**solicitud DE NUEVO REQUERIMIENTO**

Este documento tiene por objetivo registrar la solicitud de un nuevo requerimiento.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **De:** | *Julio César Ramírez Flores* | | | |
| **Para:** | *Jocelyn Hernandez* | | | |
|  |  |  | |  |
| **Folio del proyecto:** |  | | | |
| **Nombre del proyecto:** | *Plan Piso* | | | |
| **Sistema:** | *Business Pro* | | | |
| **Módulo:** | *Centralización – Plan Piso* | | | |
| **Fecha inicio:** | *14/10/2017* | **Fecha fin:** | *31/10/2017* | |

# **Objetivo**

Generar proceso que contabilice diferentes transacciones alojadas en tablas, Grupo Andrade dejara registros para que Business Pro las procese y genere la póliza contable.

# **Beneficios**

Tener en Business Pro el registro contable de todos los movimientos realizados en modulo por grupo Andrade.

# **Antecedentes o necesidades del Requerimiento**

Al ocupar un sistema externo para el cálculo de intereses en plan piso de unidades nuevas, el usuario requiere que todas las afectaciones a nivel contable se vean reflejadas en Business Pro, por tal motivo será necesario crear tablas que procesen los diferentes escenarios.

# **Proceso(s) o módulo(s) que serán afectados**

Las formas que se afectan son:

* CON\_POLIZAS
* CXC\_PedVarios

Los procesos que se afectan son:

* Pólizas y cheques.
* Orden de compra (Compras varias).

# **Descripción general del Requerimiento**

Se deberá crear un parámetro que habilite la funcionalidad de procesar las diferentes pólizas para el proceso de plan piso. Este debe ser dado de alta de la siguiente manera:

**TIPO:** POLPLPISO

**IDENTIFICADOR:** 1

**MODULO:** Contabilidad

**ESTATUS:** Activo

**DESCRIPCION1:** Proceso para generar la afectación contable por los diferentes tipos de transacción.

INSERT INTO PNC\_PARAMETR VALUES ('POLPLPISO','1','CON','Proceso para generar la afectación contable por los diferentes tipos de transacción','','','','','A','0','0','0','0','0','','','','','','','GMI',CONVERT (VARCHAR(10),GETDATE(),103),'12:00')

En caso de no existir, el sistema debe funcionar como lo hace hasta el momento, sin la necesidad de ir a buscar registros en las tablas creadas.

De la misma forma se van a crear las siguientes tablas:

* plp\_planpisoenc.
* plp\_planpisodet.
* cat\_tipoproceso.
* plp\_relordencompra.
* plp\_logplanpiso.
* plp\_bitacora

**PLP\_PLANPISOENC**

En la siguiente tabla, Grupo Andrade especifica el tipo de transacción que se va a procesar, así como la empresa y sucursal donde se está solicitando el proceso, estos valores deben coincidir con los correspondientes en las tablas [cat\_empresas] y [cat\_sucursales]. El tipo de póliza debe coincidir con el valor que le corresponde a la BDD del parámetro TIPOLI.

Es importante señalar que al procesar un registro exitosamente se deberá actualizar a 1 el campo ple\_estatus, así como también el **consecutivo de la póliza**, el **mes** y el **año** en que se generó (ple\_conspol, ple\_mes y ple\_año).

|  |  |  |
| --- | --- | --- |
| **plp\_planpisoenc** | | |
| **Campo** | **Tipo** | **comentarios** |
| ple\_idplanpiso | numeric identity | Identificador de la tabla de plan piso encabezado |
| pro\_idtipoproceso | Integer | Identificador de la tabla de catálogo de tipos de proceso [cat\_tipoproceso] |
| ple\_idempresa | Integer | Identificador de la empresa donde se generará el registro contable. Se puede obtener de la tabla cat\_empresas de control de aplicaciones |
| ple\_idsucursal | Integer | Identificador de la sucursal de la cual se hará el registro contable. Se puede obtener de la tabla cat\_sucursales de la base de control de aplicaciones, debe tener la liga a la empresa |
| ple\_tipopoliza | Varchar | Tipo de póliza a generar, depende del tipo de proceso y de la sucursal y empresa en la que se generará el registro. Es el identificador del parámetro TIPOLI de la sucursal. |
| ple\_fechapoliza | Datetime | Fecha con la que se debe generar la póliza |
| ple\_concepto | Varchar | Concepto de la póliza |
| ple\_fechageneración | Datatime | Fecha de generación real de la póliza, debe insertarse después de ejecutar el proceso |
| ple\_conspol | Integer | Consecutivo de póliza que se generó en BPro. |
| ple\_mes | Integer | Mes en el que se generó la póliza contable en BPro. |
| ple\_año | Varchar | Año en la que se generó la póliza contable en BPro. |
| ple\_estatus | Integer | Estatus del registro 0 - pro procesar 1 - procesado |

Script para creación de tabla

CREATE TABLE plp\_planpisoenc

(

ple\_idplanpiso numeric(18) IDENTITY ( 1,1 ) ,

pro\_idtipoproceso integer NOT NULL,

ple\_idempresa integer NOT NULL ,

ple\_idsucursal integer NOT NULL ,

ple\_tipopoliza varchar(20) NOT NULL ,

ple\_fechapoliza datetime NOT NULL ,

ple\_concepto varchar(200) NOT NULL ,

ple\_fechageneracion datetime NULL ,

ple\_conspol integer NULL ,

ple\_mes integer NULL ,

ple\_año varchar (4) NULL ,

ple\_estatus integer NOT NULL

)

go

ALTER TABLE plp\_planpisoenc

ADD CONSTRAINT XPKplp\_planpisoenc PRIMARY KEY (ple\_idplanpiso ASC)

go

Alter table [plp\_planpisoenc] add opp\_idordencompra [numeric](18, 0) NULL

**PLP\_PLANPISODET**

En la siguiente tabla Grupo Andrade insertara el detalle de la póliza. El campo **ple\_idplanpiso** hará referencia al movimiento generado en la tabla anterior. Esto quiere decir que el detalle del movimiento corresponde con el encabezado del mismo.

