======================================================================

INÍCIO

*\*----------------------------------------------------------------------- EXEMPLO DE ALV ORIENTADO AO OBJETO*

TYPES: BEGIN OF ty\_saida,  
         carrid   TYPE sflight-carrid,  
         currency TYPE sflight-currency,  
         price    TYPE sflight-price,  
         seatsmax TYPE sflight-seatsmax,  
       END OF ty\_saida.  
  
DATA: t\_saida  TYPE TABLE OF ty\_saida,  
      o\_alv    TYPE REF TO cl\_gui\_alv\_grid,  
      o\_cont   TYPE REF TO cl\_gui\_docking\_container,  
      o\_parent TYPE REF TO cl\_gui\_container.  
  
SELECT \*  
  FROM sflight  
  INTO TABLE @DATA(t\_sflight).  
  
IF sy-subrc IS INITIAL.  
  
  t\_saida = VALUE #( FOR wa\_sflight IN t\_sflight  
                     WHERE ( seatsmax GE 300 )  
                   ( carrid   = wa\_sflight-carrid  
                     currency = wa\_sflight-currency  
                     price    = wa\_sflight-price  
                     seatsmax = wa\_sflight-seatsmax ) ).  
  
ENDIF.  
  
IF t\_saida[] IS NOT INITIAL.  
  CALL SCREEN 9000.  
ENDIF.  
  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  STATUS\_9000  OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
MODULE status\_9000 OUTPUT.  
  SET PF-STATUS 'STATUS\_9000'.  
  SET TITLEBAR 'TITLE\_9000'.  
  
*\* Técnica para você não ter que desenhar o container na tela nova*  
*\* Essa classe cria o container pra você, e depois nós referenciamos*  
*\* estre container para o objeto do ALV OO.*  
  
*\* Cria o Container para a tela do ALV*  
  CREATE OBJECT o\_cont  
    EXPORTING  
      side      = cl\_gui\_docking\_container=>dock\_at\_top  
      repid     = sy-repid  
      dynnr     = '9000'  
      extension = 1000  
    EXCEPTIONS  
      OTHERS    = 6.  
  
*\* Casting para o tipo genério*  
  o\_parent = o\_cont.  
  
*\* Criia Instância do ALV, referenciando o Container gerado acima*  
  CREATE OBJECT o\_alv  
    EXPORTING  
      i\_parent = o\_parent.  
  
  CALL METHOD o\_alv->set\_table\_for\_first\_display  
    EXPORTING  
      i\_structure\_name = 'SFLIGHT'  
    CHANGING  
      it\_outtab        = t\_saida.  
ENDMODULE.

“FIM

======================================================================

======================================================================

INÍCIO

*\*-----------------------------------------------------------------------*  
*\*               MANIPULA TELA DE SELEÇÃO*  
*\*-----------------------------------------------------------------------*

SELECT-OPTIONS: s\_mblnr FOR zmmt0019-mblnr  MODIF ID 001.  
  
PARAMETERS:     p\_mjahr TYPE zmmt0019-mjahr MODIF ID 001.  
  
SELECT-OPTIONS: s\_werks FOR zmmt0019-werks  MODIF ID 001,  
                s\_matnr FOR zmmt0019-matnr  MODIF ID 001,  
                s\_budat FOR zmmt0019-budat\_mkpf.  
  
  
PARAMETERS: rb\_rad1 RADIOBUTTON GROUP g001 USER-COMMAND muda\_tela,  
            rb\_rad2 RADIOBUTTON GROUP g001 DEFAULT 'X'.

AT SELECTION-SCREEN OUTPUT.  
  
  PERFORM: f\_loop\_screen CHANGING rb\_rad1  
                                  s\_mblnr[]  
                                  p\_mjahr  
                                  s\_werks[]  
                                  s\_matnr[].

LOOP AT SCREEN.  
  
    IF screen-group1 EQ '001'.  
  
      IF p\_rad1 IS NOT INITIAL.  
  
        REFRESH: p\_bwart[],  
                 p\_werks[],  
                 p\_matnr[].  
  
        CLEAR: p\_mjahr.  
  
        screen-input    = 0.  
        MODIFY SCREEN.  
  
