referencia) | 24 hrs | Transforma |
| Integración del entregable | 1 hrs | Rubén Morera |
| Integración del entregable | 1 hrs | Ricardo Rojas |
| Prueba de integración | 1 hrs | Iván Trujillo |
| **6.1.3 Conciliación varios a uno** |  |  |
| Implemantar relación por referencia a un pedido (por cliente) Contrato + Dígito verificador |  |  |
| Filtro de fecha de suministro |  |  |
| Agregar al grid de Internos, Serie, Folio factura y número de cliente |  |  |
| Agregar búsqueda de pedido por folio factura |  |  |
| Agregar búsqueda por cliente |  |  |
| Quitar la columna de Ped en Internos |  |  |
| Modificar columna Pedido referencia por Documento en Archivos internos |  |  |
| Integración del entregable | 1 hrs | Rubén Morera |
| Integración del entregable | 1 hrs | Ricardo Rojas |
| Prueba de integración | 1 hrs | Iván Trujillo |
| **6.1.4 Conciliación manual** |  |  |
| Modificación a la vista del conciliador manual |  |  |
| Implementar formulario de búsqueda de créditos |  |  |
| Implementar búsqueda del cliente |  |  |
| Implementar búsqueda caso no referenciado |  |  |
| Implementar búsqueda de pedidos por rango de fecha de suministro |  |  |
| Implementar filtrado por folio de factura |  |  |
| Integrar la importación de archivos (Función reutilizada) |  |  |
| Implementar formulario de búsqueda de facturas manuales |  |  |
| Creación de procedimientos almacenados |  | Transforma |
| **6.1.5 Generación automática de relaciones de cobranza** |  |  |
| Análisis y diseño |  |  |
| Eliminación de controles implementados para evitar múltiples *TRANSBAN* |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Pruebas unitarias |  |  |
| Integración del producto |  |  |
| Pruebas funcionales |  |  |
|  |  |  |
|  |  |  |
| **Pruebas integrales del producto** |  |  |
| **Integración del producto** |  |  |
| **Liberación del producto** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Anexos

## Tabla de Referencia de Posibles Entregables

|  |  |  |
| --- | --- | --- |
| **X** | **ID** | **Nombre** |
|  | 1 | SolucionConciliacion (paquete de liberación) |
|  | 2 | Manual de usuario |
|  | 3 | Manual técnico |
|  | 4 |  |

## Anexo 1

No aplica