

**V1.6**

**Elaborado por : Latam Consulting**

**Junio 2021**

**ANÁLISIS DEL MÓDULO**

**RRHH Cheques**

**Desarrollado en PowerBuilder**

Contenido

[2 ANTECEDENTES 3](#_Toc78352157)

[3 ANÁLISIS BREVE DE FUNCIONALIDAD 3](#_Toc78352158)

[4 Módulo Interfases RH – eIBS 4](#_Toc78352159)

[4.1 Menú Principal ca\_cheques 5](#_Toc78352160)

[4.2 Opción Archivo 5](#_Toc78352161)

[4.3 Opción Parámetros 6](#_Toc78352162)

[4.4 Opción Proceso 7](#_Toc78352163)

[4.4.1 Impresión de Cheques 7](#_Toc78352164)

[4.4.2 Interfases RH - eIBS 12](#_Toc78352165)

[4.5 Opción Listados 32](#_Toc78352166)

[4.5.1 Listado de cheques 32](#_Toc78352167)

[4.5.2 Interfaces RH – eIBS – Listado de Acreditación 35](#_Toc78352168)

[4.6 Opción Ayuda 36](#_Toc78352169)

[5 COMPONENTES DE LA APLICACIÓN DE RRHH 36](#_Toc78352170)

[5.1 Windows Interfaces 36](#_Toc78352171)

[5.2 Datawindows 37](#_Toc78352172)

[5.3 Menu Frame 37](#_Toc78352173)

[5.4 Objecto de Conexión 38](#_Toc78352174)

[6 DOCUMENTACION DE STORE PROCEDURES 38](#_Toc78352175)

[6.1 Interfaz de parámetros(w\_parametro) 38](#_Toc78352176)

[7 LINKED SERVERS 41](#_Toc78352177)

[8 RIESGOS IDENTIFICADOS 41](#_Toc78352178)

[9 ANEXOS – Diseño Actual Módulo Interfaces RH – eIBS 42](#_Toc78352179)

# 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.

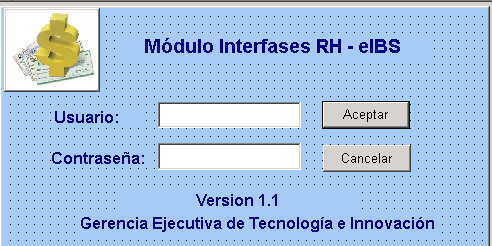
El SQL Server tiene configurados varios “linked servers” que apuntan a las respectivas base de datos ya sea ORACLE o DB2/400, quiere decir que los servidores de SQL Server deben tener instalados los driver clientes tanto de ORACLE como DB2/400 para poder realizar las conexiones respectivas; la base de 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.

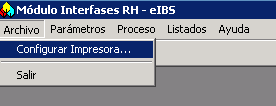
Los métodos principales programados para esta funcionalidad son:

* **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”.

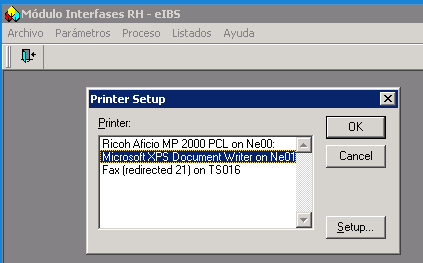
## Menú Principal ca\_cheques

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



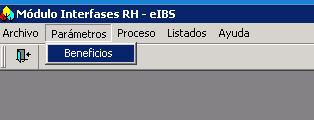
Selecciona el dispositivo a utilizarse para las impresiones locales de cheques.



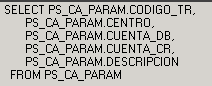
La función de PowerBuilder ejecutada se llama “PrintSetup( )”, la cual, invoca el cuadro de dialogo del sistema operativo para configuración de los dispositivos de impresión, permitiéndole al usuario que ingrese los parámetros específicos para la impresora.

## Opción Parámetros

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

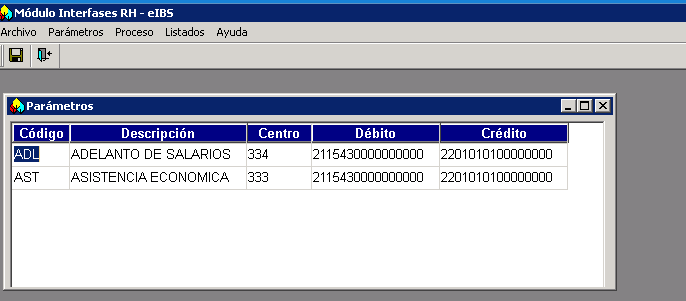


Esta opción llama al formulario ”w\_parametro” que puebla un grid con el siguiente query:



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

Visualizándose de la siguiente manera en pantalla:

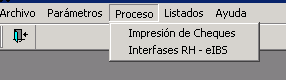


Como se puede observar es una pantalla que solamente es de uso informativo o de consulta.

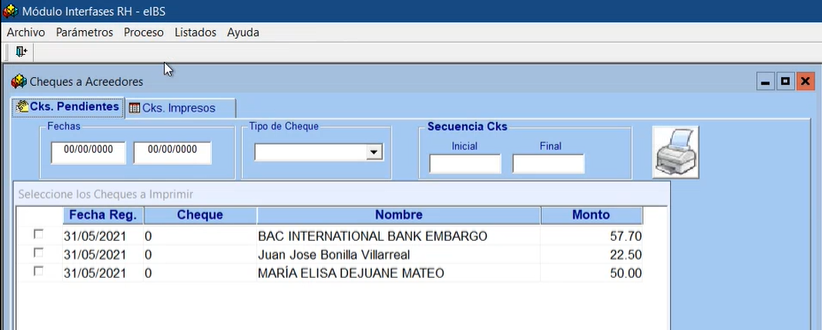
## Opción Proceso

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”.

