

# Conciliación (Carga de Datos)

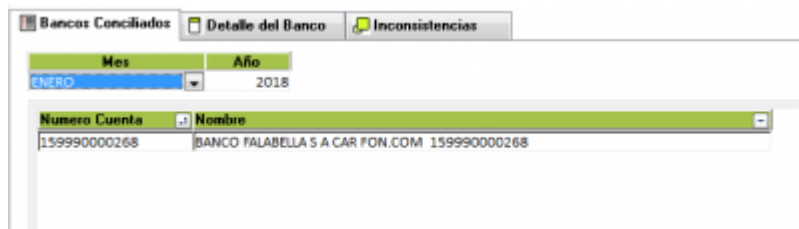
En esta ventana podemos subir al sistema la información que el banco nos ha suministrado de los movimientos realizados en nuestras cuentas, a través del archivo de extracto bancario.

Contiene tres pestañas:

## Pestaña Bancos Conciliados

Seleccionando el un mes y un año, podemos visualizar los datos de identificación de los archivos de extractos bancarios actualmente existentes en el sistema,

Usando el botón Nuevo, podemos ingresar un archivo de extracto bancario al sistema



## Pestaña Detalle del Banco

Sirve para incluir los datos de identificación el archivo de extracto bancario subido al sistema: Código de la cuenta, saldo inicial y final, período a conciliar, etc

<b>Banco</b>		
Código: 11-000268	Nombre: BANCO FALABELLA S A CAR FON.COM 159990000268	
Cuenta Bancaria: 159990000268	Tipo: Cuenta de Ahorros	
Dirección:	Teléfono:	
Saldo Inicial: 2411553777.99	Saldo Final: 9.451.788.464	Período: 2018-01
Forma de Conciliación: <input checked="" type="radio"/> Extracto <input type="radio"/> Manual	Estado de la Conciliación: INCOMPLETA	

## Pestaña Inconsistencias

Esta es la pestaña de trabajo de la conciliación bancaria, en ella podemos identificar claramente que partidas están actualmente conciliadas y cuáles faltan por conciliar, tanto de nuestro sistema como del extracto bancario.

Cuando se suben los datos de la conciliación de forma manual se puede agregar un NIT de terceros, para identificación posterior, en los reportes.

## Notas del proceso

- Cuando el mes de la anulación sea mayor al mes del documento, y ambos meses sean menores o iguales al mes a conciliar entonces se muestran ambos documentos.
- Los datos del extracto de la conciliación bancaria puede incluir terceros; en este momento, solamente se puede agregar terceros cuando se ingresan manualmente los datos de información del extracto.
- Las partidas conciliatorias se pueden conciliar así sean de meses anteriores.
- Se modifica la validación de errores al importar archivo de conciliación bancaria, para que en caso de no tener ninguna línea con la estructura, el sistema posterior a esto no debe sacar error, y posterior a esto, traer las partidas conciliatorias del mes anterior.

# Corrección de Estado en Conciliación Bancaria

1. Descripción del Problema Se detectó un comportamiento inconsistente en la ventana `w_conciliacion_bancos`. Al realizar una conciliación (ya sea manual o por importación) que resultaba en un cuadro perfecto en la primera transacción, el sistema asignaba erróneamente el estado "Incompleto" (I) en lugar de "Completado" (C).

Para corregir esto, el usuario debía: Salir de la conciliación. Volver a entrar al registro. Realizar un cambio manual trivial (desmarcar y marcar algo) para que el sistema recalculara el estado a "Completado".

## 2. Causa Raíz (Análisis Técnico)

Desincronización de Buffers: El código realizaba validaciones de totales antes de ejecutar `AcceptText()`, lo que causaba que el motor de PowerBuilder evaluara valores antiguos en los campos calculados del `DataWindow`. Campos Calculados no Refrescados: Al usar grupos y totales en el pie del `DataWindow`, estos no siempre se recalculan inmediatamente. Se requería una llamada explícita a `GroupCalc()`. Lógica de Estado Fragmentada: El evento `ue_grabar` contenía múltiples bloques IF y bucles FOR redundantes para evaluar si la conciliación estaba completa. Si el flujo de importación automática terminaba el proceso, estos bloques no siempre se ejecutaban en el orden correcto o con los datos frescos.

