

**V1.8**

**Elaborado por : Latam Consulting**

**Julio 2021**

**ANÁLISIS DEL MÓDULO**

**RRHH Cheques**

**Desarrollado en PowerBuilder**

Contenido

[2 ANTECEDENTES 3](#_Toc78897991)

[3 ANÁLISIS BREVE DE FUNCIONALIDAD 3](#_Toc78897992)

[4 Módulo Interfases RH – eIBS 4](#_Toc78897993)

[4.1 Menú Principal ca\_cheques 5](#_Toc78897994)

[4.2 Opción Archivo 5](#_Toc78897995)

[4.3 Opción Parámetros 6](#_Toc78897996)

[4.4 Opción Proceso 7](#_Toc78897997)

[4.4.1 Impresión de Cheques 7](#_Toc78897998)

[4.4.2 Interfases RH - eIBS 13](#_Toc78897999)

[4.5 Opción Listados 38](#_Toc78898000)

[4.5.1 Listado de cheques 39](#_Toc78898001)

[4.5.2 Interfaces RH – eIBS – Listado de Acreditación 42](#_Toc78898002)

[4.6 Opción Ayuda 43](#_Toc78898003)

[5 COMPONENTES DE LA APLICACIÓN DE RRHH 43](#_Toc78898004)

[5.1 Windows Interfaces 43](#_Toc78898005)

[5.2 Datawindows 44](#_Toc78898006)

[5.3 Menu Frame 44](#_Toc78898007)

[5.4 Objeto de Conexión 45](#_Toc78898008)

[6 RESUMEN DE STORED PROCEDURES 45](#_Toc78898009)

[6.1 USP\_ROL\_PAGO\_ACREEDOR 45](#_Toc78898010)

[6.2 USP\_CA\_ACREDITACION 46](#_Toc78898011)

[6.3 USP\_CA\_ACRED\_BENEFICIO 48](#_Toc78898012)

[6.4 USP\_CA\_ASIENTO 50](#_Toc78898013)

[6.5 USP\_CA\_ROL\_DESCUENTO 52](#_Toc78898014)

[7 LINKED SERVERS 54](#_Toc78898015)

[8 RIESGOS IDENTIFICADOS 54](#_Toc78898016)

[9 ANEXOS – Diseño Actual Módulo Interfaces RH – eIBS 55](#_Toc78898017)

# ANTECEDENTES

El banco requiere migrar la versión de los componentes desarrollados en PowerBuilder a una versión de desarrollo más comercial; el fuente proporcionado por el banco es del módulo de recursos humanos, el cual fue nombrado en el proyecto de PowerBuilder como **“ca\_cheques”**; el mismo tiene una interfaz de login que se conecta mediante ldap a una base de datos de SQL Server y si las credenciales son exitosas accede a un menú que puede derivar en cuatro opciones principales, Archivo, Parámetros, Proceso y listado, cuyos procesos se irán detallando conforme el desarrollo de este documento.

# ANÁLISIS BREVE DE FUNCIONALIDAD

Se revisó el funcionamiento del módulo de Recursos Humanos realizado en PowerBuilder 7; se determina que se está trabajando con tres bases de datos, SQL Server 2005, DB2/400 Y ORACLE; los accesos a la base de datos SQL Server son por Active Directory, es decir, el ingreso a las aplicaciones de PowerBuilder lo tienen configurado para que con su mismo usuario de Windows puedan conectarse al aplicativo y a su vez se relacione con la base de datos SQL Server.

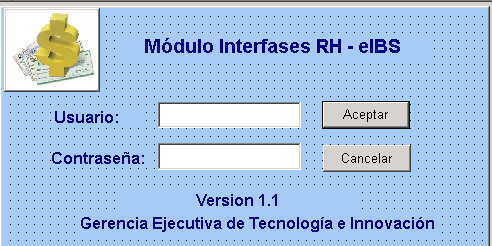
El SQL Server tiene configurados varios “linked servers” y DTS’s que apuntan a las respectivas base de datos ya sea ORACLE o DB2/400, quiere decir que los servidores de SQL Server deben tener instalados los driver clientes tanto de ORACLE como DB2/400 para poder realizar las conexiones respectivas; la base de datos SQL Server principal utilizada es **“RRHH”,** además, aquí se encuentran también todos los procedimientos almacenados (stored procedures).

A groso modo el sistema, toma la información generada por la aplicación “PeopleSoft” desde un repositorio de ORACLE valiendo de otro aplicativo llamado “**Interfaces RH”,** también denominada como **“cubitos”,**  entonces se vuelca la información desde las tablas de Oracle hacia las tablas de SQL Server mediante “stored procedures”.

En el módulo se puede aprobar los cheques e imprimirlos, también se aprueban los asientos contables, las acreditaciones y otras funciones de nómina, finalizando con las conexiones hacia DB2/400 para contabilizar la información en el sistema eIBS.

# Módulo Interfases RH – eIBS

**Login:**



Esta interfaz a groso modo realiza una conexión a una base de datos SQL Server 2008 R2 en el servidor “tsiapp209” utilizando por default “ldap” para la gestión de accesos.

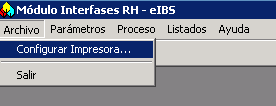
Los métodos principales programados para esta funcionalidad son:

* **Open**(). Se encarga de cargar el archivo de configuración llamado “intefases.ini” que se encuentra dentro del workspace del proyecto el cual contiene varias configuraciones entre las cuales están las configuraciones de servidores de base de datos tanto ORACLE, SQL Server y DB2/400. Adicional crea el objeto de conexión de LDAP para utilización de Active Directory.
* **Ue\_aceptar**(). Se encargar de validar las credenciales proporcionadas en el formulario por el usuario y arma las cadenas de conexión para SQL Server; por otro lado la base de dato esta hardcodeada como “RRHH”, el nombre del servidor toma del archivo de configuración “interfase.ini” y posteriormente si la conexión es satisfactoria avanza a la interfaz “w\_ca\_cheques\_frame”.

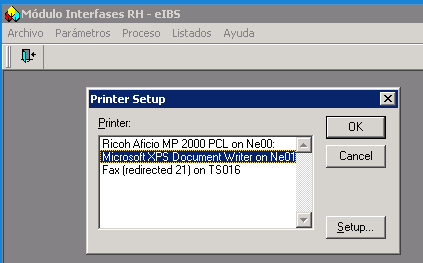
## Menú Principal ca\_cheques

Esta interfaz es el administrador de funcionalidades principal o contenedor de formularios dependiendo de la selección del usuario y cuenta diferentes opciones de menú.

## Opción Archivo



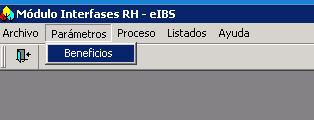
Selecciona el dispositivo a utilizarse para las impresiones locales de cheques.



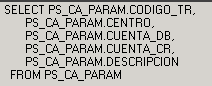
La función de PowerBuilder ejecutada se llama “PrintSetup( )”, la cual, invoca el cuadro de dialogo del sistema operativo para configuración de los dispositivos de impresión, permitiéndole al usuario que ingrese los parámetros específicos para la impresora.

## Opción Parámetros

Dentro de la opción de menú “Parámetros” tenemos el sub ítem de “Beneficios”.

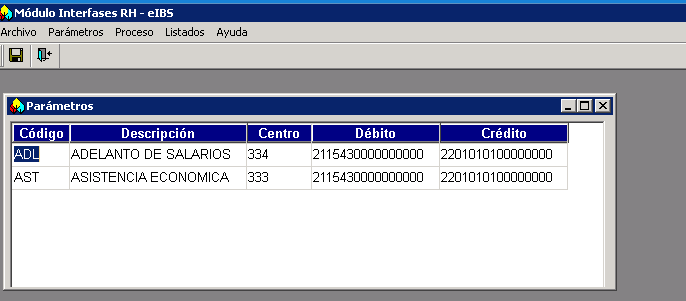


Esta opción llama al formulario ”w\_parametro” que carga un grid con el siguiente query:



Este query esta embebido en el datawindow y hace una conexión a la base de datos de SQL Server llamada “RRHH”.

Visualizándose de la siguiente manera en pantalla:

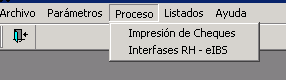


Como se puede observar es una pantalla que solamente es de uso informativo o de consulta.

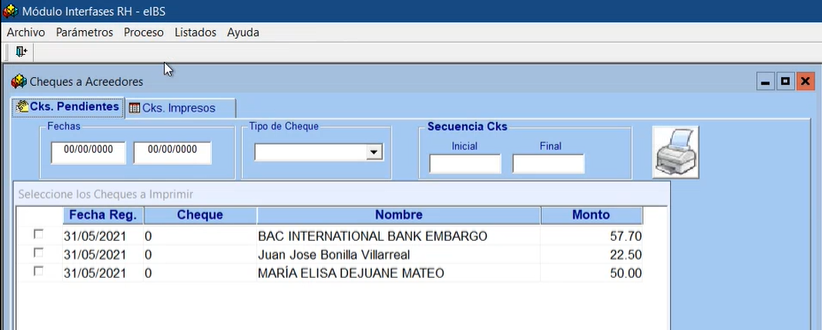
## Opción Proceso

Dentro de la opción proceso tenemos un submenú con dos (2) opciones:

* Impresión de cheques
* Interfases RH – eIBS



### Impresión de Cheques



Esta interfaz está compuesta de dos tabs uno de cheques pendientes y otro de cheques impresos su nombre físico es “w\_cheques”, esto implica que cada tab tiene un evento, es decir, si es el clickeado en el tab “Cks Pendientes” se carga los cheques con estado “P” o pendientes de impresión, sin embargo, si es clickeado el tab “Cks Impresos” entonces se cargan los cheques con estado “I” o impresos.

Contiene un método open(), que ejecuta el stored procedure “USP\_CA\_ROL\_PAGO\_ACREEDOR” con los parámetros @opcion = 0, @usuario = :Sqlca.LogId login, el cual es el usuario de active directory; el stored procedure pertenece a la base de datos RRHH del servidor SQL Server llamado “tsiapp209”.

Desde el stored procedure con los parámetros mencionados, inicialmente se carga la data de cheques desde ORACLE..SYSADM.PS\_CA\_ROL\_PAGO\_ACR a la tabla de SQL Server PS\_CA\_ROL\_PAGO\_ACREEDOR en la base de datos RRHH.

Después de ejecutar el store procedure, el método open inicializa la variable “is\_tipo = ‘AC’” para que despliegue el tab o pestaña de cheques pendientes en el “tab\_1” y el grid a través del datawindow “dw\_2” con los cheques que sean de tipo “acreedor” y que tengan el status en ‘P’ que es pendiente por imprimir.

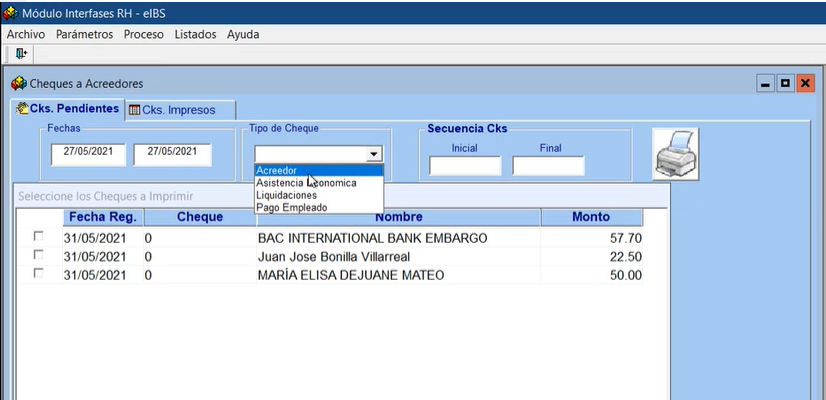
**Método open():**



**Store procedure USP\_CA\_ROL\_PAGO\_ACREEDOR:**

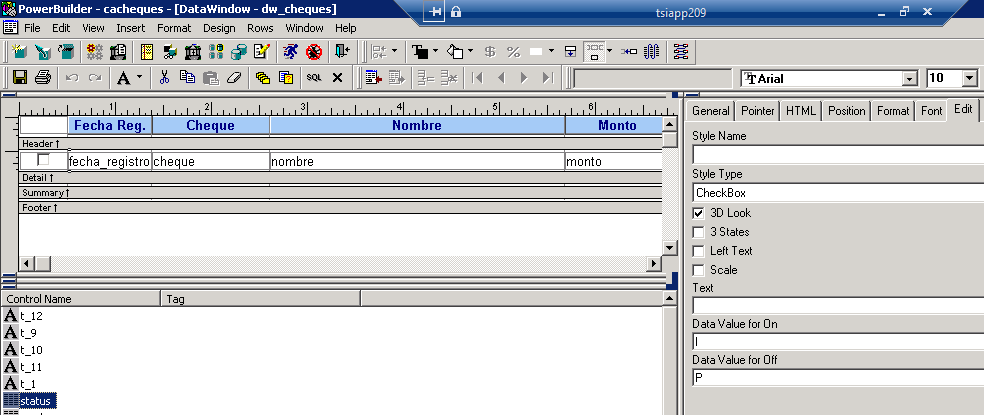


#### Pestaña Cks. Pendientes.



En esta pestaña el usuario debe rellenar manualmente las fechas, generalmente se coloca la fecha del día en que se está solicitando la impresión de los cheques; de igual forma el usuario selecciona el tipo de cheque que desea imprimir para luego indicar el número inicial del cheque a imprimir; Ese primer número corresponde al cheque de gerencia que tengan en stock y el número final se calcula automáticamente (número consecutivo) en función de la cantidad de cheques que se les haya colocado o el usuario seleccione con un check en la primera columna del grid y esos son los que se van a imprimir.