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 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 inicialmente se despliegue el tab o pestaña de cheques pendientes “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.

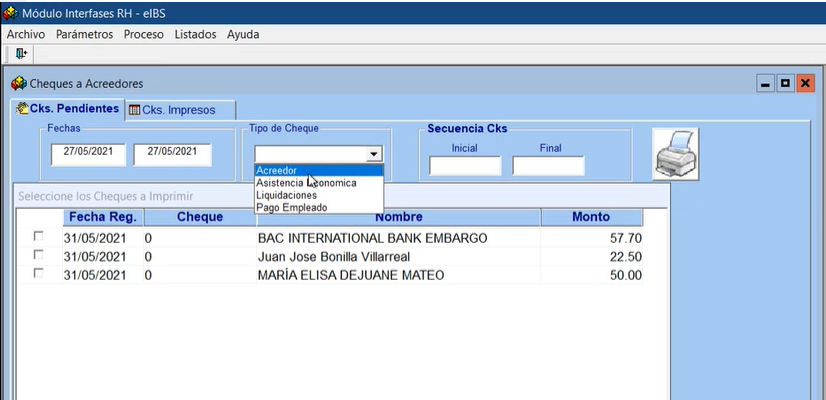
**Método open():**



**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 marcado con un checkmark previamente (lado izquierdo) y esos son los que se van a imprimir.

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:

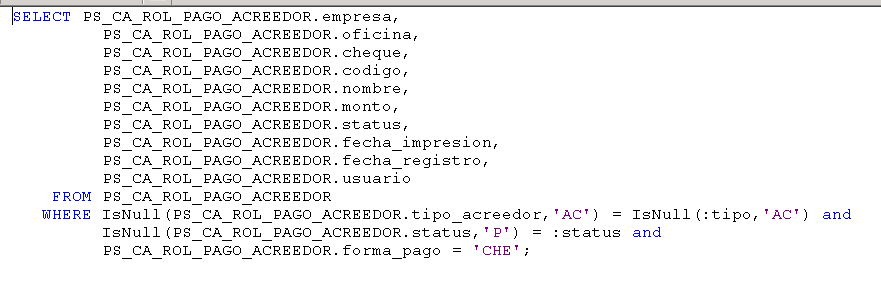
* **Acreedor**
* **Asistencia Económica**
* **Pago Empleado**
* **Liquidaciones**

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 inicializan, se jecuta un query embebido en el datawindow para cargar los cheques con estado ‘P’ y con el tipo de cheque seleccionado en el combobox y se carga el datawindow “dw\_2” del tab1 de cheques pendiente, sin embargo si viene vacío la ejecución muestra una alerta “No Existe Cheques Pendientes” concatenado la opción escogida por el usuario, ejemplo:

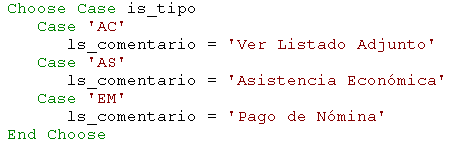
“No Existe Cheques Pendientes (“Pago Empleado”) ” y así para todas las opciones.

Cada vez que selecciona el combobox se recarga el grid con el siguiente query y los parámetros proporcionados en las condiciones previamente mencionadas.