Se deberán considerar ciertas validaciones que inclusive procesos similares ya realizan tales como:

* Que la cuenta contable exista en el catálogo y sea de detalle.
* Que si la cuenta es de cartera (parámetro cartera – descripción 5) es obligatorio el campo de persona, documento, tipo de documento y VIN.
* El tipo de documento (parámetro TIMO) debe existir en la BDD donde se está procesando la póliza, el sistema ya sabe trabajar con la regla dependiendo el valor que contenga en el importe1
  + Creación del documento original importe1 = 1
  + Afectación a documento original importe1 = 0
* Que la suma de cargos y abonos por póliza sea el mismo importe.

|  |  |  |
| --- | --- | --- |
| **plp\_planpisodet** | | |
| **Campo** | **tipo** | **Comentarios** |
| pld\_idplanpisodet | numeric identity | Identificador de la tabla de plan piso detalle |
| ple\_idplanpiso | numeric | Identificador de la tabla de plan piso encabezado |
| pld\_consecutivo | integer | Consecutivo de la póliza a generar |
| pld\_numcuenta | varchar | Número de cuenta a afectar |
| pld\_concepto | varchar | Concepto del movimiento |
| pld\_cargo | decimal (18,7) | Cargo del movimiento |
| pld\_abono | decimal (18,7) | Abono del movimiento |
| pld\_idpersona | numeric | Identificador de la persona a la que se hará el movimiento, en caso de no haber persona debe ponerse por default 0 (solo aplica para cuentas que son cartera) |
| pld\_iddocumento | varchar | Documento que se está creando o afectando, solo aplica para registros con cuentas de cartera |
| pld\_tipodocumento | varchar | Tipo de documento que se realizará. FAC, FACIVA, NCR, NCRIVA, entre otros. Continuando con la regla de lo que viene en el importe 1. |
| pld\_vin | varchar | Número de serie de la unidad mismo que se deberá almacenar en **ccp\_obsgen** para los diferentes reportes de centralización. |
| pld\_fechavencimiento | datetime | Fecha de vencimiento del documento |
| pld\_porcentajeiva | integer | Porcentaje de IVA aplicado 16 o 0 |
| pld\_afecta | varchar | Si afecta = "A" de lo contrario vacío |

Script para creación de tabla

CREATE TABLE plp\_planpisodet

(

pld\_idplanpisodet numeric(18) IDENTITY ( 1,1 ),

ple\_idplanpiso numeric(18) NOT NULL

pld\_consecutivo integer NOT NULL ,

pld\_numcuenta varchar(50) NOT NULL ,

pld\_concepto varchar(200) NOT NULL ,

pld\_cargo decimal(18,7) NOT NULL ,

pld\_abono decimal(18,7) NOT NULL ,

pld\_idpersona numeric(18) NULL ,

pld\_iddocumento varchar(20) NULL ,

pld\_tipodocumento varchar(20) NULL ,

pld\_vin varchar(17) NULL ,

pld\_fechavencimiento datetime NULL ,

pld\_porcentajeiva integer NOT NULL ,

pld\_afecta varchar(1) NULL

)

go

ALTER TABLE plp\_planpisodet

ADD CONSTRAINT XPKplp\_planpisodet PRIMARY KEY (pld\_idplanpisodet ASC)

go

El proceso para almacenar los registros contables en las tablas:

* CON\_POL + GlbEmpAct + GlbAnnAct
* CON\_MOVDET + GlbEmpAct + GlbAnnAct
* CON\_CAR + GlbEmpAct + GlbAnnAct

Deberá ser similar al procedimiento **PROCESO\_AUTOMATICO** que ya se tiene programado, considerando los campos adicionales exclusivos de centralización.

El campo pld\_vin se deberá actualizar a nivel de cartera es decir, tabla CON\_CAR + GlbEmpAct + GlbAnnAct campo CCP\_OBSGEN. Con el objetivo de que sus reportes tales como antigüedad de saldos, consulta de movimientos y neteos versión centralizada, muestre a nivel de cartera la referencia del vehículo.

**CAT\_TIPOPROCESO**

En esta tabla se estarán definiendo los diferentes tipos de transacciones. Por lo que el campo **pro\_idtipoproceso** identificara el proceso definido para cierta operación. En esta tabla se podrán dar de alta “n” procesos siempre y cuando se tenga la regla de negocio.

|  |  |  |
| --- | --- | --- |
| **cat\_tipoproceso** | | |
| **Campo** | **tipo** | **comentarios** |
| pro\_idtipoproceso | integer identity | Identificador de la tabla de catálogo de tipos de proceso |
| pro\_nombre | varchar | Nombre del proceso: 0 - No aplica, 1 - Provisión, 2 - Cancelación provisión, 3 - Provisión fin de mes, 4 - Cancelación Provisión fin de mes, 5- Conciliación, 6 - Cancelación de la conciliación, 7 - Aplicación, 8 - Cancelación de la Aplicación, 9 - Pago, 10 - Cancelación de pago, 11 – Compensación y 12 – Traspaso plan Piso |
| pro\_nombrecto | varchar | Nombre corto del tipo de proceso a generar: 0 - NA, 1 - PR, 2 - CPR, 3 - PFM, 4 - CPFM, 5 - CON, 6 - CCON, 7 - APL, 8 - CAPL, 9 –PAG, 10 – CPAG, 11 – COM y 12 - TPP. |
| pro\_usualta | integer | Identificador del usuario que da de alta el registro |
| pro\_fechaalta | datetime | Fecha de alta del registro |
| pro\_usumodifica | integer | Identificador del usuario que modifica el registro |
| pro\_fechamodifica | datetime | Fecha de modificación del registro |
| pro\_estatus | integer | Estatus del registro: 0 - inactivo 1 - activo |

Script para creación de tabla.

