

Modelos de Relatorios

Revisão: 17/07/2002

Abrangência

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

Existem várias formas de se gerar um relatório no sistema, no entanto a forma de se elaborar o programa não varia muito. Abaixo mostramos um modelo-padrão, que utiliza as funções básicas na geração de um relatório.

```

/*
+-----+
+ Funcao      | MATR425 | Autor | Rodrigo de Sartorio | Data | 11/05/95 |
+-----+-----+-----+-----+-----+
| Descricao   | Relatorio de Estoque por Lote
+-----+-----+-----+-----+
| Sintaxe     | MATR425()
+-----+-----+-----+-----+
| Uso        | Generico
+-----+-----+-----+-----+
|
|          ATUALIZACOES SOFRIDAS DESDE A CONSTRUCAO NICIAL
+-----+-----+-----+-----+
|Programador| Data    | BOPS | Motivo da Alteracao
+-----+-----+-----+-----+
|           |         |      |
+-----+-----+-----+-----+
*/
#include 'MATR425.CH'
#include 'FIVEWIN.CH'
Function MATR425()
+-----+-----+
| Define Variaveis
+-----+-----+
Local cDesc1      := STR0001 //"Este programa emitira' uma relacao com a posicao de "
Local cDesc2      := STR0002 //"estoque por Lote/Sub-Lote."
Local cDesc3      := ''
Local cString     := 'SB8'
Local Titulo      := STR0003 //"Posicao de Estoque por Lote/Sub-Lote"
Local Tamanho     := 'M'
Local wnRel       := 'MATR425'
+-----+-----+
| Variaveis Tipo Private padrao de todos os relatorios
+-----+-----+
Private aOrd      := {STR0004,STR0005} //" Por Produto"###" Por Lote/Sub-Lote"
Private aReturn   := {STR0006,1,STR0007, 1, 2, 1, '',1 } //"Zebrado"###"Administracao"
Private cPerg     := 'MR425A'
Private nLastKey  := 0
Private nTipo     := 0
+-----+-----+
| Verifica as perguntas selecionadas
+-----+-----+
Pergunte('MR425A', .F.)
+-----+-----+
| Variaveis utilizadas para parametros
| mv_par01      // Do Produto
| mv_par02      // Ate Produto
| mv_par03      // De Lote
| mv_par04      // Ate Lote
| mv_par05      // De Sub-Lote
| mv_par06      // Ate Sub-Lote
| mv_par07      // De Local
| mv_par08      // Ate Local
| mv_par09      // Lista Saldo Zerado ? Lista/Nao Lista
| mv_par10      // Do Tipo
| mv_par11      // Ate o Tipo
| mv_par12      // Do Grupo
| mv_par13      // Ate o Grupo
+-----+-----+