## 3. Solución Implementada

Se reestructuró el evento `ue_grabar` bajo los siguientes pilares:

Sincronización Total: Se agregaron llamadas a `AcceptText()` y `GroupCalc()` para todos los `DataWindows` involucrados (`idw_det_bancos`, `idw_coincidentes`, `idw_incoincidentes`, etc.) al inicio del evento.

Unificación de la Lógica de Estado:

Se reemplazaron los bucles manuales por una función `.Find()` de alto rendimiento que busca registros pendientes (`conciliado = 0`) que no sean partidas conciliatorias.

Aseguramiento de Transacción: Se garantiza que el cambio de estado a 'C' ocurra dentro del mismo bloque atómico antes del `Update()` y el `COMMIT`.

4. Cambios en el Código (`w_conciliacion_bancos.srw`) A continuación se detalla el cambio realizado en el evento `ue_grabar`:

*Si no hay registros pendientes (conciliado=0) que NO sean partidas conciliatorias, el estado es Completo ('C'), de lo contrario es Incompleto ('I')*

```
If idw_datos_incoincidentes.Find('conciliado = 0 AND conciliatoria = "N"', 1,  
idw_datos_incoincidentes.RowCount()) = 0 AND &
```

```
idw_incoincidentes.Find('conciliado = 0', 1, idw_incoincidentes.RowCount())
```

```
= 0 Then  
  ls_estado = 'C'
```

Else

```
  ls_estado = 'I'
```

End If

```
idw_det_bancos.SetItem(1, "estado", ls_estado)
```

## Corrección del Proceso de Eliminación de Conciliaciones

### 1. Resumen del Problema

Módulo: Tesorería / Conciliación Bancaria.

Bug ID: [#53269]

Descripción: El sistema no permitía eliminar conciliaciones en estado "Incompleto" ('I'). Aunque el usuario ingresaba la descripción de eliminación y guardaba, el registro persistía en la base de datos con su estado original. Además

### 2. Análisis de Causa Raíz

Se identificaron tres factores técnicos que provocaban el fallo:

Sobreescritura de Estado en Memoria: El evento ue\_grabar de la ventana w\_conciliacion\_bancos recalculaba el estado de la conciliación basándose en los registros pendientes. Este cálculo ignoraba si el usuario había marcado el registro como "Eliminado" ('E'), sobrescribiendo el cambio justo antes de enviar los datos a la DB. Persistencia de Buffers: La ventana de descripción no limpiaba los campos al inicializarse, mostrando datos residuales de sesiones previas.

### 3. Cambios Realizados

B. Lógica de la Ventana (w\_conciliacion\_bancos.srw) Evento ue\_grabar Se añadió una condicional para evitar que la lógica de cálculo automático de estado (Incompleto/Completo) se ejecute cuando el proceso actual es una eliminación.

```
// Solo recalcular estado si NO es una eliminación  
If lista_elementos <> 'eliminar_conciliacion' Then  
  If idw_datos_incoincidentes.Find('conciliado = 0 AND conciliatoria =  
"N"', 1, idw_datos_incoincidentes.RowCount()) = 0 AND &  
  idw_incoincidentes.Find('conciliado = 0', 1,  
idw_incoincidentes.RowCount()) = 0 Then
```

```
        ls_estado = 'C'  
Else  
        ls_estado = 'I'  
End If  
idw_det_bancos.SetItem(1, "estado", ls_estado)  
idw_det_bancos.accepttext( )  
End If
```

## Estándar de Persistencia y Manejo de Transacciones (Apeon)

### 1. Contexto

En aplicaciones desplegadas bajo la arquitectura Apeon PowerServer, el manejo de sesiones de base de datos es dinámico. Para garantizar que las operaciones de actualización (Update) y el SQL embebido se ejecuten dentro de la transacción correcta y no queden “colgadas” o se confirmen accidentalmente por otros procesos, es obligatorio sincronizar el contexto transaccional.

### 2. Definición: of\_sql\_embedded\_context

Esta función de objeto de usuario global (guo\_app) establece el manejador de la base de datos para las sentencias que siguen en el script. Actúa como un “puente” que le dice al servidor de aplicaciones: “Todo lo que viene a continuación debe viajar por este túnel transaccional específico”.