CREATE TABLE cat\_tipoproceso

(

pro\_idtipoproceso integer IDENTITY ( 1,1 ) ,

pro\_nombre varchar(100) NOT NULL ,

pro\_nombrecto varchar(20) NOT NULL ,

pro\_fechaalta datetime NOT NULL ,

pro\_usualta integer NOT NULL ,

pro\_usumodifica integer NULL ,

pro\_fechamodifica datetime NULL ,

pro\_estatus integer NOT NULL

)

go

ALTER TABLE cat\_tipoproceso

ADD CONSTRAINT XPKcat\_tipoproceso PRIMARY KEY (pro\_idtipoproceso ASC)

go

Sera importante considerar las siguientes reglas de negocio para procesar los archivos, considerando siempre la empresa y sucursal.

* Se inicia procesando registros con nombre corto **1 – PR – provisión**
* En caso de existir registros con nombre corto **2 – CPR – Cancelación provisión** estos se procesan sin ningún problema.
* Para ejecutar la transacción **3 – PFM – provisión de fin de mes** será obligatorio ejecutar antes el proceso **2 – CPR – Cancelación provisión.**
* Proceso **5 – CON** - **Conciliación** y proceso **6 – CCON - Cancelación conciliación** se procesan sin ninguna restricción. **\*\*\*\*\*\*descartado por completo el proceso 5 y 6 \*\*\*\*\***
* El proceso **7 – APL - Aplicación** determina que entre a ejecutarse el **proceso 4 – CPFM – Cancelación Provisión fin de mes**. Los pasos para la ejecución de la aplicación los encontramos en el siguiente apartado de **orden de compra.**
* El proceso **8 – CAPL** se ejecutara siempre y cuando exista una aplicación (**7 –APL**).
* El proceso **9 – PAG** se ejecutara considerando las siguientes reglas:
  1. En caso de existir provisiones proceso **1 – PR,** se deberá tener un proceso pendiente por ejecutar del tipo **2 - CPR** ya que por regla se deberá cancelar las provisiones antes de generar el pago
  2. En caso de que no existan provisiones no se cancela ningún proceso pero si se genera una orden de compra.
  3. En caso de existir provisiones de fin de mes proceso **3 – PFM,** se deberá tener un proceso pendiente por ejecutar del tipo **4 - CFM** ya que por regla se deberá cancelar las provisiones antes de generar el pago
  4. En caso de que no existan provisiones de fin de mes no se cancela ningún proceso pero si se genera una orden de compra.
* El proceso **10 – CPAG.** Se ejecutara siempre y cuando exista un proceso **9 – PAG** con su respectiva orden de compra.
* El proceso **11 – COM** se ejecutara siempre y cuando exista.
* El proceso **12 – TPP** se ejecutara siempre y cuando exista.

**PLP\_RELORDENCOMPRA**

La orden de compra se estará generando para los procesos:

* **7 – Aplicación – APL**.
* **9 – Pago – PAG.**
* **12 – Traspaso entre Financiera – TPP.**

Pero solo se tomara lectura de la tabla plp\_relordencompra para los procesos:

* **7 – Aplicación – APL**.
* **9 – Pago – PAG.**

Para tema de cancelaciones solo se verán afectados a nivel de estatus los procesos de:

* **8 – Cancelación aplicación – CAPL**
* **10 – Cancelación de Pago - CPAG**.

|  |  |  |
| --- | --- | --- |
| **plp\_relordencompra** | | |
| **Campo** | **tipo** | **comentarios** |
| opp\_idordencompra | numeric identity | Identificador de la tabla de conceptos para armar orden de compra de pedidos varios. |
| opp\_empresa | integer | **I**dentificador de la empresa. |
| opp\_sucursal | integer | Identificador de la sucursal. |
| opp\_area | varchar | Área de afectación, se obtiene del parámetro **AREPED** de la sucursal en la que se hará la orden. |
| opp\_concepto | varchar | Concepto que hace referencia al identificador del parámetro **CONVEN**. |

Script para la creación de la tabla

CREATE TABLE plp\_relordencompra

(

opp\_idordencompra numeric(18) IDENTITY ( 1,1 ) ,

opp\_empresa integer NOT NULL ,

opp\_sucursal integer NOT NULL ,

opp\_area varchar(20) NOT NULL ,

opp\_concepto varchar(20) NOT NULL

)

go

ALTER TABLE plp\_relordencompra

ADD CONSTRAINT XPKplp\_relordencompra PRIMARY KEY (opp\_idordencompra ASC)

go

La orden de compra que a continuación se especifica, solo se estará generando para los procesos:

* **7 – Aplicación – APL**.
* **9 – Pago – PAG.**

Para este proceso será importante considerar la relación de las tablas: [plp\_planpisoenc] y [plp\_relordencompra] por los campos que identifican a la empresa y sucursal.

|  |  |
| --- | --- |
| plp\_planpisoenc | plp\_relordencompra |
| ple\_idempresa | opp\_empresa |
| ple\_idsucursal | opp\_sucursal |

Como se desea generar una orden de compra por unidad, será necesario generar un registro en la tabla [CON\_PEDOTROS], por tal motivo se agregara una columna para no perder el origen del registro (póliza origen).

ALTER TABLE CON\_PEDOTROS add PLE\_IDPLANPISO numeric(18) NULL

Para generar el registro de la provisión será necesario insertar como a continuación se especifica:

* + poc\_idpedido: consecutivo.
  + poc\_area: se tomara del campo *opp\_area* de la tabla [plp\_relordencompra], considerando que se trate de la misma empresa y sucursal (opp\_empresa y opp\_sucursal)
  + poc\_solarea: default GMI.
  + poc\_autoarea: se tendrá que ir a validar a la tabla DIG\_ESCALAMIENTO\_AREA\_AFECT por la sucursal del usuario que está realizando dicho proceso y el valor que viene en *opp\_area* para actualizar el valor con base al campo usuario\_autoriza1.

Es importante mencionar que el campo marcado en color azul solo se debe actualizar cuando la orden de compra y la póliza son generadas correctamente.