      ELSE.  
  
        screen-input    = 1.  
        MODIFY SCREEN.  
      ENDIF.  
  
    ENDIF.  
  
  ENDLOOP.

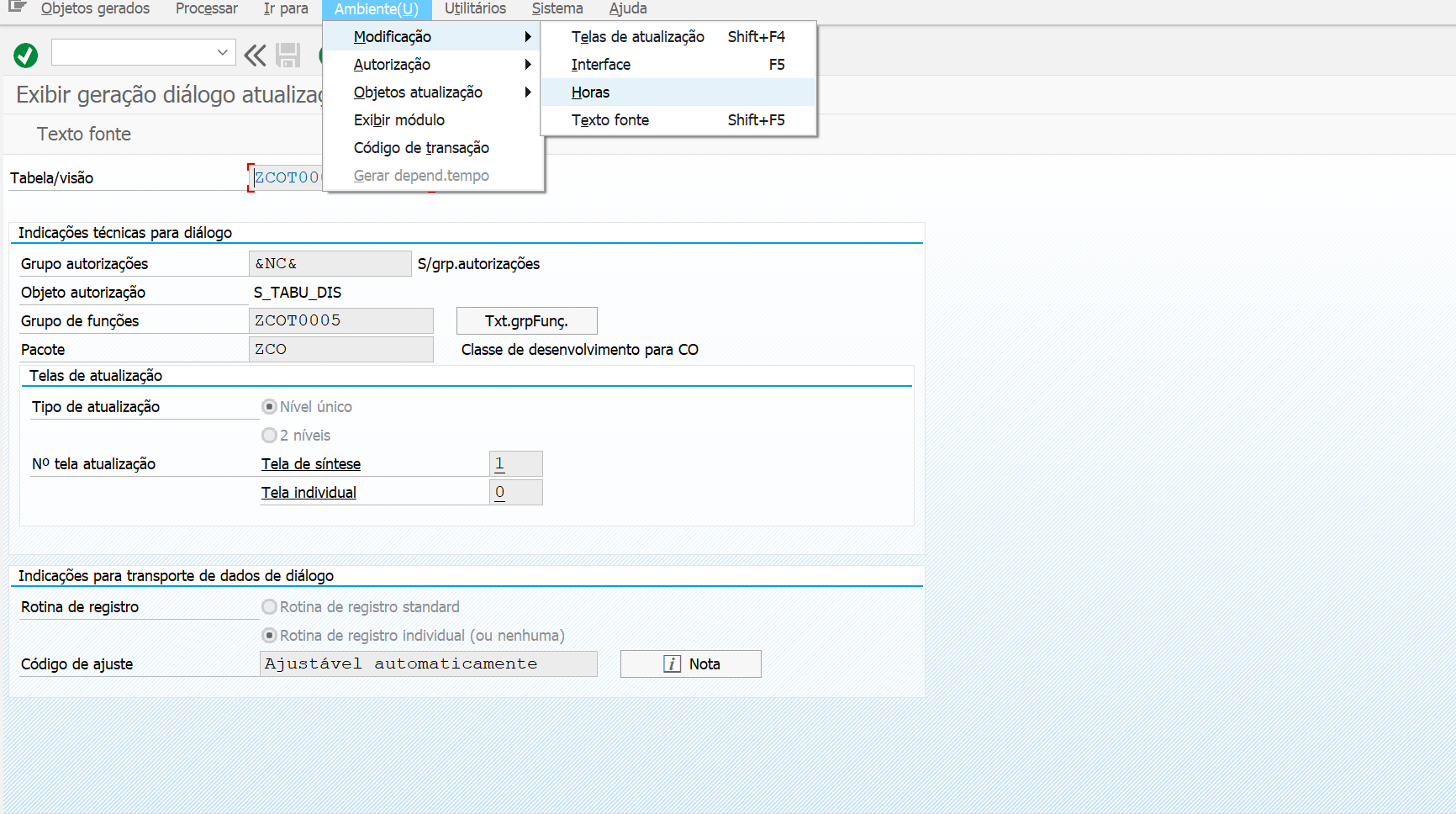
“FIM

======================================================================

======================================================================

INÍCIO

Código para validação SM30:



*\*----------------------------------------------------------------------\**  
*\*             TABELA*  
*\*----------------------------------------------------------------------\**  
  DATA: BEGIN OF t\_ativ\_maq OCCURS 0,  
          z\_ativ\_maq TYPE zcot2063-z\_ativ\_maq,  
        END OF t\_ativ\_maq.  
*\*----------------------------------------------------------------------\**  
*\*             LÓGICA*  
*\*----------------------------------------------------------------------\**  
  
*\*Agrupa linhas marcadas*  
  LOOP AT extract.  
  
