

[18/12/2023] AtualizaBaseDados INICIO.

[18/12/2023] A atualizar versão 781122 para 781123

A executar bloco 1..... OK!
A executar bloco 2..... OK!
A executar bloco 3..... OK!
A executar bloco 4..... OK!
A executar bloco 5..... OK!
A executar bloco 6..... OK!
A executar bloco 7..... OK!
A executar bloco 8..... OK!
A executar bloco 9..... OK!
A executar bloco 10..... OK!
A executar bloco 11..... OK!
A executar bloco 12..... OK!
A executar bloco 13..... OK!
A executar bloco 14..... OK!
A executar bloco 15..... OK!
A executar bloco 16..... OK!
A executar bloco 17..... OK!
A executar bloco 18..... OK!
A executar bloco 19..... OK!
A executar bloco 20..... OK!
A executar bloco 21..... OK!
A executar bloco 22..... OK!
A executar bloco 23..... OK!
A executar bloco 24..... OK!
A executar bloco 25..... OK!
A executar bloco 26..... OK!
A executar bloco 27..... OK!
A executar bloco 28..... OK!
A executar bloco 29..... OK!
A executar bloco 30..... OK!
A executar bloco 31..... OK!
A executar bloco 32..... OK!
A executar bloco 33..... OK!
A executar bloco 34..... OK!
A executar bloco 35..... OK!
A executar bloco 36..... OK!
A executar bloco 37..... OK!
A executar bloco 38..... OK!
A executar bloco 39..... OK!
A executar bloco 40..... OK!
A executar bloco 41..... OK!
A executar bloco 42..... OK!
A executar bloco 43..... OK!
A executar bloco 44..... OK!
A executar bloco 45..... OK!
A executar bloco 46..... OK!
A executar bloco 47..... OK!
A executar bloco 48..... OK!
A executar bloco 49..... OK!
A executar bloco 50..... OK!
A executar bloco 51..... OK!
A executar bloco 52..... OK!

```

A executar bloco 53..... OK!
A executar bloco 54..... OK!
A executar bloco 55..... OK!
A executar bloco 56..... OK!
A executar bloco 57..... OK!
A executar bloco 58..... OK!
A executar bloco 59..... OK!
A executar bloco 60..... OK!
A executar bloco 61..... OK!
A executar bloco 62..... OK!
A executar bloco 63..... OK!
A executar bloco 64..... OK!
A executar bloco 65..... OK!
A executar bloco 66..... OK!
A executar bloco 67..... OK!
A executar bloco 68..... OK!
A executar bloco 69..... OK!
A executar bloco 70..... OK!
A executar bloco 71..... ERRO!

```

Cannot resolve the collation conflict between "SQL_Latin1_General_CP1_CI_AS" and "Latin1_General_CI_AS" in the equal to operation.

*** LOTE INICIO ***

```

CREATE PROCEDURE [dbo].[GCP_CCT_ProcessaPendentesAData]
    @tmpTabelaPendentes      nvarchar(200),
    @DataRef                  datetime,
    @Posto                    nvarchar(3),
    @StrSQL                   TEXT,
    @ConsultaPendentes        BIT = 0,
    @pTipoDataRef              TINYINT = 1
AS

```

SET NOCOUNT ON

```

DECLARE @StrSQLAux           NVARCHAR(4000),
        @TemPendente         Bit,
        @IdMovCBLOrig         uniqueidentifier ,
        @PendenteGerado       Bit,
        @DataDoc              datetime,
        @HDataDoc             datetime,
        @Entidade             nvarchar(12),
        @TipoEntidadeQ        nvarchar(1),
        @TipoConta            nvarchar(3),
        @Estado               nvarchar(3),
        @ValorRecMoedaOrig     float,
        @ValorDescMoedaOrig    float,
        @TipoLancamento       varchar(3),
        @TipoDocumento         nvarchar(2),
        @CabTipoEntidade       nvarchar(1),
        @CabTipoEntidadeDestino nvarchar(1),
        @CabEntidade           nvarchar(12),
        @CabEntidadeDestino    nvarchar(12),
        @ModuloOrig            nvarchar(1),
        @FilialOrig            nvarchar(3),

```

@SerieOrig	nvarchar(5),
@TipoDocOrig	nvarchar(5),
@NumDocOrig	nvarchar(20),
@NumDocOrigInt	integer,
@NumPrestacao	integer,
@NumTransferencia	integer,
@HValorTotal	float,
@HTotalIva	float,
@IdHistorico	uniqueidentifier,
@HDataVenc	datetime,
@HMoedaDaUEM	bit,
@HCambioMAlt	float,
@HCambioMBase	float,
@HCambio	float,
@HMoeda	nvarchar(3),
@TemHistorico	bit,
@ValorPendenteCursor	float,
@ObraID	uniqueidentifier,
@Conta	nvarchar(5),
@RespCobranca	nvarchar(3),
@Vendedor	nvarchar(3),
@CursorOpen	bit,
@HDataIntroducao	datetime,
@TipoDoc	nvarchar(5),
@WBSItem	nvarchar(100),
@NumDoc	integer,
@Serie	nvarchar(5),
@NumPlano	INT,
@NumeroPagamento	SMALLINT,
@IdPlanoPag	UNIQUEIDENTIFIER,
@TipoContaT	nvarchar(1),
@bInLiquidado	BIT,
@TipoTerceiro	nvarchar(3),
@SQLCamposUtil	NVARCHAR(MAX),
@ModoPag	NVARCHAR(5),
@CondPag	NVARCHAR(2),
@Processo	NVARCHAR(20),
@Linhaprocesso	INT,
@ClasseEcon	NVARCHAR(30),
@OrganicaCBL	NVARCHAR(30),
@FonteFinancCBL	NVARCHAR(30),
@ActividadeCBL	NVARCHAR(30),
@ProgMedidaCBL	NVARCHAR(30)

DECLARE @NomeView AS NVARCHAR(60)

SET @NomeView = 'tmpV_GCP_CCT_ProcessaPendentesAData'