* + poc\_factura: vacío.
  + poc\_status: default **SOL** (solicitada).
  + poc\_fecfactura: vacío
  + poc\_fecorden: fecha del servidor.
  + poc\_subtotal: El importe se tomara de la tabla [plp\_planpisodet] considerando las siguientes reglas **y POR CADA VIN** almacenado en el campo [pld\_vin].
    - Se deberá considerar el registro en donde el campo [pld\_tipodocumento] sea siempre **FAC**, además que el campo [pld\_abono] sea diferente de cero y sea una cuenta de cartera (validar buscando esa cuenta en la descripción 5 del parámetro cartera y en donde el importe5 = 0).
    - Considerando que **en el campo [pld\_porcentajeiva] contiene un valor = 16**. Se deberá dividir dicho importe entre 1.16 y tomar ese cálculo para ponerlo como el subtotal correcto.
    - **Caso contrario** en donde el campo [pld\_porcentajeiva] sea igual a cero, el importe del campo [pld\_abono] deberá ser tomado tal cual.
  + poc\_iva: para el importe del IVA será importante considerar que el campo [pld\_tipodocumento] sea **FACIVA** y que en el campo [pld\_porcentajeiva] sea igual a 16 de lo contrario el valor almacenar es CERO**.** En caso de si existir registró con tipo docto FACIVA y porcentaje de IVA del 16, se deberá validar que la cuenta sea de impuestos. Para determinar si es una cuenta de impuestos podemos buscarla en la descrip5 del parámetro CARTERA en donde el importe5 siempre es igual a 1. El importe a tomar deberá de leerse del campo [pld\_cargo].
  + poc\_descuento : default 0
  + poc\_total: se deberá sumar el subtotal + el IVA
  + poc\_proveedor: Se tomara de la tabla [plp\_planpisodet] campo pld\_idpersona.
  + poc\_poriva: se tendrá que considerar por unidad – [pld\_vin] el valor que contiene el campo [pld\_porcentajeiva].
  + poc\_observaciones: vacío.
  + poc\_cvusuario: default GMI.
  + poc\_fechope: fecha servidor.
  + poc\_feccancela : vacío.
  + poc\_idcancela: vacío.
  + POC\_IDFACTUR: vacío.
  + POC\_FECFACTUR: deberá tomarse de la tabla [plp\_planpisoenc] del campo [ple\_fechapoliza] .
  + poc\_provision: NULL.
  + POC\_CARTERA: NULL.
  + poc\_ieps: default **0.**
  + poc\_comprobante: (select top 1 par\_idenpara from pnc\_parametr where par\_tipopara = ‘TCPEDVAR’ AND PAR\_DESCRIP1 LIKE ‘%FACTURA%’)
  + POC\_NUMPOLIZA: NULL.

Es importante mencionar que los campos marcados en color azul solo se deben actualizar cuando la orden de compra y la póliza son generadas correctamente.

* + POC\_EMPRESA: vacío.
  + POC\_ANNO: vacío.
  + POC\_TIPO: vacío.
  + POC\_CONSECUTIVO: NULL.
  + POC\_MES: NULL.
  + POC\_TPOCAMB: default 1
  + POC\_IDMONEDA: default PE
  + POC\_CONTRARECIBO: NULL.
  + NombrePDF: vacío.
  + nombreXML: vacío.
  + rutapdf RutaXML: vacío.
  + uuid: vacío.
  + RFC: vacío.
  + poc\_idpedidopadre: default 0.
  + poc\_folioorden: vacío.
  + poc\_folioordenpadre: vacío.
  + poc\_anticipo: default 0.
  + poc\_cantidadanticipo: default 0.
  + poc\_porcentajeanticipo: vacío.
  + poc\_fechapagoanticipo: NULL.
  + uaw\_idanticipoweb: NULL.
  + ple\_idplanpiso: Consecutivo que hace referencia al movimiento de la tabla [plp\_planpisoenc]
* Para la parte del detalle, se deberá actualizar la tabla [UNI\_PEDOTROSDET] con los siguientes criterios:
  + pod\_idpedido: consecutivo del encabezado.
  + pod\_area: se tomara del campo *opp\_area* de la tabla [plp\_relordencompra].
  + pod\_cantidad: default 1.
  + pod\_producto: siguiendo la misma regla en donde se estarán generando **órdenes de compra por unidad**. En este campo tomaremos lo que se encuentra en [pld\_concepto] de la tabla [plp\_planpisodet].
  + pod\_preuni: Para el precio unitario estaremos manejando la misma regla que se especificó en el encabezado (subtotal).
  + pod\_descuento: default 0.
  + pod\_iva: para el IVA se estará manejando la misma regla especificada en el encabezado (IVA).
  + pod\_total: misma regla que en el encabezado (TOTAL).
  + pod\_concepto: Se tendrá que ir a validar al campo [opp\_concepto] de la tabla [plp\_relordencompra], considerando que se trate de la misma empresa y sucursal (opp\_empresa y opp\_sucursal)
  + pod\_cvuusu: por default GMI.
  + pod\_fechope: fecha servidor.
  + pod\_poriva: se tendrá que considerar por unidad – [pld\_vin] el valor que contiene el campo [pld\_porcentajeiva].
  + pod\_retiva: 0
  + pod\_retisr: 0
  + POD\_IDTERCERO: 0
  + POD\_IDDOCTO: vacío.
  + pod\_Consecutivo: consecutivo detalle.
  + POD\_REFERENCIA: NULL.
  + pod\_refotrosgtos: NULL.
  + pod\_cinv: NULL.
  + pod\_folioorden: vacío.
  + pod\_cantidadrecibida: 0
  + pod\_cantidaddiferencia: 0
  + pod\_cveusurecep: 0
  + pod\_preuniinicial: 0
  + pod\_totalinicial: 0
  + pod\_idsituacionordendet: 0
  + pod\_saldocuentames: 0
  + pod\_presupuestomes: 0
  + pod\_excedentemes: 0
  + pod\_saldocuentaanio: 0
  + pod\_presupuestoanio: 0
  + pod\_excedenteanio: 0

