```
*&---------------------------------------------------------------------*
*& Include          ZI24_INTERFACES_SALESTOP_F01
*&---------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*& Form f_ajuda_pesquisa
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM f_ajuda_pesquisa .

DATA: l_filename TYPE string,
      it_upload TYPE TABLE OF ty_upload,
      it_filetable TYPE filetable,
      it_local_file TYPE TABLE OF string,
      l_rc         TYPE i,
      lwa_file TYPE string.


  CALL METHOD cl_gui_frontend_services=>file_open_dialog
    EXPORTING
      window_title            = 'File Open'
      initial_directory       = 'C:'
    CHANGING
      file_table              = it_filetable
      rc                      = l_RC
    EXCEPTIONS
      file_open_dialog_failed = 1
      cntl_error              = 2
      error_no_gui            = 3
      OTHERS                  = 4.

  IF sy-subrc = 0.
    READ TABLE it_filetable INDEX 1 INTO l_FILENAME.
    IF sy-subrc IS INITIAL.
    p_arq = l_filename.

   CALL FUNCTION 'GUI_UPLOAD'
    EXPORTING
      filename                = p_arq
    TABLES
      data_tab                = it_file
    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
      OTHERS                  = 17.

    ENDIF.
  ELSE.
      MESSAGE E003. " Error while opening the file
      EXIT.
  ENDIF.


ENDFORM.


*&---------------------------------------------------------------------*
*& Form iniciar_log
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM iniciar_log .

DATA: ls_log TYPE bal_s_log,
      ls_msg_sucess       TYPE bal_s_msg,  " Declaração da estrutura com
sucesso
      ls_msg_error      TYPE bal_s_msg.  " Declaração da estrutura com erro

*       Preenchendo a estrutura com os dados necessários
      ls_log-object    = 'ZACADEMIA'.
      ls_log-subobject = 'OUT24'.
      ls_log-extnumber = sy-cprog.
      ls_log-aldate    = sy-datum.
      ls_log-altime    = sy-uzeit.
      ls_log-aluser    = sy-uname.
      ls_log-alprog    = sy-repid.
      ls_log-altcode   = sy-tcode.


* Preenchendo a estrutura da mensagem de sucesso
ls_msg_sucess-msgty = 'S'.              " Tipo da mensagem
ls_msg_sucess-msgid = 'ZM24_CASE'.      " ID da mensagem (substitua pelo
correto)
ls_msg_sucess-msgno = '000'.            " Número da mensagem

* Preenchendo a estrutura da mensagem de error
```

```
ls_msg_error-msgty = 'S'.               " Tipo da mensagem
ls_msg_error-msgid = 'ZM24_CASE'.       " ID da mensagem (substitua pelo
correto)
ls_msg_error-msgno = '003'.             " Número da mensagem

CALL FUNCTION 'BAL_LOG_CREATE'
  EXPORTING
   i_s_log = ls_log
 IMPORTING
   E_LOG_HANDLE                  = E_log_handle
 EXCEPTIONS
   LOG_HEADER_INCONSISTENT       = 1
   OTHERS                        = 2



          .
IF sy-subrc = 0. "Se a execução anterior for bem sucedida, chamar a função
BAL_LOG_MSG_ADD

 CALL FUNCTION 'BAL_LOG_MSG_ADD'  "Execução da mensagem de sucesso
    EXPORTING
      I_LOG_HANDLE = E_log_handle
      I_S_MSG      = ls_msg_sucess.

 ELSE.

    CALL FUNCTION 'BAL_LOG_MSG_ADD'   "Execução da mensagem de error
    EXPORTING
      I_LOG_HANDLE = E_log_handle
      I_S_MSG      = ls_msg_error.



    EXIT.

ENDIF.




ENDFORM.
*&---------------------------------------------------------------------*
*& Form f_processar_data
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM f_processar_data .