    IF <xmark> IS ASSIGNED.  
  
      IF <xmark> IS NOT INITIAL.  
  
        t\_ativ\_maq-z\_ativ\_maq = extract+3.  
        APPEND t\_ativ\_maq.  
  
      ENDIF.  
  
    ENDIF.  
  
  ENDLOOP.  
  
  IF t\_ativ\_maq[] IS NOT INITIAL.  
  
*\*   Pesquisando se as chaves selecionadas existem na tabela ZCOT2063*  
    SELECT COUNT(\*)  
      FROM zcot2063  
      FOR ALL ENTRIES IN t\_ativ\_maq  
      WHERE z\_ativ\_maq EQ t\_ativ\_maq-z\_ativ\_maq.  
  
    IF sy-subrc IS INITIAL.  
  
*\*     Se existir dispara mensagem de erro*  
      MESSAGE text-001 TYPE zgt01\_c\_type\_msg2.  
  
    ENDIF.  
  
  ENDIF.

“FIM

======================================================================

======================================================================

INÍCIO

" busca SET usando o NEW:

DATA(t\_set) = NEW zcl\_utilidades( )->busca\_set\_basic( i\_setname = 'ZTESTE\_ZPP30' ).

“busca de set Hieráquico:   
      DATA(t\_set) = zcl\_utilidades=>busca\_set\_hierarchy( i\_setname = 'ZBWGS20201035' ).  
      IF t\_set[] IS NOT INITIAL.  
        SORT t\_set BY lfieldname from.  
      ENDIF.  
 "função para ler set       
   
 CALL METHOD lo\_paramcon\_fl->acessa\_parametros\_bd  
      EXPORTING  
        setname  = i\_setname  
        setparam = i\_setparam  
      CHANGING  
        tabparam = e\_tabparam  
        tabline  = e\_waparam.  
   
" LOOP para somar valores:     
   
   DATA(v\_reduce) = REDUCE s\_price( INIT v\_price TYPE s\_price  
                                    FOR <fs\_sflight> IN t\_sflight  
                                    NEXT v\_price = v\_price + <fs\_sflight>-price ).  
 "para pegar um resultado exato:   TRY.  
  
      DATA(v\_exact) = EXACT #( v\_reduce / lv\_lines ).  
  
    CATCH cx\_sy\_conversion\_rounding INTO DATA(v\_not\_exact).  
  
  ENDTRY.

"função para converter campo LIFNR :  
  
 CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'  
   EXPORTING  
     input         =  
  IMPORTING  
    OUTPUT         =

“FIM

======================================================================

======================================================================

“INÍCIO

"Exemplo de loop binário:   
  
 LOOP AT t\_vbdro INTO DATA(wa\_vbdro).  
  
      READ TABLE t\_cert TRANSPORTING NO FIELDS WITH KEY bukrs = wa\_vbdro-bukrs  
                                                        swenr = wa\_vbdro-swenr  
                                                        smenr = wa\_vbdro-smenr  
                                                        BINARY SEARCH.  
      IF sy-subrc IS INITIAL.  
  
        DATA(vl\_tabix) = sy-tabix.  
  
        LOOP AT t\_cert INTO w\_cert FROM vl\_tabix.  
  
          IF wa\_vbdro-bukrs <> w\_cert-bukrs OR  
             wa\_vbdro-swenr <> w\_cert-swenr OR  
             wa\_vbdro-smenr <> w\_cert-smenr.  
  
            EXIT.  
  
          ENDIF.  
        
      "LÓGICA  
  
          vl\_tabix = vl\_tabix + 1.  
  
        ENDLOOP.  
  
      ENDIF.  
  
    ENDLOOP.