Para poder continuar con el proceso de generar la orden de compra será necesario generar un registro en la tabla, siguiendo con la misma regla de generarlo por los diferentes números de serie [pld\_vin] que se encuentren a la póliza a procesar:

* [CXP\_ORDENCOMPRA]
  + oce\_folioorden: Se deberá continuar con las mismas reglas para el armado del folio considerando que el departamento por nombre corto siempre será **OT** y el tipo de orden siempre será pedido especial, nombre corto **PIPP.**

Ejemplo: AU-AU-CUA-OT-PIPP-1

* + oce\_iddivision: mismas reglas para generar la división.
  + oce\_idempresa: mismas reglas para generar la empresa.
  + oce\_idsucursal: mismas reglas para generar la sucursal.
  + oce\_iddepartamento: mismas reglas para generar departamento (nombre corto OT dependiendo la sucursal).
  + oce\_idtipoorden: se crea nuevo tipo de orden con nombre corto **PIPP**

INSERT INTO [cuentasxpagar].[dbo].[cat\_tiposorden]

([tip\_nombre]

,[tip\_nombrecorto]

,[tip\_status]

,[tip\_fechaalta]

,[tip\_idusuarioalta]

,[tip\_fechamodifica]

,[tip\_idusuariomodifica])

VALUES

('PAGO DE INTERESES PLAN PISO','PIPP',1,GETDATE() ,1,GETDATE() ,1)

GO

* + oce\_numcotizacion: en blanco
  + oce\_numeroflotilla: default 0.
  + oce\_imposicionplanta: default 0.
  + oce\_fechaorden: fecha servidor.
  + oce\_horaorden: hora servidor.
  + oce\_importetotal: Deberá ser el mismo importe que se está ingresando en la tabla [CON\_PEDOTROS] campo [poc\_total].
  + oce\_idusuario: mismas reglas para generar usuario.
  + oce\_motivocancelacion: vacío.
  + oce\_motivorechazo: vacío.
  + sod\_idsituacionorden: se genera nueva situación con nombre corto **ATTP**

INSERT INTO [cuentasxpagar].[dbo].[cat\_situacionorden]

([sod\_nombresituacion]

,[sod\_nombrecto]

,[sod\_statussituacion]

,[sod\_fechaalta]

,[sod\_idusuarioalta]

,[sod\_fechamodifica]

,[sod\_idusuariomodifica])

VALUES

('Autorizado por Traspaso de Plan Piso','ATTP',1,GETDATE() , 1,GETDATE(),1)

* + oce\_anticipo: default 0.
  + oce\_cantidadanticipo: default 0.
  + oce\_porcentajeanticipo: default 0.
  + oce\_idusuariomovimiento: mismas reglas para generar usuario.
  + oce\_idproveedor: Se tomara de la tabla [plp\_planpisodet] campo pld\_idpersona dependiendo el VIN que se esté procesando.
  + oce\_uuid: vacío.
  + oce\_factura: capturar el mismo valor que se encuentra en oce\_folioorden - considerando que el departamento por nombre corto siempre será **OT** y el tipo de orden siempre será pedido especial, nombre cortó **PIPP.**

Ejemplo: AU-AU-CUA-OT-PIPP-1

* + oce\_consefolio: mismas reglas para generar consecutivo. Prácticamente siempre será registro y orden de compra por póliza.
  + oce\_foliopadre: vacío.
  + oce\_imptotalrecibido: Deberá ser el mismo importe almacenado en este mismo evento campo [oce\_importetotal].
  + oce\_antaplicado: NULL.
  + oce\_numerolote: NULL.
  + oce\_txtgenerado: NULL.
  + oce\_idtipomoneda: default **PESOS**.
  + oce\_fechafactura: fecha servidor.
  + oce\_fechapagounidad: misma regla para generar registro.
  + oce\_numautorizaplanta: NULL.
  + oce\_fechapagoanticipo : misma regla.
  + oce\_polanticipo: NULL.

De la misma forma se deberá generar registro en la tabla [CXP\_DETALLEOTROSCONCEPTOS]:

* otc\_iddetalle: consecutivo.
* otc\_area: se tomara del campo *opp\_area* de la tabla [plp\_relordencompra].
* otc\_cantidad: default 1.
* otc\_cantidadrecibida: default 1.
* otc\_cveusurecep: deberá funcionar de la misma forma.
* otc\_producto: tomar el mismo contenido que se dejó al momento de crear la compra varia en la tabla [UNI\_PEDOTROSDET] campo [pod\_producto]
* otc\_preuni: se deberá tomar de la tabla [UNI\_PEDOTROSDET] el valor almacenado en el campo [pod\_preuni]
* otc\_preuniinicial: repetir el valor del campo anterior.
* otc\_descuento: default 0.
* otc\_iva: se deberá tomar de la tabla [UNI\_PEDOTROSDET] el valor almacenado en el campo [pod\_iva]
* otc\_total: se deberá tomar de la tabla [UNI\_PEDOTROSDET] el valor almacenado en el campo [pod\_total]
* otc\_totalinicial: se deberá tomar de la tabla [UNI\_PEDOTROSDET] el valor almacenado en el campo [pod\_total]
* otc\_concepto: colocar descripción 1 del parámetro CONVEN, mismo que se encuentra en el campo [pod\_concepto] de la tabla [UNI\_PEDOTROSDET].
* oce\_folioorden: deberá funcionar de la misma forma.
* otc\_cveusu: deberá funcionar de la misma forma.
* otc\_fechope: fecha servidor.
* otc\_poriva: se deberá tomar de la tabla [UNI\_PEDOTROSDET] el valor almacenado en el campo [pod\_poriva]
* otc\_retiva: default 0.
* otc\_retisr: default 0.
* otc\_iddocto: vacío.
* otc\_idsituacionordendet: se genera nueva situación con nombre corto **ATTP** (en la tabla anterior se menciona como debe ir).
* otc\_fechapromentrega: deberá funcionar de la misma forma.
* otc\_horapromentrega: deberá funcionar de la misma forma.
* otc\_cantidaddiferencia: default 0.
* otc\_consecutivo: consecutivo, por lo general será 1 ya que se genera un registro
* otc\_saldocuentames: deberá funcionar de la misma forma.
* otc\_presupuestomes: deberá funcionar de la misma forma.
* otc\_excedentemes: deberá funcionar de la misma forma.
* otc\_saldocuentaanio: deberá funcionar de la misma forma.
* otc\_presupuestoanio: deberá funcionar de la misma forma.
* otc\_excedenteanio: deberá funcionar de la misma forma.
* otc\_comentarios: vacío.
* otc\_nopedido: Se deberá colocar el número del pedido generado en [CON\_PEDOTROS].

