\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

User Function FatCompleto(cNumPed) // Faturamento Completo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Local aAreaSC5 := SC5->(GetArea())

Local lFaturou := .f.

if !empty(SC5->C5\_NOTA)

msgStop("","Pedido de vendas "+cNumPed+" já foi faturado !")

return

endif

DbSelectArea("SC5") ; dbSetOrder(1) ; SC5->(Dbseek(xfilial("SC5")+cNumPed))

if !SC5->C5\_TIPO $ "B-D-" .and. posicione("SA1",1,xfilial("SA1")+SC5->C5\_CLIENTE+SC5->C5\_LOJACLI,"A1\_MSBLQL") == '1' .and. !MsgYesNo("Continuar mesmo assim?", "O Cliente encontra-se Bloqueado")

RestArea(aAreaSC5)

return

endif

BEGIN TRANSACTION

if SC5->C5\_LIBEROK <> "" ; reclock("SC5") ; SC5->C5\_LIBEROK := "" ; endif

dbSelectArea("SC9") ; DBSetOrder(1) ; DbSeek(xFilial("SC9")+cNumPed)

While SC9->(!EOF()) .and. SC9->C9\_FILIAL+SC9->C9\_PEDIDO = xFilial("SC9")+cNumPed // Se já estiver liberado, estorna para liberar novamente.....

A460Estorna(.T.)

SC9->(dbSkip())

Enddo

//Tb pode ser --> StaticCall(FATXFUN, MaGravaSC9, SC6->C6\_QTDVEN, cBlqCred, cBlqEst, aLocal)

SC6->(dbsetorder(1)) ; SC6->(dbseek(SC5->C5\_FILIAL+SC5->C5\_NUM))

while !SC6->(eof()) .and. SC6->C6\_FILIAL+SC6->C6\_NUM = SC5->C5\_FILIAL+SC5->C5\_NUM

RecLock("SC6")

MaLibDoFat(SC6->(RecNo()),SC6->C6\_QTDVEN,.t.,.t.,.f.,.f.,.T.,.F.) // Libera o Pedido de vendas

SC6->(MaLiberOk({SC5->C5\_NUM},.T.))

SC6->(MsUnLock())

SC6->(dbskip())

enddo

if !u\_FatPedVen(SC5->C5\_NUM,.t.,.f.) // Fatura o pedido de Vendas

DISARMTRANSACTION()

MsgStop("<h2>favor acionar o setor de faturamento !</h2>","Não consegui faturar o pedido de vendas")

else

lFaturou := .t.

endif

END TRANSACTION

MsUnLockall()

if !lFaturou ; frestSX1() ; return .f. ; endif

SpedNFe6Mnt(SF2->F2\_SERIE,SF2->F2\_DOC,SF2->F2\_DOC, .t.) // A Transmissão é automática então só faz o monitor Faixa, .t. não pede os parâmetros...

sleep(10000) // Aguarda 10 segundos para receber a autorização do SEFAZ

do while SF2->F2\_FIMP <> "S" // Se não autorizou a nota, faz enquanto Sefaz não autorizou....

aviso("Aguardando autorização de Uso do Sefaz....", "Nota Fiscal: " + SF2->F2\_SERIE + "-" + SF2->F2\_DOC + " gerada !",,,,,,, 10 )

SpedNFe6Mnt(SF2->F2\_SERIE,SF2->F2\_DOC,SF2->F2\_DOC, .t.) // Monitor Faixa, .t. não pede os parâmetros...

if SF2->F2\_FIMP <> "S" .and. !msgYesNo("<h2> continua tentando? </h2>","Falha no retorno do Sefaz,") ; exit ; endif

enddo

if empty(SF2->F2\_CHVNFE)

MsgStop("<h2>Reenviar XML ou acionar o faturamento !</h2>","Não consegui transmitir a NF "+SF2->F2\_DOC+" ao Sefaz,")

endif

frestSX1()

If Select("SC5") <> 0 ; SC5->(dbclosearea()) ; endif

If Select("SC6") <> 0 ; SC6->(dbclosearea()) ; endif

If Select("SC9") <> 0 ; SC9->(dbclosearea()) ; endif

If Select("SF2") <> 0 ; SF2->(dbclosearea()) ; endif

If Select("SD2") <> 0 ; SD2->(dbclosearea()) ; endif

If Select("SX5") <> 0 ; SX5->(dbclosearea()) ; endif

return NIL

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

User Function FatPedVen(cC5Num,lSefaz,lConfirma)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Local aPvlDocS := {}

Local nPrcVen := 0

Local cSerie := "001"

Local cEmbExp := ""

Local cDoc := ""

Local aArea := getarea()

DEFAULT lSefaz := .f.

DEFAULT lConfirma := .t.

