

Modelos de Calculos

Revisão: 23/02/2003

Abrangência

Versão 5.07	Versão 5.08	Versão 6.09	Versão 7.10	Versões Anteriores
-------------	-------------	-------------	-------------	--------------------

Existem diversas formas de se gerar uma rotina de cálculo no sistema, no entanto o modelo básico sempre se mantém. Abaixo mostramos um modelo-padrão que utiliza as funções básicas para tal.

```

/**
+-----+-----+-----+-----+-----+-----+
+ Funcao   | FINA210 | Autor | Wagner Xavier           | Data | 01/12/92 |
+-----+-----+-----+-----+-----+-----+
| Descricao | Recalcula saldos bancarios dentro de um determinado periodo |
+-----+-----+-----+-----+-----+-----+
| Sintaxe   | FINA210()
+-----+-----+-----+-----+-----+-----+
| Uso       | Generico
+-----+-----+-----+-----+-----+-----+
|          | ATUALIZACOES SOFRIDAS DESDE A CONSTRUCAO NICIAL
+-----+-----+-----+-----+-----+-----+
|Programador| Data    | BOPS | Motivo da Alteracao
+-----+-----+-----+-----+-----+-----+
|          |         |      |
+-----+-----+-----+-----+-----+-----+
#include "FINA210.CH"
#include "PROTHEUS.CH"
Function FinA210()
LOCAL nOpcA :=0
+-----+-----+-----+-----+-----+-----+
| Define Variaveis
+-----+-----+-----+-----+-----+-----+
LOCAL oDlg, aSays:={}, aButtons:={}
Private cCadastro := OemToAnsi(STR0004) //"Reconciliacao de Saldos Bancarios"
Pergunte("FIN210",.F.)
AADD (aSays,;
    OemToAnsi(STR0005))//"Este programa tem como objetivo recalculer e analisar os saldos"
AADD (aSays,;
    OemToAnsi(STR0006)) //"Bancarios dia a dia de um determinado periodo ate a data base do      "
AADD (aSays,;
    OemToAnsi(STR0007)) //"sistema. Utilizando no caso de haver necessidade de retroagir a      "
AADD (aSays,;
    OemToAnsi(STR0008)) //"movimentacao bancaria. Use como referencia a data em que o saldo      "
AADD (aSays,;
    OemToAnsi(STR0009)) //"ficou defasado. "
AADD(aButtons, { 1,.T.,{|o| nOpcA:= 1,o:oWnd:End()}} )
AADD(aButtons, { 2,.T.,{|o| o:oWnd:End()}} )
AADD(aButtons, { 5,.T.,{|| Pergunte("FIN210",.T. ) } } )
FormBatch( cCadastro, aSays, aButtons )
If nOpcA == 1
    #IFDEF TOP
        If TcSrvType() == 'AS/400'
            Processa({|lEnd| FA210Proc()}) // Chamada da funcao de reconciliacao
        Else
            Processa({|lEnd| FA211Proc()}) // Chamada da funcao de reconciliacao
        EndIf
    #ELSE
        Processa({|lEnd| FA210Proc()}) // Chamada da funcao de reconciliacao
    #ENDIF
EndIf
Return
/**
+-----+-----+-----+-----+-----+-----+
| Funcao   | Fa210Proc | Autor | Wagner Xavier           | Data | 01.12.1992|
+-----+-----+-----+-----+-----+-----+
| Descricao | Funcao de calculo dos saldos bancarios - CODEBASE
+-----+-----+-----+-----+-----+-----+
| Sintaxe   | Fa210Proc ( )
+-----+-----+-----+-----+-----+-----+