Al involucrar una orden de compra, será necesario guardar registro en la tabla [CXP\_MOVIMIENTOSORDEN].

* mov\_idmovimiento: consecutivo.
* mov\_idusuariomovimiento: deberá funcionar de la misma forma.
* mov\_fechamovimiento: deberá funcionar de la misma forma.
* mov\_horamovimiento: deberá funcionar de la misma forma.
* oce\_folioorden: folio generado al **Aplicar la conciliación.**
* sod\_idsituacionorden: se genera nueva situación con nombre corto **ATTP** (en eventos anteriores se menciona como debe ir).

Puntos a considerar en la creación exitosa de la orden de compra:

* La afectación a nivel de cartera (CON\_CAR + GlbEmpAct + GlbAnnAct) debe almacenar en el campo **ccp\_fechprompag** el siguiente día natural a la creación.
* Actualizar en la tabla [CON\_PEDOTROS] el campo [poc\_status]: default **FIN** (finalizada), POC\_EMPRESA, POC\_ANNO, POC\_TIPO, POC\_CONSECUTIVO y POC\_MES.
* Actualizar de la tabla [CXP\_ORDENCOMPRA] el campo [sod\_idsituacionorden] a situación con nombre corto **RCO**.
* Actualizar de la tabla [CXP\_DETALLEOTROSCONCEPTOS] el campo [otc\_idsituacionordendet] a situación con nombre corto **RCO**.
* Generar nuevo registro en la tabla [CXP\_MOVIMIENTOSORDEN]:
  + mov\_idmovimiento: consecutivo.
  + mov\_idusuariomovimiento: deberá funcionar de la misma forma.
  + mov\_fechamovimiento: deberá funcionar de la misma forma.
  + mov\_horamovimiento: deberá funcionar de la misma forma.
  + oce\_folioorden: folio generado al aplicar conciliación.
  + sod\_idsituacionorden: se genera nueva situación con nombre corto **RCO**.

Es importante mencionar que el siguiente proceso también depende del número de VIN que contenga la póliza: [plp\_planpisodet] – [pld\_vin]. Las especificaciones para generar la orden de compra del proceso **12 – TPP – Traspaso Plan Piso** son las siguientes. :

Para generar la nueva orden de compra será necesario:

* [CXP\_ORDENCOMPRA]
  + oce\_folioorden: Se deberá continuar con las mismas reglas para el armado del folio considerando que el departamento por nombre corto siempre será **UN** y el tipo de orden siempre será Traspaso Plan Piso, nombre corto **TPP.**

Ejemplo: AU-AU-CUA-**UN**-**TPP**-1

* + oce\_iddivision: mismas reglas para generar la división.
  + oce\_idempresa: mismas reglas para generar la empresa.
  + oce\_idsucursal: mismas reglas para generar la sucursal.
  + oce\_iddepartamento: mismas reglas para generar departamento.
  + oce\_idtipoorden: se ocupa el tipo de orden ya existente con nombre corto **TPP**
  + oce\_numcotizacion: en blanco.
  + oce\_numeroflotilla: default 0.
  + oce\_imposicionplanta: default 0.
  + oce\_fechaorden: fecha servidor.
  + oce\_horaorden: hora servidor.
  + oce\_importetotal: El importe se tomara de la tabla [plp\_planpisodet] considerando las siguientes reglas **y POR CADA VIN** almacenado en el campo [pld\_vin].
    - Se deberá considerar el registro en donde el campo [pld\_tipodocumento] sea siempre **FAC**, además que el campo [pld\_abono] sea diferente de cero y sea una cuenta de cartera (validar buscando esa cuenta en la descripción 5 del parámetro cartera y en donde el importe5 = 0).
  + oce\_idusuario: mismas reglas para generar usuario.
  + oce\_motivocancelacion: vacío.
  + oce\_motivorechazo: vacío.
  + sod\_idsituacionorden: se genera nueva situación con nombre corto **ATTP**

INSERT INTO [cuentasxpagar].[dbo].[cat\_situacionorden]

([sod\_nombresituacion]

,[sod\_nombrecto]

,[sod\_statussituacion]

,[sod\_fechaalta]

,[sod\_idusuarioalta]

,[sod\_fechamodifica]

,[sod\_idusuariomodifica])

VALUES

('Autorizado por Traspaso de Plan Piso','ATTP',1,GETDATE() , 1,GETDATE(),1)

* + oce\_anticipo: default 0.
  + oce\_cantidadanticipo: default 0.
  + oce\_porcentajeanticipo: default 0.
  + oce\_idusuariomovimiento: mismas reglas para generar usuario.
  + oce\_idproveedor: Se tomara de la tabla [plp\_planpisodet] campo pld\_idpersona
  + oce\_uuid: vacío.
  + oce\_factura: capturar el mismo valor que se encuentra en oce\_folioorden - considerando que el departamento por nombre corto siempre será **UN** y el tipo de orden siempre será pedido especial, nombre cortó **TPP.**

Ejemplo: AU-AU-CUA-UN-TPP-1

* + oce\_consefolio: mismas reglas para generar consecutivo. Prácticamente siempre será registro y orden de compra por póliza.
  + oce\_foliopadre: vacío.
  + oce\_imptotalrecibido: El importe se tomara de la tabla [plp\_planpisodet] considerando las siguientes reglas **y POR CADA VIN** almacenado en el campo [pld\_vin].
    - Se deberá considerar el registro en donde el campo [pld\_tipodocumento] sea siempre **FAC**, además que el campo [pld\_abono] sea diferente de cero y sea una cuenta de cartera (validar buscando esa cuenta en la descripción 5 del parámetro cartera y en donde el importe5 = 0).
  + oce\_antaplicado: NULL.
  + oce\_numerolote: NULL.
  + oce\_txtgenerado: NULL.
  + oce\_idtipomoneda: default PESOS.
  + oce\_fechafactura: fecha servidor.
  + oce\_fechapagounidad: misma regla para generar registro.
  + oce\_numautorizaplanta: NULL.
  + oce\_fechapagoanticipo : misma regla.
  + oce\_polanticipo: NULL.

