! Cancelacion de facturas de activos y anticipos

logout(1, ElFac, ElDFac, ElCaa, ElExa, ElCte, ElFof, ElADxC, ElCxC, ElIns, ElNsf, ElInv, ElFna, ElDFna, ElEga, ElFan, ElFanD, ElPdc, ElDpdc, ElBVi, ELHBVI, ELCHC, ELDCHC)

if errorcode()

message('No se Puede Iniciar la Transacción.|' & |

'Error Reportado: ' & error() , 'Error de Proceso', icon:hand, button:ok)

Loc:Error\_Transaccion = false

Loc:Transaccion\_Activa= false

select(?Loc:FcCanc)

cycle

end

!! completar los datos restantes de la cancelación de la factura

!1. estado fact. en: cancelada

!2. fecha de la factura

FAC:Estado= 0

FAC:FcCanc= Loc:FcCanc

FAC:Edo\_Entrega = 'Entregada'

FAC:FcCambio\_E = TODAY()

FAC:HrCambio\_E = CLOCK()

FAC:IdUsuario\_Ent = Das\_GroupUser.NetId

!!! actualizar el saldo del cliente

CTE:NCte= FAC:NCte

get(ElCte, Cte:LLP\_NCliente)

if errorcode()

Loc:Nivel\_Error= 9

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!1. actualizar los abonos del cliente

!2. grabar registro

CTE:Abonos\_Periodo+= FAC:ImpTotal

put(ElCte)

if errorcode()

Loc:Nivel\_Error= 9

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!! Si la factura es de anticipo y es de hoy, borrar pagos realizados

IF FAC:Bnd\_AntFac AND FAC:FcFactura = TODAY() THEN

!! Eliminar Pagos Virtuales

Lof:BndError# = False !! Bandera de Error

Clear(BVI:Record)

BVI:LFolio = FAC:LFolio

BVI:NFolio = FAC:NFolio

SET(BVI:PLL\_Folios,BVI:PLL\_Folios)

Loop

Next(ELBVI)

if ErrorCode() Then Break.

If BVI:LFolio <> FAC:LFolio Or |

BVI:NFolio <> FAC:NFolio Then Break.

!! Eliminar Registro

Delete(ELBVI)

if errorcode() !! Bandera de error

Lof:BndError# = True

Break

End

End

! Su hubo Error

if Lof:BndError# = True

Loc:Nivel\_Error= 22

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!! Eliminar Pagos Historico Virtuales

Lof:BndError# = False !! Bandera de Error

Clear(HBVI:Record)

HBVI:LFolio = FAC:LFolio

HBVI:NFolio = FAC:NFolio

SET(HBVI:PLL\_Folios,HBVI:PLL\_Folios)

Loop

Next(ELHBVI)

if ErrorCode() Then Break.

If HBVI:LFolio <> FAC:LFolio Or |

HBVI:NFolio <> FAC:NFolio Then Break.

!! Buscar Cheque y Detalle del Cheque

CLEAR(CHC:Record)

CHC:LFolioC = HBVI:LFolioC

CHC:NFolioC = HBVI:NFolioC

GET(ELCHC, CHC:LLP\_FolioC)

IF ~ERRORCODE() THEN

DELETE(ELCHC)

IF ERRORCODE() THEN

Lof:BndError# = True

Break

END

END

CLEAR(DCHC:Record)

DCHC:LFolioC = HBVI:LFolioC

DCHC:NFolioC = HBVI:NFolioC

GET(ELDCHC, DCHC:LLS\_FoliosC)

IF ~ERRORCODE() THEN

DELETE(ELDCHC)

IF ERRORCODE() THEN

Lof:BndError# = True

Break

END

END

!! Eliminar Registro

Delete(ELHBVI)

if errorcode()

Lof:BndError# = True

End

End

! Caso de Error

If Lof:BndError# = True

Loc:Nivel\_Error= 23

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

End

End

if FAC:Bnd\_AntFac

!! anticipo facturado

!! buscar el anticipo

clear(Fan:Record)

FAN:LFolio= FAC:LFolio

FAN:NFolio= FAC:NFolio

get(ElFan, FAN:LLP\_Folio)

if errorcode()

Loc:Nivel\_Error= 120

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

FAN:Estado= false

put(ElFan)

if errorcode()

Loc:Nivel\_Error= 20

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

if FAN:FcPago

!! anular el anticipo

clear(Cxc:Record)

CXC:LFolio= FAN:LFolio

CXC:NFolio= FAN:NFolio

get(ElCxc, CXC:LLP\_LFolio\_NFolio)

if ~errorcode()