```

```

+-----+-----+
| Parametros | Nenhum |
+-----+-----+
| Retorno    | Nenhum |
+-----+-----+
| Uso        | FINA210 |
+-----+-----+
/*
Function FA210Processa()
LOCAL nSaldoIni
LOCAL nEntradas
LOCAL nSaidas, nData, cCond
LOCAL dDataMovto
LOCAL cFil      := ""
LOCAL lAllFil   := .F.
LOCAL cChave
LOCAL cIndex    := ""
LOCAL lSaida    := .F.
+-----+-----+
| Variaveis utilizadas para parametros |
| mv_par01 // Do Banco                |
| mv_par02 // Ate o Banco              |
| mv_par03 // Da Agencia               |
| mv_par04 // Ate a Agencia            |
| mv_par05 // Da Conta                 |
| mv_par06 // Ate a Conta              |
| mv_par07 // A partir da Data         |
+-----+-----+
DbSelectArea( "SA6" )
DbSeek( cFilial+mv_par01 , .T.)
ProcRegua(RecCount())
If Empty(xFilial( "SA6")) .AND. !Empty(xFilial("SE5"))
+-----+-----+
| Filtra o arquivo pro tipo e vencimento |
+-----+-----+
DbSelectArea("SE5")
cIndex := CriaTrab(nil,.f.)
cChave  := "E5_BANCO+E5_AGENCIA+E5_CONTA+DTOS(E5_DTDISPO)"
cCond   := 'dtos(E5_DTDISPO)>='+'dtos(mv_par07)+'
IndRegua("SE5",cIndex,cChave,,cCond,OemToAnsi(STR0015)) //"Selecionando Registros..."
nIndexSE5 := RetIndex("SE5")
#IFDEF TOP
    dbSetIndex(cIndex+OrdBagExt())
#ENDIF
DbSetOrder(nIndexSE5+1)
lSaida := .T.
dbGoTop()
lAllFil:= .T.
Else
    DbSelectArea("SE5")
    cIndex := CriaTrab(nil,.f.)
    cChave  := "E5_FILIAL+E5_BANCO+E5_AGENCIA+E5_CONTA+DTOS(E5_DTDISPO)"
    cCond   := 'dtos(E5_DTDISPO)>='+'dtos(mv_par07)+' .and. E5_FILIAL == '"+xFilial("SE5")+"'
    IndRegua("SE5",cIndex,cChave,,cCond,OemToAnsi(STR0015)) //"Selecionando Registros..."
    nIndexSE5 := RetIndex("SE5")
    #IFDEF TOP
        dbSetIndex(cIndex+OrdBagExt())
    #ENDIF
    DbSetOrder(nIndexSE5+1)
    dbGoTop()
EndIf
+-----+-----+
| Inicia recalculo dos saldos atraves da movimentacao bancaria |
+-----+-----+
DbSelectArea( "SA6" )
DbSeek( cFilial+mv_par01 , .T.)
While !Eof() .and. A6_FILIAL == cFilial .and. A6_COD <= mv_par02
    IncProc()
    // Alteracao para nao recalculer o saldo dos caixas do Loja, pois
    // estes devem ser recalculados atraves da opcao "Recalculo de Caixa" - Adriano
    DbSelectArea("SX5")
    If (dbSeek(xFilial("SX5")+"23"+SA6->A6_COD)) .or. (SA6->A6_Cod == "CL1")
        DbSelectArea("SA6")
        dbSkip()
        Loop
    EndIf
    DbSelectArea("SA6")
    cBanco    := A6_COD
    cAgencia  := A6_AGENCIA
    cConta    := A6_NUMCON