SC5->(DbSetOrder(1)) ; SC5->(MsSeek(xFilial("SC5")+cC5Num))

if !empty(SC5->C5\_NOTA) ; msgStop("<h2> já foi faturado!</h2>","Pedido de vendas "+cC5Num) ; return .f. ; endif

If ( ExistBlock("M410PVNF") ) // Executa o P.E.

lContinua := ExecBlock("M410PVNF",.f.,.f.,SC5->(recno()))

EndIf

if !lContinua .or. (lConfirma .and. !msgYesNo("<h2>do pedido "+cC5Num+"?</h2>","Confirma o faturamento")) ; Return .f. ; endif

SC6->(dbSetOrder(1)) ; SC6->(MsSeek(xFilial("SC6")+SC5->C5\_NUM))

//É necessário carregar o grupo de perguntas MT460A, se não será executado com os valores default.

Pergunte("MT460A",.F.)

aviso("Obtendo dados do pedido de vendas....","Pedido: " +SC9->C9\_PEDIDO,,,,,,, 10 )

// Obter os dados de cada item do pedido de vendas liberado para gerar o Documento de Saída

While !SC6->(Eof()) .And. SC6->C6\_FILIAL == xFilial("SC6") .And. SC6->C6\_NUM == SC5->C5\_NUM

SC9->(DbSetOrder(1))

if !SC9->( MsSeek(xFilial("SC9") + SC6->C6\_NUM + SC6->C6\_ITEM) ) //FILIAL+NUMERO+ITEM

msgStop("<h2>não liberado !</h2>","Pedido "+SC6->C6\_NUM+" Item "+SC6->C6\_ITEM)

return .f.

endif

SE4->(DbSetOrder(1)) ; SE4->(MsSeek(xFilial("SE4")+SC5->C5\_CONDPAG) ) //FILIAL+CONDICAO PAGTO

SB1->(DbSetOrder(1)) ; SB1->(MsSeek(xFilial("SB1")+SC6->C6\_PRODUTO)) //FILIAL+PRODUTO

SB2->(DbSetOrder(1)) ; SB2->(MsSeek(xFilial("SB2")+SC6->(C6\_PRODUTO+C6\_LOCAL))) //FILIAL+PRODUTO+LOCAL

SF4->(DbSetOrder(1)) ; SF4->(MsSeek(xFilial("SF4")+SC6->C6\_TES)) //FILIAL+TES

nPrcVen := SC9->C9\_PRCVEN

If ( SC5->C5\_MOEDA <> 1 ) ; nPrcVen := xMoeda(nPrcVen,SC5->C5\_MOEDA,1,dDataBase) ; EndIf

AAdd(aPvlDocS,{ SC9->C9\_PEDIDO,;

SC9->C9\_ITEM,;

SC9->C9\_SEQUEN,;

SC9->C9\_QTDLIB,;

nPrcVen,;

SC9->C9\_PRODUTO,;

.F.,;

SC9->(RecNo()),;

SC5->(RecNo()),;

SC6->(RecNo()),;

SE4->(RecNo()),;

SB1->(RecNo()),;

SB2->(RecNo()),;

SF4->(RecNo())})

SC6->(DbSkip())

EndDo

if empty(aPvlDocS) ; msgStop("<h2>para Faturar !</h2>","Não há pedidos liberados") ; return .f. ; endif

aviso("Gerando Nota Fiscal....","Pedido: "+SC9->C9\_PEDIDO,,,,,,, 1 )

cDoc := MaPvlNfs( /\*aPvlNfs\*/ aPvlDocS,; // 01 - Array com os itens a serem gerados

/\*cSerieNFS\*/ cSerie,; // 02 - Serie da Nota Fiscal

/\*lMostraCtb\*/ .F.,; // 03 - Mostra Lançamento Contábil

/\*lAglutCtb\*/ .F.,; // 04 - Aglutina Lançamento Contábil

/\*lCtbOnLine\*/ .F.,; // 05 - Contabiliza On-Line

/\*lCtbCusto\*/ .T.,; // 06 - Contabiliza Custo On-Line

/\*lReajuste\*/ .F.,; // 07 - Reajuste de preço na Nota Fiscal

/\*nCalAcrs\*/ 0,; // 08 - Tipo de Acréscimo Financeiro

/\*nArredPrcLis\*/ 0,; // 09 - Tipo de Arredondamento

/\*lAtuSA7\*/ .T.,; // 10 - Atualiza Amarração Cliente x Produto

/\*lECF\*/ .F.,; // 11 - Cupom Fiscal

/\*cEmbExp\*/ cEmbExp,; // 12 - Número do Embarque de Exportação

/\*bAtuFin\*/ {||},; // 13 - Bloco de Código para complemento de atualização dos títulos financeiros

/\*bAtuPGerNF\*/ {||},; // 14 - Bloco de Código para complemento de atualização dos dados após a geração da Nota Fiscal

/\*bAtuPvl\*/ {||},; // 15 - Bloco de Código de atualização do Pedido de Venda antes da geração da Nota Fiscal

/\*bFatSE1\*/ {|| .T. },; // 16 - Bloco de Código para indicar se o valor do Titulo a Receber será gravado no campo F2\_VALFAT quando o parâmetro MV\_TMSMFAT estiver com o valor igual a "2".