Al hacer click en la primera columna del grid, el valor de la data para el campo “status” de esa fila se setea en “ON” y le asigna el valor de ‘I’ mientras que los no marcados quedan seteados en “OFF” y con valor ‘P’ por defecto, como se puede ver en las propiedades del datawindow:



Con respecto al combo box “Tipo de cheque” tenemos que es alimentado dentro del código, no de la base de datos, con los siguientes valores:

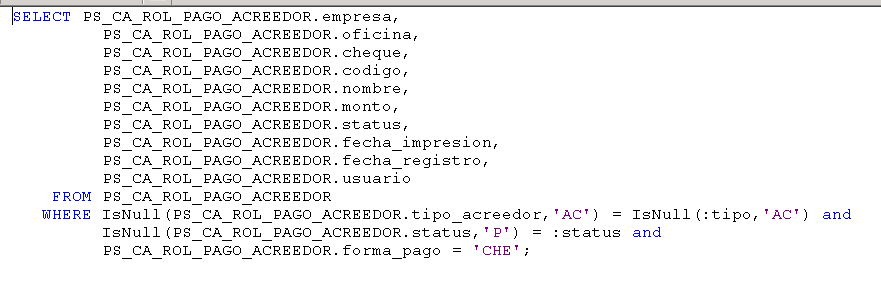
* **Acreedor**
* **Asistencia Económica**
* **Pago Empleado**
* **Liquidaciones**

Cuando selecciona una de las opciones del combo box se setea la variable “is\_tipo” dependiendo de los valores seleccionados; si selecciona “Acreedor” se asigna el valor istipo=”AC”; si selecciona “Asistencia Económica” se asigna la variable con el valor istipo=”AS”; cuando se selecciona la opción “Pago Empleado” se asigna el valor istipo=”EM” o cuando se selecciona la opción “Liquidaciones” se asigna el valor istipo =“LQ”.

Luego los valores de la secuencia de los cheques de inicial y final se limpia la caja de texto, seguido se ejecuta un query embebido en el datawindow para cargar los cheques con estado ‘P’ y con el tipo de cheque seleccionado en el combobox se carga el datawindow “dw\_2” del tab1 de cheques pendientes, sin embargo si viene vacío la ejecución muestra una alerta “No Existe Cheques Pendientes” concatenado la opción escogida por el usuario, ejemplo:

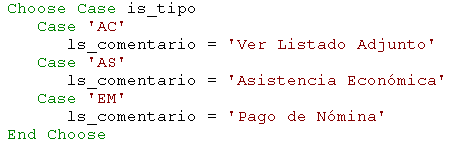
“No Existe Cheques Pendientes (“Pago Empleado”)” y así para todas las opciones.

En resumen cada vez que selecciona el combobox se recarga el grid con el siguiente query embebido en el datawindow y los parámetros proporcionados en las condiciones previamente mencionadas.

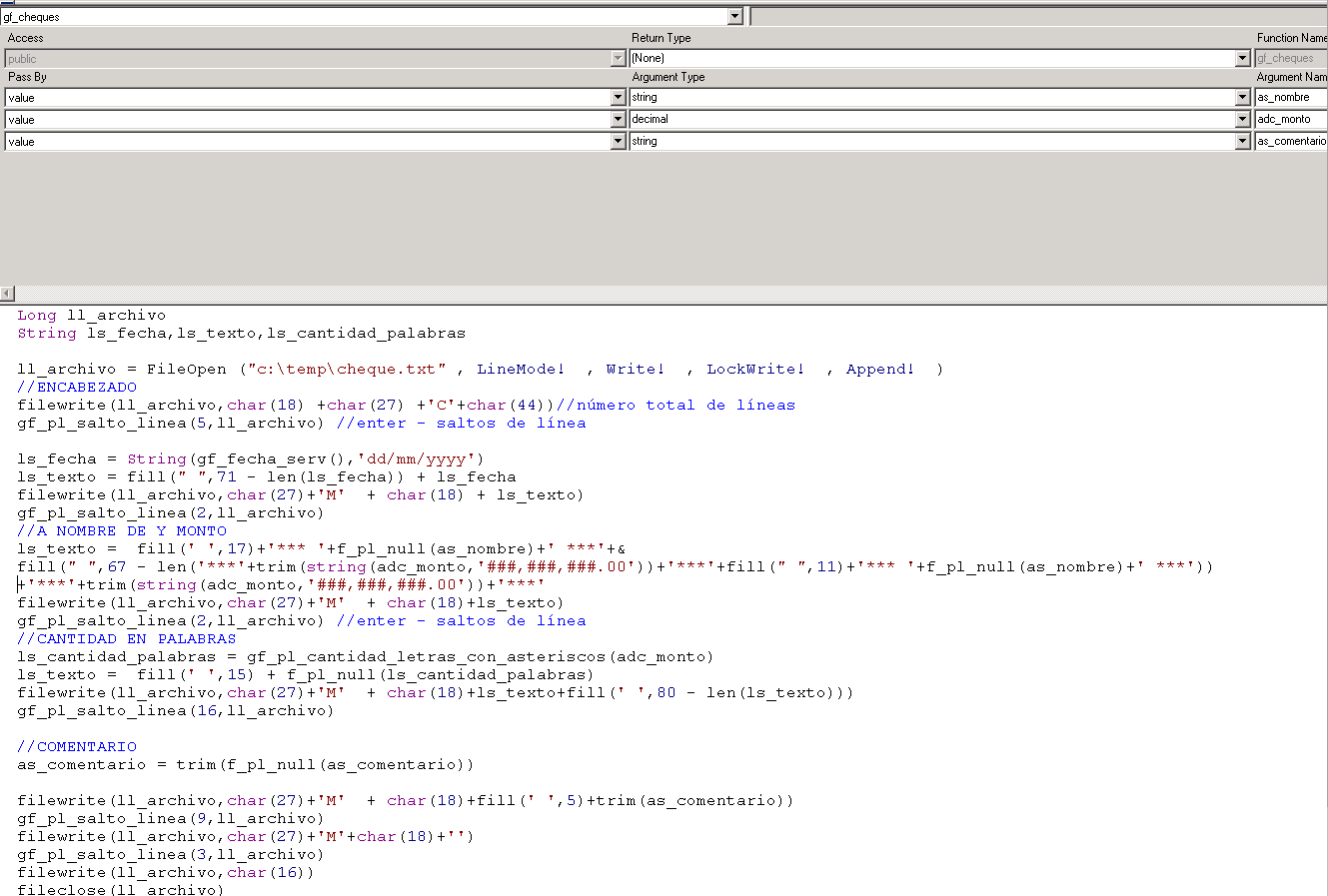


Al presionar el botón de la impresora se realiza una carga de datos con los filtros seleccionados por los clientes y a los registros seleccionados con checkmark (on), es decir con estatus = ‘I’, de esta forma selecciona del datawindow solo los cheques marcados.

Luego toma el valor de la caja de texto de la secuencia inicial y lo asigna a una variable ldc\_num\_ck de tipo decimal, luego crea un “bucle for” para barrer los registros que tiene el datacontrol dw2, desde 1 hasta el total de registros, por cada iteración se modifican las columnas “fecha\_impresion” con su fila iterada se asigna el valor de la fecha del servidor de SQL Server utilizando una función “gf\_fecha\_serv” donde obtiene el getdate() de base de datos, luego se procede con la fila y columna “cheque”, toma el valor de la variable ldc\_num\_ck se le asigna en su posición respectiva y esta variable ira cambiando por cada iteración de las filas que contenga el datacontrol, luego en la fila columna “usuario” se asigna el usuario que se encuentra logeado en el sistema en ese momento, seguido se asigna el valor de la fila columna “nombre” a la variable ls\_acreedor, también se asigna el valor de la fila columna “monto” a la variable ldc\_monto. Seguido utiliza un case dependiendo de lo que seleccionó en el tipo de cheque dejando asignado en una variable global “is\_tipo” y realiza la siguiente asignación a la variable ls\_comentario:



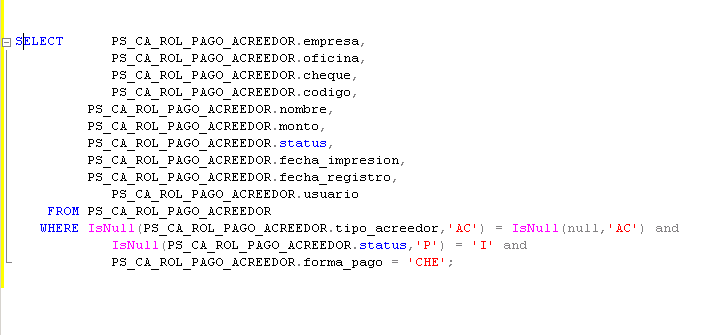
A continuación, se llama a una función global “gf\_cheques” que recibe tres parámetros de entrada el nombre, el monto, y un comentario dependiendo del tipo de cheque seleccionado, esta función arma el formato del cheque a imprimirse en un archivo plano base llamado “cheque.txt” para luego enviarlo a la impresora correspondiente a través del archivo batch “cheque.bat”.



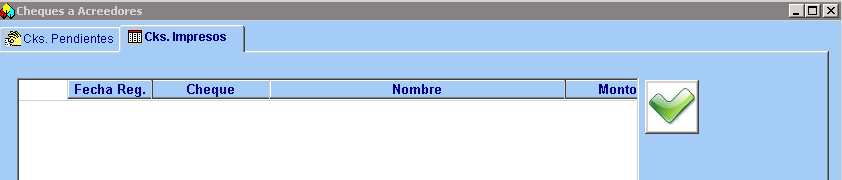
Luego de la ejecución de la función anterior se asigna a la caja de texto de la secuencia de cheques -1 “String(ldc\_num\_ck -1)” seguido se llamada la función “of\_getapppath (ls\_path,ls\_app)”

#### Pestaña o Tab Cks. Impresos

En esta ventana, el datawindow se carga con todos los cheques que han sido impresos y cuyo status es igual a ‘”I”, mediante la ejecución del siguiente query embebido:

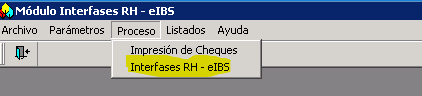
****

Al pulsar el boton con el CheckMark verde, se confirma que los cheques fueron impresos y que la data esta lista para que sea transmitida al Sistema eIBS a través de la interface correspondiente (punto 4.4.2).

****

**Pestaña Interfaz IBS:** Desactivada por código. No se utiliza ni se muestra al ejecutar la aplicación.

### Interfases RH - eIBS

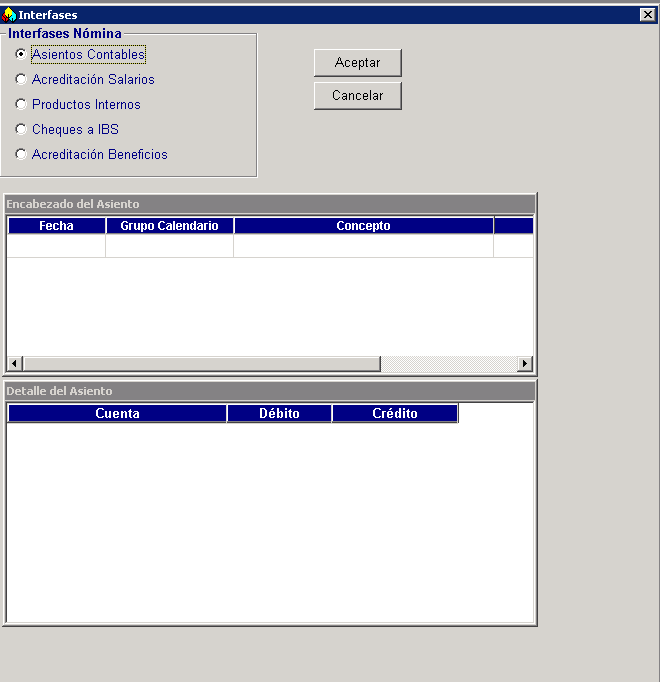


Esta opción hace uso de linked servers tanto para Oracle como para DB2/400 por ende se debe contar con un SQL Server que tenga configurado correctamente los linked server que son llamados directo en el código como llamados dentro de los procedimientos almacenados.

El nombre físico de esta interfaz es “w\_interfaces”.

Contiene cuatro métodos:

* Open(). Carga los datawindows de encabezado asiento y detalle asiento contable llamados dw2,dw3
* Close(). Cierra la aplicación.
* Ue\_aceptar(). Carga esta función cuando presiona el botón aceptar.
* Wf\_ejecutar(integer ai\_opcion). Es llamado desde el botón aceptar y dependiendo del RadioButton clickleado desde el GroupBox.



Maneja un GroupBox con 5 opciones:

* Asientos Contables
* Acreditación de salarios
* Productos Internos
* Cheques a IBS
* Acreditación de Beneficios

Inicialmente se ejecuta el método “open” que llama a la función “wf\_ejecutar(ll\_row,0) dentro de un bucle que varía en el primer parámetro llamado “ll\_row” de 1 a 4 y manteniendo siempre el segundo parámetro de la función “valor = 0”, para de esta forma llamar a los stored procedures:

* ll\_row = 1 invoca al sp USP\_CA\_ACREDITACION
* ll\_row = 2 invoca al sp USP\_CA\_ROL\_DESCUENTO
* ll\_row = 3 invoca al sp USP\_CA\_ASIENTO
* ll\_row = 4 invoca al sp USP\_CA\_ACRED\_BENEFICIO

Para los cuatro (4) stored procedures mencionados, se pasan los parámetros siguientes:

* valor = 0
* @usuario = :Sqlca.LogId USING SQLCA

El parámetro “valor=0” le indica al sp que solo debe ejecutar la carga de datos en las tablas correspondientes una sola vez.