--Verificar se na tabela temporária vem a info da view
(nomeTemp%NomeView)

IF CHARINDEX('%', @tmpTabelaPendentes) > 0

BEGIN

SET @NomeView = SUBSTRING(@tmpTabelaPendentes, CHARINDEX('%',
@tmpTabelaPendentes) + 1, LEN(@tmpTabelaPendentes))

SET @tmpTabelaPendentes = SUBSTRING(@tmpTabelaPendentes, 0,

```

CHARINDEX('%', @tmpTabelaPendentes))
END

-- Destroi Tabela
SET @StrSQLAux = 'EXEC STD_DropTempTable ' + @tmpTabelaPendentes
EXEC(@StrSQLAux)

SET @CursorOpen = 0

-- Garante que o cursor é removido --
IF CURSOR_STATUS('global','Cursor_Liqs')>=-1
BEGIN
    DEALLOCATE Cursor_Liqs
END

IF NOT EXISTS (SELECT [ID] FROM tempdb..sysobjects WHERE [ID] =
Object_ID('tempdb..#TmpErros') AND Type = 'U')
    CREATE TABLE [dbo].[#TmpErros] ([Tipo] VARCHAR(1), [ID] INT,
[Mensagem] VARCHAR(512), [Parametro1] VARCHAR(512), [Parametro2] VARCHAR(512),
[Parametro3] VARCHAR(512), [Parametro4] VARCHAR(512), [Parametro5] VARCHAR(512))
ELSE
    DELETE FROM [dbo].[#TmpErros]

EXEC('STD_DropView ' + @NomeView)

EXEC(@StrSQL)

--Para poder usar sql dinamico, tenho tenho que criar a temporaria antes
CREATE TABLE #tmpTabelaPendentes(
    [Modulo] [nvarchar](1) NOT NULL,
    [TipoEntidade] [nvarchar](1) NOT NULL,
    [Entidade] [nvarchar](12) NULL,
    [Nome] [nvarchar](100) NULL,
    [Filial] [nvarchar](3) NOT NULL,
    [Serie] [nvarchar](5) NOT NULL,
    [TipoDoc] [nvarchar](5) NOT NULL,
    [NumDoc] [nvarchar](60) NULL,
    [NumDocInt] [int] NOT NULL,
    [NumPrestacao] [smallint] NOT NULL,
    [NumTransferencia] [int] NOT NULL,
    [DataDoc] [datetime] NULL,
    [DataVenc] [datetime] NULL,
    [DataIntroducao] [datetime] NULL,
    [ValorTotal] [float] NULL,
    [ValorPendente] [float] NULL,
    [Moeda] [nvarchar](3) NULL,
    [Cambio] [float] NULL,
    [CambioMBase] [float] NOT NULL,
    [CambioMAlt] [float] NOT NULL,
    [MoedaDaUEM] [bit] NULL,
    [TipoConta] [nvarchar](3) NULL,
    [Estado] [nvarchar](4) NOT NULL,
    [TipoTerceiro] [nvarchar](3) NULL,

```

```

[NumAvisos] [smallint] NULL,
[Posto] [varchar](1) NOT NULL,
[IDHistorico] [uniqueidentifier] NULL,
[DataPrevTes] [datetime] NULL,
[ObraID] [uniqueidentifier] NULL,
[Conta] [nvarchar](5) NULL,
[RespCobranca] [nvarchar](3) NULL,
[Vendedor] [nvarchar](3) NULL,
[IdMovCBL] [uniqueidentifier] NULL,
[WBSItem] [nvarchar](100) NULL,
[ModoPag] [nvarchar](5) NULL,
[CondPag] [nvarchar](2) NULL,
[TotalIva] [float] NULL,
[ValorRetencaoPendente] [float] NULL,
[ValorRetencaoGarantiaPendente] [float] NULL,
[TotalRecargo] [float] NULL,
[TipoLancamento] [varchar](3) NULL,
[TipoEntidadeComercial] [nvarchar](1) NULL,
[EntidadeComercial] [nvarchar](12) NULL,
[IdCessao] [nvarchar](50) NULL,
[DiasVenc] [int] NULL,
[NumPlano] [int] NULL,
[NumeroPagamento] [smallint] NULL,
[IdPlanoPag] [uniqueidentifier] NULL,
[NumeroCheque] [nvarchar](15) NULL,
[IdLinhaTesChq] [uniqueidentifier] NULL,
[Liquidado] [int] NOT NULL,
[Processo] [nvarchar](20) NULL,
[Linhaprocesso] [int] NULL,
[ClassEcon] [nvarchar](15) NULL,
[OrganicaCBL] [nvarchar](15) NULL,
[FonteFinancCBL] [nvarchar](10) NULL,
[ActividadeCBL] [nvarchar](10) NULL,
[ProgMedidaCBL] [nvarchar](10) NULL
) ON [PRIMARY]

```

```

DECLARE @Cdus as NVARCHAR(MAX)
SET @Cdus = ''
IF @ConsultaPendentes = 1 AND EXISTS (SELECT TOP 1 1 FROM StdCamposVar
WHERE Tabela = 'Historico')
BEGIN
    --Adicionar à tabela temporária os campos do utilizador
    DECLARE @SqlCDU varchar(8000)
    SET @SqlCDU = 'ALTER TABLE #tmpTabelaPendentes ADD '

    SELECT @Cdus = @Cdus + ', ' + sc.Name , @SqlCDU = @SqlCDU +
sc.Name + ' ' +
        st.Name + CASE WHEN st.Name in
('varchar','nvarchar','char','nchar') then '(' + CAST(sc.max_length as varchar)
+ ')' ' ELSE ' ' END +
        CASE WHEN st.allownulls = 1 THEN 'NULL' ELSE 'NOT NULL'
END + ', ' + CHAR(10)
FROM sys.columns sc
INNER JOIN systypes st on st.xusertype = sc.user_type_id