DATA: lv_line TYPE string,
```

```
       lwa_header TYPE ty_header,
       lwa_itens TYPE ty_itens,
       lwa_precos TYPE ty_precos,
       lwa_parceiro TYPE ty_parceiro.


IF it_file is NOT INITIAL.
  LOOP AT it_file INTO lv_line.
    CASE lv_line(1).

     WHEN 'H'.
     SPLIT lv_line AT ';' INTO lwa_header-ident
          lwa_header-agrupador
          lwa_header-doc_tipo
          lwa_header-org_vendas
          lwa_header-canal
          lwa_header-setor
          lwa_header-ext_pedido.
       APPEND lwa_header TO it_header.

     WHEN 'I'.
        SPLIT lv_line AT ';' INTO lwa_itens-ident
          lwa_itens-agrupador
          lwa_itens-posicao
          lwa_itens-material
          lwa_itens-quantidade
          lwa_itens-unid_medida
          lwa_itens-planta.
       APPEND lwa_itens TO it_itens.

     WHEN 'V'.
        SPLIT lv_line AT ';' INTO lwa_precos-ident
          lwa_precos-agrupador
          lwa_precos-posicao
          lwa_precos-preco.
       APPEND lwa_precos TO it_precos.

     WHEN 'P'.
        SPLIT lv_line AT ';' INTO lwa_parceiro-ident
         lwa_parceiro-agrupador
         lwa_parceiro-tipo
         lwa_parceiro-cliente.
       APPEND lwa_parceiro TO it_parceiro.

    ENDCASE.

  ENDLOOP.

ENDIF.



ENDFORM.
```

```
*&---------------------------------------------------------------------*
*& Form f_validar_data
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM f_validar_data .

DATA: ls_header TYPE ty_header,
      ls_msg_error      TYPE bal_s_msg.   " Declaração da estrutura com erro

*LOOP AT it_header INTO ls_header.

 perform f_validar_itens.

 perform f_validar_precos USING ls_header.

 perform f_validar_parceiro USING ls_header.

*ENDLOOP.


IF v_check_itens = abap_true AND v_check_precos = abap_true AND
v_check_parceiros = abap_true.


  perform f_criar_ordem_vendas.


  ELSE.
  CALL FUNCTION 'BAL_LOG_MSG_ADD'  "Execução da mensagem de error
  EXPORTING
    I_LOG_HANDLE = E_log_handle
    I_S_MSG      = ls_msg_error.

    MESSAGE S007.

  EXIT.
ENDIF.




ENDFORM.
*&---------------------------------------------------------------------*
*& Form f_validar_itens
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
```

```
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM f_validar_itens.

DATA: lc_header TYPE ty_header,
      lwa_itens TYPE ty_itens,
      ls_msg_error      TYPE bal_s_msg.   " Declaração da estrutura com erro


LOOP AT it_header INTO lc_header.


* Preenchendo a estrutura da mensagem de error
ls_msg_error-msgty = 'E'.               " Tipo da mensagem
ls_msg_error-msgid = 'ZM24_CASE'.       " ID da mensagem (substitua pelo
correto)
ls_msg_error-msgno = '004'.             " Número da mensagem
ls_msg_error-msgv1 = lc_header-ext_pedido.


LOOP AT it_itens INTO lwa_itens.

READ TABLE it_itens INTO lwa_itens WITH KEY agrupador = lc_header-agrupador
BINARY SEARCH.

IF it_itens IS INITIAL OR ( it_itens IS NOT INITIAL AND lwa_itens-ident IS
INITIAL OR lwa_itens-agrupador IS INITIAL OR lwa_itens-material
  IS INITIAL OR lwa_itens-posicao IS INITIAL OR lwa_itens-quantidade IS
INITIAL OR lwa_itens-unid_medida IS INITIAL OR
  lwa_itens-planta IS INITIAL ).