La finalidad es extraer la información nueva o más reciente o aún no transmitida contenida las tablas de Oracle para cargarla en las tablas respectivas en MS SQL Server para el procesamiento posterior de la información y visualizaciones en pantalla.

**IMPORTANTE**: En este método NO se ejecuta el stored procedure correspondiente a la interfaz de cheques (USP\_CA\_ROL\_PAGO\_ACREEDOR), porque la carga de la data se realiza previamente en la ejecución del proceso de Impresión de Cheques.

**Método open():**



**Función wf\_ejecutar():**

****

**Store procedure USP\_CA\_ACREDITACION:**

****

**Store procedure USP\_CA\_ROL\_DESCUENTO:**

****

**Store procedure USP\_CA\_ASIENTO:**

****

**Store Procedure USP\_CA\_ACRED\_BENEFICIO:**

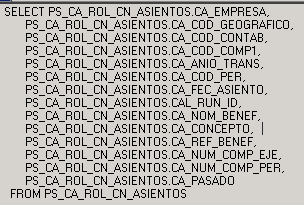
****

#### Opción del RadioButton “Asientos Contables”.

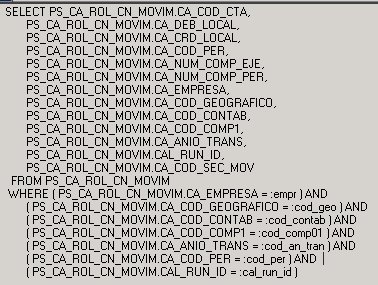
#### 

Al seleccionar esta opción se activa un evento clicked que básicamente inicializa la variable “ll\_opcion = 3”, setea los datawindows de asientos contables, “dw2 = dw\_asiento\_contable” con su título “Encabezado del Asiento” y “dw\_3 = dw\_detalle\_ac” con su título “Detalle del Asiento” para recibir información de la base de datos y llenar los grids respectivos, con los siguiente querys embebidos:

dw\_asiento\_contable (encabezado):



dw\_detalle\_ac (detalle):



Al colocarse sobre algún registro del encabezado y darle click se activa el evento respectivo que rellena la ventana de detalle con cada uno de los asientos contables correspondientes al encabezado seleccionado.

Seguidamente posicionado sobre algún registro de la cabecera y darle click al botón “Aceptar”, se ejecuta la función “ue\_aceptar” (sin parámetros) que a su vez llama a la función “wf\_ejecutar(ll\_opcion,valor)”, en este caso, el valor del parámetro ll\_opcion = 3 y el parámetro valor = 1, con el objetivo de contabilizar esos asientos en el Sistema eIBS, y desde donde inicialmente se ejecutan una serie de instrucciones y accesos a la base de datos, para validar los períodos contables, como se ve en el siguiente extracto de la función:

Long ll\_anio,ll\_periodo,ll\_comp\_eje,ll\_comp\_per,ll\_cod\_per,ll\_cod\_ejer

dw\_2.AcceptText()

ll\_anio = dw\_2.GetItemNumber(dw\_2.GetRow(),'ca\_anio\_trans')

ll\_periodo = dw\_2.GetItemNumber(dw\_2.GetRow(),'ca\_cod\_per')

SELECT B.seq\_comp ,C.seq\_comp ,B.COD\_EJER ,C.COD\_PER

INTO :ll\_COMP\_EJE, :ll\_COMP\_PER, :ll\_cod\_ejer, :ll\_COD\_PER

FROM TEST\_CONTA.DBO.CN\_SEQEJER B, TEST\_CONTA.DBO.CN\_SEQPER C

WHERE B.COD\_EJER=C.COD\_EJER AND B.COD\_COMP=C.COD\_COMP AND

B.COD\_COMP = '01' and B.COD\_EJER = :ll\_anio and

C.COD\_PER = :ll\_periodo USING sqlca;

If sqlca.SqlCode = -1 Then ls\_error = sqlca.SqlErrtext;

RollBack;

Messagebox('Atención','Error al buscar los periodos en contabilidad...' + ls\_error)

disconnect using sqlca; Return -1

End If

If sqlca.SqlCode = 100 Then

MessageBox('Atención', 'No existe el período creado en contabilidad.')

Return -1

End If

dw\_2.SetItem(dw\_2.GetRow(),'CA\_NUM\_COMP\_EJE',ll\_COMP\_EJE)

dw\_2.SetItem(dw\_2.GetRow(),'CA\_NUM\_COMP\_PER',ll\_COMP\_PER +1)

dw\_2.AcceptText()

If dw\_2.Update() = -1 Then

MessageBox('Atención','Error al actualizar los períodos')

Return -1

End If

For ll\_row = 1 to dw\_3.RowCount()

dw\_3.SetItem(ll\_row,'CA\_NUM\_COMP\_EJE', ll\_COMP\_EJE)

dw\_3.SetItem(ll\_row,'CA\_NUM\_COMP\_PER', ll\_COMP\_PER +1)

Next

dw\_3.AcceptText()

If dw\_3.Update() = -1 Then

MessageBox('Atención','Error al actualizar los períodos')

Return -1

End If

Seguidamente, se invoca al sp ”USP­\_CA\_ASIENTO” con los parámetros de entrada

* valor = 1
* @usuario = :Sqlca.LogId USING SQLCA

El parámetro “valor=1” ejecuta un llamado a un DTS definido en la siguiente ruta del servidor:

* D:\DTS\_PROD\ASIENTO\_CONTABLE\ASIENTO\_CONTABLE\ASIENTO\_CONTABLE.dtsx

Extracto del store procedure con las instrucciones del DTS: 

Al finalizar la ejecución del stored procedure y para finalizar el proceso, se realizan unas actualizaciones en la base de datos, como se observa en el siguiente extracto de la función wf\_ejecutar():

If ai\_valor = 1 Then MessageBox('Atención','Generación de Asientos Contables Realizados')

Sqlca.AutoCommit = False;

Sqlca.AutoCommit = True;

UPDATE TEST\_CONTA.DBO.CN\_SEQPER

SET seq\_comp = seq\_comp + 1

FROM TEST\_CONTA.DBO.CN\_SEQPER C

WHERE C.COD\_COMP = '01' and

C.COD\_EJER = :ll\_anio and

C.COD\_PER = :ll\_periodo;

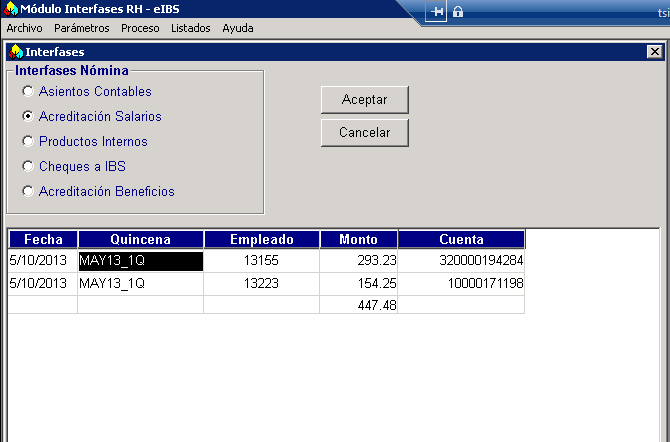
If Sqlca.SqlCode = -1 Then

ls\_error = Sqlca.SqlErrText;

RollBack;

MessageBox('Atención','Error al Actualizar la Secuencia...' + ls\_error) end If

#### Opción del RadioButton “Acreditación de salarios”.



Como se comentó previamente (punto 4.4.2), la tabla de SQL Server con la información de las acreditaciones llamada “RRHH..PS\_CA\_ROL\_ACRED\_EMPL” ha sido precargada por el store procedure “USP\_CA\_ACREDITACION” a partir de la tabla de Oracle “ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM” cuyo tipo de pago “CA\_TP\_PAGO” **NO** se inicie con los caracteres ‘BE’, es decir “CA\_TP\_PAGO not like ‘BE%’”.

Adicionalmente, se asegura que el registro de la tabla origen “ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM A” no haya sido creado en la tabla destino “RRHH..PS\_CA\_ROL\_ACRED\_EMPL B” mediante la comparación de los siguientes campos:

99999 = B.CA\_COD\_COMP AND

B.CA\_FEC\_PAGO = A.CA\_FEC\_PAGO AND

B.CA\_MONTO\_PG = A.CA\_MONTO\_PG AND

B.CA\_CTA\_ACRE = A.CA\_CTA\_ACRE AND

B.CA\_NUM\_EMPL = A.CA\_NUM\_EMPL AND

B.CA\_TP\_PAGO = A.CA\_TP\_PAGO and

A.CA\_FEC\_PAGO > dateadd(day,-10,GETDATE())

Lo anteriormente dicho lo podemos observar en el siguiente extracto del stored procedure USP\_CA\_ACREDITACION, donde el parámetro que recibe en el campo @opción es igual a cero (0) para que realice solamente la carga desde las tablas de Oracle:

IF @opcion = 0

BEGIN

INSERT dbo.PS\_CA\_ROL\_ACRED\_EMPL

SELECT 99999,

CA\_FEC\_PAGO,

CA\_MONTO\_PG,

CA\_CTA\_ACRE,

CA\_NUM\_EMPL,

CA\_TP\_PAGO,

CA\_ESTATUS,'',NULL,NULL,NULL,NULL,NULL,NULL

FROM ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM A

WHERE CA\_TP\_PAGO not like 'BE%'

AND NOT EXISTS (SELECT 1 FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE 99999 = B.CA\_COD\_COMP AND

B.CA\_FEC\_PAGO = A.CA\_FEC\_PAGO AND

B.CA\_MONTO\_PG = A.CA\_MONTO\_PG AND

B.CA\_CTA\_ACRE = A.CA\_CTA\_ACRE AND

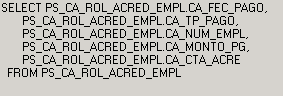
B.CA\_NUM\_EMPL = A.CA\_NUM\_EMPL AND

B.CA\_TP\_PAGO = A.CA\_TP\_PAGO)

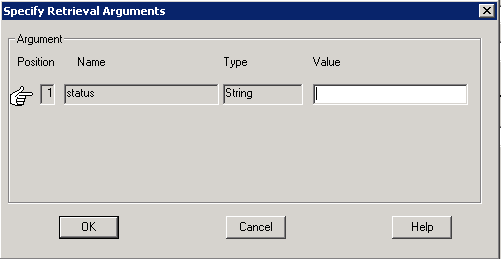
and A.CA\_FEC\_PAGO > dateadd(day,-10,GETDATE())

END

Ahora bien, una vez que aparece la pantalla de Interfaces de Nómina y se selecciona la opción del RadioButton “Acreditación Salarios”, se inicializa la variable “ll\_opcion = 1” y se setea el datawindow de acreditación “dw\_1 = dw\_acreditacion” para recibir información de la base de datos y llenar el grid respectivo, con el siguiente sql embebido en el datawindow:



La carga del grid adicionalmente tiene un filtro al campo “CA\_STATUS” con el valor de ‘P’ para recuperar los registros.



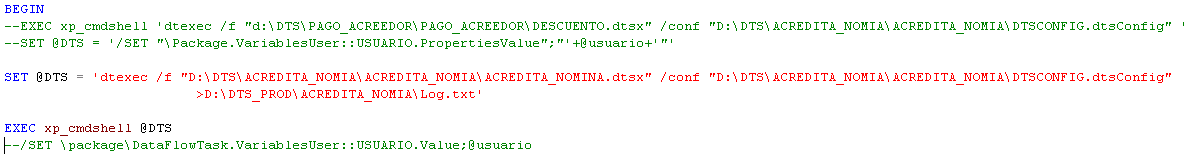
Seguidamente al darle click al botón “Aceptar” en la pantalla de Interfaces de Nómina con la opción del RadioButton “Acreditación Salarios” ya seleccionada, se ejecuta la función “ue\_aceptar” (sin parámetros) que a su vez llama a la función “wf\_ejecutar(ll\_opcion,valor)”, en este caso, el valor del parámetro ll\_opcion = 1 y el parámetro valor = 1, por lo que se invoca nuevamente al sp ”USP­\_CA\_ACREDITACION” con los parámetros de entrada

* valor = 1
* @usuario = :Sqlca.LogId USING SQLCA

El parámetro “valor=1” ejecuta un llamado a un DTS definido en la siguiente ruta del servidor:

* D:\DTS\_PROD\ACREDITA\_NOMIA\ACREDITA\_NOMIA\ACREDITA\_NOMINA.dtsx

Extracto del store procedure con las instrucciones del DTS:



De igual manera, si el parámetro “valor = 1” y el resultado de la ejecución del store procedure es satisfactorio, seguidamente en la función “wf\_ejecutar()” se ejecuta una actualización de estatus en el servidor de SQL SERVER base de datos RRHH, según vemos en el siguiente extracto de la función:

if ai\_opcion = 1 Then

ls\_sql = " UPDATE PS\_CA\_ROL\_ACRED\_EMPL " +&

" SET CA\_ESTATUS = 'A' " +&

" FROM [" +ls\_lync +"].[" + ls\_catalogo + "].[" +ls\_library + "].INPAS inp, dbo.PS\_CA\_ROL\_ACRED\_EMPL acr " +&

" Where inp.ipgacc = acr.CA\_CTA\_ACRE and " +&

" ACR.ca\_fec\_pago =CONVERT(datetime,cast (inp.ipgvdm as CHAR(2)) +'/'+ cast(inp.ipgvdd as CHAR(2)) + '/' + cast(inp.ipgvdy as CHAR(2)) ) and " +&

" acr.CA\_MONTO\_PG = inp.ipgcr1 and " +&

" acr.CA\_ESTATUS = 'P'"

Execute Immediate :ls\_sql;

If Sqlca.SqlCode = -1 Then

ls\_error = Sqlca.SqlErrText;

Rollback;

MessageBox('Atención', 'Error al Actualizar el Status de Acreditación...'+ ls\_error)

Return -1

End If

commit;

If ai\_valor = 1 Then

MessageBox('Atención','Acreditación Realizada')

End If

Las variables escritas en la sentencia anterior, ls\_lync, ls\_catalogo y ls\_library se toman del archivo llamado “interfaces.ini”, tomando los siguientes valores:

Ls\_lync=INTERFACES\_ACREDIT, ls\_catalogo=TSILPRU1 y ls\_library=CAHCYFILES

De acuerdo a lo antes mencionado, así queda la variable ls\_sql (en ambiente de desarrollo):

UPDATE PS\_CA\_ROL\_ACRED\_EMPL SET CA\_ESTATUS = 'A'

FROM [INTERFACES\_ACREDIT].[TSILPRU1].[CAHCYFILES].INPAS inp ,dbo.PS\_CA\_ROL\_ACRED\_EMPL acr

Where inp.ipgacc = acr.CA\_CTA\_ACRE

And ACR.ca\_fec\_pago = CONVERT(datetime,cast (inp.ipgvdm as CHAR(2)) +'/' + cast(inp.ipgvdd as CHAR(2)) + '/' + cast(inp.ipgvdy as CHAR(2)) )

and acr.CA\_MONTO\_PG = inp.ipgcr1

and acr.CA\_ESTATUS = 'P'

Si los valores o el archivo interfaces.ini está incompleto muestra el siguiente mensaje: “Error al Actualizar el Status de Acreditación.”