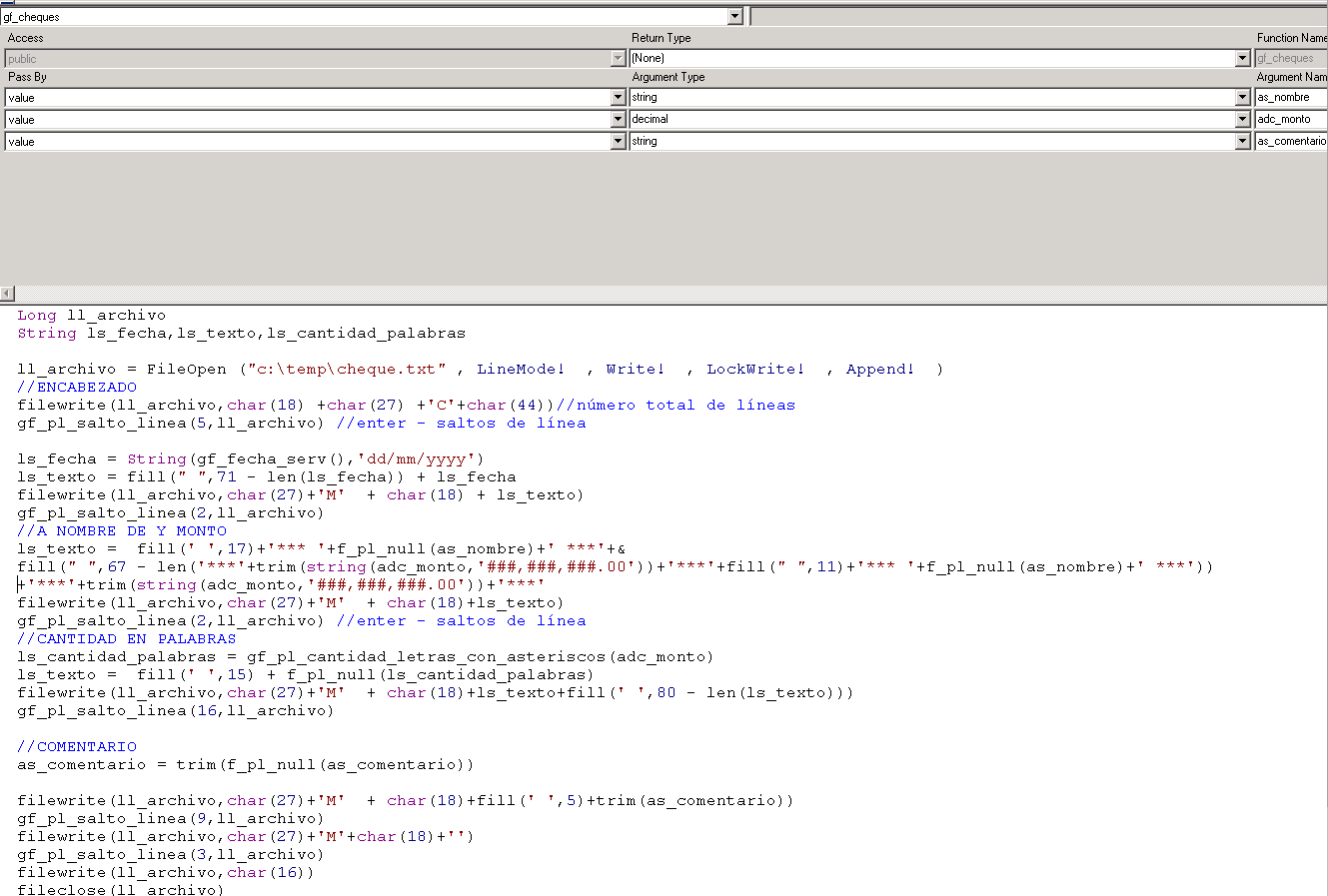


El botón de la impresora hace una carga de datos con los filtros seleccionados por los clientes y carga en el datawindow con estatus impreso “I”.

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 sqlserver utilizando una función “gf\_fecha\_serv” donde saca 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 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 selecciono en el tipo de cheque dejo asignado en una variable global is\_tipo realiza la siguiente asignación a la variable ls\_comentario:



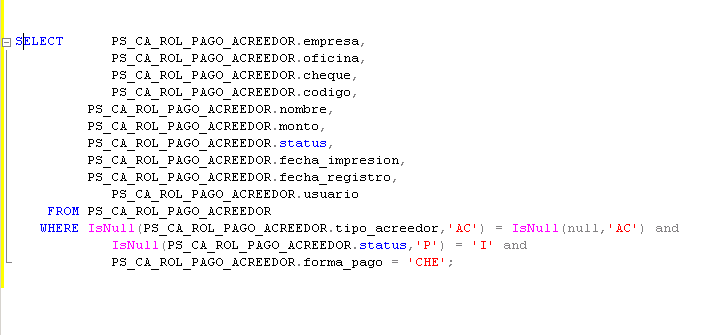
Seguido se llama a una función global “gf\_cheques” que recibe tres parámetros de entrada el nombre, el monto, y un comentario dependiendo del tipo de cheque seleccionado, esta función arma el formato del cheque a imprimirse en un archivo plano base llamado “cheque.txt” para luego enviarlo a la impresora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

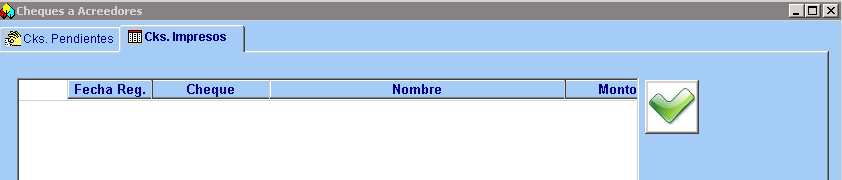


Luego de la ejecución de la función anterior se asigna a la caja de texto de la secuencia de cheques -1 “String(ldc\_num\_ck -1)” seguido se llamada la función “of\_getapppath (ls\_path,ls\_app)”

#### Pestaña o Tab Cks. Impresos

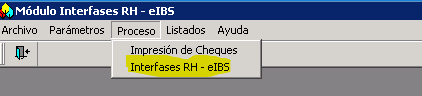
Cada tab tiene un evento si es el clickeado en el tab “chk Pendientes” se carga los datos con el estado P si es el tab “Chks Impresos” clickeado coloca el estado “I” y ejecuta el siguiente query:

****

****

**Pestaña Interfaz IBS:** Desactivada por código

### Interfases RH - eIBS

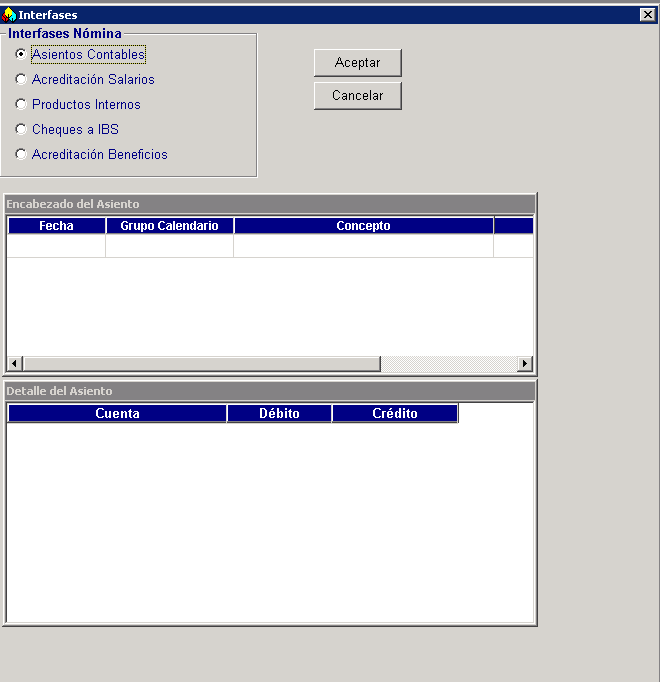


Esta opción hace uso de linkedservers tanto para Oracle como para DB2/400 por ende se debe contar con un sqlserver que tenga configurado correctamente los linkedserver que son llamados directo en el código como llamados dentro de los procedimientos almacenados.