“FIM

======================================================================

======================================================================

“INÍCIO

"Funções para ler SETS parametros:      
  
 CALL FUNCTION 'G\_SET\_GET\_ID\_FROM\_NAME'  
      EXPORTING  
        shortname                = c\_set  
      IMPORTING  
        new\_setid                = v\_set\_id  
      EXCEPTIONS  
        no\_set\_found             = 1  
        no\_set\_picked\_from\_popup = 2  
        wrong\_class              = 3  
        wrong\_subclass           = 4  
        table\_field\_not\_found    = 5  
        fields\_dont\_match        = 6  
        set\_is\_empty             = 7  
        formula\_in\_set           = 8  
        set\_is\_dynamic           = 9  
        OTHERS                   = 10.  
  
    IF v\_set\_id IS NOT INITIAL.  
  
      CALL FUNCTION 'G\_SET\_GET\_ALL\_VALUES'  
        EXPORTING  
          setnr         = v\_set\_id  
        TABLES  
          set\_values    = t\_set  
        EXCEPTIONS  
          set\_not\_found = 1  
          OTHERS        = 2.  
  
    ENDIF.

“FIM

======================================================================

======================================================================

“INÍCIO

"Exemplo de  Batch Input:   
  
\* Carrega modo de shdb  
  wa\_options-dismode  = 'N'. " Modo de processamento = Background  
  wa\_options-updmode  = 'S'. " Modo de atualização = Sincrono  
  wa\_options-defsize  = 'X'. " Tamanho standard de tela  
  wa\_options-racommit = 'X'. " COMMIT WORK não finalizado  
  
  PERFORM f\_bdc USING:   'X'   'SAPMZCO2005'      '9000'  ,  
                         ' '   'BDC\_OKCODE'       '=ENTER',  
                         ' '   'ZCOTBAL002-VSTEL' i\_vstel ,  
                         ' '   'ZCOTBAL014-EVENT' i\_event .  
  
  IF i\_ticketrec IS NOT INITIAL.  
  
    PERFORM f\_bdc USING:   ' '   'ZCOTBAL014-TICKETREC'   i\_ticketrec.  
  ENDIF.  
  
  
  
  PERFORM f\_bdc USING:   'X'   'SAPMZCO2005'        '9000'   ,  
                         ' '   'BDC\_OKCODE'         '=ENTER' ,  
                         ' '   'ZCOTBAL014-PLACA'   i\_placa  ,  
                         ' '   'ZCOTBAL014-TARA'    i\_tara   ,  
                         ' '   'ZCOTBAL014-CNTAI'   i\_cntai  ,  
                         ' '   'ZCOTBAL014-NLACR'   i\_nlacr  ,  
                         ' '   'ZCOTBAL014-PAYLOAD' i\_payload.  
  
  IF v\_exibe\_popup IS NOT INITIAL.  
  
    PERFORM f\_bdc USING: 'X'   'SAPLCOPU'    '0600'    ,  
                         ' '   'BDC\_OKCODE'     '=OPT1'.  
  
  ENDIF.  
  
  PERFORM f\_bdc USING:    'X'   'SAPMZCO2005'        '9000'   ,  
                          ' '   'BDC\_OKCODE'         '=SAVE'  ,  
                          ' '   'ZCOTBAL014-PLACA'   i\_placa  ,  
                          ' '   'ZCOTBAL014-TARA'    i\_tara   ,  
                          ' '   'ZCOTBAL014-CNTAI'   i\_cntai  ,  
                          ' '   'ZCOTBAL014-NLACR'   i\_nlacr  ,  
                          ' '   'ZCOTBAL014-PAYLOAD' i\_payload,  
                          ' '   'ZCOTBAL006-LIFNR'   i\_lifnr   ,  
                          ' '   'ZCOTBAL014-NUOFE'   I\_NUOFE  ,  
                          ' '   'ZCOTBAL016-OBS'     I\_OBS    .  
  
  
  IF v\_exibe\_popup IS NOT INITIAL.  
  
    PERFORM f\_bdc USING:   'X'   'SAPLCOPU'    '0600'    ,  
                           ' '   'BDC\_OKCODE'     '=OPT1'.  
  
  ENDIF.  
  
  PERFORM f\_bdc USING: 'X' 'SAPMZCO2005' '9000' ,  
  ' ' 'BDC\_OKCODE' '/EBACK' .  
  