Insertar en la tabla [cxp\_detalleautosnuevos]:

Apoyándonos del valor del VIN que colocaran a nivel de detalle en la tabla [plp\_planpisodet] buscaremos en la tabla [cxp\_detalleautosnuevos] campo [anu\_numeroserie] el más actual para insertar un registro idéntico pero con un folio nuevo.

Cuando la unidad no exista en la tabla [cxp\_detalleautosnuevos] se tendrá que crear el registro con ayuda del siguiente script, prácticamente la variable serie el número de serie que viene en el detalle de la póliza [plp\_planpisodet] – pld\_vin.

SELECT VEH\_CATALOGO,

VEH\_NUMSERIE,

VEH\_ANMODELO,

MARCA.PAR\_DESCRIP1,

COLEXTE.COL\_DESCRIPCION,

COLINTE.COL\_DESCRIPCION,

CVE.UNC\_CVEVEHIC,

ROUND(VEH\_IMPFACTPLAN / 1.16,2),

ROUND(VEH\_IMPFACTPLAN,2),

TIPENT.PAR\_DESCRIP1,

1,

FINAN.PAR\_DESCRIP1,

0,

convert(varchar, getdate(), 111),

convert(varchar, getdate(), 114),

'' as comentarios,

VEH\_CATALOGO,

'FOLIO DEL TIPO TPP',

'GASOLINA',

'TRANSMISION',

DIF.PAR\_DESCRIP1,

VEH\_ORGUNIDAD,

ORIGEN.PAR\_DESCRIP1,

VEH\_TIPENTRADA,

VEH\_TRASIMP

FROM SER\_VEHICULO

INNER JOIN PNC\_PARAMETR AS MARCA ON MARCA.PAR\_IDENPARA = VEH\_SMARCA AND MARCA.PAR\_TIPOPARA = 'MCAV'

INNER JOIN UNI\_CATACOLOR AS COLEXTE ON COLEXTE.COL\_CATALOGO = VEH\_CATALOGO AND COLEXTE.COL\_MODELO = VEH\_ANMODELO AND COLEXTE.COL\_TIPO = 'EXTERIOR' AND COLEXTE.COL\_CLAVE = VEH\_COLOEXTE

INNER JOIN UNI\_CATACOLOR AS COLINTE ON COLINTE.COL\_CATALOGO = VEH\_CATALOGO AND COLINTE.COL\_MODELO = VEH\_ANMODELO AND COLINTE.COL\_TIPO = 'INTERIOR' AND COLINTE.COL\_CLAVE = VEH\_COLOINTE

INNER JOIN UNI\_CATALOGO AS CVE ON CVE.UNC\_IDCATALOGO = VEH\_CATALOGO AND CVE.UNC\_MODELO = VEH\_ANMODELO

INNER JOIN PNC\_PARAMETR AS TIPENT ON TIPENT.PAR\_IDENPARA = VEH\_TIPENTRADA AND TIPENT.PAR\_TIPOPARA = 'TIPENT'

INNER JOIN PNC\_PARAMETR AS FINAN ON FINAN.PAR\_IDENPARA = VEH\_FINANCIERA AND FINAN.PAR\_TIPOPARA = 'FINAN'

INNER JOIN PNC\_PARAMETR AS DIF ON DIF.PAR\_IDENPARA = VEH\_TIPFINANC AND DIF.PAR\_TIPOPARA = 'TIPFIN'

INNER JOIN PNC\_PARAMETR AS ORIGEN ON ORIGEN.PAR\_IDENPARA = VEH\_ORGUNIDAD AND ORIGEN.PAR\_TIPOPARA = 'OR'

WHERE VEH\_NUMSERIE = 'TSMYD21S6HM335448' --- AQUÍ SE DEBERA DEFINIR EL NÚMERO DE SERIE

Insertar en la tabla [cxp\_movimientosorden]:

* mov\_idmovimiento = consecutivo
* mov\_idusuariomovimiento = usuario logeado
* mov\_fechamovimiento = fecha movimiento
* mov\_horamovimiento = hora movimiento
* oce\_folioorden = poner el nuevo tipo de orden- nomenclatura TPP
* sod\_idsituacionorden = situación AUTORIZADO POR TPP

Sera necesario e importante que al tipo de orden compra padre:

* Se actualice en la tabla [cxp\_ordencompra] actualizar la situación [sod\_idsituacionorden] por: Pagado por TPP
* Se genere nuevo registro en la tabla [cxp\_movimientosorden] con situación [sod\_idsituacionorden] por: Pagado por TPP

**AFECTACION CONTABLE.**

El proceso para contabilizar los tres procesos anteriores, será el mismo que ya se tiene programado, considerando que las tablas [plp\_planpisoenc] y [plp\_planpisodet] tienen relación con el campo **ple\_idplanpiso.**

Se deberán considerar las mismas validaciones:

* Que la cuenta contable exista en el catálogo y sea de detalle.
* Que si la cuenta es de cartera (parámetro cartera – descripción 5) es obligatorio el campo de persona, documento, tipo de documento y VIN.
* El tipo de documento (parámetro TIMO) debe existir en la BDD donde se está procesando la póliza, el sistema ya sabe trabajar con la regla dependiendo el valor que contenga en el importe1
  + Creación del documento original importe1 = 1
  + Afectación a documento original importe1 = 0
* Que la suma de cargos y abonos por póliza sea el mismo importe.