CXC:NCtaB= 0

CXC:Saldo= 0

put(ElCxc)

if errorcode()

Loc:Nivel\_Error= 7

do Sr\_Error\_Transaccion

select(?Loc:FcCanc)

cycle

end

!! agregar el registro

clear(ADxc:Record)

ADXC:CDoc = 'NEA'

ADXC:CConcepto = 16 ! eliminación de anticipos

ADXC:LFolio = Loc:LFolio

ADXC:NFolio = Loc:NFolio

ADXC:FcDoc = Loc:FcCanc

ADXC:Referencia = 'CANCELACION FACT.: '&FAC:LFolio & ' ' &FORMAT(FAC:NFolio, @N06)

ADXC:NCte = FAN:NCte

ADXC:Importe = FAN:Importe

ADXC:ACDoc = CXC:CDoc

ADXC:ACConcepto = CXC:CConcepto

ADXC:ALFolio = FAN:LFolio

ADXC:ANFolio = FAN:NFolio

ADXC:ImpAplicar = FAN:Importe

ADXC:Aplicado = 0

ADXC:Origen = 2

ADXC:Procedencia= ' '

ADXC:FcRegistro= Today()

ADXC:IdUsuario = Das\_GroupUser.NetId

add(ElADxc)

if errorcode()

Loc:Nivel\_Error= 8

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!! agregar el registro del abono para equilibrar los movs.

clear(ADxc:Record)

ADXC:CDoc = 'NA'

ADXC:CConcepto = 16 ! eliminación de anticipos

ADXC:LFolio = Loc:LFolio

ADXC:NFolio = Loc:NFolio

ADXC:FcDoc = Loc:FcCanc

ADXC:Referencia = 'CANCELACION FACT.: '&FAC:LFolio & ' ' &FORMAT(FAC:NFolio, @N06)

ADXC:NCte = FAN:NCte

ADXC:Importe = FAN:Importe

ADXC:ACDoc = CXC:CDoc

ADXC:ACConcepto = CXC:CConcepto

ADXC:ALFolio = FAN:LFolio

ADXC:ANFolio = FAN:NFolio

ADXC:ImpAplicar = FAN:Importe

ADXC:Aplicado = 0

ADXC:Origen = 2

ADXC:Procedencia= ' '

ADXC:FcRegistro= Today()

ADXC:IdUsuario = Das\_GroupUser.NetId

add(ElADxc)

if errorcode()

Loc:Nivel\_Error= 8

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

end

end

else

!!checar si se genera el anticipo en CxC

! Loc:Saldo = CxC:Saldo, por si se pierde el valor de la cuenta que se

if (FAC:ImpTotal - Loc:Saldo) > 0

!! generar anticipo

clear(Cxc:Record)

CXC:CDoc = 'AN'

CXC:CConcepto = 2

CXC:LFolio = Loc:LFolio

CXC:NFolio = Loc:NFolio

CXC:FcDoc = Loc:FcCanc

CXC:FcVenc = Loc:FcCanc

CXC:Referencia = 'CANCEL. FACT. DIA ANT.: '&FAC:LFolio & ' ' &FORMAT(FAC:NFolio, @N06)

CXC:NCte = FAC:NCte

CXC:Importe = (FAC:ImpTotal - Loc:Saldo)

CXC:Saldo = (FAC:ImpTotal - Loc:Saldo)

CXC:Aplicado = 0

CXC:Tipo\_Recibo= 1 ! recibo no fiscal

CXC:DiasGracia = FAC:DiaGracia

CXC:Procedencia= 'FE'

CXC:IdUsuario = Das\_GroupUser.NetId

add(ElCxC)

if errorcode()

Loc:Nivel\_Error= 7

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

end

!! checar si se genera la nota de abono del cliente

if Loc:Imp\_NA > 0

!! generar nota de abono

!! checar que no se duplique el docto.

if errorcode()

Loc:Nivel\_Error= 81

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

! checar que no exista el folio

clear(ADxc:Record)

ADXC:LFolio= Loc:LFolio

ADXC:NFolio= Loc:NFolio

get(ElADxc, ADXC:LLP\_LFolio\_NFolio)

if ~errorcode()

Loc:Nivel\_Error= 82

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

! grabar el docto.

clear(ADxc:Record)

ADXC:CDoc = 'CA' !! canc. días ant.

ADXC:CConcepto = 10 !! cancelación días ant.