```

```

INNER JOIN StdCamposVar s ON s.Tabela = 'Historico' AND s.Campo
= sc.name
WHERE sc.object_id = object_id('Historico')

SET @SqlCDU = SUBSTRING(@SqlCDU,0, LEN(@SqlCDU)- 1)
EXEC(@SqlCDU)

END

CREATE INDEX #tmpTabelaPendentes_i ON #tmpTabelaPendentes
(Modulo,Filial,Serie,TipoDoc,NumDoc,NumDocInt,NumPrestacao,TipoEntidade,Entidade
,TipoConta,Estado,NumTransferencia, ValorTotal, ValorPendente)

DECLARE @Campos as NVARCHAR(MAX)
SET @Campos = 'Modulo, TipoEntidade, Entidade, Nome, Filial, Serie,
TipoDoc, NumDoc, NumDocInt, NumPrestacao, NumTransferencia, DataDoc, DataVenc,
DataIntroducao, ValorTotal,
ValorPendente, Moeda, Cambio, CambioMBase, CambioMAlt, MoedaDaUEM, TipoConta,
Estado, TipoTerceiro, NumAvisos, Posto, IDHistorico, DataPrevTes, ObraID, Conta,
RespCobranca, Vendedor, IdMovCBL, WBSItem, ModoPag, CondPag, TotalIva,
ValorRetencaoPendente, ValorRetencaoGarantiaPendente, TotalRecargo,
TipoLancamento,
TipoEntidadeComercial, EntidadeComercial, IdCessao, DiasVenc, NumPlano,
NumeroPagamento, IdPlanoPag, NumeroCheque, IdLinhaTesChq, Liquidado, Processo,
Linhaprocessos,
ClassEcon, OrganicaCBL, FonteFinancCBL, ActividadeCBL, ProgMedidaCBL' + @Cdus

DECLARE @SqlInsert as NVARCHAR(MAX)
SET @SqlInsert = 'INSERT INTO #tmpTabelaPendentes WITH (TABLOCKX) (' +
@Campos + ') SELECT '+' @Campos +' FROM ' + @NomeView

EXEC (@SqlInsert)

-- BID 596574
SELECT DISTINCT C.Entidade AS CabEntidade, C.EntidadeDestino AS
CabEntidadeDestino, C.TipoEntidade AS CabTipoEntidade, C.TipoEntidadeDestino AS
CabTipoEntidadeDestino, D.TipoDocumento as TipoDocumento,
CASE WHEN H.Modulo IS NULL AND H2.Modulo IS NULL THEN 0 ELSE
1 END AS TemHistorico,
CASE WHEN H.DataDoc IS NULL THEN H2.DataDoc ELSE H.DataDoc
END AS HDataDoc,
CASE WHEN H.DataVenc IS NULL THEN H2.DataVenc ELSE H.DataVenc
END AS HDataVenc,
CASE WHEN H.DataIntroducao IS NULL THEN H2.DataIntroducao
ELSE H.DataIntroducao END AS HDataIntroducao,
CASE WHEN H.ValorTotal IS NULL
THEN H2.ValorTotal
ELSE ( SELECT TOP 1 ValorTotal
FROM Historico WITH(NOLOCK)
WHERE L.TipoDocOrig =
TipoDoc
AND L.ModuloOrig =
Modulo
AND L.SerieOrig =
Serie

```

```

AND L.FilialOrig      =
Filial
AND L.NumDocOrigInt   =
NumdocInt
AND L.NumPrestacao    =
NumPrestacao

ORDER BY NumTransferencia)

END

AS HValorTotal,
CASE WHEN H.TotalIva IS NULL THEN H2.TotalIva ELSE H.TotalIva
END AS HTotalIva,
CASE WHEN H.Moeda IS NULL THEN H2.Moeda ELSE H.Moeda END AS
HMoeda,
CASE WHEN H.Cambio IS NULL THEN H2.Cambio ELSE H.Cambio END
AS HCambio,
CASE WHEN H.CambioMBase IS NULL THEN H2.CambioMBase ELSE
H.CambioMBase END AS HCambioMBase,
CASE WHEN H.CambioMAlt IS NULL THEN H2.CambioMAlt ELSE
H.CambioMAlt END AS HCambioMAlt,
CASE WHEN H.MoedaDaUEM IS NULL THEN H2.MoedaDaUEM ELSE
H.MoedaDaUEM END AS HMoedaDaUEM,
CASE WHEN H.Conta IS NULL THEN H2.Conta ELSE H.Conta END AS
Conta,
CASE WHEN H.RespCobranca IS NULL THEN H2.RespCobranca ELSE
H.RespCobranca END AS RespCobranca,
CASE WHEN H.Vendedor IS NULL THEN H2.Vendedor ELSE H.Vendedor
END AS Vendedor,
L.IdMovCBLOrig, L.PendenteGerado, L.Entidade, L.TipoEntidade,
L.TipoConta, L.Estado,
(Case When H.ValorTotal >= 0 OR H2.ValorTotal >= 0 then
ABS(L.ValorRecMoedaOrig) ELSE ABS(L.ValorRecMoedaOrig) * -1 END * CASE WHEN
RE.Id IS NULL THEN 1 ELSE -1 END) as ValorRecMoedaOrig,
L.ValorDescMoedaOrig, L.ModuloOrig, L.FilialOrig,
L.SerieOrig, L.TipoDocOrig, L.NumDocOrig, L.NumDocOrigInt, L.NumPrestacao,
L.NumTransferencia, L.IdHistorico, L.ObraID,
C.DataDoc, C.TipoDoc, C.NumDoc, C.Serie, L.WBSItem,
TC.TipoConta TipoContaT,
NumPlano = ISNULL(CPP.NumPlano,0), NumeroPagamento =
ISNULL(PPP.NumPagamento,0), IdPlanoPag = PPP.IdPlano, C.DataUltimaAtualizacao,
CASE WHEN H.TipoTerceiro IS NULL THEN H2.TipoTerceiro ELSE
H.TipoTerceiro END AS TipoTerceiro,
ModoPag = H.ModoPag, CondPag = H.CondPag
INTO #tmpLiquidacoes
FROM CabLiq C WITH(NOLOCK)
INNER JOIN LinhasLiq L WITH(NOLOCK) ON C.Id=L.IdCabLiq
LEFT JOIN Historico H WITH(NOLOCK) ON L.TipoDocOrig      = H.TipoDoc
AND

L.ModuloOrig      = H.Modulo AND

L.SerieOrig       = H.Serie AND

L.FilialOrig      = H.Filial AND