El proceso para almacenar los registros contables en las tablas:

* CON\_POL + GlbEmpAct + GlbAnnAct
* CON\_MOVDET + GlbEmpAct + GlbAnnAct
* CON\_CAR + GlbEmpAct + GlbAnnAct

Deberá ser similar al procedimiento **PROCESO\_AUTOMATICO** que ya se tiene programado, considerando los campos adicionales exclusivos de centralización.

El campo [pld\_vin] se deberá actualizar a nivel de cartera es decir, tabla CON\_CAR + GlbEmpAct + GlbAnnAct campo CCP\_OBSGEN. Con el objetivo de que sus reportes tales como antigüedad de saldos, consulta de movimientos y neteos versión centralizada, muestre a nivel de cartera la referencia del vehículo.

\*\*NOTA\*\* Es muy importante mencionar que la afectación contable a **nivel de documento** tanto en CON\_MOVDET + GlbEmpAct + GlbAnnAct como en CON\_CAR + GlbEmpAct + GlbAnnAct se deberá seguir haciendo de la misma forma, esto quiere decir que el **folio generado por la orden de compra** es el mismo que se tiene que almacenar en los campo **ccp\_iddocto** y **mov\_iddocto.**

**PLP\_LOGPLANPISO**

|  |  |  |
| --- | --- | --- |
| **plp\_logplanpiso** | | |
| **Campo** | **tipo** | **comentarios** |
| lpp\_idlogplanpiso | numeric identity | Identificador de la tabla de log de plan piso |
| lpp\_idempresa | integer | Identificador de la empresa en la que se originó el error |
| lpp\_idsucursal | integer | Identificador de la sucursal en la que se originó el error |
| lpp\_iddocumento | varchar | Documento con error |
| lpp\_tipopoliza | varchar | Tipo de póliza que mando el error |
| lpp\_error | varchar | Descripción del error |
| lpp\_fecha | datetime | Fecha |

CREATE TABLE plp\_logplanpiso

(

lpp\_idlogplanpiso numeric(18) IDENTITY ( 1,1 ) ,

lpp\_idempresa integer NOT NULL ,

lpp\_sucursal integer NOT NULL ,

lpp\_iddocumento varchar(50) NOT NULL ,

lpp\_tipopoliza varchar(20) NOT NULL ,

lpp\_error varchar(500) NOT NULL ,

lpp\_fecha datetime NOT NULL

)

go

ALTER TABLE plp\_logplanpiso

ADD CONSTRAINT XPKplp\_logplanpiso PRIMARY KEY (lpp\_idlogplanpiso ASC)

go

Se estarán generando registros en esta tabla cuando:

* El mes se encuentre cerrado y no se pueda afectar.
* La cuenta no exista o no sea de cartera.
* Que no cuadren los importes entre cargos y abonos de las pólizas a procesar.
* Que no exista la persona a la que s ele va a generar el movimiento contable.
* Que no exista el identificador en el parámetro TIMO.
* Que no se encuentre la empresa y/o sucursal.
* Que no exista el tipo de póliza.

De la misma forma se estarán generando registros en esta tabla [plp\_logplanpiso] cuando existan los siguientes problemas al generar la orden de compra:

* No existe equivalencia alguna de empresa – sucursal contra el catalogo para generar la orden de compra (Enviar nombre de la empresa y sucursal).
* No se puede elaborar Orden de compra por falta de VIN.
* No se puede elaborar Orden de compra para el número de serie: [pld\_vin], debido a que no existe importe para el subtotal.

**PLP\_BITACORA**

|  |  |  |
| --- | --- | --- |
| **plp\_bitacora** | | |
| **Campo** | **tipo** | **comentarios** |
| bit\_idbitacora | numeric identity | Identificador de la tabla de bitácora. |
| ple\_idplanpiso | numeric | identificador de la tabla de plan piso encabezado |
| pro\_nombrecorto | varchar | Nombre corto del tipo de proceso a generar: 0 - NA, 1 - PR, 2 - CPR, 3 - PFM, 4 - CPFM, 5 - CON, 6 - CCON, 7 - APL, 8 - CAPL, 9 –PAG, 10 – CPAG, 11 – COM y 12 - TPP. |
| pld\_vin | varchar | VIN de la unidad que se afectó en la póliza |
| bit\_mes | numeric | Mes en el que se está generando la póliza |
| bit\_año | varchar | Año en el que se generó la póliza |
| bit\_fechageneracion | Date time | Fecha en la que se generó registro |

CREATE TABLE plp\_bitacora

(

bit\_idbitacora numeric(18) IDENTITY ( 1,1 ) ,

ple\_idplanpiso numeric(18) NOT NULL ,

pro\_nombrecorto varchar(20) NOT NULL ,

pld\_vin varchar(17) NOT NULL ,

bit\_fechageneracion datetime NOT NULL ,

bit\_mes integer NOT NULL ,

bit\_año varchar(4) NOT NULL

)

go

ALTER TABLE plp\_bitacora

ADD CONSTRAINT XPKplp\_bitacora PRIMARY KEY CLUSTERED (bit\_idbitacora ASC)

go

Estaremos generando registros en esta tabla cada vez que se procesen las diferentes transacciones. En caso de que la póliza contenga un conjunto de unidades es importante generar los mismos registros para tener la referencia de la unidad, esto quiere decir; que se puede tomar el detalle de la póliza generada que identifica a las unidades involucradas en ese proceso.

# **Material visual de referencia**

N/A

# **Archivos relacionados**

|  |  |
| --- | --- |
| AUTORIZACIÓN PARA INICIO DE ATENCIÓN DE NUEVO REQUERIMIENTO | |
| *[Firma autógrafa]* | |
| Nombre y puesto de quien autoriza | **Fecha de Autorización**  *[DD-MM-AAAA]* |