**Archivo de configuración interfaces.ini:**



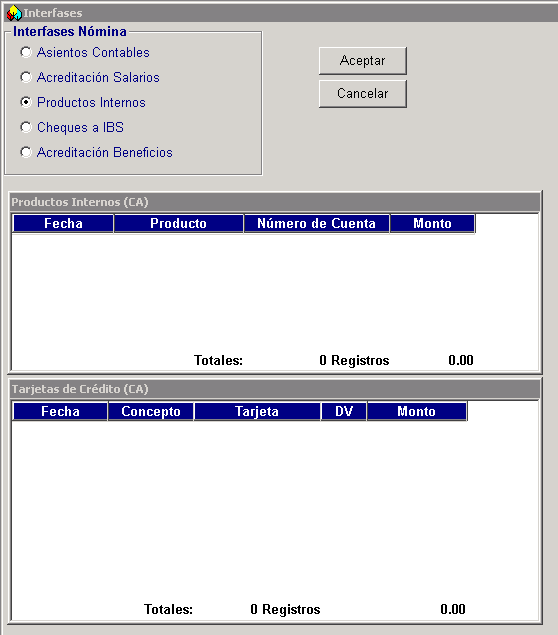
**Función wf\_ejecutar():**

****

**Store procedure USP\_CA\_ACREDITACION:**

****

#### Opción del RadioButton “Productos internos”.



Como se comentó previamente (punto 4.4.2), las tablas de SQL Server con la información de las acreditaciones llamada RRHH..PS\_CA\_ROL\_DESCUEN\_PROD y RRHH..PS\_CA\_ROL\_DESCUEN\_TC han sido precargadas por el store procedure USP\_CA\_ROL\_DESCUENTO a partir de las tablas de Oracle ORACLE..SYSADM.PS\_CA\_ROL\_DCTO\_PRO y ORACLE..SYSADM.PS\_CA\_ROL\_DCTO\_TC respectivamente, básicamente bajo el criterio principal que la información o registros de las tablas de origen no existan en las tablas destino.

Importante destacar que el valor ‘P’ en el campo CA\_ESTATUS se asigna directamente en el “select” de la sentencia SQL que inserta los registros a la tabla RRHH..PS\_CA\_ROL\_DESCUEN\_PROD.

Lo anteriormente dicho lo podemos observar en el siguiente extracto del stored procedure USP\_CA\_ROL\_DESCUENT0, donde el parámetro que recibe en el campo @opción es igual a cero (0) para que realice solamente la carga desde las tablas de Oracle:

IF @opcion = 0

BEGIN

INSERT dbo.PS\_CA\_ROL\_DESCUEN\_PROD

SELECT CA\_AINO\_PROC,

CA\_MES\_PROC,

CA\_DIA\_PROC,

CA\_AINO\_VR,

CA\_MES\_VR,

CA\_DIA\_VALOR,

CA\_CTR\_COSTO,

CA\_NUM\_CUENTA,

CA\_MONTO\_PAG,

CA\_USUARIO\_A,CA\_TP\_PROD,'P',NULL,NAME,National\_ID

FROM ORACLE..SYSADM.PS\_CA\_ROL\_DCTO\_PRO A

WHERE NOT EXISTS(SELECT 1 FROM dbo.PS\_CA\_ROL\_DESCUEN\_PROD B

WHERE A.CA\_NUM\_CUENTA = B.CA\_NUM\_CUENTA AND

A.CA\_CTR\_COSTO = B.CA\_CTR\_COSTO AND

A.CA\_AINO\_VR = B.CA\_AINO\_VR AND

A.CA\_MES\_VR = B.CA\_MES\_VR AND

A.CA\_DIA\_VALOR = B.CA\_DIA\_VALOR)

INSERT dbo.PS\_CA\_ROL\_DESCUEN\_TC

SELECT CA\_ID\_CAJERO,

CA\_MES\_TRANS,

CA\_DIA\_TRANS,

CA\_ANIO\_TRANS,

CA\_COD\_TRANS,

CA\_CONC\_PAGO,

CA\_NUM\_TARJ,

CA\_DIG\_VERIF,

CA\_MONTO\_PAG,

CA\_CTA\_CONT,

CA\_CTR\_COSTO,

CA\_ESTATUS

FROM ORACLE..SYSADM.PS\_CA\_ROL\_DESC\_TC A

WHERE NOT EXISTS(SELECT 1 FROM dbo.PS\_CA\_ROL\_DESCUEN\_TC B

WHERE A.CA\_ID\_CAJERO = B.CA\_ID\_CAJERO AND

A.CA\_MES\_TRANS = B.CA\_MES\_TRANS AND

A.CA\_DIA\_TRANS = B.CA\_DIA\_TRANS AND

A.CA\_ANIO\_TRANS = B.CA\_ANIO\_TRANS AND

A.CA\_COD\_TRANS = B.CA\_COD\_TRANS AND

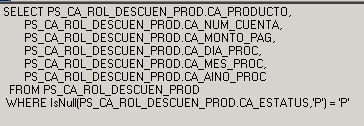
A.CA\_CONC\_PAGO = B.CA\_CONC\_PAGO AND

A.CA\_NUM\_TARJ = B.CA\_NUM\_TARJ)

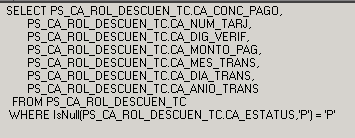
END

Ahora bien, una vez que aparece la pantalla de Interfaces de Nómina y se selecciona en cualquier momento la opción de RadioButton “Productos Internos”, se activa un evento clicked que inicializa la variable “ll\_opcion = 2” y se setean los datawindows de productos internos llamado “dw\_2 = dw\_productos\_ca” y el de tarjetas de crédito llamado “dw\_3 = dw\_tarjetas\_ca” para recibir información de la base de datos y llenar los grids respectivos, con los siguientes sql embebidos en los datawindows:

dw\_productos\_ca:



dw\_tarjetas\_ca:



Seguidamente al darle click al botón “Aceptar”, se ejecuta la función “ue\_aceptar” (sin parámetros) que a su vez llama a la función “wf\_ejecutar(ll\_opcion,valor)”, en este caso, el valor del parámetro ll\_opcion = 2 y el parámetro valor = 1, por lo que, en primer lugar se ejecuta la pantalla “w\_entrada\_ibs” para que se introduzca y valide un usuario y contraseñas del sistema eIBS.



Si la validación de usuario y contraseña de eIBS resulta satisfactoria se procede a realizar una actualización en los campos de estatus en la tabla RRHH..PS\_CA\_ROL\_DESCUEN\_PROD, como se ve en el siguiente extracto de la función “wf\_ejecutar()”:

If ai\_valor = 1 Then

OpenWithParm(w\_entrada\_ibs,1)

ls\_user\_ibs = Message.StringParm

If ls\_user\_ibs = '\*' Then

MessageBox('Atención','Usuario de IBS No Autorizado')

Return -1

Else

Update PS\_CA\_ROL\_DESCUEN\_PROD

Set CA\_USUARIO\_IBS = :ls\_user\_ibs

Where CA\_ESTATUS = 'P';

If Sqlca.SqlCode = -1 Then

ls\_error = Sqlca.SqlErrText;

RollBack Using SQLCA;

return messageBox("Error", "Error al Actualizar el Usuario

en Productos Internos..." +ls\_error, StopSign!) Else

commit;

End If

End If

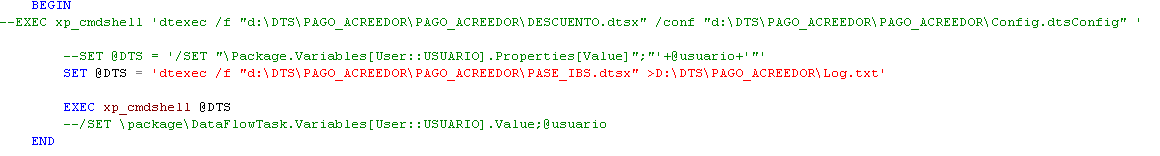
A continuación se invoca al stored procedure ”USP­\_CA\_ROL\_DESCUENTO” con los parámetros de entrada

* valor = 1
* @usuario = :Sqlca.LogId USING SQLCA

El parámetro “valor = 1” ejecuta un llamado a un DTS definido en la siguiente ruta del servidor

* "d:\DTS\_PROD\PAGO\_ACREEDOR\PAGO\_ACREEDOR\PASE\_IBS.dtsx"

Extracto del store procedure con las instrucciones del DTS:



De igual manera, si el parámetro “valor = 1” y el resultado de la ejecución del store procedure es satisfactorio, seguidamente en la función “wf\_ejecutar()” se ejecuta una actualización de los campos de estatus a valor ‘C’ en las tablas de SQL Server de la base de datos RRHH, según vemos en el siguiente extracto de la función:

If ai\_valor = 1 Then

commit;

MessageBox('Atención','Aplicación de Productos Internos Realizados')

Update dbo.PS\_CA\_ROL\_DESCUEN\_PROD

Set CA\_ESTATUS = 'C'

Where CA\_ESTATUS = 'P';

If Sqlca.SqlCode = -1 Then

ls\_error = Sqlca.SqlErrText;

RollBack Using SQLCA;

return messageBox("Error", "Error al Actualizar Productos Internos..."

+ls\_error, StopSign!)

End If

Update dbo.PS\_CA\_ROL\_DESCUEN\_TC

Set CA\_ESTATUS = 'C'

Where CA\_ESTATUS = 'P';

If Sqlca.SqlCode = -1 Then

ls\_error = Sqlca.SqlErrText;

RollBack Using SQLCA;

return messageBox("Error", "Error al Actualizar Tarjetas de Crédito..."

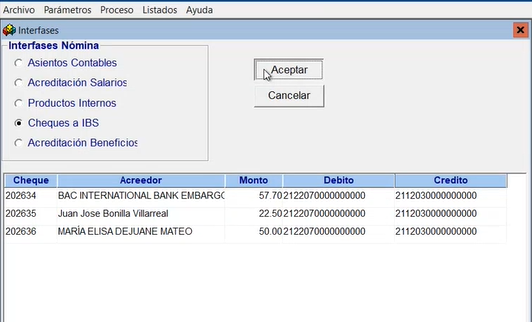
+ls\_error, StopSign!) End If

End If

**Store procedure USP\_CA\_ROL\_DESCUENTO:**

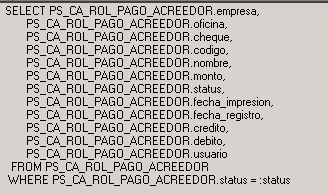
****

#### Opción del RadioButton “Cheques a IBS”.

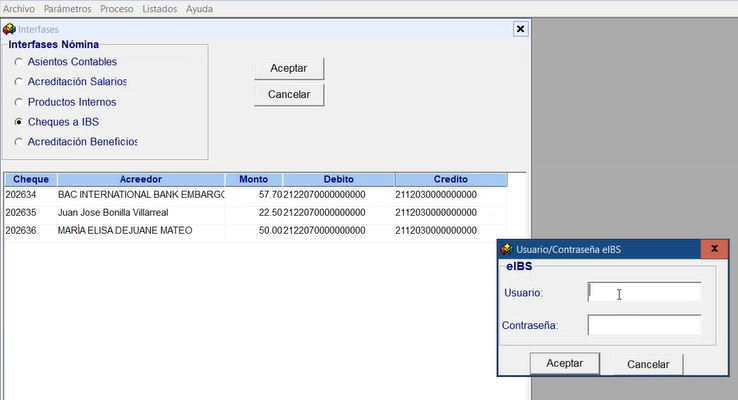


Para esta interfaz, la tablas de SQL Server con la información de los cheques llamada RRHH..PS\_CA\_ROL\_PAGO\_ACREEDOR ha sido ya cargada por el proceso de impresión de cheques, es decir, con aquellos cheques que ya han sido impresos y están pendientes por contabilizar en el Sistema eIBS.