```

```

nSaldoIni:= 0
nEntradas:= 0
nSaidas := 0
If cAgencia < mv_par03 .or. cAgencia > mv_par04 .or. cConta < mv_par05 .or. cConta > mv_par06
  dbSkip( )
  Loop
EndIf
+-----+
| Localiza Saldo de Partida.                                     |
| Observe que o programa retorna um registro no banco de      |
| dados, portanto a data de referencia , a data em que        |
| o saldo ficou errado, nao a data correta do saldo.          |
+-----+
DbSelectArea( "SE8" )
DbSeek( cFilial+cBanco+cAgencia+cConta+Dtos(mv_par07),.T. )
dbSkip( -1 )
If E8_BANCO != cBanco .or. E8_AGENCIA != cAgencia .or. E8_CONTA != cConta .or. BOF() .or. EOF()
  nSaldoIni := 0
Else
  nSaldoIni := E8_SALATUA
End
+-----+
| Localiza movimentacao bancaria                               |
+-----+
DbSelectArea( "SE5" )
DbSetOrder(nIndexSE5+1)
cFil := Iif(1AllFil,"",xFilial("SE5"))
DbSeek(cFil+cBanco+cAgencia+cConta+Dtos(mv_par07),.T.)
While !Eof() .and. E5_BANCO+E5_AGENCIA+E5_CONTA == cBanco+cAgencia+cConta
  If !1AllFil .and. E5_FILIAL != xFilial("SE5")
    Exit
  EndIf
  dDataMovto := E5_DTDISPO
  While !Eof() .and. E5_BANCO+E5_AGENCIA+E5_CONTA+dtos(E5_DTDISPO)== ;
    cBanco+cAgencia+cConta+dtos(dDataMovto)
    If !1AllFil .and. E5_FILIAL != xFilial("SE5")
      Exit
    EndIf
    If E5_TIPODOC $ "DC/JR/MT/CM/D2/J2/M2/C2/V2/CP/TL" //Valores de Baixas
      dbSkip()
      Loop
    EndIf
    If E5_VENCTO > E5_DATA // Ignora pre' datados - gerou titulo
      dbSkip()
      Loop
    EndIf
    If E5_SITUACA = "C" //Cancelado
      dbSkip()
      Loop
    EndIf
    If SE5->E5_MOEDA $ "C1/C2/C3/C4/C5" .and. Empty(SE5->E5_NUMCHEQ)
      dbSkip()
      Loop
    EndIf
    +-----+
    | Na transferencia somente considera nestes numerarios      |
    | No Final00 , tratado desta forma.                          |
    | As transferencias TR de titulos p/ Desconto/Cau#e0 (FINA060) |
    | n#e0 sofrem mesmo tratamento dos TR bancarias do FINA100  |
    | Aclaracao : Foi incluido o tipo $ para os movimentos en di-- |
    | nheiro em QUALQUER moeda, pois o R$ nao e representativo   |
    | fora do BRASIL.                                           |
    +-----+
    If SE5->E5_TIPODOC $ "TR/TE" .and. Empty(SE5->E5_NUMERO)
      If !(E5_MOEDA $ " $ /R$/DO/TB/TC/CH"+Iif(cPaisLoc=="BRA","","/$ "))
        dbSkip()
        Loop
      EndIf
    EndIf
    If E5_TIPODOC $ "TR/TE" .and. (Substr(E5_NUMCHEQ,1,1)=="*" ;
      .or. Substr(E5_DOCUMEN,1,1) == "*" )
      dbSkip()
      Loop
    EndIf
    If SE5->E5_MOEDA == "CH" .and. IsCaixaLoja(SE5->E5_BANCO) //Sangria
      dbSkip()
      Loop
    EndIf
    If SubStr(E5_NUMCHEQ,1,1)=="*" //cheque para juntar (PA)
      dbSkip()

```

```

        Loop
    EndIf
    If !Empty(SE5->E5_MOTBX)
        If !MovBcoBx(SE5->E5_MOTBX)
            dbSkip()
        Loop
    EndIf
EndIf
+-----+
| Baixa automatica |
+-----+
If E5_TIPODOC = "BA"
    dbSkip()
    Loop
EndIf
DbSelectArea("SE5")
If E5_RECPAG = "R"
    nEntradas += E5_VALOR
Else
    nSaidas += E5_VALOR
EndIf
dbSkip()
End // Quebra da data
DbSelectArea("SE8")
DbSeek(cFilial + cBanco+cAgencia+cConta+Dtos(dDataMovto))
If Eof()
    RecLock("SE8",.t.)
Else
    RecLock("SE8",.f.)
EndIf
Replace E8_FILIAL With cFilial,;
E8_BANCO With cBanco,;
E8_AGENCIA With cAgencia,;
E8_CONTA With cConta,;
E8_DTSALATU With dDataMovto
Replace E8_SALATUA With nSaldoIni+nEntradas-nSaidas
MsUnlock()
DbSelectArea("SE5")
If !(SE5->(Eof())) .and. E5_BANCO+E5_AGENCIA+E5_CONTA == cBanco+cAgencia+cConta
    If !lAllFil .and. E5_FILIAL != xFilial("SE5")
        Exit
    EndIf
    For nData := dDataMovto+1 to ((SE5->E5_DTDISPO) - 1)
        DbSelectArea("SE8")
        If dbSeek(cFilial + cBanco+cAgencia+cConta+dtos(nData))
            RecLock("SE8",.F.,.T.)
            dbDelete()
            MsUnlock()
            SX2->(MsUnlock())
        EndIf
        DbSelectArea("SE5")
    Next
EndIf
If SE5->(Eof()) .or. E5_BANCO+E5_AGENCIA+E5_CONTA != cBanco+cAgencia+cConta
    DbSelectArea("SE8")
    DbSeek(cFilial + cBanco+cAgencia+cConta+dtos(dDataMovto+1),.t.)
    While !Eof() .and. cFilial + cBanco+cAgencia+cConta == ;
        E8_FILIAL+E8_BANCO+E8_AGENCIA+E8_CONTA
        RecLock("SE8",.F.,.T.)
        dbDelete()
        MsUnlock()
        SX2->(MsUnlock())
        DbSelectArea("SE8")
        dbSkip()
    End
EndIf
DbSelectArea("SE5")
End // Fecha Primeiro Loop do SE5
DbSelectArea("SA6")
RecLock("SA6")
Replace A6_SALATU With nSaldoIni+nEntradas-nSaidas
MsUnlock()
dbSkip()
End
DbSelectArea("SE5")
RetIndex("SE5")
Set Filter To
If !Empty(cIndex)
    ferase( cIndex+OrdBagExt())
EndIf