```

```

| Envia controle para SETPRINT                                     |
+-----+
wnRel := SetPrint(cString,wnRel,cPerg,@Titulo,cDesc1,cDesc2,cDesc3,.F.,aOrd,,Tamanho)
nTipo := If(aReturn[4]==1,GetMv('MV_COMP'),GetMv('MV_NORM'))
If nLastKey == 27
    dbClearFilter()
    Return Nil
Endif
SetDefault(aReturn,cString)
If nLastKey == 27
    dbClearFilter()
    Return Nil
Endif
RptStatus({|lEnd| C425Imp(@lEnd,wnRel,Tamanho,Titulo)},Titulo)
Return Nil
/*
+-----+-----+-----+-----+-----+-----+
| Funcao      | C425Imp  | Autor |Rodrigo Sartorio      | Data | 14/11/95 |
+-----+-----+-----+-----+-----+-----+
| Descricao   | Chamada do Relatorio
+-----+-----+-----+-----+-----+-----+
| Uso         | MATR425
+-----+-----+-----+-----+-----+-----+
*/
Static Function C425Imp(lEnd, wnRel, Tamanho, Titulo)
+-----+
| Variaveis especificas dos relatorios                               |
+-----+
Local cIndex      := ''
Local cCond       := ''
Local cLoteAnt    := ''
Local cProdAnt    := ''
Local cDescAnt    := ''
Local cSLOTAnt    := ''
Local cAlmoAnt    := ''
Local cSeekSB8    := ''
Local cCondSB8    := ''
Local cNomArq     := ''
Local cPicSld     := PesqPict('SB8', 'B8_SALDO', 12)
Local cPicEmp     := PesqPict('SB8', 'B8_EMPENHO', 12)
Local dDataAnt    := CtoD(' / / ')
Local dValiAnt    := CtoD(' / / ')
Local nSaldo      := 0
Local nEmpenho    := 0
Local nSaldoT     := 0
Local nEmpenhoT   := 0
Local nCntImpr    := 0
Local nIndSB8     := 0
Local lSubLote    := .F.
+-----+
| Variaveis utilizadas para Impressao do Cabecalho e Rodape       |
+-----+
Private aLinha     := {}
Private Cabec1     := ''
Private Cabec2     := ''
Private cBtxt      := Space(10)
Private cBCont     := 0
Private Li         := 80
Private M_PAG      := 01
/-- Condicao de Filtragem da IndRegua
cCond := 'B8_FILIAL<='+xFilial('SB8')+'.And.'
cCond += 'B8_PRODUTO>='+mv_par01+'.And.B8_PRODUTO<='+mv_par02+'.And.'
cCond += 'B8_LOTECTL>='+mv_par03+'.And.B8_LOTECTL<='+mv_par04+'.And.'
cCond += 'B8_NUMLOTE>='+mv_par05+'.And.B8_NUMLOTE<='+mv_par06+'.And.'
cCond += 'B8_LOCAL>='+mv_par07+'.And.B8_LOCAL<='+mv_par08+'.And.'
If aReturn[8]==1
    cIndex := 'B8_FILIAL+B8_PRODUTO+B8_LOCAL+B8_LOTECTL+B8_NUMLOTE'
    Titulo := STR0008 //"POSICAO DE ESTOQUE POR LOTE/SUBLOTE (POR PRODUTO)"
    Cabec1 := STR0009 //"PRODUTO DESCRICAO SUB-LOTE LOTE AL SALDO EMPENHO DATA DATA "
    Cabec2 := STR0014 //"VALIDADE "
ElseIf aReturn[8] == 2
    cIndex := 'B8_FILIAL+B8_LOTECTL+B8_NUMLOTE+B8_PRODUTO+B8_LOCAL'
    Titulo := STR0010 //"POSICAO DE ESTOQUE POR LOTE/SUB-LOTE (POR LOTE)"
    Cabec1 := STR0011 //"SUB-LOTE LOTE PRODUTO DESCRICAO AL SALDO EMPENHO DATA DATA "
    Cabec2 := STR0014 //"
Endif
+-----+
| Pega o nome do arquivo de indice de trabalho                     |
+-----+
cNomArq := CriaTrab('', .F.)

```

```

/-- Seta a Ordem Correta no Arquivo SB1
dbSelectArea('SB1')
dbSetOrder(1)
+-----+
| Cria Indice de Trabalho |
+-----+
dbSelectArea('SB8')
IndRegua('SB8', cNomArq, cIndex,, cCond, STR0017) //"Selecionando Registros..."
#IFDEF TOP
    dbSetIndex(cNomArq+OrdBagExt())
#ENDIF
dbGoTop()
SetRegua>LastRec()
+-----+
| Processa o Laco de impressao |
+-----+
Do While !Eof()
+-----+
| Cancela a impressao |
+-----+
If lEnd
    @ PRow()+1, 001 PSay STR0012 //"CANCELADO PELO OPERADOR"
    Exit
EndIf
lSubLote := Rastro(B8_PRODUTO,'S')
/-- Define a Quebra por Produto ou Lote
If aReturn[8] == 1
    cSeekSB8 := B8_FILIAL+B8_PRODUTO+B8_LOCAL
    cCondSB8 := 'B8_FILIAL+B8_PRODUTO+B8_LOCAL'
Else
    cSeekSB8 := B8_FILIAL+B8_LOTECTL+If(lSubLote,B8_NUMLOTE,'')+B8_PRODUTO+B8_LOCAL
    cCondSB8 := 'B8_FILIAL+B8_LOTECTL'+If(lSubLote,'B8_NUMLOTE+',')+B8_PRODUTO+B8_LOCAL'
EndIf
nSaldo := 0
nEmpenho := 0
nSaldoT := 0
nEmpenhoT := 0

/-- Processa o Laco da Quebra
Do While !Eof() .And. cSeekSB8 == &(cCondSB8)

    /-- Atualiza a Regua de Impressao
    IncRegua()

+-----+
| Cancela a Impressao |
+-----+
If lEnd
    @ PRow()+1, 001 PSay STR0012 //"CANCELADO PELO OPERADOR"
    Exit
EndIf

/-- Saldo do Lote ou Lote/Sublote
nSaldo += B8_SALDO
nEmpenho += B8_EMPENHO

/-- Saldo Total da Quebra
nSaldoT += B8_SALDO
nEmpenhoT += B8_EMPENHO

/-- Posiciona-se na Descricao Correta do SB1
If !(cProdAnt==B8_PRODUTO)
    SB1->(dbSeek(xFilial('SB1')+SB8->B8_PRODUTO, .F.))
EndIf

If SB1->B1_TIPO < mv_par10 .Or. SB1->B1_TIPO > mv_par11
    dbSkip()
    Loop
EndIf

If SB1->B1_GRUPO < mv_par12 .Or. SB1->B1_GRUPO > mv_par13
    dbSkip()
    Loop
EndIf

/-- Salva Dados do Registro Atual / Passa para o Proximo Registro
cProdAnt := B8_PRODUTO
cDescAnt := SubS(SB1->B1_DESC,1,30)
cSLotAnt := If(lSubLote,B8_NUMLOTE,Space(Len(B8_NUMLOTE)))
cLoteAnt := B8_LOTECTL