  IF v\_exibe\_popup IS NOT INITIAL.  
    PERFORM f\_bdc USING: 'X' 'SAPLCOPU' '0600' ,  
   ' ' 'BDC\_OKCODE' '=OPT1'.  
  ENDIF.  
  
  
\*IMPORT será executado na transação ZCOMBAL013  
  FREE MEMORY ID 'FTI.20.001009'.  
  
  EXPORT v\_exec\_rfc TO MEMORY ID 'FTI.20.001009'.  
  
\* Executa ZCOMBAL013  
  CALL TRANSACTION c\_transacao                           "#EC CI\_CALLTA  
    USING t\_bdcdata  
    OPTIONS FROM wa\_options  
    MESSAGES INTO t\_msgtab.  
  
  IF sy-subrc = 0 AND  
     t\_msgtab[] IS NOT INITIAL.  
  
    SORT t\_msgtab BY msgid msgnr.  
    READ TABLE t\_msgtab INTO DATA(wa\_msgtab) WITH KEY msgid = 'ZCO'  
                                                      msgnr = '000'  
                                                      BINARY SEARCH.  
  
    IF sy-subrc = 0.  
  
      e\_message = wa\_msgtab-msgv1.  
  
    ENDIF.  
  
  ELSE.  
  
    READ TABLE t\_msgtab INTO wa\_msgtab INDEX 1.  
  
    IF sy-subrc = 0.  
  
      CALL FUNCTION 'MESSAGE\_TEXT\_BUILD'  
        EXPORTING  
          msgid               = wa\_msgtab-msgid  
          msgnr               = wa\_msgtab-msgnr  
          msgv1               = wa\_msgtab-msgv1  
          msgv2               = wa\_msgtab-msgv2  
          msgv3               = wa\_msgtab-msgv3  
          msgv4               = wa\_msgtab-msgv4  
        IMPORTING  
          message\_text\_output = e\_message.  
  
    ENDIF.  
  
  ENDIF.  
  
\*EXPORT executado na transação ZCOMBAL013  
  IMPORT v\_ticket FROM MEMORY ID 'FTI.20.001009\_2'.  
  
  IF v\_ticket IS NOT INITIAL.  
  
    e\_ticket = v\_ticket.  
  
  ENDIF.

FORM f\_bdc USING:   p\_begin  
                    p\_nam  
                    p\_val.  
  
  DATA: wa\_bdcdata TYPE bdcdata.  
  
  CLEAR: wa\_bdcdata.  
  IF p\_begin IS INITIAL.  
    wa\_bdcdata-fnam = p\_nam.  
    wa\_bdcdata-fval = p\_val.  
  ELSE.  
    wa\_bdcdata-program  = p\_nam.  
    wa\_bdcdata-dynpro   = p\_val.  
    wa\_bdcdata-dynbegin = p\_begin.  
  ENDIF.  
  
  APPEND wa\_bdcdata TO t\_bdcdata.  
  CLEAR wa\_bdcdata.  
  
ENDFORM.

FUNÇÃO ONDE EU USEI: Z\_SDF0073

“FIM  
======================================================================

======================================================================

“INÍCIO

"conversão para data invertida  
  
CALL FUNCTION 'CONVERSION\_EXIT\_INVDT\_OUTPUT'  
  EXPORTING  
    input         =  
 IMPORTING  
   OUTPUT        =  
          .

“FIM

======================================================================

“INÍCIO

"LOOP INLINE COM CONDICIONAL "Pegar somente registros onde a moeda for USD"  
  
t\_saida = VALUE #( FOR wa\_sflight IN t\_sflight  
                   WHERE ( currency = 'USD' )  
                 ( carrid   = wa\_sflight-carrid  
                   currency = wa\_sflight-currency  
                   price    = wa\_sflight-price ) ).

“FIM

======================================================================

“INÍCIO

"procurar conteúdo na tabela "Verificando se no campo "currency" da tabela currency existe USD:  
  
IF line\_exists( t\_sflight[ currency = 'USD' ] ).  
  
  
   BREAK-POINT.  
  "existe  
  
ENDIF.