Al seleccionar la opción del RadioButton “Cheques a IBS, se activa un evento clicked que básicamente inicializa la variable “ll\_opcion = 5”, setea el datawindow de cheques a IBS llamado “dw\_1 = dw\_interfaz\_contable para recibir información de la base de datos y llenar el grid respectivo, con el siguiente sql embebido, con valor de status = ‘I’ (impreso) :



Seguidamente al darle click al botón “Aceptar”, se ejecuta la función “ue\_aceptar” (sin parámetros) que a su vez llama a la función “wf\_ejecutar(ll\_opcion,valor)”, en este caso, el valor del parámetro ll\_opcion = 5 y el parámetro valor = 1, por lo que, en primer lugar se ejecuta la pantalla “w\_entrada\_ibs” para que se introduzca y valide un usuario y contraseñas del sistema eIBS.



Si la validación de usuario y contraseña resulta satisfactoria se procede a realizar una actualización en la base de datos, específicamente el campo “usuario” de la tabla RRHH..PS\_CA\_ROL\_PAGO\_ACREEDOR con el perfil con que se validó el eIBS como se ve en el siguiente extracto de la función “wf\_ejecutar()”:

Update PS\_CA\_ROL\_PAGO\_ACREEDOR

Set usuario = :ls\_user\_ibs

Where status = 'I';

If Sqlca.Sqlcode = -1 Then

ls\_error = Sqlca.SqlErrText;

RollBack;

MessageBox('Atención', 'Error al Actualizar el usuario...'+ ls\_error)

End If

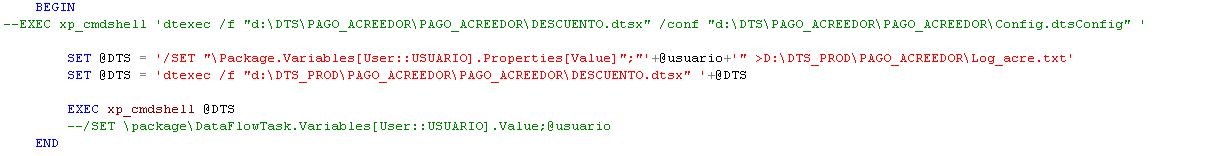
A continuación, la función ejecuta el sp ”USP­\_CA\_ROL\_PAGO\_ACREEDOR” con los parámetros de entrada

* valor = 1
* @usuario = :Sqlca.LogId USING SQLCA

El parámetro “valor=1” ejecuta un llamado a un DTS definido en la siguiente ruta del servidor:

* d:\DTS\_PROD\PAGO\_ACREEDOR\PAGO\_ACREEDOR\DESCUENTO.dtsx

Extracto del store procedure con las instrucciones del DTS:



De igual manera, si el parámetro “valor = 1” y el resultado de la ejecución del stored procedure es satisfactorio, seguidamente en la función “wf\_ejecutar()” se ejecuta una validación de que el registro de cheques haya sido satisfactorio, llamando a la función wf\_valida\_ck() (sin parámetros).

**Función wf\_valida\_ck():**



**Store procedure USP\_CA\_ROL\_PAGO\_ACREEDOR**

****

#### Opción del RadioButton “Acreditación Beneficios”.



Como se comentó previamente (punto 4.4.2), la tabla de SQL Server con la información de las acreditaciones de beneficios llamada RRHH..PS\_CA\_ROL\_ACRED\_EMPL ha sido precargada por el store procedure USP\_CA\_ACRED\_BENEFICIO a partir de la tabla de Oracle ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM cuyo tipo de pago (campo CA\_TP\_PAGO) se inicie con los caracteres ‘BE’, es decir CA\_TP\_PAGO like ‘BE%’.

Adicionalmente, se asegura que el registro de la tabla origen “ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM A” NO haya sido creado en la tabla destino “RRHH..PS\_CA\_ROL\_ACRED\_EMPL B” mediante la comparación de los siguientes campos:

99999 = B.CA\_COD\_COMP AND

B.CA\_FEC\_PAGO = A.CA\_FEC\_PAGO AND

B.CA\_MONTO\_PG = A.CA\_MONTO\_PG AND

B.CA\_CTA\_ACRE = A.CA\_CTA\_ACRE AND

B.CA\_NUM\_EMPL = A.CA\_NUM\_EMPL AND

B.CA\_TP\_PAGO = A.CA\_TP\_PAGO

Lo anteriormente dicho lo podemos observar en el siguiente extracto del stored procedure USP\_CA\_ACRED\_BENEFICIO, donde el parámetro que recibe en el campo @opción es igual a cero (0) para que realice solamente la carga desde las tablas de Oracle:

IF @opcion = 0

BEGIN

INSERT dbo.PS\_CA\_ROL\_ACRED\_EMPL

SELECT 99999,

CA\_FEC\_PAGO,

CA\_MONTO\_PG,

CA\_CTA\_ACRE,

CA\_NUM\_EMPL,

CA\_TP\_PAGO,

CA\_ESTATUS,'',NULL,NULL,NULL,NULL,NULL,NULL

FROM ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM A

WHERE CA\_TP\_PAGO not like 'BE%'

AND NOT EXISTS (SELECT 1 FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE 99999 = B.CA\_COD\_COMP AND

B.CA\_FEC\_PAGO = A.CA\_FEC\_PAGO AND

B.CA\_MONTO\_PG = A.CA\_MONTO\_PG AND

B.CA\_CTA\_ACRE = A.CA\_CTA\_ACRE AND

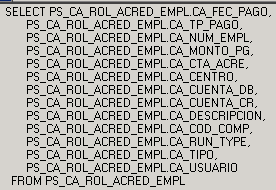
B.CA\_NUM\_EMPL = A.CA\_NUM\_EMPL AND

B.CA\_TP\_PAGO = A.CA\_TP\_PAGO)

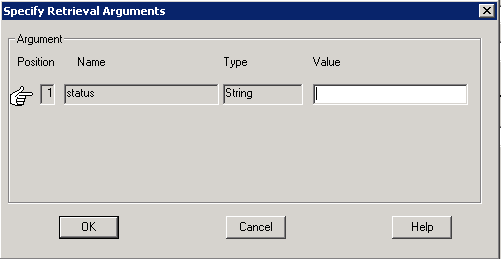
and A.CA\_FEC\_PAGO > dateadd(day,-10,GETDATE())

END

Ahora bien, una vez que aparece la pantalla de Interfaces de Nómina y se selecciona en cualquier momento la opción de RadioButton “Acreditación Beneficios”, se inicializa la variable “ll\_opcion = 4” y se setea el datawindow de acreditación llamado “dw\_1 = dw\_acreditacion\_benef” para recibir información de la base de datos y llenar el grid respectivo, con el siguiente sql embebido en el datawindow:



La carga del grid adicionalmente tiene un filtro relacionado con el campo CA\_STATUS, y el cual hace la carga a los registros con el valor de ‘P’ en dicho campo.



Seguidamente al darle click al botón “Aceptar” en la pantalla de Interfaces de Nómina con la opción del RadioButton “Acreditación Beneficios” ya seleccionada, se ejecuta la función “ue\_aceptar” (sin parámetros) que a su vez llama a la función “wf\_ejecutar(ll\_opcion,valor)”, en este caso, el valor del parámetro ll\_opcion = 4 y el parámetro valor = 1, lo que, en primer lugar se ejecuta la pantalla “w\_entrada\_ibs” para que se introduzca y valide un usuario y contraseñas del sistema eIBS.



Si la validación de usuario y contraseña de eIBS resulta satisfactoria se procede a realizar una actualización en la base de datos, en la que básicamente se graba el perfil de usuario con el que se conectó al eIBS en el campo “ca\_usuario” de la tabla RRHH..PS\_CA\_ROL\_ACRED\_EMPL para cada uno de los registros cargados en pantalla, como se ve en el siguiente extracto de la función “wf\_ejecutar()”:

If ai\_valor = 1 Then

OpenWithParm(w\_entrada\_ibs,1)

ls\_user\_ibs = Message.StringParm

If ls\_user\_ibs = '\*' Then

MessageBox('Atención','Usuario de IBS No Autorizado')

Return -1

End If

For ll\_row = 1 to dw\_1.RowCount()

dw\_1.SetItem(ll\_row,'ca\_usuario',ls\_user\_ibs)

Next

dw\_1.AcceptText()

If dw\_1.Update()= -1 Then

ls\_error = Sqlca.SqlErrText;

RollBack;

MessageBox('Atención', 'Error al Actualizar Acreditación de

Beneficios...'+ ls\_error)

Return -1

End If

End If

A continuación, la función ejecuta el sp ”USP­\_CA\_ACRED\_BENEFICIO con los parámetros de entrada

* valor = 1
* @usuario = :Sqlca.LogId USING SQLCA

El parámetro “valor=1” ejecuta dentro del stored procedure una serie de operaciones a la base de datos SQL Server como a DB2/400 , como se muestra en el siguiente extracto:

BEGIN

--DECLARE @DTS VARCHAR(200),@lote decimal(5,0),@centro decimal(4,0),@banco varchar(2),@sec int,@usuario\_ibs varchar(20)

DELETE FROM dbo.WKINPUT

SELECT @usuario\_ibs = MAX(CA\_USUARIO)

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE CA\_ESTATUS='P'

AND CA\_TP\_PAGO like 'BE%'

set @usuario\_ibs=upper(@usuario\_ibs)

select @lote =BTHDIB ,@centro=BTHUBR,@banco=BTHUBK

from INTERFACES\_BENEFCO.TSILPRU1.CAHCYFILES.CNTRLBTH

WHERE BTHUSR=@usuario\_ibs

/\*\*\*\*\*\*\*\*\*\*\*\*\*DEBITO\*\*\*\*\*\*\*\*\*\*\*/

INSERT dbo.WKINPUT

SELECT 'T' as WINSTA, 0 AS WINRTE, @banco AS WINOBK, @centro AS WINOBR,

'01' AS WINBNK, @centro AS WINBRN, 'USD' AS WINCCY, CA\_CUENTA\_DB AS WINGLN,

0 AS ASWINACC, 'DR ' AS WINTCD, 0 AS WINVDM, 0 AS WINVDD, 0 AS WINVDY,

'BENEFICIOS ASISTENCIA ECONOMIC' AS WINTDS, CA\_MONTO\_PG AS WINDRA,

0 AS WINCR1, 0 AS WINUN1, 0 AS WINCR2, 0 AS WINUN2, 0 AS WINCR3,

0 AS WINUN3, 0 AS WINCR4, 0 AS WINUN4, 0 AS WINMDM, 0 AS WINMDD,

0 AS WINMDY, @lote AS WINKBT,

ROW\_NUMBER() OVER(ORDER BY CA\_NUM\_EMPL DESC) AS WINKSQ,

0 AS WINPYT,@centro AS WINCCN, 1 AS WINFER, '' AS WINMEF, 0 AS WINNID, 0 AS WINTRN,

0 AS WINRTN, month(GETDATE()) AS WINBDM, day(GETDATE()) AS WINBDD,

year(GETDATE()) AS WINBDY, 0 AS WINTTR, @usuario\_ibs AS WINUSO, '' AS WINARF,

'' AS WINMBT, 0 AS WINCUN, 0 AS WINCKN, '' AS WINSVF, 0 AS WINDRR,

CA\_MONTO\_PG AS WINEQV, 0 AS WINACR, 0 AS WINCNU, 0 AS WINCST, 0 AS WINDSQ,

0 AS WINIVP, 0 AS WINIVB

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE CA\_ESTATUS='P'

AND CA\_TP\_PAGO like 'BE%'

select @sec=COUNT(1)

from dbo.PS\_CA\_ROL\_ACRED\_EMPL

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%'

/\*\*\*\*\*\*\*\*CREDITO\*\*\*\*\*\*\*\*\*\*/

INSERT dbo.WKINPUT

SELECT 'T'as WINSTA,0 AS WINRTE,ACMBNK AS WINOBK,@centro AS WINOBR,'01' AS WINBNK, ACMBRN AS WINBRN, 'USD' AS WINCCY, ACMGLN AS WINGLN, CA\_CTA\_ACRE AS WINACC,

'CR ' AS WINTCD, 0 AS WINVDM, 0 AS WINVDD, 0 AS WINVDY,

'BENEFICIOS ASISTENCIA ECONOMIC' AS WINTDS, 0 AS WINDRA, CA\_MONTO\_PG AS WINCR1,

0 AS WINUN1, 0 AS WINCR2, 0 AS WINUN2, 0 AS WINCR3, 0 AS WINUN3, 0 AS WINCR4,

0 AS WINUN4, 0 AS WINMDM, 0 AS WINMDD, 0 AS WINMDY,

@lote AS WINKBT,((ROW\_NUMBER() OVER(ORDER BY CA\_NUM\_EMPL DESC)) + @sec) AS WINKSQ,

0 AS WINPYT,ACMBRN AS WINCCN, 1 AS WINFER, '' AS WINMEF, 0 AS WINNID, 0 AS WINTRN,

0 AS WINRTN, month(GETDATE()) AS WINBDM, day(GETDATE()) AS WINBDD,

year(GETDATE()) AS WINBDY, 0 AS WINTTR, @usuario\_ibs AS WINUSO, '' AS WINARF,

'' AS WINMBT, ACMCUN AS WINCUN, 0 AS WINCKN, '' AS WINSVF, 0 AS WINDRR,

CA\_MONTO\_PG AS WINEQV, 0 AS WINACR, 0 AS WINCNU, 0 AS WINCST, 0 AS WINDSQ,

0 AS WINIVP,0 AS WINIVB

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B,(SELECT \* FROM OPENQUERY(INTERFACES\_BENEFCO,

'SELECT ACMACC,ACMBNK,ACMBRN,ACMGLN,ACMCUN FROM CAHCYFILES.ACMST,CAHCYFILES.CUMST

WHERE ACMCUN=CUSCUN AND (CUSSTF =''3'' OR ACMUC4=''0004'') ')) AS ACMST

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%' AND ACMACC = CA\_CTA\_ACRE

set @sec =@sec \* 2

UPDATE WKINPUT

SET WINSVF = 'Y'