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 radio button clickleado desde el group box.



Maneja un group box 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 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:**

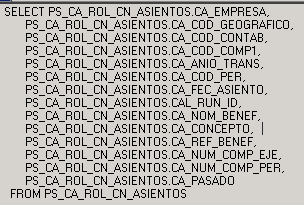
****

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

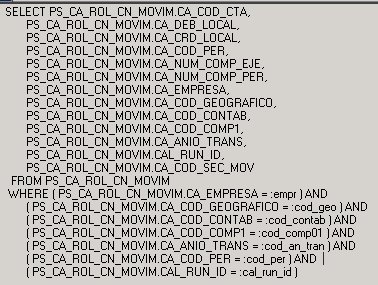
#### 

Al seleccionar esta opción se activa un evento clicked que básicamente inicializa la variable “ll\_opcion = 3”, setea los datawindows de asientos contables llamados “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 sqls embebidos en:

dw\_asiento\_contable (encabezado):



dw\_detalle\_ac (detalle):



Al darle 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

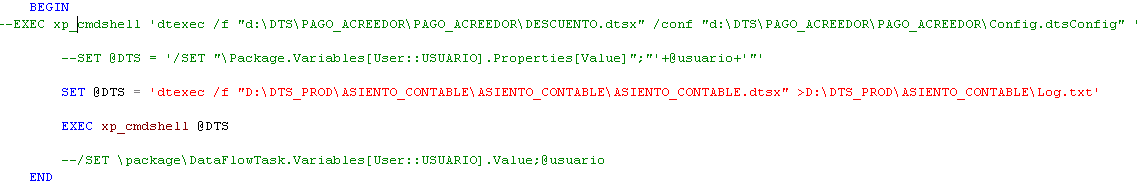
Seguidamente, se invoca al sp ”USP­\_CA\_ASIENTO” con los parámetros de entrada

* 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;

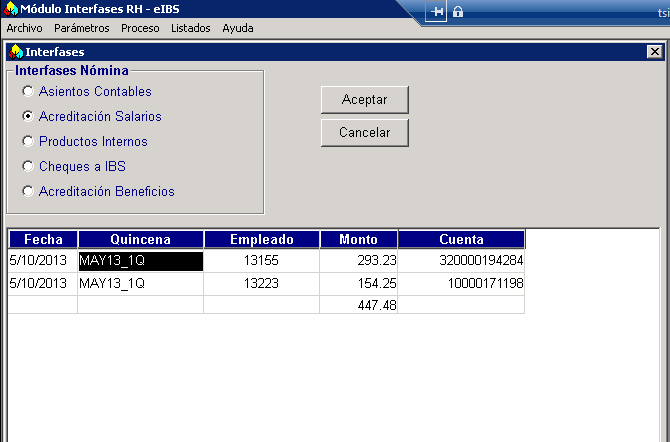
If Sqlca.SqlCode = -1 Then

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, la tabla de SQL Server RRHH..PS\_CA\_ROL\_ACRED\_EMPL ha sido precargada por el store procedure USP\_CA\_ACREDITACION a partir de la tabla de Oracle PS\_CA\_ROL\_ACRED\_EMPL cuyo tipo de pago (campo CA\_TP\_PAGO) **NO** se inicie con los caracteres ‘BE’ y que el estatus (CA\_ESTATUS) sea igual al carácter ‘P’, es decir CA\_TP\_PAGO not like ‘BE%’.

Una vez que se han cargado los registro en la tabla RRHH..PS\_CA\_ROL\_ACRED\_EMPL se procede a cambiarle en la respectiva tabla de Oracle PS\_CA\_ROL\_ACRED\_EMPL el campo que representa al estatus (CA\_ESTATUS) con el valor ‘A’

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 PPS..PS\_CA\_ROL\_ACRED\_EMPL

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

ISNULL(CA\_ESTATUS,'P') = 'P'

If @@ROWCOUNT > 0

Begin

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

SET CA\_ESTATUS = 'A'

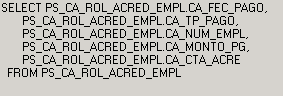
WHERE CA\_TP\_PAGO not like 'BE%' and

ISNULL(CA\_ESTATUS,'P') = 'P'

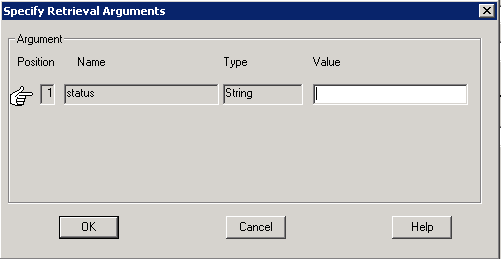
End

End

Ahora bien, una vez que aparece la pantalla de Interfaces de Nómina y se selecciona la opción de RadioButton “Acreditación Salarios”, se inicializa la variable “ll\_opcion = 1” y se setea el datawindow de acreditación llamado “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 argumento de recuperación relacionado con el campo CA\_STATUS, y el cual se le hace retrieve 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 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

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 SQLSERVER 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 l ls\_library=CAHCYFILES

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.”