```

```

L.NumDocOrigInt      = H.NumdocInt AND

L.NumTransferencia = H.NumTransferencia AND

L.NumPrestacao      = H.NumPrestacao
      INNER JOIN DocumentosCCT D WITH(NOLOCK) ON D.Documento = C.Tipodoc
      INNER JOIN TiposConta TC WITH(NOLOCK) ON TC.Conta = L.TipoConta
      LEFT JOIN   [PendentesPlanosPagamentos] PPP WITH (NOLOCK)      ON H.Id
= PPP.IdHistorico
      LEFT JOIN   [CabecPlanosPagamentos]      CPP WITH (NOLOCK)
ON PPP.IdPlano = CPP.Id
      LEFT JOIN   [RastreabilidadeEstornos]      RE WITH (NOLOCK)
ON RE.IdDocDestino = C.Id
      LEFT JOIN   [PendentesPlanosPagamentos] PPP2 WITH (NOLOCK)      ON
PPP2.Filial = L.FilialOrig AND PPP2.Modulo = L.ModuloOrig AND PPP2.TipoDoc =
L.TipoDocOrig AND PPP2.Serie = L.SerieOrig AND PPP2.NumDoc = L.NumDocOrigInt AND
PPP2.NumTransferenciaAnt = L.NumTransferencia AND PPP2.NumPrestacaoAnt =
L.NumPrestacao AND PPP2.TipoConta = L.TipoConta AND PPP2.Estado = L.Estado
      LEFT JOIN   [Historico]                      H2 WITH (NOLOCK)
ON H2.Id = PPP2.IdHistorico
      WHERE ISNULL(C.Anulado, 0) = 0 AND CASE WHEN @pTipoDataRef = 1 THEN
C.DataDoc ELSE C.DataIntroducao END > @DataRef

      DECLARE Cursor_Liqs CURSOR FORWARD_ONLY STATIC FOR
      SELECT      L.CabEntidade, L.CabEntidadeDestino,
L.CabTipoEntidade, L.CabTipoEntidadeDestino,
      L.TipoDocumento, CASE WHEN P.Modulo IS
NULL THEN 0 ELSE 1 END AS TemPendente,
      L.TemHistorico, L.HDataDoc, L.HDataVenc,
L.HDataIntroducao, L.HValorTotal, L.HTotalIva, L.HMoeda,
      L.HCambio, L.HCambioMBase,
L.HCambioMAlt, L.HMoedaDaUEM, L.DataDoc, L.Conta, L.RespCobranca, L.Vendedor,
      L.IdMovCBLOrig, L.PendenteGerado,
L.Entidade, L.TipoEntidade, L.TipoConta, L.Estado, L.ValorRecMoedaOrig,
      L.ValorDescMoedaOrig, L.ModuloOrig,
L.FilialOrig, L.SerieOrig, L.TipoDocOrig, L.NumDocOrig, L.NumDocOrigInt,
      L.NumPrestacao, L.NumTransferencia,
L.IdHistorico, L.ObraID, L.TipoDoc, L.NumDoc, L.Serie, L.WBSItem,
      L.TipoContaT, L.NumPlano,
L.NumeroPagamento, L.IdPlanoPag,
      Liquidado = CASE WHEN P.IdHistorico IS
NOT NULL THEN 0 ELSE 1 END, L.TipoTerceiro,
      L.Modopag, L.CondPag,
      Processo = ISNULL(LC.Processo,
LD.Processo), Linhaprocesso = ISNULL(LC.Linhaprocesso, LD.Linhaprocesso),
      ClasseEcon = ISNULL(LC.ClasseEcon, LD.ClasseEcon), OrganicaCBL =
ISNULL(LC.OrganicaCBL, LD.OrganicaCBL), FonteFinancCBL =
ISNULL(LC.FonteFinancCBL, LD.FonteFinancCBL), ActividadeCBL =
ISNULL(LC.ActividadeCBL, LD.ActividadeCBL), ProgMedidaCBL =
ISNULL(LC.ProgMedidaCBL, LD.ProgMedidaCBL)
      FROM      #tmpLiquidacoes L
      LEFT JOIN   Pendentes P WITH(NOLOCK) ON      P.Modulo
= L.ModuloOrig      AND

```

```

P.Filial          = L.FilialOrig          AND
P.Serie           = L.SerieOrig           AND
P.TipoDoc         = L.TipoDocOrig        AND
P.NumDoc          = L.NumDocOrig         AND
P.NumDocInt       = L.NumDocOrigInt      AND
P.NumPrestacao    = L.NumPrestacao      AND
P.TipoEntidade    = L.TipoEntidade      AND
P.Entidade        = L.Entidade           AND
P.TipoConta       = L.TipoConta         AND
P.Estado          = L.Estado             AND
P.NumTransferencia = L.NumTransferencia
LEFT JOIN Historico H (NOLOCK) ON
H.Id = L.IdHistorico

LEFT JOIN LinhasCompras LC WITH
(NOLOCK) ON LC.ID = H.IdLinhaDoc
LEFT JOIN LinhasDoc LD WITH (NOLOCK) ON LD.ID = H.IdLinhaDoc

ORDER BY L.DataUltimaAtualizacao DESC
-- ^BID 596574

OPEN Cursor_Liqs
SET @CursorOpen = 1
FETCH NEXT FROM Cursor_Liqs INTO @CabEntidade, @CabEntidadeDestino,
@CabTipoEntidade, @CabTipoEntidadeDestino, @TipoDocumento, @TemPendente,

@TemHistorico, @HDataDoc, @HDataVenc, @HDataIntroducao, @HValorTotal,
@HTotalIva, @HMoeda, @HCambio, @HCambioMBase, @HCambioMAlt,

@HMoedaDaUEM, @DataDoc, @Conta, @RespCobranca, @Vendedor,
@IdMovCBLOrig, @PendenteGerado, @Entidade,

@TipoEntidadeQ, @TipoConta, @Estado, @ValorRecMoedaOrig, @ValorDescMoedaOrig,
@ModuloOrig, @FilialOrig,

@SerieOrig, @TipoDocOrig, @NumDocOrig, @NumDocOrigInt, @NumPrestacao,
@NumTransferencia, @IdHistorico, @ObraID, @TipoDoc,

@NumDoc, @Serie, @WBSItem, @TipoContaT, @NumPlano, @NumeroPagamento,
@IdPlanoPag, @bInLiquidado, @TipoTerceiro, @ModoPag, @CondPag,

@Processo, @Linhaprocesso, @ClasseEcon, @OrganicaCBL, @FonteFinancCBL,
@ActividadeCBL, @ProgMedidaCBL