WHERE WINKSQ = @sec

INSERT dbo.WKINPUT

SELECT 'B'as WINSTA, 0 AS WINRTE, @banco AS WINOBK, @centro AS WINBRN, '01' AS WINBNK,

@centro AS WINOBR, 'USD' AS WINCCY, 0 AS WINGLN,0 AS ASWINACC,'' AS WINTCD,

0 AS WINVDM, 0 AS WINVDD, 0 AS WINVDY, '' AS WINTDS, SUM(CA\_MONTO\_PG) AS WINDRA, SUM(CA\_MONTO\_PG) AS WINCR1,0 AS WINUN1,

0 AS WINCR2, 0 AS WINUN2, 0 AS WINCR3, 0 AS WINUN3, 0 AS WINCR4,0 AS WINUN4,

0 AS WINMDM, 0 AS WINMDD, 0 AS WINMDY, @lote AS WINKBT, 0 AS WINKSQ,

0 AS WINPYT, 0 AS WINCCN, 1 AS WINFER, '' AS WINMEF, 0 AS WINNID, 0 AS WINTRN,

0 AS WINRTN, month(GETDATE()) AS WINBDM, day(GETDATE()) AS WINBDD,

year(GETDATE()) AS WINBDY, @sec AS WINTTR, @usuario\_ibs AS WINUSO,'A' AS WINARF,

'Y' AS WINMBT, 0 AS WINCUN, 0 AS WINCKN, 'Y' AS WINSVF, 0 AS WINDRR,

0.00 AS WINEQV, 0 AS WINACR, 0 AS WINCNU,0 AS WINCST,0 AS WINDSQ,0 AS WINIVP,

0 AS WINIVB

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%'

GROUP BY CA\_COD\_COMP

INSERT INTERFACES\_BENEFCO.TSILPRU1.CAHCYFILES.WKINPUT(WINSTA,WINRTE,WINOBK,WINOBR,

WINBNK,WINBRN,WINCCY,WINGLN,WINACC,WINTCD,WINVDM,WINVDD,WINVDY,WINTDS, WINDRA,WINCR1,WINUN1,WINCR2,WINUN2,WINCR3,WINUN3,WINCR4,WINUN4,WINMDM,

WINMDD,WINMDY,WINKBT,WINKSQ, WINPYT,WINCCN,WINFER,WINMEF,WINNID,WINTRN,

WINRTN,WINBDM, WINBDD,WINBDY,WINTTR,WINUSO,WINARF,WINMBT,WINCUN,

WINCKN,WINSVF,WINDRR,WINEQV,WINACR,WINCNU,WINCST,WINDSQ,WINIVP,WINIVB)

SELECT \* FROM dbo.WKINPUT

---Actualizar Estatus de los registros

UPDATE dbo.PS\_CA\_ROL\_ACRED\_EMPL

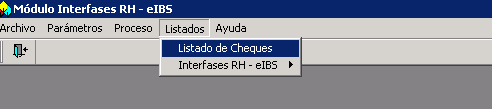
SET CA\_ESTATUS = 'A'

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%'

END

Como se puede observar al final del stored procedure, se actualiza el estatus de los registros procesados de la tabla RRHH..PS\_CA\_ROL\_ACRED\_EMPL asignándole “CA\_STATUS=A”.

## Opción Listados

 El menú de listados tiene dos submenús de “Listado de Cheques” e “Interfases de RH – eIBS” dentro del sub menú anterior existe un sub menú “Acreditación”, vamos a empezar con la primera opción.

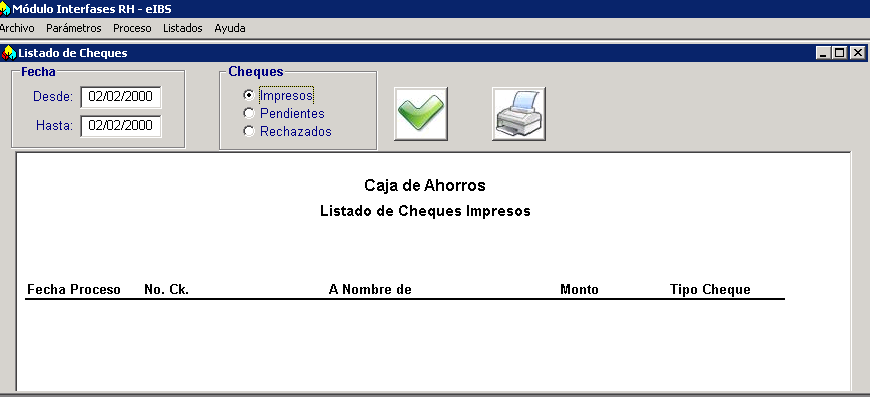
### Listado de cheques

Esta interfaz filtra los cheques por medio de un rango de fechas desde y hasta y el estado de cheques pendientes, impresos y rechazados con el botón con un visto verde ejecuta la consulta, el formulario se llama w\_list\_cheques, consta de dos group box llamados “Fecha” y “Cheques”. **Fecha**. Tiene el filtro de fecha inicio y fin.

**Cheques**.

Tiene 3 opciones de Radio Button:

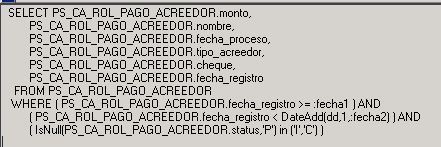
* Impresos
* Pendientes
* Rechazados



#### RadioButtom Impresos.

Sí selecciona el radio button “Impresos” se muestra el datawindow “dw\_list\_ck\_impresos” y carga los valores por default de la plantilla en blanco como se muestra en la pantalla previa.

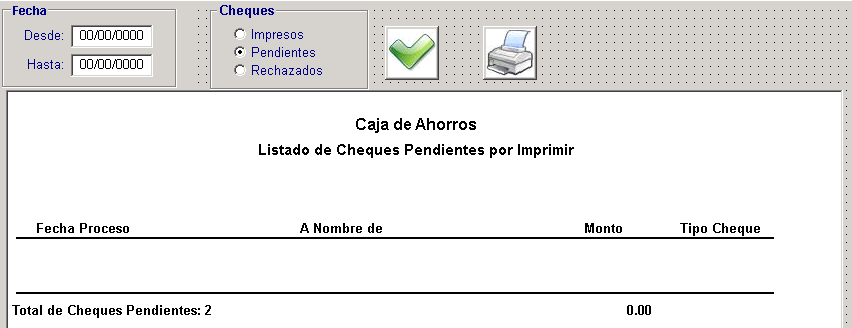
Luego si presiona el botón con la imagen de un visto verde ejecuta el filtro de la información con la fecha inicio y fin de la siguiente manera:

****

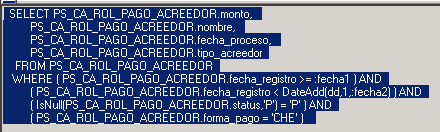
Query basado en los parámetros proporcionados por el usuario es decir las fechas inicio y fin para la consulta de cheques con estado “I” y/o “C”, si el estado es nulo por default se coloca “P” para evitar errores en caso de valores nulos. Finalmente el usuario puede imprimir el reporte en el botón de la impresora.

#### RadioButtom Pendientes.

En esta opción se carga el datawindow “dw\_list\_ck\_pend” con una sentencia doomy para armar los campos del datawindow.



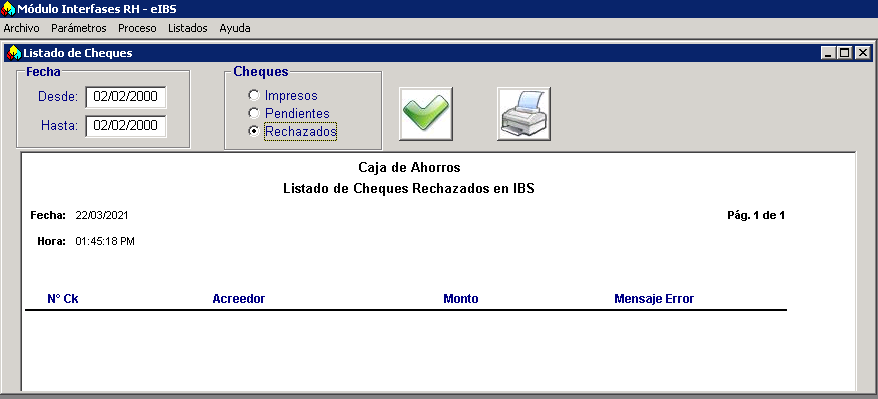
Luego de seleccionar el check de pendientes y haber colocado el rango de fechas presiona el botón con el tick verde para ejecutar el reporte el cual llama al siguiente query:



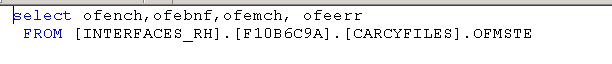
Extrae los datos de los cheques pendientes a pagar a los acreedores dentro del rango de fecha proporcionado por el usuario suma un día más a la fecha hasta para considerar el rango que envió el usuario y filtra por estados pendientes y que la forma de pago sea cheque.

#### Radio Buttom Rechazados.

Aquí se redibuja el datacontrol “dw\_err\_ck\_ibs” con los filtros de fecha para preparar antes de presionar el botón con el tick verde.



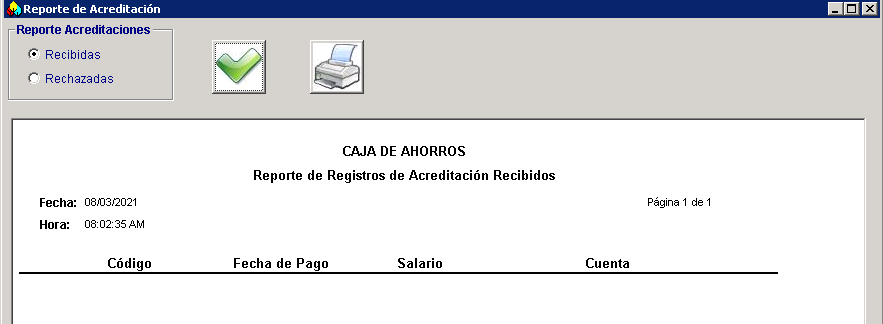
Si desea ejecutar la transacción de consulta de listado de cheques rechazados en IBS al presionar el botón con el icono de un visto verde ejecuta una consulta a base de datos en el servidor tsiapp209 de SQL Server el cual ejecuta el filtro de por fecha por estado, utilizando un linked server llamado “INTERFASES\_RH” que se re direcciona hacia un servidor de AS400 A LA BASE DE DATOS DB2/400 con el siguiente Query:



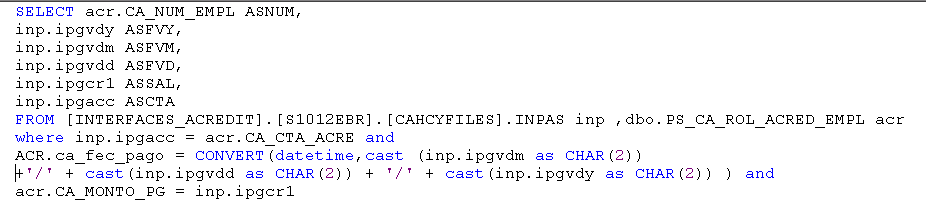
### Interfaces RH – eIBS – Listado de Acreditación

### 

En esta opción por default hace la carga de los reportes de acreditaciones recibidas donde se le asigna al datawindow el dataobject “dw\_rep\_acreditacion”

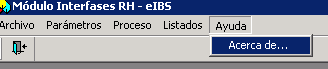


Aquí se puede visualizar las acreditaciones realizadas correctamente o rechazados mediante una conexión a AS400, al momento utiliza el siguiente query como filtro de las consultas:



## Opción Ayuda

En la opción ayuda tenemos el sub menú acerca de:



Aquí muestra la información de la aplicación.

# COMPONENTES DE LA APLICACIÓN DE RRHH

## Windows Interfaces

Existen 16 pantallas dentro del módulo de RRHH que son los siguientes:

* w\_cacheques\_about
* w\_cacheques\_sheet1
* w\_cacheques\_sheet2
* w\_cacheques\_sheet3
* w\_cacheques\_toolbars
* w\_cheques
* w\_dw\_print\_options
* w\_entrada
* w\_entrada\_ibs
* w\_interfaces
* w\_list\_cheques
* w\_parametro
* w\_rep\_acreditaciones
* w\_tombola\_tab

## Datawindows

Existen 20 DATAWIDOWS que se encargan de mostrar la información de las estructuras de las base de datos por sql o store procedures intercalados entre SQL Server o DB2/400 de AS/400.

