Gráfico, Gráfico de barras, Histograma

O conteúdo gerado por IA pode estar incorreto.

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

User Function GRDEV  // Gráfico Devoluções de Vendas

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

Local nTipoGraph := 2   // Indica o tipo inicial do gráfico(1 - LINHA, 2-BARRAS, 5-PIZZA, 6-AREA).

Local cTitulo     := ""

Private cPerg     := PadR ("XXGRDEV", Len (SX1->X1\_GRUPO))

IF !Pergunte(cPerg,.T.) ; return ; endif

cQuery := "SELECT SUBSTRING(D1\_EMISSAO,1,4)+SUBSTRING(D1\_EMISSAO,5,2) ,SUBSTRING(D1\_EMISSAO,5,2)+ '/' +SUBSTRING(D1\_EMISSAO,1,4)"

cQuery += "' - '+FORMAT(SUM(D1\_QUANT),'N0') as bar, SUM(D1\_QUANT) as value  FROM "+retSqlName("SD1")+" SD1 "

cQuery += "INNER JOIN "+retSqlName("SB1")+" B ON B.B1\_FILIAL='" +xFilial("SB1")+ "' AND B.B1\_COD = SD1.D1\_COD AND B.D\_E\_L\_E\_T\_ <> '\*' AND B.B1\_TIPO IN ('PA','SP','ME','MP') "

cQuery += "INNER JOIN "+retSqlName("SA1")+" C ON C.A1\_FILIAL='" +xFilial("SA1")+ "' AND C.A1\_COD = SD1.D1\_FORNECE AND C.A1\_LOJA = SD1.D1\_LOJA AND C.D\_E\_L\_E\_T\_ <> '\*'  "

cQuery += "WHERE D1\_FILIAL='"+xfilial("SD1")+"'  AND SD1.D\_E\_L\_E\_T\_ <> '\*' AND SD1.D1\_TIPO = 'D' AND D1\_EMISSAO BETWEEN '" + dtos(MV\_PAR01) + "' AND '" + dtos(MV\_PAR02) + "' "

cQuery += "GROUP BY SUBSTRING(D1\_EMISSAO,1,4)+SUBSTRING(D1\_EMISSAO,5,2) , SUBSTRING(D1\_EMISSAO,5,2) + '/' + SUBSTRING(D1\_EMISSAO,1,4) "

cQuery += "ORDER BY 1 "

cQuery := ChangeQuery(cQuery)

If Select("QRY") > 0 ; Dbselectarea("QRY") ; QRY->(DbClosearea()) ; EndIf

TcQuery cQuery New Alias "QRY"

if QRY->(eof()) ; msgStop("<h2>dados não localizados!</h2>","Atenção,") ; QRY->( dbCloseArea() ) ; return ; endif

cTitulo := "Devoluções no periodo de " + dtoc(MV\_PAR01) + " a " + dtoc(MV\_PAR02)

u\_GeraGraf(cTitulo,"\_grafDev.png",.f.,nTipoGraph)

QRY->( dbCloseArea() )

Return

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

User Function GeraGraf(cTitulo,cArqGraf,lCompMes,nTipoGraph)

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

    Local aArea       := GetArea()

    Local cNomeRel    := "GRP\_"+dToS(Date())+StrTran(Time(), ':', '-')

    Local cDiretorio  := GetTempPath() // "\system\temp\"

    Local nLinCab     := 025

    Local nAltur      := 550

    Local nLargur     := 1050

    Local aRand       := {}

    Local aCor        := {"171,225,108", "017,019,010"}

    Local aCor1       := {"171,225,108", "017,019,010"}

    Local aCor2       := {"084,120,164", "007,013,017"}

    Default nTipoGraph := 2  // Barras

    Private cHoraEx    := Time()

    Private nPagAtu    := 1

    Private oPrintPvt

    //Fontes

    Private cNomeFont  := "Arial"

    Private oFontRod   := TFont():New(cNomeFont, , -06, , .F.)