“FIM

======================================================================

“INÍCIO

"Conversões de variáveis:  
  
DATA(v\_days\_string) = '15081998'.  
  
DATA(v\_days\_date)   = CONV d( v\_days\_string ).

“FIM

======================================================================

“INÍCIO

"Função para buscar diretório na AL11:  
  
CALL FUNCTION 'EPS\_GET\_DIRECTORY\_LISTING'  
  EXPORTING  
    dir\_name                     =  
\*   FILE\_MASK                    = ' '  
\* IMPORTING  
\*   DIR\_NAME                     =  
\*   FILE\_COUNTER                 =  
\*   ERROR\_COUNTER                =  
  tables  
    dir\_list                     =  
\* EXCEPTIONS  
\*   INVALID\_EPS\_SUBDIR           = 1  
\*   SAPGPARAM\_FAILED             = 2  
\*   BUILD\_DIRECTORY\_FAILED       = 3  
\*   NO\_AUTHORIZATION             = 4  
\*   READ\_DIRECTORY\_FAILED        = 5  
\*   TOO\_MANY\_READ\_ERRORS         = 6  
\*   EMPTY\_DIRECTORY\_LIST         = 7  
\*   OTHERS                       = 8  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here

“FIM

======================================================================

“INÍCIO

"método para saber se os caracteres são UTF8:  
 cl\_abap\_file\_utilities=>check\_utf8.  
"Método para comparar o caractere com UTF8:   
cl\_abap\_file\_utilities=>encoding\_utf8.  
"Método para fazer upload do desktop:  
 cl\_gui\_frontend\_services=>gui\_upload.  
"para pesquisar arquivo no desktop do usuário:   
cl\_gui\_frontend\_services=>file\_open\_dialog.

cl\_gui\_frontend\_services=>directory\_browse.

" mostra a tabela tipo um ALV  
cl\_demo\_output=>display( it\_mara ).

Exemplo baixar arquivo excel para o desktop:

  TYPES: BEGIN OF ty\_output,  
           z\_cc\_maq   TYPE string,  
           z\_cc\_col   TYPE string,  
           z\_ativ\_maq TYPE string,  
         END OF ty\_output.  
  
  DATA: vl\_folder TYPE string,  
        wa\_output TYPE zgt01\_ty\_set\_par,  
        t\_output  TYPE TABLE OF zgt01\_ty\_set\_par.

*\* Escolha do diretório para o Download*  
  CALL METHOD cl\_gui\_frontend\_services=>directory\_browse  
    EXPORTING  
      window\_title         = 'Escolha o diretório'  
    CHANGING  
      selected\_folder      = vl\_folder  
    EXCEPTIONS  
      cntl\_error           = 1  
      error\_no\_gui         = 2  
      not\_supported\_by\_gui = 3  
      OTHERS               = 4.  
  
  IF sy-subrc IS NOT INITIAL.  
  
    *"mssg de erro*  
  
  ELSE.  
  
*\* Monta cabaçalho*  
    CONCATENATE text-027 text-028 text-029  
    INTO wa\_output-parametro SEPARATED BY zgt01\_c\_sep6.  
  
    APPEND wa\_output TO t\_output.  
    CLEAR wa\_output.  
  
    CONCATENATE vl\_folder '/' 'Layout ZCOT2063' '.CSV' INTO DATA(vl\_filename).  
  
*\* Baixa o Arquivo para o diretório selecionado*  
    CALL FUNCTION 'GUI\_DOWNLOAD'  
      EXPORTING  
        filename                = vl\_filename  
        filetype                = 'ASC'  
      TABLES  
        data\_tab                = t\_output  
      EXCEPTIONS  
        file\_write\_error        = 1  
        no\_batch                = 2  
        gui\_refuse\_filetransfer = 3  
        invalid\_type            = 4  
        no\_authority            = 5  
        unknown\_error           = 6  
        header\_not\_allowed      = 7  
        separator\_not\_allowed   = 8  
        filesize\_not\_allowed    = 9  
        header\_too\_long         = 10  
        dp\_error\_create         = 11  
        dp\_error\_send           = 12  
        dp\_error\_write          = 13  
        unknown\_dp\_error        = 14  
        access\_denied           = 15  
        dp\_out\_of\_memory        = 16  
        disk\_full               = 17  
        dp\_timeout              = 18  
        file\_not\_found          = 19  
        dataprovider\_exception  = 20  
        control\_flush\_error     = 21  
        OTHERS                  = 22.  
  