```

Return NIL

/*/

Funcao	Fa211Proc	Autor	Wagner Xavier	Data	09.03.2000
Descricao	Funcao de recalculo dos saldos bancarios - TOP CONNECT				
Sintaxe	Fa211Proc ()				
Parametros	Nenhum				
Retorno	Nenhum				
Uso	FINA210				

/*/

#IFDEF TOP

Function FA211Proc()

LOCAL nSaldoIni, nEntradas

LOCAL nSaidas, nData

LOCAL cQuery

LOCAL dDataMovto

Variaveis utilizadas para parametros

mv_par01 // Do Banco

mv_par02 // Ate o Banco

mv_par03 // Da Agencia

mv_par04 // Ate a Agencia

mv_par05 // Da Conta

mv_par06 // Ate a Conta

mv_par07 // A partir da Data

Inicia recalculo de saldos atraves da movimentacao bancaria

cQuery := "SELECT SA6.R_E_C_N_O_ A6_RECNO,"

cQuery += " E5_BANCO, E5_AGENCIA, E5_CONTA, E5_DTDISPO, E5_TIPODOC, E5_MOEDA,"

cQuery += " E5_NUMCHEQ, E5_MOTBX, E5_NUMERO, E5_RECPAG, E5_VALOR, E5_DOCUMEN"

cQuery += " FROM " + RetSqlName("SA6") + " SA6, " + RetSqlName("SE5") + " SE5"

cQuery += " WHERE A6_FILIAL = " + xFilial("SA6") + "'"

cQuery += " AND A6_COD between " + mv_par01 + "' AND " + mv_par02 + "'"

cQuery += " AND A6_AGENCIA between " + mv_par03 + "' AND " + mv_par04 + "'"

cQuery += " AND A6_NUMCON between " + mv_par05 + "' AND " + mv_par06 + "'"

cQuery += " AND SA6.D_E_L_E_T_ <> '*'"

cQuery += " AND E5_VENCTO <= E5_DATA"

cQuery += " AND A6_COD = E5_BANCO"

cQuery += " AND A6_AGENCIA = E5_AGENCIA"

cQuery += " AND A6_NUMCON = E5_CONTA"

cQuery += " AND E5_SITUACA <> 'C'"

cQuery += " AND E5_TIPODOC <> 'BA'"

cQuery += " AND SE5.D_E_L_E_T_ <> '*'"

If Empty(xFilial("SA6")) .AND. !Empty(xFilial("SE5"))

cQuery += " AND E5_DTDISPO >= " + dtos(mv_par07) + "'"

//O filtro de filial entre ' ' e 'ZZ' foi adicionado para o SQL utilizar indice

cQuery += " AND E5_FILIAL between ' ' AND 'ZZ'"

Else

cQuery += " AND E5_DTDISPO >= " + dtos(mv_par07) + "' AND E5_FILIAL = " + xFilial("SE5") + "'"

EndIf

cQuery += " ORDER BY E5_BANCO, E5_AGENCIA, E5_CONTA, E5_DTDISPO"

cQuery := ChangeQuery(cQuery)

dbUseArea(.T., "TOPCONN", TcGenQry(, cQuery), "TRB210", .T., .T.)

TcSetField("TRB210", "E5_DTDISPO", "D")

While !Eof()

DbSelectArea("TRB210")

cBanco := E5_BANCO

cAgencia := E5_AGENCIA

cConta := E5_CONTA

nSaldoIni:= 0

nEntradas:= 0

nSaidas := 0

Localiza Saldo de Partida.

Observe que o programa retorna um registro no banco de dados, portanto a data de referencia , a data em que o saldo ficou errado, nao a data correta do saldo.