ADXC:LFolio = Loc:LFolio

ADXC:NFolio = Loc:NFolio

ADXC:FcDoc = Loc:FcCanc

ADXC:Referencia= 'CANCELACION FACT.: '& FAC:LFolio & ' ' &FORMAT(FAC:NFolio, @N06)

ADXC:NCte = CXC:NCte

ADXC:Importe = Loc:Imp\_NA

ADXC:ACDoc = CXC:CDoc

ADXC:ACConcepto= CXC:CConcepto

ADXC:ALFolio = CXC:LFolio

ADXC:ANFolio = CXC:NFolio

ADXC:ImpAplicar= Loc:Imp\_NA

ADXC:Aplicado = 0

ADXC:Origen = 0

ADXC:Procedencia= 'FC'

ADXC:FcRegistro= Today()

ADXC:IdUsuario = Das\_GroupUser.NetId

add(ElADxC)

if errorcode()

Loc:Nivel\_Error= 8

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!Buscar CXC

clear(Cxc:Record)

CXC:LFolio= FAC:LFolio

CXC:NFolio= FAC:NFolio

get(ElCxC, CXC:LLP\_LFolio\_NFolio)

if errorcode()

Loc:Nivel\_Error= 7

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!1. igualar el saldo a cero

!2. grabar registro

CXC:Saldo= 0

put(ElCxC)

if errorcode()

Loc:Nivel\_Error= 7

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

end

end !! Fin de Procedimiento Para Anticipo / Activo, Inicia Procedimiento Común para ambos

! grabar notas de abono de facturación

clear(Fna:Record)

FNA:LFolio = Loc:LFolio

FNA:NFolio = Loc:NFolio

FNA:FcMov = Loc:FcCanc

FNA:Concepto = FAC:Conc\_Cancelacion

FNA:NCte = FAC:NCte

FNA:NAlm = FAC:NAlm

FNA:Consecutivo = FAC:Consecutivo

FNA:SubTotal = FAC:SubTotal

FNA:Imp\_Impuestos = FAC:ImpImpuesto

FNA:Importe\_Total = FAC:ImpTotal

FNA:Anticipo\_Fiscal= 0

FNA:TMov = 2 ! días anteriores

FNA:Procedencia= 'FC' ! Factura/Cancelación

FNA:HrMov = clock()

FNA:IdUsuario = Das\_GroupUser.NetId

!! Si está configurado en CFDi, guardar el dato en la nota

IF CTRL:Grl\_VersionCFD = 2 THEN

FNA:TipoCFD = 1

ELSIF CTRL:Grl\_VersionCFD = 3 THEN

FNA:TipoCFD = 2

END

add(ElFna)

if errorcode()=40

!! llave duplicada

Loc:Nivel\_Error= 131

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

elsif errorcode()

!! cualquier otro error

Loc:Nivel\_Error= 13

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

!! buscar el folio en el catálogo

FOF:CFolio= Loc:CFolio

get(ElFof, Fof:LLP\_Cfolio)

if errorcode()

Loc:Nivel\_Error= 2

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

! checar si se debe act. el folio

if Fof:NFolio<(Loc:NFolio+1)

!! actualizar el núm. de folio

!1. incr. el folio

!2. grabar el reg.

!3. checar si hay error

Fof:NFolio= Loc:NFolio+1

put(Elfof)

if errorcode()

Loc:Nivel\_Error= 2

do sr\_error\_transaccion

select(?Loc:FcCanc)

cycle

end

end

!! Actualizar los datos del Pedido

! Do Sr\_ActPedido

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

!! tomar los datos para imprimir la nota de abono

Loc:LFolIni= Loc:LFolio

Loc:NFolIni= Loc:NFolio

! End of "When completed, before writing to disk"

History::FAC:Record = FAC:Record

CASE LocalRequest

OF ChangeRecord

LOOP

LocalResponse = RequestCancelled

SETCURSOR(Cursor:Wait)

RecordChanged=false

IF (SAV::FAC:Record <> FAC:Record)

RecordChanged = True

END

! Start of "Before call to RIUpdate if Record Changed"

! [Priority 5000]

! End of "Before call to RIUpdate if Record Changed"

IF RecordChanged THEN

Update::Error = RIUpdate:ELFAC(1)

ELSE

Update::Error = 0

END

SETCURSOR()

IF Update::Error THEN

IF Update::Error = 1 THEN

! Start of "Despues Llamar RIUptdate; Error de Modificación"

! [Priority 4000]

!! llama al rollback si hay error

Loc:Nivel\_Error= 3

do sr\_error\_transaccion

! End of "Despues Llamar RIUptdate; Error de Modificación"