```

```

cAlmoAnt := B8_LOCAL
dDataAnt := B8_DATA
dValiAnt := B8_DTVALID
dbSkip()

/-- Imprime Saldo do Lote ou Lote/Sublote
If !(cSeekSB8=&(cCondSB8)) .Or. lSubLote .Or. !(cLoteAnt==B8_LOTECTL)
  /-- Verifica se Lista Saldo Zerado
  If mv_par09==2 .And. QtdComp(nSaldo)==QtdComp(0)
    Loop
  EndIf
  If Li > 58
    Cabec(Titulo,Cabec1,Cabec2,wnRel,Tamanho,nTipo)
  EndIf
  nCntImpr ++
  If aReturn[8] == 1
    @ Li, 000 PSay cProdAnt
    @ Li, 016 PSay cDescAnt
    @ Li, 047 PSay cSLotAnt
    @ Li, 054 PSay cLoteAnt
  ElseIf aReturn[8] == 2
    @ Li, 000 PSay cSLotAnt
    @ Li, 007 PSay cLoteAnt
    @ Li, 018 PSay cProdAnt
    @ Li, 034 PSay cDescAnt
  EndIf
  @ Li, 065 PSay cAlmoAnt
  @ Li, 068 PSay nSaldo Picture cPicSld
  @ Li, 081 PSay nEmpenho Picture cPicEmp
  @ Li, 094 PSay dDataAnt
  @ Li, 105 PSay dValiAnt
  Li ++
  nSaldo := 0
  nEmpenho := 0
EndIf
EndDo

/-- Imprime Saldo Total da Quebra
If nCntImpr > 0
  If Li > 58
    Cabec(Titulo,Cabec1,Cabec2,wnRel,Tamanho,nTipo)
  EndIf
  @ Li, 000 PSay If(aReturn[8]==1,STR0013,If(lSubLote,STR0016,STR0015)) //"Total do Produto -
  @ Li, 081 PSay nEmpenhoT Picture cPicEmp
  Li++
  @ Li, 000 PSay __PrtThinLine()
  Li++
  nCntImpr := 0
  nSaldoT := 0
  nEmpenhoT := 0
EndIf
EndDo
If !(Li==80)
  Roda(cBCont,cBtxt,Tamanho)
EndIf
/-- Restaura a Integridade do SB8
dbSelectArea('SB8')
RetIndex('SB8')
dbClearFilter()
If File(cNomArq+OrdBagExt())
  fErase(cNomArq+OrdBagExt())
EndIf
If aReturn[5] == 1
  Set Printer To
  dbCommitAll()
  OurSpool(wnRel)
EndIf
MS_Flush()
Return Nil

```

Grupos Relacionados



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

[Topo da Página](#)