    IF sy-subrc IS INITIAL.  
      DATA(vl\_message) = text-023 && | | && vl\_folder.  
      MESSAGE: vl\_message TYPE zgt01\_c\_type\_msg1.  
    ENDIF.  
  
  ENDIF.

Exemplo: para fazer UPLOAD de um arquivo do diretório/PC.

  DATA: t\_file      TYPE TABLE OF zgt01\_ty\_set\_par,  
        t\_filetable TYPE filetable,  
        t\_zcot2063  TYPE TABLE OF zcot2063.  
  
  DATA: vl\_filename  TYPE string,  
        wa\_filetable TYPE LINE OF filetable,  
        wa\_zcot2063  TYPE zcot2063,  
        vl\_rc        TYPE i,  
        vl\_saida     TYPE string,  
        vl\_answer    TYPE string.

CLEAR: t\_saida[],  
v\_filename.  
  
CALL METHOD cl\_gui\_frontend\_services=>file\_open\_dialog  
  EXPORTING  
    window\_title            = 'Escolha de Arquivo'  
    initial\_directory       = ''  
  CHANGING  
    file\_table              = t\_filetable  
    rc                      = vl\_rc  
  EXCEPTIONS  
    file\_open\_dialog\_failed = 1  
    cntl\_error              = 2  
    error\_no\_gui            = 3  
    not\_supported\_by\_gui    = 4  
    OTHERS                  = 5.  
  
IF sy-subrc IS INITIAL.  
  READ TABLE t\_filetable INTO wa\_filetable INDEX 1.  
  MOVE wa\_filetable TO v\_filename.  
ENDIF.  
  
CALL METHOD cl\_gui\_frontend\_services=>gui\_upload  
  EXPORTING  
    filename                = v\_filename  
    filetype                = 'ASC'  
    has\_field\_separator     = 'X'  
  CHANGING  
    data\_tab                = t\_saida[]  
  EXCEPTIONS  
    file\_open\_error         = 1  
    file\_read\_error         = 2  
    no\_batch                = 3  
    gui\_refuse\_filetransfer = 4  
    invalid\_type            = 5  
    no\_authority            = 6  
    unknown\_error           = 7  
    bad\_data\_format         = 8  
    header\_not\_allowed      = 9  
    separator\_not\_allowed   = 10  
    header\_too\_long         = 11  
    unknown\_dp\_error        = 12  
    access\_denied           = 13  
    dp\_out\_of\_memory        = 14  
    disk\_full               = 15  
    dp\_timeout              = 16  
    not\_supported\_by\_gui    = 17  
    error\_no\_gui            = 18  
    OTHERS                  = 19.  
  
IF NOT sy-subrc IS INITIAL.  
  MESSAGE s398(00) WITH text-e01 DISPLAY LIKE 'E'.  
  LEAVE LIST-PROCESSING.  
  
ELSEIF NOT t\_saida[] IS INITIAL.  
*\*-> Deleta cabeçalho do Arquivo*  
  DELETE t\_saida INDEX 1.  
ENDIF.

“FIM

======================================================================

“INÍCIO

DATA(wa\_sflight) = t\_sflithg[ carrid = 'AA' ].  
  
 IF wa\_sflight IS NOT INITIAL.  
  
   BREAK-POINT.  
  
 ENDIF.

“FIM

======================================================================

“INÍCIO

"CONCATENATE INLINE

DATA(v\_result) = 'Eu'  && | | &&  'Sou' && | | && 'Lindo' .  
  
WRITE: v\_result.

“FIM

======================================================================

“INÍCIO

"Translate letras maiúsculas/ minuscula é só trocar o upper po lower  
  
DATA(v\_ent) = 'ALCIR'.  
  
DATA(v\_saida) = to\_lower( v\_ent ).  
  
WRITE: v\_saida.

“FIM

======================================================================

“INÍCIO

"APPEND INLINE:  
  
TYPES: BEGIN OF ty\_saida,  
        field\_1 TYPE string,  
        field\_2 TYPE string,  
        field\_3 TYPE string,  
        field\_4 TYPE string,  
      END OF ty\_saida.  
  