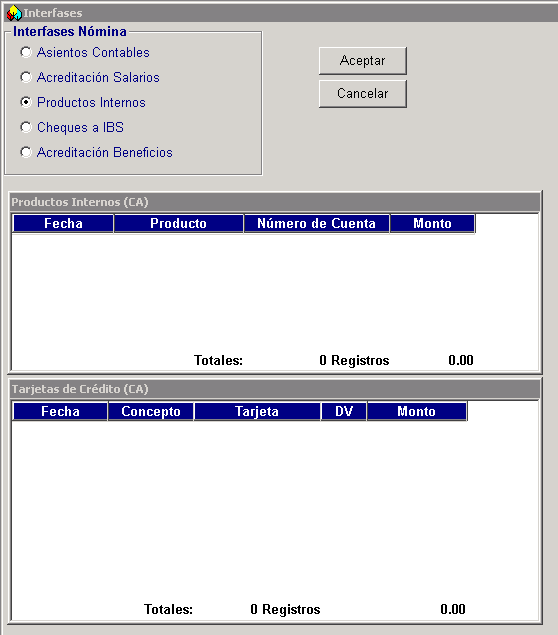
**Archivo de configuración interfaces.ini:**



**Función wf\_ejecutar():**

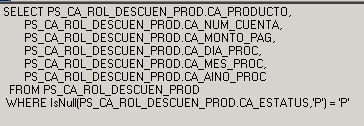
****

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

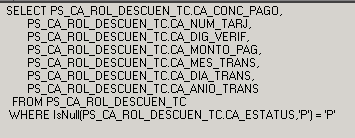


Al seleccionar esta opción se activa un evento clicked que básicamente inicializa la variable “ll\_opcion = 2”, setea los datawindows de productos internos llamados “dw\_2 = dw\_productos\_ca” y “dw\_3 = dw\_tarjetas\_ca” para recibir información de la base de datos y llenar los grids respectivos, con los siguiente sqls embebidos en:

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 resulta satisfactoria se procede a realizar una actualización en la base de datos, como se ve en el siguiente extracto de la función “wf\_ejecutar()”:

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

A continuación se invoca al sp ”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"

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 SQLSERVER base de datos RRHH, según vemos en el siguiente extracto de la función:

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';

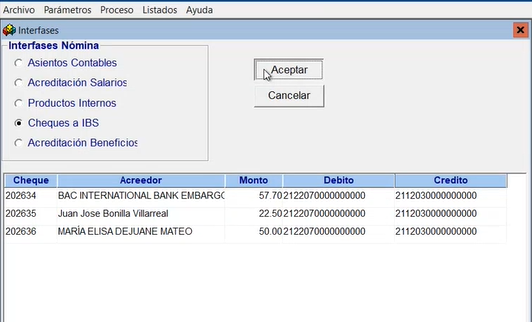
If Sqlca.SqlCode = -1 Then

ls\_error = Sqlca.SqlErrText;

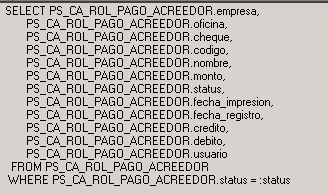
RollBack Using SQLCA;

return messageBox("Error", "Error al Actualizar Tarjetas de Crédito..." +ls\_error, StopSign!) End If

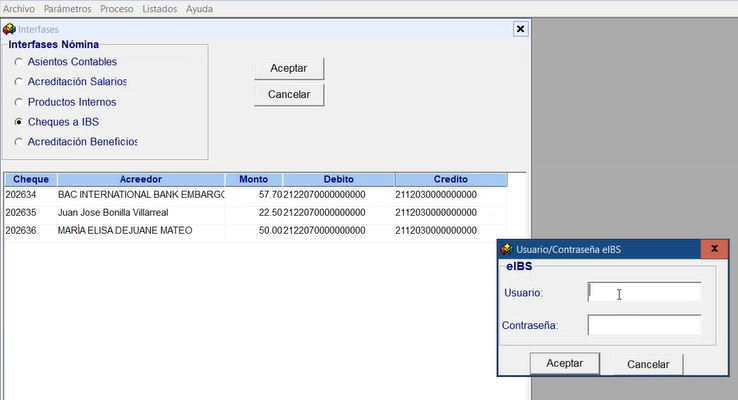
#### Opción del Radio Button “Cheques a IBS”.



Al seleccionar esta opción 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:



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, 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

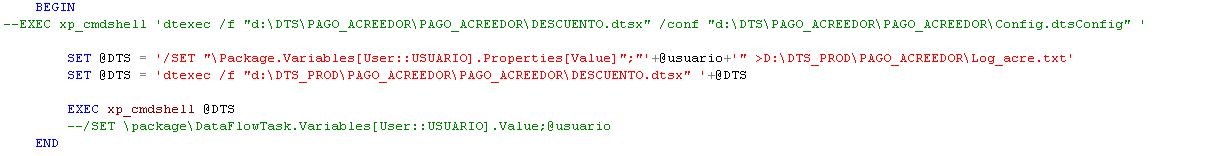
A continuación, la función ejecuta el sp ”USP­\_CA\_ROL\_PAGO\_ACREEDOR” con los parámetros de entrada

* 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():**



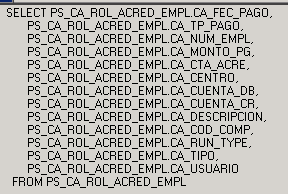
**Store procedure USP\_CA\_ROL\_PAGO\_ACREEDOR**