* dw\_acreditacion
* dw\_acreditacion\_benef
* dw\_asiento\_contable
* dw\_cheques
* dw\_dd\_tipos
* dw\_detalle\_ac
* dw\_err\_ck\_ibs
* dw\_interfaz\_contable
* dw\_list\_ck\_pend
* dw\_listado\_cheques\_acreedores
* dw\_master\_ac
* dw\_param\_fecha\_ck
* dw\_param\_fecha\_sa
* dw\_param\_ps
* dw\_productos\_ca
* dw\_rep\_acreditacion
* dw\_rep\_ck\_impreso
* dw\_rep\_ck\_pendientes
* dw\_rep\_rechazadas
* dw\_tarjetas\_ca

## Menu Frame

Está compuesto de 5 menús principales sobre el componente frame menú :

* Archivo
* Parámetros
* Proceso
* Listados
* Ventana
* Ayuda

## Objeto de Conexión

Este objeto realiza una lectura del archivo de conexión del ini para posteriormente pasar los valores a la cadena de conexión hacia las bases de datos.

# RESUMEN DE STORED PROCEDURES

Esta aplicación utiliza los siguientes stored procedures:

* USP\_CA\_ROL\_PAGO\_ACREEDOR: Proceso de Impresión de Cheque e Interface de Cheques a IBS.
* USP\_CA\_ACREDITACION: Interface de Acreditación Salarios.
* USP\_CA\_ACRED\_BENEFICIO: Interface de Acreditación Beneficios.
* USP\_CA\_ASIENTO: Interface de Asientos Contables.
* USP\_CA\_ROL\_DESCUENTO: Interface de Productos Internos.

## USP\_ROL\_PAGO\_ACREEDOR

Es utilizado tanto por el proceso de Impresión de Cheques como la Interface de Cheques a IBS

Este stored procedure recibe dos parámetros de entrada: @opción y @usuario.

Sí recibe el valor de “opción =0” se procede a volcar la información de una tabla de Oracle nombrada “PS\_CA\_ROL\_PAGO\_ACR” de la instancia “SYSADM” mediante el linked server “ORACLE” que es alimentada de una interfaz del “Sistema PeopleSoft” hacia la tabla de SQL Server “PS\_CA\_ROL\_PAGO\_ACREEDOR de la base de datos RRHH.

El filtro principal para la obtención de la información radica en que el registro no exista en la tabla destino, comparando los siguientes campos (tabla ORACLE..SYSADM.PS\_CA\_ROL\_PAGO\_ACR A vs. RRHH..PS\_CA\_ROL\_PAGO\_ACREEDOR B):

A.CA\_EMPRESA = B.empresa AND

A.CA\_OFICINA = B.oficina AND

A.RECIPIENT\_ID = B.CODIGO AND

A.CA\_MONTO = B.monto AND

convert(datetime,A.DATE\_POSTED) = convert(datetime,B.fecha\_registro)) and convert(datetime,A.DATE\_POSTED) <> '2019-12-17 00:00:00'

Por otro lado, si el parámetro @opcion es diferente de cero ejecuta un DTS, denominado DESCUENTO.dtsx, utilizando una variable @dts para armar todo el string a ejecutarse desde una línea de comandos mediante un xmdshell utilizando el usuario enviado desde el stored procedure @usuario de la siguiente manera:

SET @DTS = '/SET "\Package.Variables[User::USUARIO].Properties[Value]";"'+@usuario+'" >D:\DTS\_PROD\PAGO\_ACREEDOR\Log\_acre.txt'

SET @DTS = 'dtexec /f "d:\DTS\_PROD\PAGO\_ACREEDOR\PAGO\_ACREEDOR\DESCUENTO.dtsx" '+@DTS

EXEC xp\_cmdshell @DTS



## USP\_CA\_ACREDITACION

Este stored procedure recibe dos parámetros de entrada @opción y @usuario.

Sí le recibe el valor de “opción =0” se procede a volcar la información de una tabla de Oracle nombrada “PS\_CA\_ACRED\_EM” de la instancia “SYSADM” mediante el linked server “ORACLE” que es alimentada de una interfaz del “Sistema PeopleSoft” hacia la tabla de SQL Server “PS\_CA\_ROL\_ACRED\_EMPL de la base de datos RRHH.

El filtro principal para la obtención de la información es que el campo de tipo de pago “CA\_TP\_PAGO” NO comience con los caracteres ‘BE’.

Adicionalmente, se asegura que el registro de la tabla origen “ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM A” NO haya sido creado en la tabla destino “RRHH..PS\_CA\_ROL\_ACRED\_EMPL B” mediante la comparación de los siguientes campos:

99999 = B.CA\_COD\_COMP AND

B.CA\_FEC\_PAGO = A.CA\_FEC\_PAGO AND

B.CA\_MONTO\_PG = A.CA\_MONTO\_PG AND

B.CA\_CTA\_ACRE = A.CA\_CTA\_ACRE AND

B.CA\_NUM\_EMPL = A.CA\_NUM\_EMPL AND

B.CA\_TP\_PAGO = A.CA\_TP\_PAGO and

A.CA\_FEC\_PAGO > dateadd(day,-10,GETDATE())

Lo anteriormente dicho lo podemos observar en el siguiente extracto del stored procedure USP\_CA\_ACREDITACION, donde el parámetro que recibe en el campo @opción es igual a cero (0) para que realice solamente la carga desde las tablas de Oracle:

IF @opcion = 0

BEGIN

INSERT dbo.PS\_CA\_ROL\_ACRED\_EMPL

SELECT 99999,

CA\_FEC\_PAGO,

CA\_MONTO\_PG,

CA\_CTA\_ACRE,

CA\_NUM\_EMPL,

CA\_TP\_PAGO,

CA\_ESTATUS,'',NULL,NULL,NULL,NULL,NULL,NULL

FROM ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM A

WHERE CA\_TP\_PAGO not like 'BE%'

AND NOT EXISTS (SELECT 1 FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE 99999 = B.CA\_COD\_COMP AND

B.CA\_FEC\_PAGO = A.CA\_FEC\_PAGO AND

B.CA\_MONTO\_PG = A.CA\_MONTO\_PG AND

B.CA\_CTA\_ACRE = A.CA\_CTA\_ACRE AND

B.CA\_NUM\_EMPL = A.CA\_NUM\_EMPL AND

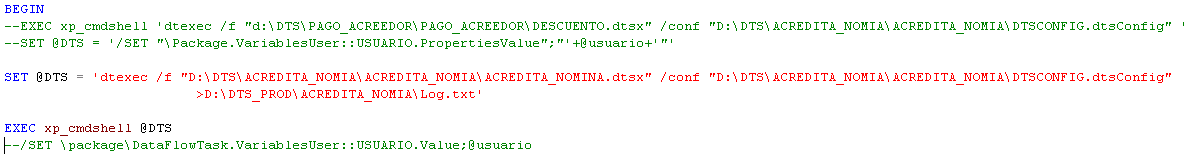
B.CA\_TP\_PAGO = A.CA\_TP\_PAGO)

and A.CA\_FEC\_PAGO > dateadd(day,-10,GETDATE())

END

Por otro lado, en caso contrario que la opción sea diferente de cero ejecuta un DTS llamado “ACREDITA\_NOMINA.dtsx” mediante un xmdshell, el dts debe estar en la ruta:

D:\DTS\ACREDITA\_NOMIA\ACREDITA\_NOMIA\ACREDITA\_NOMINA.dtsx





## USP\_CA\_ACRED\_BENEFICIO

Ahora bien, cuando la variable @opcion es diferente de 0 entonces se realiza un borrado completo de la tabla “dbo.WKINPUT”; seguidamente se saca el máximo campo “CA\_USUARIO)” de la tabla “dbo.PS\_CA\_ROL\_ACRED\_EMPL” donde el “CA\_ESTATUS = 'P'” incluso si existe estatus en nulo y “CA\_TP\_PAGO” encuentra un tipo de pago que inicie con las iniciales “BE”, luego asigno en una variable “@usuario\_ibs” el valor obtenido del campo “CA\_USUARIO” en el query que antecede y lo coloca en mayúsculas.

A continuación, se asigna las variables @lote=BTHDIB, @centro=BTHUBR, @banco=BTHUBK de la instancia “CAHCYFILES.CNTRLBTH“ que está en un servidor AS/400 de una base de datos DB2/400 mediante un linked server llamado “INTERFACES\_BENEFCO” donde el usuario sea igual al previamente extraído.

Después se procede a insertar dos registros uno para débito y otro para crédito en la tabla local de “SQLSERVER dbo.WKINPUT” finalmente se vuelva la tabla capturada con los débitos y créditos a la tabla de DB2/400 del sistema eIBS “INTERFACES\_BENEFCO.TSILPRU1.CAHCYFILES.WKINPUT”.

BEGIN

--DECLARE @DTS VARCHAR(200),@lote decimal(5,0),@centro decimal(4,0),@banco varchar(2),@sec int,@usuario\_ibs varchar(20)

DELETE FROM dbo.WKINPUT

SELECT @usuario\_ibs = MAX(CA\_USUARIO)

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE CA\_ESTATUS='P'

AND CA\_TP\_PAGO like 'BE%'

set @usuario\_ibs=upper(@usuario\_ibs)

select @lote =BTHDIB ,@centro=BTHUBR,@banco=BTHUBK

from INTERFACES\_BENEFCO.TSILPRU1.CAHCYFILES.CNTRLBTH

WHERE BTHUSR=@usuario\_ibs

/\*\*\*\*\*\*\*\*\*\*\*\*\*DEBITO\*\*\*\*\*\*\*\*\*\*\*/

INSERT dbo.WKINPUT

SELECT 'T' as WINSTA, 0 AS WINRTE, @banco AS WINOBK, @centro AS WINOBR,

'01' AS WINBNK, @centro AS WINBRN, 'USD' AS WINCCY, CA\_CUENTA\_DB AS WINGLN,

0 AS ASWINACC, 'DR ' AS WINTCD, 0 AS WINVDM, 0 AS WINVDD, 0 AS WINVDY,

'BENEFICIOS ASISTENCIA ECONOMIC' AS WINTDS, CA\_MONTO\_PG AS WINDRA,

0 AS WINCR1, 0 AS WINUN1, 0 AS WINCR2, 0 AS WINUN2, 0 AS WINCR3,

0 AS WINUN3, 0 AS WINCR4, 0 AS WINUN4, 0 AS WINMDM, 0 AS WINMDD,

0 AS WINMDY, @lote AS WINKBT,

ROW\_NUMBER() OVER(ORDER BY CA\_NUM\_EMPL DESC) AS WINKSQ,

0 AS WINPYT,@centro AS WINCCN, 1 AS WINFER, '' AS WINMEF, 0 AS WINNID, 0 AS WINTRN,

0 AS WINRTN, month(GETDATE()) AS WINBDM, day(GETDATE()) AS WINBDD,

year(GETDATE()) AS WINBDY, 0 AS WINTTR, @usuario\_ibs AS WINUSO, '' AS WINARF,

'' AS WINMBT, 0 AS WINCUN, 0 AS WINCKN, '' AS WINSVF, 0 AS WINDRR,

CA\_MONTO\_PG AS WINEQV, 0 AS WINACR, 0 AS WINCNU, 0 AS WINCST, 0 AS WINDSQ,

0 AS WINIVP, 0 AS WINIVB

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE CA\_ESTATUS='P'

AND CA\_TP\_PAGO like 'BE%'

select @sec=COUNT(1)

from dbo.PS\_CA\_ROL\_ACRED\_EMPL

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%'

/\*\*\*\*\*\*\*\*CREDITO\*\*\*\*\*\*\*\*\*\*/

INSERT dbo.WKINPUT

SELECT 'T'as WINSTA,0 AS WINRTE,ACMBNK AS WINOBK,@centro AS WINOBR,'01' AS WINBNK, ACMBRN AS WINBRN, 'USD' AS WINCCY, ACMGLN AS WINGLN, CA\_CTA\_ACRE AS WINACC,

'CR ' AS WINTCD, 0 AS WINVDM, 0 AS WINVDD, 0 AS WINVDY,

'BENEFICIOS ASISTENCIA ECONOMIC' AS WINTDS, 0 AS WINDRA, CA\_MONTO\_PG AS WINCR1,

0 AS WINUN1, 0 AS WINCR2, 0 AS WINUN2, 0 AS WINCR3, 0 AS WINUN3, 0 AS WINCR4,

0 AS WINUN4, 0 AS WINMDM, 0 AS WINMDD, 0 AS WINMDY,

@lote AS WINKBT,((ROW\_NUMBER() OVER(ORDER BY CA\_NUM\_EMPL DESC)) + @sec) AS WINKSQ,

0 AS WINPYT,ACMBRN AS WINCCN, 1 AS WINFER, '' AS WINMEF, 0 AS WINNID, 0 AS WINTRN,

0 AS WINRTN, month(GETDATE()) AS WINBDM, day(GETDATE()) AS WINBDD,

year(GETDATE()) AS WINBDY, 0 AS WINTTR, @usuario\_ibs AS WINUSO, '' AS WINARF,

'' AS WINMBT, ACMCUN AS WINCUN, 0 AS WINCKN, '' AS WINSVF, 0 AS WINDRR,

CA\_MONTO\_PG AS WINEQV, 0 AS WINACR, 0 AS WINCNU, 0 AS WINCST, 0 AS WINDSQ,

0 AS WINIVP,0 AS WINIVB

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B,(SELECT \* FROM OPENQUERY(INTERFACES\_BENEFCO,

'SELECT ACMACC,ACMBNK,ACMBRN,ACMGLN,ACMCUN FROM CAHCYFILES.ACMST,CAHCYFILES.CUMST

WHERE ACMCUN=CUSCUN AND (CUSSTF =''3'' OR ACMUC4=''0004'') ')) AS ACMST

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%' AND ACMACC = CA\_CTA\_ACRE

set @sec =@sec \* 2

UPDATE WKINPUT

SET WINSVF = 'Y'