### 3. Directriz de Implementación

Se establece como estándar obligatorio invocar la sentencia antes de cualquier operación de persistencia de DataWindows o DataStores que utilice objetos de transacción distintos a ts\_transaccion o cuando se requiera asegurar la integridad en procesos críticos (como Conciliación Bancaria).

Patrón de Código Sugerido Fragmento de código 1. *Sincronizar el contexto antes de la persistencia guo\_app.of\_sql\_embedded\_context(ts\_transaccion)* 2. Ejecutar la actualización con control de flags (Recomendado para visualización previa) IF idw\_datos\_conciliacion.Update(True, False) = 1 THEN

```
// La operación fue exitosa en el contexto de ts_transaccion  
// NOTA: El COMMIT se debe realizar solo cuando el usuario confirme la  
acción final
```

ELSE

```
// En caso de error, limpiar el contexto inmediatamente  
ROLLBACK USING ts_transaccion;  
MessageBox("Error", "Falla al actualizar el contexto de datos.")
```

END IF

#### 4. Problemas que resuelve esta implementación

Aislamiento de Transacciones: Evita que un COMMIT en una ventana B confirme datos cargados erróneamente en la ventana A (Conciliación).

Compatibilidad Apeon: Corrige errores donde el servidor de aplicaciones pierde el rastro de la transacción activa tras procesos largos de importación.

Integridad en el “Deshacer”: Al estar el contexto correctamente mapeado, un ROLLBACK USING ts\_transaccion limpiará efectivamente todas las inserciones temporales realizadas por las funciones de importación (wf\_import\_extracto, etc.).

## Optimización y Corrección del Proceso de Conciliación Bancaria

Bug ID: [#53442]

### 1. Descripción del Problema

Se detectó que movimientos contables de meses anteriores (ej. enero/febrero) reaparecían en la conciliación del mes de abril a pesar de haber sido conciliados previamente. Adicionalmente, se identificaron bloqueos en la interfaz de usuario que impedían la gestión ágil de partidas conciliatorias y errores en la determinación del estado final de la conciliación.

### 2. Causa Raíz

Falta de Estampa de Tiempo: Las funciones de cruce (automático y manual) no estaban asignando el valor a la columna periodo\_conciliado en la tabla DET\_ASIENTO\_CONTABLE, dejando el registro como “huérfano” para procesos futuros. Limpieza Indiscriminada: La función de validación previa al guardado (wf\_puede\_guardar) realizaba una limpieza basada solo en periodo y banco, lo que bajo ciertas condiciones de concurrencia desmarcaba registros válidos. Bloqueo de UI: La lógica de protección de columnas no incluía la columna conciliatoria, y el evento itemchanged de los cuadrantes no notificaba a la ventana la necesidad de habilitar el botón “Grabar”.

### 3. Soluciones Implementadas

A. Integridad de Datos (Periodo de Conciliación) Se modificaron los procesos de cruce para asegurar que cada registro marcado como conciliado reciba el periodo actual:

Objeto n\_cst\_conciliacion\_bancaria: En udf\_conciliar\_faltante, se añadió la asignación de is\_periodo\_conciliando. Ventana w\_conciliacion\_bancos: En wf\_conciliar, se añadió la asignación de

is\_periodo. B. Lógica de Estado (Completa vs. Incompleta) Se refinó la validación en el evento ue\_grabar para evitar que una conciliación se marque como "Completa" si no hay actividad real.

Actividad Válida: Se considera actividad si existe al menos un registro con conciliado = 1 O una partida del extracto marcada como conciliatoria = 'S'. C. Mejoras en la Interfaz de Usuario (UX) Edición Fluida: Se actualizó wf\_bloquea\_campos\_detalle para que la columna conciliatoria sea editable cuando el estado es 'I' (Incompleta). Habilidad Dinámica: Se insertó el evento ue\_enable\_grabar en los itemchanged de los DataWindows de la pestaña 3. Esto elimina la necesidad de salir y volver a entrar a la ventana para guardar cambios manuales. D. Seguridad Transaccional Se ajustó el SQL de la función wf\_puede\_guardar para incluir el identificador único id\_cconciliacion en las cláusulas de exclusión, garantizando que el proceso de guardado actual no borre accidentalmente sus propias marcas.