```

```

WHILE @@FETCH_STATUS = 0
BEGIN
    IF @TemHistorico = 1 BEGIN
        --Transferências de conta/estado
        IF @TipoDocumento = 5 OR @TipoDocumento = 7 BEGIN
            --Nos descontos de Títulos a coluna
            ValorRecMoedaOrig não é preenchida no pendente gerado
            IF @TipoContaT = 'T' OR @ValorRecMoedaOrig = 0
            BEGIN
                SELECT @ValorRecMoedaOrig =
                ValorRecMoedaOrig, @ValorDescMoedaOrig = ValorDescMoedaOrig FROM LinhasLiq
                WITH(NOLOCK) WHERE TipoDocOrig = @TipoDocOrig AND NumDocOrigInt = @NumDocOrigInt
                AND SerieOrig = @SerieOrig AND FilialOrig = @FilialOrig AND TipoDoc = @TipoDoc
                AND Numdoc = @NumDoc AND Serie = @Serie AND NumPrestacao = @NumPrestacao AND
                PendenteGerado = 0

                END

                IF @TemPendente = 1 BEGIN

                    IF @PendenteGerado = 1 BEGIN

                        UPDATE #tmpTabelaPendentes SET
                        ValorPendente=ValorPendente - (@ValorRecMoedaOrig + @ValorDescMoedaOrig)
                        WHERE Modulo=@ModuloOrig AND
                        Filial=@FilialOrig AND Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND
                        NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND

                        NumPrestacao=@NumPrestacao AND NumTransferencia=@NumTransferencia AND
                        TipoConta=@TipoConta AND Estado=@Estado

                        END ELSE BEGIN

                            UPDATE #tmpTabelaPendentes SET
                            ValorPendente=ValorPendente + (@ValorRecMoedaOrig + @ValorDescMoedaOrig)
                            WHERE Modulo=@ModuloOrig AND
                            Filial=@FilialOrig AND Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND
                            NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND

                            NumPrestacao=@NumPrestacao AND NumTransferencia=@NumTransferencia AND
                            TipoConta=@TipoConta AND Estado=@Estado

                            END

                        END ELSE BEGIN

                            IF @PendenteGerado = 1 BEGIN

                                UPDATE #tmpTabelaPendentes SET
                                ValorPendente=ValorPendente - (@ValorRecMoedaOrig + @ValorDescMoedaOrig)
                                WHERE Modulo=@ModuloOrig AND
                                Filial=@FilialOrig AND Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND
                                NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND

```

```
NumPrestacao=@NumPrestacao AND NumTransferencia=@NumTransferencia AND
TipoConta=@TipoConta AND Estado=@Estado
```

```
END ELSE BEGIN
```

```
UPDATE #tmpTabelaPendentes SET
ValorPendente=ValorPendente + (@ValorRecMoedaOrig + @ValorDescMoedaOrig)
WHERE Modulo=@ModuloOrig AND
Filial=@FilialOrig AND Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND
NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND
```

```
NumPrestacao=@NumPrestacao AND NumTransferencia=@NumTransferencia AND
TipoConta=@TipoConta AND Estado=@Estado
```

```
END
```

```
IF @@ROWCOUNT = 0
```

```
--Caso não exista só é
```

```
inserido o pendente origem
```

```
IF @PendenteGerado = 0
```

```
INSERT INTO
```

```
#tmpTabelaPendentes (Modulo, TipoEntidade, Entidade, Nome, Filial, Serie,
TipoDoc, NumDoc, NumDocInt, NumPrestacao, NumTransferencia, DataDoc, DataVenc,
DataIntroducao, ValorTotal, ValorPendente, Moeda, Cambio, CambioMBase,
CambioMAlt, MoedaDaUEM, TipoConta, Estado, NumAvisos, Posto, IDHistorico,
DataPrevTes, ObraID, Conta, RespCobranca, Vendedor, IdMovCBL, WBSItem, ModoPag,
CondPag, TipoEntidadeComercial, EntidadeComercial, TotalIva, TotalRecargo,
ValorRetencaoGarantiaPendente, ValorRetencaoPendente, TipoLancamento, DiasVenc,
IdCessao, NumPlano, NumeroPagamento, IdPlanoPag, NumeroCheque, IdLinhaTesChq,
Liquidado, TipoTerceiro, Processo, Linhaprocesso, ClasseEcon, OrganicaCBL,
FonteFinancCBL, ActividadeCBL, ProgMedidaCBL)
```

```
VALUES
```

```
(@ModuloOrig,@TipoEntidadeQ,@Entidade,'',@FilialOrig,@SerieOrig,@TipoDocOrig,@Nu
mDocOrig,@NumDocOrigInt,@NumPrestacao,@NumTransferencia,CASE WHEN
@PendenteGerado = 1 THEN @DataDoc ELSE @HDataDoc
END,@HDataVenc,@HDataIntroducao,@HValorTotal,@ValorRecMoedaOrig +
@ValorDescMoedaOrig,@HMoeda,@HCambio,@HCambioMBase,@HCambioMAlt,@HMoedaDaUEM,@Ti
poConta,@Estado,0,@Posto, @IdHistorico, NULL, @ObraID, @Conta, @RespCobranca,
@Vendedor, @IdMovCBLOrig, @WBSItem, @ModoPag, @CondPag, NULL, NULL, @HTotalIva,
NULL, NULL, NULL, NULL, 0, NULL, @NumPlano, @NumeroPagamento, @IdPlanoPag,
NULL, NULL, @bInLiquidado, @TipoTerceiro, @Processo, @Linhaprocesso, @ClasseEcon,
@OrganicaCBL, @FonteFinancCBL, @ActividadeCBL, @ProgMedidaCBL)
```

```
END
```

```
END
```

```
--Neste else entram as operações de
liquidação/regularizações...
```

```
ELSE BEGIN
```

```
IF ((SELECT LiquidacaoAutomatica FROM
DocumentosVenda WHERE Documento = @TipoDocOrig) <> 1 AND @ModuloOrig = 'V') OR
((SELECT LiquidacaoAutomatica FROM
```

```

DocumentosCompra WHERE Documento = @TipoDocOrig) <> 1 AND @ModuloOrig = 'C') OR
                                @ModuloOrig NOT IN ('C', 'V') BEGIN
                                IF @TemPendente = 1 BEGIN
                                    UPDATE #tmpTabelaPendentes SET
ValorPendente=ValorPendente + (@ValorRecMoedaOrig + @ValorDescMoedaOrig)
                                    WHERE Modulo=@ModuloOrig AND
Filial=@FilialOrig AND Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND
NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND

NumPrestacao=@NumPrestacao AND NumTransferencia=@NumTransferencia AND
TipoEntidade=@TipoEntidadeQ AND Entidade=@Entidade AND TipoConta=@TipoConta AND
Estado=@Estado

                                END ELSE BEGIN
                                    UPDATE #tmpTabelaPendentes SET
ValorPendente=ValorPendente + (@ValorRecMoedaOrig + @ValorDescMoedaOrig)
                                    WHERE Modulo=@ModuloOrig AND
Filial=@FilialOrig AND Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND
NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND

NumPrestacao=@NumPrestacao AND NumTransferencia=@NumTransferencia AND
TipoEntidade=@TipoEntidadeQ AND Entidade=@Entidade AND TipoConta=@TipoConta AND
Estado=@Estado

                                IF @@ROWCOUNT = 0

                                    INSERT INTO
#tmpTabelaPendentes (Modulo, TipoEntidade, Entidade, Nome, Filial, Serie,
TipoDoc, NumDoc,NumDocInt, NumPrestacao, NumTransferencia, DataDoc, DataVenc,
DataIntroducao, ValorTotal, ValorPendente, Moeda, Cambio,CambioMBase,
CambioMAlt, MoedaDaUEM, TipoConta, Estado, NumAvisos, Posto, IDHistorico,
DataPrevTes, ObraID, Conta, RespCobranca, Vendedor, IdMovCBL, WBSItem, ModoPag,
CondPag, TipoEntidadeComercial, EntidadeComercial, TotalIva, TotalRecargo,
ValorRetencaoGarantiaPendente, ValorRetencaoPendente, TipoLancamento, DiasVenc,
IdCessao, NumPlano, NumeroPagamento, IdPlanoPag, NumeroCheque, IdLinhaTesChq,
Liquidado, TipoTerceiro, Processo, Linhaprocesso, ClasseEcon, OrganicaCBL,
FonteFinancCBL, ActividadeCBL, ProgMedidaCBL)

                                    VALUES
(@ModuloOrig,@TipoEntidadeQ,@Entidade,'',@FilialOrig,@SerieOrig,@TipoDocOrig,@Nu
mDocOrig,@NumDocOrigInt,@NumPrestacao,@NumTransferencia,CASE WHEN
@PendenteGerado = 1 THEN @DataDoc ELSE @HDataDoc
END,@HDataVenc,@HDataIntroducao,@HValorTotal,@ValorRecMoedaOrig +
@ValorDescMoedaOrig,@HMoeda,@HCambio,@HCambioMBase,@HCambioMAlt,@HMoedaDaUEM,@Ti
poConta,@Estado,0,@Posto, @IdHistorico, NULL, @ObraID, @Conta, @RespCobranca,
@Vendedor, @IdMovCBLOrig, @wbsitem, @ModoPag, @CondPag, NULL, NULL, @HTotalIva,
NULL, NULL, NULL, NULL, 0, NULL, @NumPlano, @NumeroPagamento, @IdPlanoPag,
NULL, NULL, @bInLiquidado, @TipoTerceiro, @Processo, @Linhaprocesso, @ClasseEcon,
@OrganicaCBL, @FonteFinancCBL, @ActividadeCBL, @ProgMedidaCBL)

                                END

                                END

                                END