WHERE WINKSQ = @sec

INSERT dbo.WKINPUT

SELECT 'B'as WINSTA, 0 AS WINRTE, @banco AS WINOBK, @centro AS WINBRN, '01' AS WINBNK,

@centro AS WINOBR, 'USD' AS WINCCY, 0 AS WINGLN,0 AS ASWINACC,'' AS WINTCD,

0 AS WINVDM, 0 AS WINVDD, 0 AS WINVDY, '' AS WINTDS, SUM(CA\_MONTO\_PG) AS WINDRA, SUM(CA\_MONTO\_PG) AS WINCR1,0 AS WINUN1,

0 AS WINCR2, 0 AS WINUN2, 0 AS WINCR3, 0 AS WINUN3, 0 AS WINCR4,0 AS WINUN4,

0 AS WINMDM, 0 AS WINMDD, 0 AS WINMDY, @lote AS WINKBT, 0 AS WINKSQ,

0 AS WINPYT, 0 AS WINCCN, 1 AS WINFER, '' AS WINMEF, 0 AS WINNID, 0 AS WINTRN,

0 AS WINRTN, month(GETDATE()) AS WINBDM, day(GETDATE()) AS WINBDD,

year(GETDATE()) AS WINBDY, @sec AS WINTTR, @usuario\_ibs AS WINUSO,'A' AS WINARF,

'Y' AS WINMBT, 0 AS WINCUN, 0 AS WINCKN, 'Y' AS WINSVF, 0 AS WINDRR,

0.00 AS WINEQV, 0 AS WINACR, 0 AS WINCNU,0 AS WINCST,0 AS WINDSQ,0 AS WINIVP,

0 AS WINIVB

FROM dbo.PS\_CA\_ROL\_ACRED\_EMPL B

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%'

GROUP BY CA\_COD\_COMP

INSERT INTERFACES\_BENEFCO.TSILPRU1.CAHCYFILES.WKINPUT(WINSTA,WINRTE,WINOBK,WINOBR,

WINBNK,WINBRN,WINCCY,WINGLN,WINACC,WINTCD,WINVDM,WINVDD,WINVDY,WINTDS, WINDRA,WINCR1,WINUN1,WINCR2,WINUN2,WINCR3,WINUN3,WINCR4,WINUN4,WINMDM,

WINMDD,WINMDY,WINKBT,WINKSQ, WINPYT,WINCCN,WINFER,WINMEF,WINNID,WINTRN,

WINRTN,WINBDM, WINBDD,WINBDY,WINTTR,WINUSO,WINARF,WINMBT,WINCUN,

WINCKN,WINSVF,WINDRR,WINEQV,WINACR,WINCNU,WINCST,WINDSQ,WINIVP,WINIVB)

SELECT \* FROM dbo.WKINPUT

---Actualizar Estatus de los registros

UPDATE dbo.PS\_CA\_ROL\_ACRED\_EMPL

SET CA\_ESTATUS = 'A'

WHERE CA\_ESTATUS='P' AND CA\_TP\_PAGO like 'BE%'

END



## USP\_CA\_ASIENTO

Este stored procedure recibe dos parámetros de entrada @opción y @usuario.

Sí recibe el valor de “opción =0” se procede a volcar la información de dos (2) tablas de Oracle nombradas “PS\_CA\_ROL\_CN\_ASIEN” y PS\_CA\_ROL\_CN\_MOVIM de la instancia “SYSADM” mediante el linked server “ORACLE” que son alimentadas de una interfaz del “Sistema PeopleSoft” hacia las tablas de SQL Server “PS\_CA\_ROL\_CN\_ASIENTOS” y “PS\_CA\_ROL\_CN\_MOVIM” de la base de datos RRHH respectivamente.

Para ambas tablas, se asegura que los registros de las tablas origen en Oracle NO hayan sido creados en las tablas destino SQL Server, como se puede apreciar en el siguiente extracto del stored procedure donde el campo @opción es igual a cero (0) para que realice solamente la carga desde las tablas de Oracle:

IF @opcion = 0

BEGIN

INSERT dbo.PS\_CA\_ROL\_CN\_ASIENTOS

SELECT

'01',--CA\_EMPRESA,

CA\_COD\_GEOGRAFICO,

CA\_COD\_CONTAB,

'01',--CA\_COD\_COMP1,

CA\_ANIO\_TRANS,

CA\_COD\_PER,

CA\_NUM\_COMP\_EJE,

CA\_NUM\_COMP\_PER,

CA\_COD\_EMPL,

CA\_FEC\_ASIENTO,

CA\_NOM\_BENEF,

CA\_REF\_BENEF,

CA\_CONCEPTO,

CA\_STATUS\_ASI,

CA\_COD\_MODULO,

CA\_COD\_MONEDA,

CA\_COD\_VALIDA,

CA\_OBSERV,

CA\_FEC\_CONTB,

CA\_TP\_ASIENTO,

CA\_MON\_ORIGEN,

CA\_COMP\_EXT,'N',CAL\_RUN\_ID

FROM ORACLE..SYSADM.PS\_CA\_ROL\_CN\_ASIEN A

WHERE NOT EXISTS(SELECT 1 FROM dbo.PS\_CA\_ROL\_CN\_ASIENTOS B

WHERE '01' = B.CA\_EMPRESA

AND A.CA\_COD\_GEOGRAFICO = B.CA\_COD\_GEOGRAFICO

AND A.CA\_COD\_CONTAB = B.CA\_COD\_CONTAB

AND '01' = B.CA\_COD\_COMP1

AND A.CA\_ANIO\_TRANS = B.CA\_ANIO\_TRANS

AND A.CA\_COD\_PER = B.CA\_COD\_PER

-- AND A.CA\_COD\_EMPL = B.CA\_COD\_EMPL

AND A.CA\_FEC\_ASIENTO = B.CA\_FEC\_ASIENTO

AND A.CAL\_RUN\_ID = B.CAL\_RUN\_ID )

INSERT dbo.PS\_CA\_ROL\_CN\_MOVIM

SELECT '01',CA\_COD\_GEOGRAFICO,CA\_COD\_CONTAB,

'01',CA\_ANIO\_TRANS,CA\_COD\_PER,CA\_NUM\_COMP\_EJE,

CA\_NUM\_COMP\_PER,

ROW\_NUMBER() OVER(ORDER BY CA\_COD\_SEC\_MOV DESC) AS CA\_COD\_SEC\_MOV,CA\_COD\_UGEO,

CA\_PCTJE,SUBSTRING(CA\_CTR\_COSTO1,1,3),

CA\_COD\_MONEDA,CA\_COD\_CTA,CA\_CRD\_LOCAL,CA\_DEB\_LOCAL,

CA\_CRD\_DOLAR,CA\_DEB\_DOLAR,CA\_CRD\_OTRA,CA\_DEB\_OTRA,

CA\_FEC\_CONTB,CA\_COMMENT,CA\_DEBCRE,CAL\_RUN\_ID

FROM ORACLE..SYSADM.PS\_CA\_ROL\_CN\_MOVIM A

WHERE NOT EXISTS ( SELECT 1 FROM dbo.PS\_CA\_ROL\_CN\_MOVIM B

WHERE '01' = B.CA\_EMPRESA

AND A.CA\_COD\_GEOGRAFICO = B.CA\_COD\_GEOGRAFICO

AND A.CA\_COD\_CONTAB = B.CA\_COD\_CONTAB

AND '01' = B.CA\_COD\_COMP1

AND A.CA\_ANIO\_TRANS = B.CA\_ANIO\_TRANS

-- AND A.CA\_COD\_PER = B.CA\_COD\_PER

--AND A.CA\_COD\_SEC\_MOV = B.CA\_COD\_SEC\_MOV

AND A.CAL\_RUN\_ID = B.CAL\_RUN\_ID)

and A.CA\_FEC\_CONTB > dateadd(day,-10,GETDATE())

END

Por otro lado, en caso contrario que la opción sea diferente de cero ejecuta un DTS llamado “ACREDITA\_NOMINA.dtsx” mediante un xmdshell, el dts debe estar en la ruta:

* D:\DTS\_PROD\ASIENTO\_CONTABLE\ASIENTO\_CONTABLE\ASIENTO\_CONTABLE.dtsx

Extracto del store procedure con las instrucciones del DTS: 



## USP\_CA\_ROL\_DESCUENTO

Este stored procedure recibe dos parámetros de entrada @opción y @usuario.

Sí recibe el valor de “opción =0” se procede a volcar la información de dos (2) tablas de Oracle nombradas “PS\_CA\_ROL\_DCTO\_PRO” y “PS\_CA\_ROL\_DESC\_TC” de la instancia “SYSADM” mediante el linked server “ORACLE” que son alimentadas de una interfaz del “Sistema PeopleSoft” hacia las tablas de SQL Server “PS\_CA\_ROL\_DESCUEN\_PROD” y “PS\_CA\_ROL\_DESCUEN\_TC” de la base de datos RRHH respectivamente.

Para ambas tablas, se asegura que los registros de las tablas origen en Oracle NO hayan sido creados en las tablas destino SQL Server, como se puede apreciar en el siguiente extracto del stored procedure donde el campo @opción es igual a cero (0) para que realice solamente la carga desde las tablas de Oracle:

IF @opcion = 0

BEGIN

INSERT dbo.PS\_CA\_ROL\_DESCUEN\_PROD

SELECT CA\_AINO\_PROC, CA\_MES\_PROC, CA\_DIA\_PROC, CA\_AINO\_VR,

CA\_MES\_VR, CA\_DIA\_VALOR, CA\_CTR\_COSTO, CA\_NUM\_CUENTA,

CA\_MONTO\_PAG, CA\_USUARIO\_A,

CA\_TP\_PROD,'P',NULL,NAME,National\_ID

FROM ORACLE..SYSADM.PS\_CA\_ROL\_DCTO\_PRO A

WHERE NOT EXISTS(SELECT 1 FROM dbo.PS\_CA\_ROL\_DESCUEN\_PROD B

WHERE A.CA\_NUM\_CUENTA = B.CA\_NUM\_CUENTA

AND A.CA\_CTR\_COSTO = B.CA\_CTR\_COSTO

AND A.CA\_AINO\_VR = B.CA\_AINO\_VR

AND A.CA\_MES\_VR = B.CA\_MES\_VR

AND A.CA\_DIA\_VALOR = B.CA\_DIA\_VALOR)

INSERT dbo.PS\_CA\_ROL\_DESCUEN\_TC

SELECT CA\_ID\_CAJERO, CA\_MES\_TRANS, CA\_DIA\_TRANS, CA\_ANIO\_TRANS,

CA\_COD\_TRANS, CA\_CONC\_PAGO, CA\_NUM\_TARJ, CA\_DIG\_VERIF,

CA\_MONTO\_PAG, CA\_CTA\_CONT, CA\_CTR\_COSTO, CA\_ESTATUS

FROM ORACLE..SYSADM.PS\_CA\_ROL\_DESC\_TC A

WHERE NOT EXISTS(SELECT 1 FROM dbo.PS\_CA\_ROL\_DESCUEN\_TC B

WHERE A.CA\_ID\_CAJERO = B.CA\_ID\_CAJERO

AND A.CA\_MES\_TRANS = B.CA\_MES\_TRANS

AND A.CA\_DIA\_TRANS = B.CA\_DIA\_TRANS

AND A.CA\_ANIO\_TRANS = B.CA\_ANIO\_TRANS

AND A.CA\_COD\_TRANS = B.CA\_COD\_TRANS

AND A.CA\_CONC\_PAGO = B.CA\_CONC\_PAGO

AND A.CA\_NUM\_TARJ = B.CA\_NUM\_TARJ

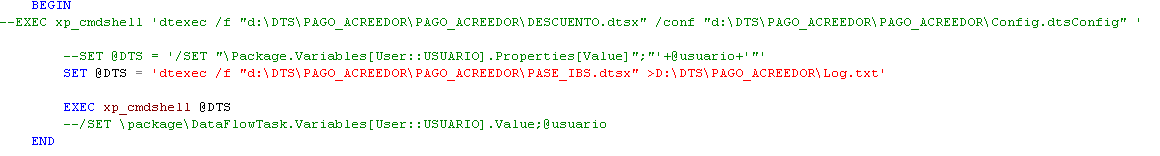
)

END

El parámetro “valor = 1” ejecuta un llamado a un DTS definido en la siguiente ruta del servidor

* "d:\DTS\_PROD\PAGO\_ACREEDOR\PAGO\_ACREEDOR\PASE\_IBS.dtsx"

Extracto del store procedure con las instrucciones del DTS:





# LINKED SERVERS

Detalle de linked servers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Server | Base de Datos | Servidor(variable por ambiente) | instancia | Store Procedure |
| ORACLE | ORACLE | TSIAP209 | SYSADM | USP\_CA\_ACRED\_BENEFICIO  USP\_CA\_ACREDITACION  USP\_CA\_ROL\_PAGO\_ACREEDOR |
| INTERFACES\_BENEFCO | AS400 | TSILPRU1 | CAHCYFILES | USP\_CA\_ACRED\_BENEFICIO |
| INTERFACES\_RH | AS400 | TSILPRU1 | CAHCYFILES | Módulo Interfaz RH |
| INTERFACES\_ACREDIT | AS400 | TSILPRU1 | CAHCYFILES | Módulo Interfaz RH |
|  |  |  |  |  |

# RIESGOS IDENTIFICADOS

Se identifica que las fuentes proporcionadas del módulo de interfaces RH (“Cubito”) no están acorde a las versiones liberadas a los usuarios, los linked server se encuentran quemados en los datawindows de la aplicación para tomar en cuenta con la migración, los ambientes o servidores no mantienen las mismas configuraciones.

# ANEXOS – Diseño Actual Módulo Interfaces RH – eIBS