 DATA: t\_saida TYPE TABLE OF ty\_saida.  
  
DATA(var\_1) = '1'.  
DATA(var\_2) = '2'.  
DATA(var\_3) = '3'.  
  
APPEND VALUE #( field\_1 = var\_1  
                field\_2 = var\_2  
                field\_3 = var\_3 ) TO t\_saida.

“FIM

"trocar ponto por vírugla ou qualquer outro caractere  
  
DATA(v\_ent) = '129,99'.  
  
DATA(v\_saida) = translate( val = v\_ent from = ',' to = '.' ).  
  
WRITE: v\_saida.

“FIM

======================================================================

“INÍCIO

"código de dia útil:    
  
 CLEAR lv\_nr\_dia\_util.  
  
  DO.  
  
    CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
      EXPORTING  
        date                       = lv\_ultil  
        factory\_calendar\_id        = '32'  
        message\_type               = 'E'  
      EXCEPTIONS  
        date\_after\_range           = 1  
        date\_before\_range          = 2  
        date\_invalid               = 3  
        date\_no\_workingday         = 4  
        factory\_calendar\_not\_found = 5  
        message\_type\_invalid       = 6  
        OTHERS                     = 7.  
  
    IF sy-subrc = 0.  
      ADD 1 TO lv\_nr\_dia\_util.  
  
      IF lv\_nr\_dia\_util EQ c\_2.  
        EXIT.  
      ENDIF.  
  
    ENDIF.  
  
    ADD 1 TO lv\_ultil.  
  
  ENDDO.  
  WRITE: 'Data de vencimento: ', lv\_ultil.

“FIM

======================================================================

“INÍCIO

Código para remover botao Executar(F8) de um Report

*\*  DATA vl\_comandos TYPE TABLE OF sy-ucomm.*  
*\**  
*\*  APPEND 'ONLI' TO vl\_comandos.*  
*\**  
*\*  CALL FUNCTION 'RS\_SET\_SELsN\_STATUS'*  
*\*    EXPORTING*  
*\*      p\_status  = sy-pfkey*  
*\*    TABLES*  
*\*      p\_exclude = vl\_comandos.*

“FIM

======================================================================

“INÍCIO

Conversão campo timestampss

GET TIME.  
CONVERT DATE sy-datum TIME sy-uzeit   
INTO TIME STAMP vl\_timestamp   
TIME ZONE c\_timezone.

“FIM

======================================================================

“INÍCIO

*"Conversão Data e Tempo*  
  
*"datas*  
 |{ sy-datum DATE = ENVIRONMENT }|   
  
*"tempo*  
 |{ sy-uzeit TIME = ISO }|.

“FIM

======================================================================

“INÍCIO

Bloqueia tabela ZCOT2063 para edição  
    CALL FUNCTION 'ENQUEUE\_E\_TABLE'  
      EXPORTING  
        mode\_rstable   = 'E'  
        tabname        = 'ZCOT2063'  
      EXCEPTIONS  
        foreign\_lock   = 1  
        system\_failure = 2  
        OTHERS         = 3.  
    IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
    ENDIF.

*\*Desbloqueia tabela ZCOT2063 para edição*  
    CALL FUNCTION 'DEQUEUE\_E\_TABLE'  
      EXPORTING  
        mode\_rstable = 'E'  
        tabname      = 'ZCOT2063'.  
  
    IF sy-subrc IS INITIAL.  
  
    ENDIF.

“FIM

**Links Últeis**

**modificar request liberada**: https://blogs.sap.com/2016/01/06/change-released-task-to-unreleased-modifiable/

**SHDB**: https://servicos.celpe.com.br/servicos-ao-cliente/Pages/consulta-historico-consumo.aspx

**WSDL**: <https://sapyard.com/wsdl-file-in-sap/>

**abap inline com Delmas:** https://connectedlearning.accenture.com/learningboard/861315-abap-inline

**Criar SM30:** <https://www.trailsap.com/dev/sap-dictionary/?topic=table-maintenance>

**Problema/Ajuste de tabelas:** http://www.kodyaz.com/sap-abap/convert-the-table-using-the-database-utility-se14.aspx