```

```

        IF @@ERROR <> 0 BEGIN
            EXEC [dbo].[GCP_InsererErro] 'E', 92101, 'Erro no
processamento'
            GOTO Cancelar
        END

    END ELSE BEGIN
        EXEC [dbo].[GCP_InsererErro] 'E', 92100, 'O registro não existe
no histórico', @TipoDocOrig, @NumDocOrigInt, @SerieOrig
        GOTO Cancelar
    END

    FETCH NEXT FROM Cursor_Liqs INTO @CabEntidade,
@CabEntidadeDestino, @CabTipoEntidade, @CabTipoEntidadeDestino, @TipoDocumento,
@TemPendente,

    @TemHistorico, @HDataDoc, @HDataVenc, @HDataIntroducao, @HValorTotal,
@HTotalIva, @HMoeda, @HCambio, @HCambioMBase, @HCambioMAlt,

    @HMoedaDaUEM, @DataDoc, @Conta, @RespCobranca, @Vendedor,
@IdMovCBLOrig,@PendenteGerado, @Entidade,

    @TipoEntidadeQ, @TipoConta, @Estado, @ValorRecMoedaOrig, @ValorDescMoedaOrig,
@ModuloOrig, @FilialOrig,

    @SerieOrig, @TipoDocOrig, @NumDocOrig, @NumDocOrigInt, @NumPrestacao,
@NumTransferencia, @IdHistorico, @ObraID, @TipoDoc,

    @NumDoc, @Serie, @WBSItem, @TipoContaT,@NumPlano, @NumeroPagamento,
@IdPlanoPag, @bInLiquidado, @TipoTerceiro, @ModoPag, @CondPag,

    @Processo, @Linhaprocesso, @ClasseEcon, @OrganicaCBL, @FonteFinancCBL,
@ActividadeCBL, @ProgMedidaCBL
    END

    CLOSE Cursor_Liqs
    DEALLOCATE Cursor_Liqs
    SET @CursorOpen = 0

    --Adiantamentos utilizados nos documentos de compra ou venda
    DECLARE Cursor_ADI CURSOR FORWARD_ONLY STATIC FOR
        SELECT Modulo, TipoEntidade, Entidade, Filial, Serie, TipoDoc,
NumDoc, NumDocInt,

NumPrestacao,NumTransferencia,DataDoc,DataVenc,DataIntroducao,ValorTotal,TotalIv
a,

        AdiDoc, Moeda, Cambio, CambioMBase, CambioMAlt,

MoedaDaUEM, TipoConta,

        id, ObraID, Conta, RespCobranca, Vendedor,

IdMovCBL, WBSItem
    FROM
        (SELECT H.Modulo, H.TipoEntidade, H.Entidade, H.Filial, H.Serie,
H.TipoDoc, H.NumDoc, H.NumDocInt,

```

H.NumPrestacao,H.NumTransferencia,H.DataDoc,H.DataVenc,H.DataIntroducao,H.ValorTotal,H.TotalIva,

CASE WHEN H.ValorTotal < 0 THEN
-1*(L.TotalIliquido + L.TotalIva) ELSE (L.TotalIliquido + L.TotalIva) END
AdiDoc,

H.Moeda, H.Cambio, H.CambioMBase, H.CambioMAlt,
H.MoedaDaUEM, H.TipoConta,

H.id, H.ObraID, H.Conta, H.RespCobranca,
H.Vendedor, H.IdMovCBL, L.WBSItem

FROM CabecDoc C WITH(NOLOCK)

INNER JOIN LinhasDoc L WITH(NOLOCK) on L.IdCabecDoc = C.Id

INNER JOIN Historico H WITH(NOLOCK) on H.id = L.IdHistorico

INNER JOIN Historico H1 WITH(NOLOCK) on H1.IdDoc = C.Id

WHERE C.Data > @DataRef AND H1.TotalAdiantamentosCC <> 0

UNION ALL

SELECT H.Modulo, H.TipoEntidade, H.Entidade, H.Filial, H.Serie,
H.TipoDoc, H.NumDoc, H.NumDocInt,

H.NumPrestacao,H.NumTransferencia,H.DataDoc,H.DataVenc,H.DataIntroducao,H.ValorTotal,H.TotalIva,

CASE WHEN H.ValorTotal < 0 THEN (L.TotalIliquido +
L.TotalIva) ELSE -1 * (L.TotalIliquido + L.TotalIva) END AdiDoc,

H.Moeda, H.Cambio, H.CambioMBase, H.CambioMAlt,
H.MoedaDaUEM, H.TipoConta,

H.id, H.ObraID, H.Conta, H.RespCobranca, H.Vendedor,
H.IdMovCBL, L.WBSItem

FROM CabecCompras C WITH(NOLOCK)

INNER JOIN LinhasCompras L WITH(NOLOCK) on L.IdCabecCompras =
C.Id

INNER JOIN Historico H WITH(NOLOCK) on H.id = L.IdHistorico

INNER JOIN Historico H1 WITH(NOLOCK) on H1.IdDoc = C.Id

LEFT JOIN CnfDocLigCBL Cnf WITH(NOLOCK) on C.TipoDoc =
Cnf.Documento and year(c.DataDoc) = Cnf.Ano and Cnf.Modulo = 'C'

WHERE (CASE WHEN ISNULL(Cnf.LigCBLData,1) = 1 THEN
C.DataIntroducao ELSE C.DataDoc END) > @DataRef

AND H1.TotalAdiantamentosCC <> 0) as Adi

OPEN Cursor_ADI

SET @CursorOpen = 1

FETCH NEXT FROM Cursor_ADI INTO @ModuloOrig, @TipoEntidadeQ, @Entidade,
@FilialOrig, @SerieOrig, @TipoDocOrig, @NumDocOrig, @NumDocOrigInt,

@NumPrestacao, @NumTransferencia, @HDataDoc, @HDataVenc, @HDataIntroducao,
@HValorTotal, @HTotalIva,

@ValorRecMoedaOrig, @HMoeda, @HCambio, @HCambioMBase, @HCambioMAlt,

@HMoedaDaUEM, @TipoConta, @IdHistorico, @ObraID, @Conta, @RespCobranca,
@Vendedor, @IdMovCBLOrig, @WBSItem