    Private oFontTit   := TFont():New(cNomeFont, , -11, , .T.)

    Private oFontSubN  := TFont():New(cNomeFont, , -17, , .T.)

    //Linhas e colunas

    Private nLinAtu     := 0

    Private nLinFin     := 820

    Private nColIni     := 010

    Private nColFin     := 550

    Private nColMeio    := (nColFin-nColIni)/2

    #Define PAD\_LEFT    0

    #Define PAD\_RIGHT   1

    #Define PAD\_CENTER  2

    //Criando o objeto de impressão

    oPrintPvt := FWMSPrinter():New(cNomeRel, IMP\_PDF, .F., /\*cStartPath\*/, .T., , @oPrintPvt, , , , , .T.)

    oPrintPvt:cPathPDF := GetTempPath() // "\system\temp\"

    oPrintPvt:SetResolution(72)

    oPrintPvt:SetPortrait()

    oPrintPvt:SetPaperSize(DMPAPER\_A4)

    oPrintPvt:SetMargin(60, 60, 60, 60)

    oPrintPvt:StartPage()

    //Cabeçalho

    oPrintPvt:SayAlign(nLinCab, nColMeio-150, cTitulo, oFontTit, 450, 20, RGB(0,0,255), PAD\_CENTER, 0)

    nLinCab += 35

    nLinAtu := nLinCab

    //Se o arquivo existir, exclui ele

    If File(cDiretorio+cArqGraf)

        FErase(cDiretorio+cArqGraf)

    EndIf

    //Cria a Janela

    DEFINE MSDIALOG oDlgChar PIXEL FROM 0,0 TO nAltur,nLargur

        //Instância a classe

        if nTipoGraph = 2      ; oChart := FWChartBar():New()

        elseif nTipoGraph = 5  ; oChart := FWChartPie():New()

        endif

        //Inicializa pertencendo a janela

        oChart:Init(oDlgChar, .T., .T. )

        //Seta o título do gráfico

//        oChart:SetTitle("Título", CONTROL\_ALIGN\_CENTER)

       //Define que a legenda será mostrada na esquerda

//       oChart:setLegend( CONTROL\_ALIGN\_LEFT )

       //Seta a máscara mostrada na régua

       oChart:cPicture := "@e 99,999,999"

        QRY->(dbgotop())

        do while !QRY->(eof())

            //Adiciona as séries, com as descrições e valores

           oChart:addSerie(QRY->BAR, QRY->VALUE)

           if lCompMes

                if "01/" $ QRY->BAR     .or. "07/" $ QRY->BAR ; aCor := aCor1

                elseif "02/" $ QRY->BAR .or. "08/" $ QRY->BAR ; aCor := aCor2

                elseif "03/" $ QRY->BAR .or. "09/" $ QRY->BAR ; aCor := aCor1

                elseif "04/" $ QRY->BAR .or. "10/" $ QRY->BAR ; aCor := aCor2

                elseif "05/" $ QRY->BAR .or. "11/" $ QRY->BAR ; aCor := aCor1

                elseif "06/" $ QRY->BAR .or. "12/" $ QRY->BAR ; aCor := aCor2

                endif

           endif

            aAdd(aRand, aCor)

           QRY->(dbskip())

        enddo

        //Seta as cores utilizadas

        oChart:oFWChartColor:aRandom := aRand

        oChart:oFWChartColor:SetColor("Random")

        //Constrói o gráfico

        oChart:Build()

    ACTIVATE MSDIALOG oDlgChar CENTERED ON INIT (oChart:SaveToPng(0, 0, nLargur, nAltur, cDiretorio+cArqGraf), oDlgChar:End())

    oPrintPvt:SayBitmap(nLinAtu, nColIni, cDiretorio+cArqGraf, nLargur/2, nAltur/1.6)

    nLinAtu += nAltur/1.6 + 3

    //Gera o pdf para visualização

    oPrintPvt:Preview()

    RestArea(aArea)

Return