CASE StandardWarning(Warn:UpdateError)

OF Button:Yes

CYCLE

OF Button:No

POST(Event:CloseWindow)

BREAK

END

END

DISPLAY

SELECT(?Loc:LFolio:Prompt)

! Start of "(Eo) Después de Completar Modificación, Error de Modificación"

! [Priority 4000]

!! llama al rollback si hay error

Loc:Nivel\_Error= 3

do sr\_error\_transaccion

! End of "(Eo) Después de Completar Modificación, Error de Modificación"

ELSE

IF RecordChanged OR VCRRequest = VCRNone THEN

LocalResponse = RequestCompleted

END

! Start of "Evento Completo; Modificación sin Error"

! [Priority 4000]

!! dar por terminada la transacción. imprimir docto.

commit

END

BREAK

END

END

END

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! procedimiento del rollback

sr\_error\_transaccion routine

data

Loc:Arch String(90)

Loc:Merror String(120)

code

!unlock(ElADxc)

!! actualizar banderas de control

!1. bandera en: no hay error en la transacción

!2. bandera en: no hay transacción activa

!3. determina el archivo en donde ocurrió el error

Loc:Error\_Transaccion = false

Loc:Transaccion\_Activa= false

case Loc:Nivel\_Error

of 1

Loc:Arch= 'No Existe el Tipo de Documento para ''Notas de Abono''.|' & |

'Defina Primero el Tipo de Documento'

of 2

Loc:Arch= 'Folos Fiscales'

of 3

Loc:Arch= 'Facturación'

of 4

Loc:Arch= 'Detalle de Facturación'

of 5

Loc:Arch= 'Existencia de Artículos por Almacén'

of 6

Loc:Arch= 'Capas de Artículos'

of 7

Loc:Arch= 'Movimientos de Cuentas Por Cobrar'

of 8

Loc:Arch= 'Afectación a Documentos Por Cobrar'

of 9

Loc:Arch= 'Clientes'

of 10

Loc:Arch= 'Movimientos al Invetario'

of 11

Loc:Arch= 'Control de Inventario de Números de Serie por Almacén'

of 12

Loc:Arch= 'Número de Serie en Facturación'

of 13

Loc:Arch= 'Notas de Abono en Facturación'

of 14

Loc:Arch= 'Detalle de Notas de Abono en Facturación'

of 17

Loc:Arch= 'Anticipos Facturados'

of 18

Loc:Arch= 'Detalle de Anticipos Facturados'

of 19

Loc:Arch= 'Anticipos en Cuentas por Cobrar'

of 20

Loc:Arch= 'Cabecera de Pedidos'

of 21

Loc:Arch= 'Detalle de Pedidos'

of 22

Loc:Arch= 'Pagos Virtuales'

Of 23

Loc:Arch= 'Historico de Pagos'

end

Loc:Merror= error()

!! iniciar rollback

!1. timbre

!2. realizar el rollback

!3. timbre

!4.. mensajes de error

beep

rollback

beep

if Loc:Nivel\_Error=81

message('No se Puede Utilizar el Archivo de Doctos. de Afectación|' & |

'para Revisar que No se Duplique el Folio del Docto.|' & |

'Espere un Momento y Vuelva a Grabar el Docto.', 'Error de Proceso', icon:hand, button:ok)

elsif Loc:Nivel\_Error=82

message('Ya Existe un Documento en Doctos. de Afectación con|' & |

'el Mismo Folio de la Cancelación que está Capturando.|' & |

'Cambie el Folio y Vuelva a Grabar el Docto.', 'Error de Proceso', icon:hand, button:ok)

elsif Loc:Nivel\_Error=131

message('Ya Existe un Documento en la Notas de Abonoecta con el|' & |

'Mismo Folio de la Cancelación que está Capturando.|' & |

'Cambie el Folio y Vuelva a Grabar el Docto.', 'Error de Proceso', icon:hand, button:ok)

elsif Loc:Nivel\_Error=120

message('No Existe el Anticipo Facturado Relacionado con la Factura.|' & |

'Revise el Control de Anticipos Facturados.', 'Error de Proceso', icon:hand, button:ok)

else

message('Ha Ocurrido un Error al Realizar la Transacción.|' & |

'|Archivo: ' & clip(Loc:Arch) & |

'|Artículo: ' & Loc:Art\_Nivel & |

'|Error: ' & clip(Loc:Merror), 'Error de Proceso', icon:hand, button:ok)

end

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!