/\*dDataMoe\*/ dDatabase,; // 17 - Data da cotação para conversão dos valores da Moeda do Pedido de Venda para a Moeda Forte

/\*lJunta\*/ .F.) // 18 - Aglutina Pedido Iguais

restarea(aArea)

msunlockall()

SF2->(DbSetOrder(1)) ; SF2->(MsSeek(xFilial("SF2")+cDoc+cSerie))

Return .T.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Static Function LiberaPV(cNumPed)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Local aArea := GetArea()

Local lLiberou := .t.

Local nItem := 0

Private Inclui := .F.

Private Altera := .T.

Private nOpca := 1

Private cCadastro := "Pedido de Vendas - Liberar"

Private aRotina := {}

DbSelectArea("SC5") ; dbSetOrder(1) ; SC5->(Dbseek(xfilial("SC5")+cNumPed))

if !SC5->C5\_TIPO $ "B-D-" .and. posicione("SA1",1,xfilial("SA1")+SC5->C5\_CLIENTE+SC5->C5\_LOJACLI,"A1\_MSBLQL") == '1' .and. !MsgYesNo("Continuar mesmo assim?", "O Cliente encontra-se Bloqueado")

RestArea(aArea)

return

endif

if empty(SC5->C5\_NOTA)

if SC5->C5\_LIBEROK <> " " ; reclock("SC5") ; SC5->C5\_LIBEROK := " " ; SC5->(msunlock()) ; endif

dbSelectArea("SC9") ; DBSetOrder(1)

DbSeek(xFilial("SC9")+cNumPed)

While SC9->(!EOF()) .and. SC9->C9\_FILIAL+SC9->C9\_PEDIDO = xFilial("SC9")+cNumPed

A460Estorna(.T.)

SC9->(dbSkip())

Enddo

DbSelectArea("SC6") ; dbSetOrder(1) ; SC6->(Dbseek(xfilial("SC6")+cNumPed))

While SC6->(!EOF()) .and. SC6->C6\_FILIAL+SC6->C6\_NUM = xFilial("SC6")+cNumPed // Libera todos os itens do pedido de vendas posicionado

RecLock("SC6")

MaLibDoFat(SC6->(RecNo()),SC6->C6\_QTDVEN,.t.,.t.,.f.,.f.,.T.,.F.) // Liberação do Pedido de vendas

SC6->(MaLiberOk({SC5->C5\_NUM},.T.))

MsUnLockall()

if posicione("SC9",1,xfilial("SC9") + SC6->C6\_NUM + SC6->C6\_ITEM , "C9\_BLEST") <> "10" .and. !empty(SC9->C9\_BLEST) ; lLiberou := .f. ; endif

SC6->(dbSkip())

Enddo

if lLiberou

msgInfo("<h2>Liberado !</h2>","Pedido de vendas "+cNumPed)

endif

else

msgStop("<h2>não pode ser Liberado!</h2>","Pedido de vendas "+cNumPed+" já foi faturado e")

endif

RestArea(aArea)

Return

Faturamento Automático no Protheus !  
  
  
Quem trabalha com Protheus reconhece que o setor que fatura vários pedidos por dia sofre um pouco.  
  
Incluir, Liberar, Faturar, Transmitir, aguardar Autorização....  
  
Pensando nisto e com empatia aos Faturistas, criei há alguns anos atrás uma função de Faturamento Completo, chamando as funções padrão do Protheus e vou compartilhar com vocês.  
  
Veja como é simples, basta chamar:  
  
u\_fatcompleto(Numero do Pedido de Vendas)  
  
Esta função, libera, fatura e transmite o pedido de vendas informado no parâmetro.  
  
Pode ser acionada pelo Ponto de Entrada MA410MNU, (exemplo abaixo) e será criada a opção no menu do browse de Pedido de vendas.  
  
Também pode ser acionada pelo SIGAOMS ou alguma rotina customizada de controle de carregamento após a pesagem de saída da balança rodoviária, por exemplo:   
Entrou com 10 toneladas e saiu com 30, então vendi 20 toneladas, por isto o nome "Faturamento Automático".  
  
  
A todos uma boa implantação !  