#### 4. Objetos Modificados

Objeto Tipo Método/Evento n\_cst\_conciliacion\_bancaria SRU udf\_conciliar\_faltante  
w\_conciliacion\_bancos SRW wf\_conciliar, ue\_grabar, wf\_puede\_guardar, wf\_bloquea\_campos\_detalle  
d\_incoincidentes SRD Evento itemchanged d\_incoincidentes\_extracto SRD Evento itemchanged

#### A continuación presento los cambios realizados en los archivos:

```
if ll_encuentra_igual > 0 then
    idw_incoincidentes.SetItem          (ll_encuentra_igual,'conciliado',1)
    idw_incoincidentes.SetItem
    (ll_encuentra_igual,'periodo_conciliado',is_periodo_conciliando)
    idw_datos_incoincidentes.SetItem(ll_cont_ext,'conciliado',1)
    idw_datos_incoincidentes.SetItem(ll_cont_ext,'usuario_concilia',cod_usuario)
    idw_datos_incoincidentes.SetItem(ll_cont_ext,'fecha_conciliado',id_fecha)
```

w\_conciliacion\_bancos.srw

```
    If ll_docto_lib = ll_docto_ext And ld_valor_lib = ld_valor_ext Then
        idw_coincidentes.SetItem          (ll_cont_lib,'conciliado',1)
        idw_coincidentes.SetItem
        (ll_cont_lib,'periodo_conciliado',is_periodo)
        idw_datos_coincidentes.SetItem(ll_cont_ext,'conciliado',1)
        idw_datos_coincidentes.SetItem(ll_cont_ext,'usuario_concilia',cod_usuario)
        idw_datos_coincidentes.SetItem(ll_cont_ext,'fecha_conciliado',id_fecha)
        idw_det_bancos.accepttext( )
```

```
If lista_elementos <> 'eliminar_conciliacion' Then
    // Garantizar que no se guarde como completa si no hay ningún registro
    // marcado en los cuadrantes de la pestaña 3
    // Garantizar que no se guarde como completa si no hay actividad (marcas
    // de conciliado o partidas conciliatorias)
    If idw_det_bancos.GetItemString(1, "estado") = 'C' Then
        If idw_coincidentes.Find("conciliado = 1", 1,
        idw_coincidentes.RowCount()) <= 0 And &
```

```
        idw_incoincidentes.Find("conciliado = 1", 1,
idw_incoincidentes.RowCount()) <= 0 And &
        idw_datos_coincidentes.Find("conciliado = 1", 1,
idw_datos_coincidentes.RowCount()) <= 0 And &
        idw_datos_incoincidentes.Find("conciliado = 1", 1,
idw_datos_incoincidentes.RowCount()) <= 0 Then
        idw_datos_incoincidentes.Find("conciliado = 1 OR conciliatoria =
'S'", 1, idw_datos_incoincidentes.RowCount()) <= 0 Then
        idw_det_bancos.SetItem(1, "estado", 'I')
    End If
End If
idw_datos_incoincidentes.Modify('ajuste.protect = ' + ps_protect)
```

End If

If not( isnull(idw\_datos\_incoincidentes)) And isvalid(idw\_datos\_incoincidentes) Then

```
idw_datos_incoincidentes.Modify('conciliatoria.protect = ' + ps_protect)
```

End If

end subroutine

public function boolean wf\_puede\_guardar ()

```
        (SELECT 1
        FROM TESORE01.MAE_CONCILIACION MC
        WHERE MC.CODIGO_BANCO = :ld_cod_inter_banco
        AND MC.PERIODO = :ls_periodo
        AND MC.ESTADO <> 'E'
        AND MC.ESTADO <> 'E'
        OR MC.CODIGO_CONCILIACION = :id_cconciliacion
        )
USING sqlca;
```

From:  
<http://wiki.adacsc.co/> - Wiki

Permanent link:  
<http://wiki.adacsc.co/doku.php?id=ada:sicoferp.financiero.tesoreria.conciliacionbancaria:conciliacioncargadedatos>

Last update: 2026/04/20 13:46