WHILE @@FETCH_STATUS = 0

BEGIN

```
IF EXISTS (SELECT 1 FROM #tmpTabelaPendentes WHERE
Modulo=@ModuloOrig AND Filial=@FilialOrig AND Serie=@SerieOrig AND
TipoDoc=@TipoDocOrig AND
```

```
NumDoc=@NumDocOrig AND NumDocInt=@NumDocOrigInt AND
NumPrestacao=@NumPrestacao AND TipoEntidade=@TipoEntidadeQ AND
Entidade=@Entidade AND ValorTotal=@HValorTotal) BEGIN
```

```
UPDATE #tmpTabelaPendentes SET
ValorPendente=ValorPendente + @ValorRecMoedaOrig
WHERE Modulo=@ModuloOrig AND Filial=@FilialOrig AND
Serie=@SerieOrig AND TipoDoc=@TipoDocOrig AND NumDoc=@NumDocOrig AND
NumDocInt=@NumDocOrigInt AND
```

```
NumPrestacao=@NumPrestacao AND
TipoEntidade=@TipoEntidadeQ AND Entidade=@Entidade AND TipoConta=@TipoConta
```

```
END ELSE BEGIN
```

```
INSERT INTO #tmpTabelaPendentes (Modulo, TipoEntidade,
Entidade, Nome, Filial, Serie, TipoDoc, NumDoc, NumDocInt, NumPrestacao,
NumTransferencia, DataDoc, DataVenc, DataIntroducao, ValorTotal, ValorPendente,
Moeda, Cambio, CambioMBase, CambioMAlt, MoedaDaUEM, TipoConta, Estado, NumAvisos,
Posto, IDHistorico, DataPrevTes, ObraID, Conta, RespCobranca, Vendedor,
IdMovCBL, WBSItem, ModoPag, CondPag, TipoEntidadeComercial, EntidadeComercial,
TotalIva, TotalRecargo, ValorRetencaoGarantiaPendente, ValorRetencaoPendente,
TipoLancamento, DiasVenc, IdCessao, NumPlano, NumeroPagamento, IdPlanoPag,
NumeroCheque, IdLinhaTesChq, Liquidado)
```

```
VALUES
```

```
(@ModuloOrig, @TipoEntidadeQ, @Entidade, '', @FilialOrig, @SerieOrig, @TipoDocOrig, @NumDocOrig, @NumDocOrigInt, @NumPrestacao, @NumTransferencia, @HDataDoc, @HDataVenc, @HDataIntroducao, @HValorTotal, @ValorRecMoedaOrig, @HMoeda, @HCambio, @HCambioMBase, @HCambioMAlt, @HMoedaDaUEM, @TipoConta, 'PEN', 0, @Posto, @IdHistorico, NULL, @ObraID, @Conta, @RespCobranca, @Vendedor, @IdMovCBLOrig, @WBSItem, NULL, NULL, NULL, NULL, @HTotalIva, NULL, NULL, NULL, NULL, 0, NULL, @NumPlano, @NumeroPagamento, @IdPlanoPag, NULL, NULL, @bInLiquidado)
```

```
END
```

```
FETCH NEXT FROM Cursor_ADI INTO @ModuloOrig, @TipoEntidadeQ,
@Entidade, @FilialOrig, @SerieOrig, @TipoDocOrig, @NumDocOrig, @NumDocOrigInt,
```

```
@NumPrestacao, @NumTransferencia, @HDataDoc, @HDataVenc, @HDataIntroducao,
@HValorTotal, @HTotalIva,
```

```
@ValorRecMoedaOrig, @HMoeda, @HCambio, @HCambioMBase, @HCambioMAlt,
```

```
@HMoedaDaUEM, @TipoConta, @IdHistorico, @ObraID, @Conta, @RespCobranca,
@Vendedor, @IdMovCBLOrig, @WBSItem
```

```
END
```

```
CLOSE Cursor_ADI
```

```
DEALLOCATE Cursor_ADI
```

```
SET @CursorOpen = 0
```

```

-- inserir os CDUs
IF @ConsultaPendentes = 1 AND EXISTS (SELECT TOP 1 1 FROM StdCamposVar
WHERE Tabela = 'Historico')
BEGIN
    SET      @SQLCamposUtil = ''

    SELECT  @SQLCamposUtil = @SQLCamposUtil + CASE WHEN
EXISTS(SELECT TOP 1 1 FROM tempdb.sys.columns WHERE [object_id] =
object_id('tempdb..#tmpTabelaPendentes') and [name] = Campo)

        THEN ', TMP.' + Campo + ' = H.' + Campo

        ELSE ''

    END
    FROM    StdCamposVar
    WHERE   Tabela = 'Historico'

    SET      @SQLCamposUtil = ISNULL(@SQLCamposUtil, '')
    SET      @SQLCamposUtil = SUBSTRING(@SQLCamposUtil, 2,
LEN(@SQLCamposUtil))
    SET      @SQLCamposUtil = RTRIM(LTRIM(@SQLCamposUtil))

    IF LEN(@SQLCamposUtil) > 0
    BEGIN
        SET @StrSQLAux = '

UPDATE

TMP

SET

' + @SQLCamposUtil + '

FROM

#tmpTabelaPendentes TMP

INNER JOIN

Historico H ON H.Id = TMP.IdHistorico'

        EXEC(@StrSQLAux)
    END
END

SET @StrSQLAux = ' SELECT * INTO ' + @tmpTabelaPendentes + ' FROM
#tmpTabelaPendentes '

EXEC(@StrSQLAux)

IF @ConsultaPendentes = 0
DROP TABLE #tmpTabelaPendentes

Cancelar:

SET NOCOUNT OFF
IF @CursorOpen = 1 BEGIN
    CLOSE Cursor_Liqs

```

```

        DEALLOCATE Cursor_Liqs
        SET @CursorOpen = 0
    END
    IF EXISTS (SELECT [ID] FROM tempdb..sysobjects WHERE [ID] =
Object_ID('tempdb..#TmpErros') AND Type = 'U')
    BEGIN
        -- Apenas se tiver erros limpa a temporária
        IF EXISTS (SELECT [ID] FROM tempdb..sysobjects WHERE [ID] =
Object_ID('tempdb..#tmpTabelaPendentes') AND Type = 'U') AND EXISTS (SELECT TOP
1 1 FROM #TmpErros)
        BEGIN
            SET @StrSQLAux = ' IF (SELECT
COALESCE(OBJECT_ID(''tempdb..' + @tmpTabelaPendentes + ''),0)) = 0 SELECT *
INTO ' + @tmpTabelaPendentes + ' FROM #tmpTabelaPendentes WHERE 1 = 0'
            EXEC(@StrSQLAux)
        END

        IF (EXISTS (SELECT TOP 1 1 FROM #TmpErros)) OR
@ConsultaPendentes = 0
        SELECT [Tipo], [ID], [Mensagem], [Parametro1], [Parametro2],
[Parametro3], [Parametro4], [Parametro5] FROM #TmpErros
    END

*** LOTE FIM ***

...ERRO!
0: 9999, ExecutaBatchSQL, Erro ao efetuar atualização

```