DbSelectArea("SE8")

DbSeek(cFilial+cBanco+cAgencia+cConta+Dtos(mv_par07),.T.)

dbSkip(-1)

If E8_BANCO != cBanco .or. E8_AGENCIA != cAgencia .or. E8_CONTA != cConta .or. BOF() .or. EOF()

```

        nSaldoIni := 0
    Else
        nSaldoIni := E8_SALATUA
    EndIf
+-----+
| Localiza movimentacao bancaria |
+-----+
DbSelectArea("TRB210")
While !Eof() .and. E5_BANCO+E5_AGENCIA+E5_CONTA == cBanco+cAgencia+cConta
    dDataMovto := E5_DTDISPO
    While !Eof() .and. E5_BANCO+E5_AGENCIA+E5_CONTA+DTOS(E5_DTDISPO) == ;
        cBanco+cAgencia+cConta+dtos(dDataMovto)
        If E5_TIPODOC $ "DC/JR/MT/CM/D2/J2/M2/C2/V2/CP/TL" //Valores de Baixas
            dbSkip()
            Loop
        EndIf
        If E5_MOEDA $ "C1/C2/C3/C4/C5" .and. Empty(E5_NUMCHEQ)
            dbSkip()
            Loop
        EndIf
+-----+
| Na transferencia somente considera nestes numerarios |
| No Final100 , tratado desta forma. |
| As transferencias TR de titulos p/ Desconto/Cau#o (FINA060) |
| n#o sofrem mesmo tratamento dos TR bancarias do FINAL100 |
+-----+
        If E5_TIPODOC $ "TR/TE" .and. Empty(E5_NUMERO)
            If !(E5_MOEDA $ "R$/DO/TB/TC/CH")
                dbSkip()
                Loop
            EndIf
        EndIf
        If E5_TIPODOC $ "TR/TE" .and. (Substr(E5_NUMCHEQ,1,1)=="*" ;
            .or. Substr(E5_DOCUMEN,1,1) == "*" )
            dbSkip()
            Loop
        EndIf
        If E5_MOEDA == "CH" .and. IsCaixaLoja(E5_BANCO) //Sangria
            dbSkip()
            Loop
        EndIf
        If SubStr(E5_NUMCHEQ,1,1)=="*" //cheque para juntar (PA)
            dbSkip()
            Loop
        EndIf
        If !Empty(E5_MOTBX)
            If !MovBcoBx(E5_MOTBX)
                dbSkip()
                Loop
            EndIf
        EndIf
        If E5_RECPAG = "R"
            nEntradas += E5_VALOR
        Else
            nSaidas += E5_VALOR
        EndIf
        dbSkip()
    End // Quebra da data
    DbSelectArea("SE8")
    DbSeek(cFilial+cBanco+cAgencia+cConta+Dtos(dDataMovto))
    If Eof()
        RecLock("SE8",.t.)
    Else
        RecLock("SE8",.f.)
    EndIf
    Replace E8_FILIAL With cFilial,;
        E8_BANCO With cBanco,;
        E8_AGENCIA With cAgencia,;
        E8_CONTA With cConta,;
        E8_DTSALATU With dDataMovto
    Replace E8_SALATUA With nSaldoIni+nEntradas-nSaidas
    MsUnlock()
    DbSelectArea("TRB210")
    If !(TRB210->(Eof())) .and. E5_BANCO+E5_AGENCIA+E5_CONTA == cBanco+cAgencia+cConta
        For nData := dDataMovto+1 to ((E5_DTDISPO) - 1)
            DbSelectArea("SE8")
            If dbSeek(cFilial + cBanco+cAgencia+cConta+dtos(nData))
                RecLock("SE8",.F.,.T.)
                dbDelete()
                MsUnlock()
            EndIf
        EndFor
    EndIf
End

```

```

        SX2->(MsUnlock())
    EndIf
    DbSelectArea("TRB210")
Next
EndIf
If TRB210->(Eof()) .or. E5_BANCO+E5_AGENCIA+E5_CONTA != cBanco+cAgencia+cConta
    DbSelectArea("SE8")
    DbSeek(cFilial+cBanco+cAgencia+cConta+dtos(dDataMovto+1),.t.)
    While !Eof() .and. cFilial+cBanco+cAgencia+cConta == ;
        E8_FILIAL+E8_BANCO+E8_AGENCIA+E8_CONTA
        RecLock("SE8",.F.,.T.)
        dbDelete()
        MsUnlock()
        SX2->(MsUnlock())
        DbSelectArea("SE8")
        dbSkip()
    EndDo
EndIf
DbSelectArea("TRB210")
EndDo // Fecha Primeiro Loop do SE5
DbSelectArea("SA6")
If dbSeek( xFilial("SA6")+cBanco+cAgencia+cConta)
    RecLock("SA6")
    Replace A6_SALATU With nSaldoIni+nEntradas-nSaidas
    MsUnLock()
EndIf
DbSelectArea("TRB210")
dbSkip()
EndDo
DbSelectArea("TRB210")
DbCloseArea()
Return NIL
#ENDIF

```

Grupos Relacionados



[Principal / Guias de Referência / Como programar Advpl no ERP / Modelos Padronizados](#)

[Topo da Página](#)