  CALL FUNCTION 'BAL_LOG_MSG_ADD'  "Execução da mensagem de error
  EXPORTING
    I_LOG_HANDLE = E_log_handle
    I_S_MSG      = ls_msg_error.

  v_check_itens = abap_false.

ELSE.
  v_check_itens = abap_true.

  EXIT.

ENDIF.
ENDLOOP.

ENDLOOP.



ENDFORM.
```

```
*&---------------------------------------------------------------------*
*& Form f_validar_precos
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM f_validar_precos USING ls_header.

DATA: lc_header TYPE ty_header,
      ls_itens TYPE ty_itens,
      lwa_precos TYPE ty_precos,
      ls_msg_error      TYPE bal_s_msg.   " Declaração da estrutura com erro


LOOP AT it_header INTO lc_header.


ls_msg_error-msgty = 'E'.              " Tipo da mensagem
ls_msg_error-msgid = 'ZM24_CASE'.      " ID da mensagem (substitua pelo
correto)
ls_msg_error-msgno = '005'.            " Número da mensagem
ls_msg_error-msgv1 = lc_header-ext_pedido.



LOOP AT it_precos INTO lwa_precos.

READ TABLE it_precos INTO lwa_precos WITH KEY agrupador = lc_header-agrupador
BINARY SEARCH.

  IF it_precos IS INITIAL OR
  ( it_precos IS NOT INITIAL AND
  lwa_precos-ident IS INITIAL OR lwa_precos-agrupador IS INITIAL OR
lwa_precos-posicao IS INITIAL OR
  lwa_precos-preco IS INITIAL ).


  CALL FUNCTION 'BAL_LOG_MSG_ADD'  "Execução da mensagem de error
  EXPORTING
   I_LOG_HANDLE = E_log_handle
   I_S_MSG      = ls_msg_error.

  v_check_itens = abap_false.

  ELSE.
    v_check_precos = abap_true.
    EXIT.

  ENDIF.
ENDLOOP.
ENDLOOP.
```

```
ENDFORM.


FORM f_validar_parceiro USING ls_header.

DATA: lc_header TYPE ty_header,
      lwa_parceiro TYPE ty_parceiro,
      ls_msg_error      TYPE bal_s_msg.   " Declaração da estrutura com erro


LOOP AT it_header INTO lc_header.


ls_msg_error-msgty = 'E'.               " Tipo da mensagem
ls_msg_error-msgid = 'ZM24_CASE'.       " ID da mensagem (substitua pelo
correto)
ls_msg_error-msgno = '006'.             " Número da mensagem
ls_msg_error-msgv1 = lc_header-ext_pedido.


LOOP AT it_parceiro INTO lwa_parceiro.

  READ TABLE it_parceiro INTO lwa_parceiro WITH KEY agrupador =
lc_header-agrupador BINARY SEARCH.

  IF it_parceiro IS INITIAL OR ( it_parceiro IS NOT INITIAL AND
lwa_parceiro-ident IS INITIAL OR lwa_parceiro-cliente IS INITIAL OR
lwa_parceiro-agrupador
  IS INITIAL OR lwa_parceiro-tipo IS INITIAL ).


  CALL FUNCTION 'BAL_LOG_MSG_ADD'  "Execução da mensagem de error
  EXPORTING
   I_LOG_HANDLE = E_log_handle
   I_S_MSG      = ls_msg_error.

  v_check_itens = abap_false.

  ELSE.
    v_check_parceiros = abap_true.
   EXIT.

  ENDIF.
ENDLOOP.
ENDLOOP.

ENDFORM.
*&---------------------------------------------------------------------*
*& Form f_criar_ordem_vendas
*&---------------------------------------------------------------------*
*& text
```

```
*&---------------------------------------------------------------------*
*& -->  p1        text
*& <--  p2        text
*&---------------------------------------------------------------------*
FORM f_criar_ordem_vendas .

WRITE 'TO-DO'.




ENDFORM.
```