****

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



Al seleccionar esta opción se activa un evento clicked que básicamente inicializa la variable “ll\_opcion = 4”, setea el datawindows 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 datawindows:



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 = 4 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 aserie de instrucciones, como se ve en el siguiente extracto de la función “wf\_ejecutar()”:

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

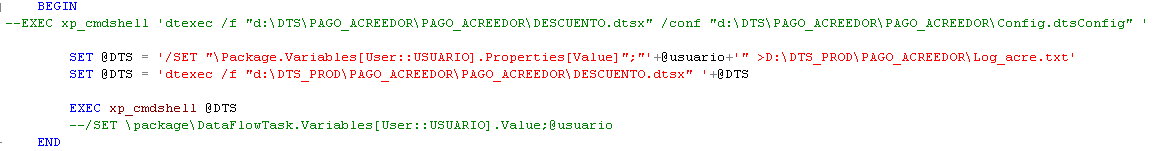
A continuación, la función ejecuta el sp ”USP­\_CA\_ACRED\_BENEFICIO” con los parámetros de entrada

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

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

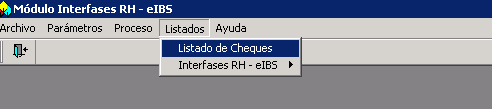
* d:\DTS\PAGO\_ACREEDOR\PAGO\_ACREEDOR\DESCUENTO.dtsx

Extracto del store procedure con las instrucciones del DTS



## 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.

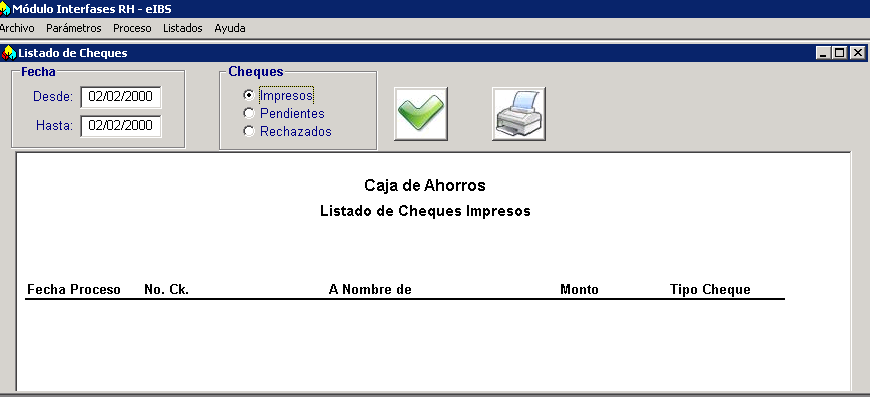


### Listado de cheques

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 radio buttons:

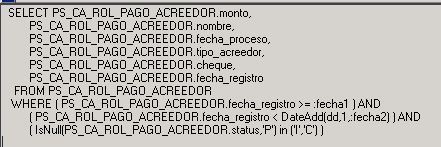
* Impresos
* Pendientes
* Rechazados



#### RadioButtom Impresos.

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

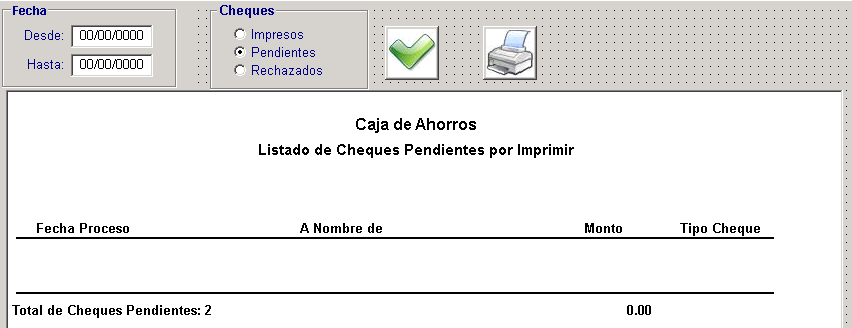
Luego si presiona el botón con la imagen de un visto verde ejecuta el filtro de la información con la fecha inicio y fin de la siguiente manera:

****

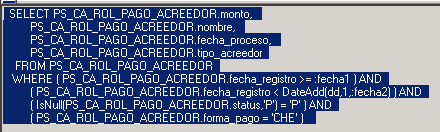
Query basado en los parámetros proporcionados por el usuario es decir las fechas inicio y fin para la consulta de cheques con estado “I” y/o “C”, si el estado es nulo por default se coloca “P” para evitar errores en caso de valores nulos. Finalmente el usuario puede imprimir el reporte en el botón de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

#### RadioButtom Pendientes.

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



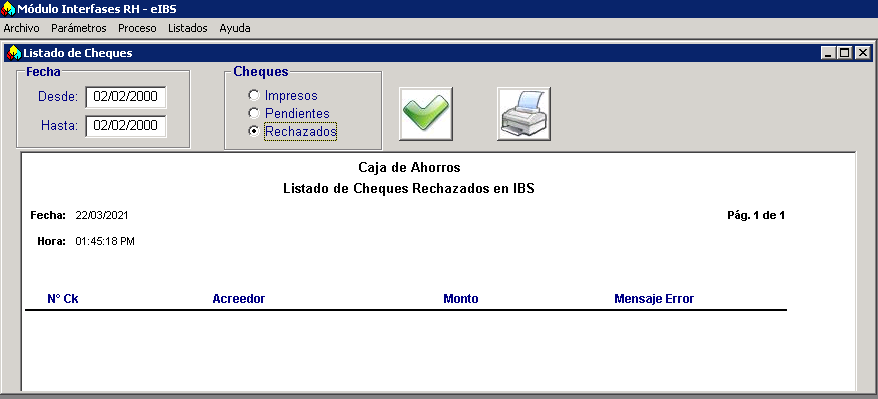
Luego de seleccionar el check de pendientes y haber colocado el rango de fechas presiona el botón con el tick verde para ejecutar el reporte el cual llama al siguiente query:



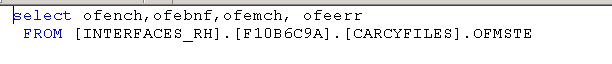
Extrae los datos de los cheques pendientes a pagar a los acreedores dentro del rango de fecha proporcionado por el usuario suma un día más a la fecha hasta para considerar el rango que envió el usuario y filtra por estados pendientes y que la forma de pago sea cheque.

#### 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.



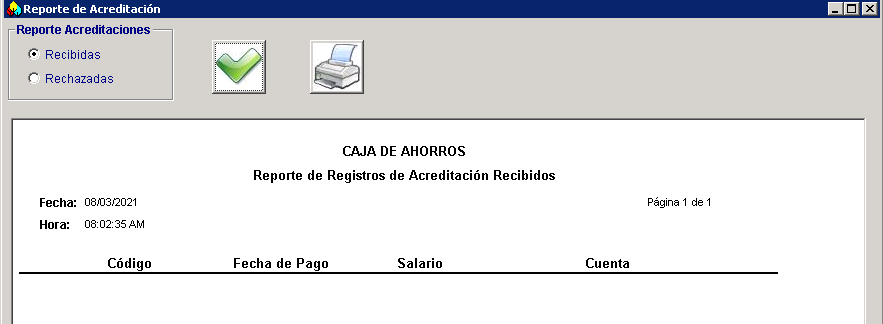
Si desea ejecutar la transacción de consulta de listado de cheques rechazados en IBS al presionar el botón con el icono de un visto verde ejecuta una consulta a base de datos en el servidor tsiapp209 de sqlserver el cual ejecuta el filtro de por fecha por estado, utilizando un linked server llamado “INTERFASES\_RH”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



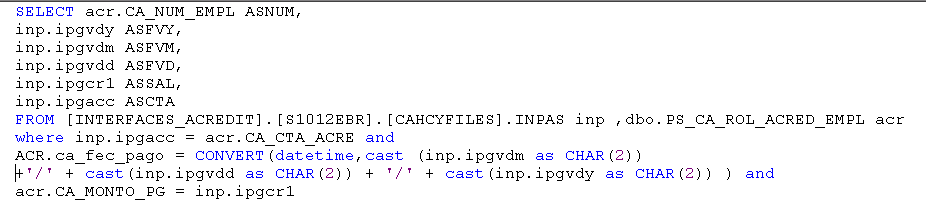
### Interfaces RH – eIBS – Listado de Acreditación

### 

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”

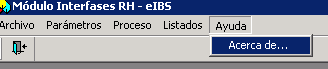


Aquí se puede visualizar las acreditaciones realizadas correctamente o rechazados mediante una conexión a AS400, al momento utiliza el siguiente query como filtro de las consultas:



## 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 Sqlserver o DB2/400 de AS400.

* 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

## Objecto 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.

# DOCUMENTACION DE STORE PROCEDURES

## Interfaz de parámetros(w\_parametro)

Esta interfaz utiliza los siguientes sps:

* USP\_CA\_ROL\_PAGO\_ACREEDOR
* USP\_CA\_ACREDITACION
* USP\_CA\_ACRED\_BENEFICIO

**USP\_CA\_ROL\_PAGO\_ACREEDOR.-**  Este sp recibe dos parámetros de entrada @opción y @usuario la opción =0 es la única programada se encarga de volvar la información de una tabla de Oracle llamada PS\_CA\_ROL\_PAGO\_ACR de la instancia “SYSADM” mediante el linkedserver “ORACLE” que es alimentada de una interfaz de peoplesoft, la información es filtrada por las siguientes condiciones:

* ISNULL(CA\_STATUS\_ASI,'P') = 'P' or
* LEN(Ltrim(Rtrim(CA\_STATUS\_ASI))) = 0

Si la condiciones expuestas anteriormente son satisfactorias y existen registros procesados al menos uno se actualiza la tabla anteriormente expuesta cambiando el estado a “A” donde el estado era “P” asi:

UPDATE ORACLE..SYSADM.PS\_CA\_ROL\_PAGO\_ACR

SET CA\_STATUS\_ASI = 'A'

WHERE ISNULL(CA\_STATUS\_ASI,'P') = 'P'

Caso contrario si el parámetro @opcion es diferente de cero ejecuta un DTS, llamado 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 sp @usuario de la siguiente manera:

SET @DTS = '/SET "\Package.Variables[User::USUARIO].Properties[Value]";"'+@usuario+'"'

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

EXEC xp\_cmdshell @DTS

**USP\_CA\_ACREDITACION.-** Este sp recibe dos parámetros de entrada @opción y @usuario la opción =0 realiza un volcado de la información de desde la estructura de Oracle ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM que es llamada mediante un linkedserver, llamado “ORACLE” las condiciones de filtro son que el tipo de pago no inicialice con “BE” y que el status sea igual a P, los campos seleccionado para la inserción en la tabla de dbo.PS\_CA\_ROL\_ACRED\_EMPL que pertenece al esquema “RRHH” SON:

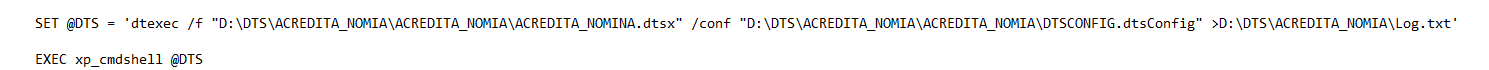
* + 99999,
  + CA\_FEC\_PAGO,
  + CA\_MONTO\_PG,
  + CA\_CTA\_ACRE,
  + CA\_NUM\_EMPL,
  + CA\_TP\_PAGO,
  + CA\_ESTATUS,'',NULL,NULL,NULL,NULL,NULL,NULL

SI encuentra algún registro actualiza el estado de la tabla **PS\_CA\_ROL\_ACRED\_EM** del servidor de Oracle peoplesoft cambiando el estado de “P” a “A” donde el tipo de pago no inicialice con las letras “BE” y el status sea igual a P.

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:

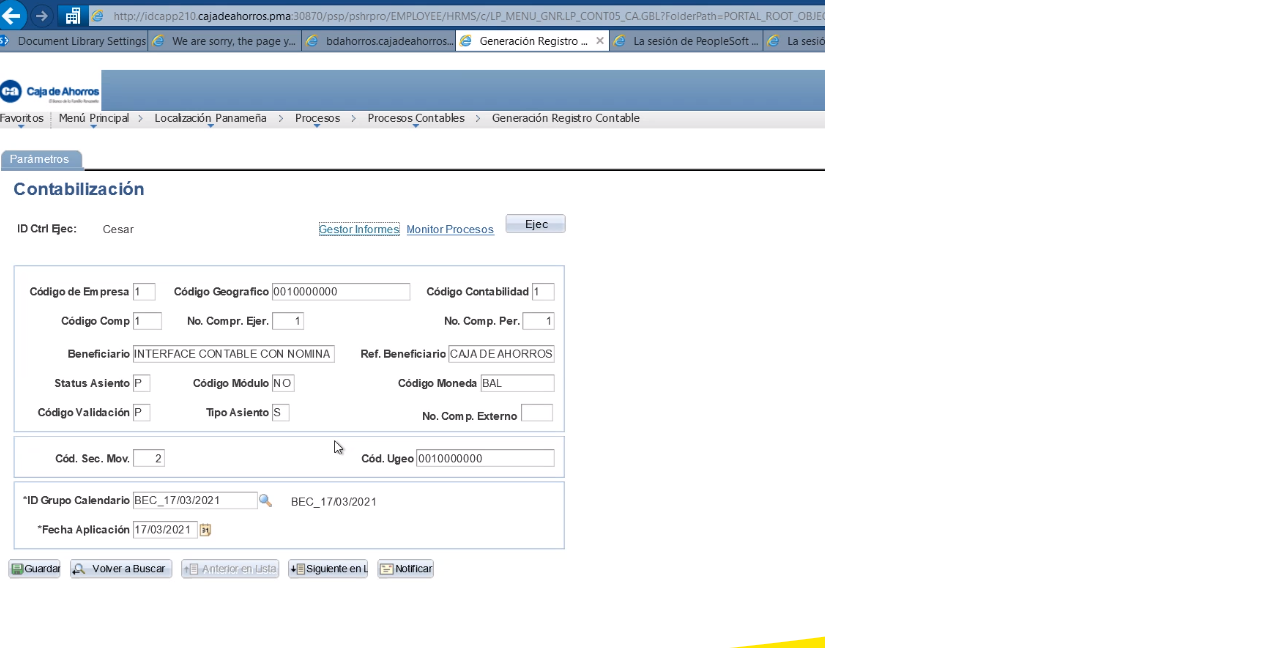
D:\DTS\ACREDITA\_NOMIA\ACREDITA\_NOMIA\ACREDITA\_NOMINA.dtsx

Ejemplo:



Pantalla que alimenta las tablas de nómina:

g



**USP\_CA\_ACRED\_BENEFICIO.-** recibe dos parámetros de entrada @opcion int, @usuario varchar(20) cuando el parámetro @opcion =0 realiza un volcado de la información de la estructura ORACLE..SYSADM.PS\_CA\_ROL\_ACRED\_EM que se encuentra en un servidor Oracle conectado desde SQL Server mediante un linked server hacia la tabla PS\_CA\_ROL\_ACRED\_EMPL de SQLSERVER bajo las condiciones de tipo de pago encuentre las iniciales “BE” y que el estado sea igual “P” incluso si el campo esta nulo los campos seleccionados son:

99999,

CA\_FEC\_PAGO,

CA\_MONTO\_PG,

CA\_CTA\_ACRE,

CA\_NUM\_EMPL,

CA\_TP\_PAGO,

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

Si se procesa al menos un registro se actualiza el estado a “A” donde el tipo de pago comience con la iniciales “BE” y el status sea igual a “P”.

Cuando la variable @opcion es diferente de 0 entonces se realiza un borrado completo de la tabla dbo.WKINPUT, seguido 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 optenido del campo CA\_USUARIO en el mquery que antecede y lo coloca en mayúsculas.

Ahora se asigna las variables @lote =BTHDIB ,@centro=BTHUBR,@banco=BTHUBK de la instancia “CAHCYFILES.CNTRLBTH “ que está en un servidor AS400 de una base de datos DB2/400 mediante un linkedserver llamado “INTERFACES\_BENEFCO” donde el usuario sea igual al previamente extraido.

Seguido 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 INTERFACES\_BENEFCO.TSILPRU1.CAHCYFILES.WKINPUT.

# LINKED SERVERS

Detalle de linkedservers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 linkedserver 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

