

\*--------------------------------------------------------------------\*

\*\*\* LZRLES\_008TOP

\*--------------------------------------------------------------------\*

\*--------------------------------------------------------------------  
\* Constantes  
\*--------------------------------------------------------------------  
CONSTANTS:  
 cc\_uname       TYPE field  
               VALUE 'UNAME',  
 cc\_udate       TYPE field  
               VALUE 'DATUM',  
 cc\_uzeit       TYPE field  
               VALUE 'UZEIT'  
               .  
\*--------------------------------------------------------------------  
\* Variáveis  
\*--------------------------------------------------------------------  
DATA:  
 gw\_name       TYPE field,  
 gw\_value      TYPE value  
 .  
\*--------------------------------------------------------------------  
\* Ponteiros  
\*--------------------------------------------------------------------  
FIELD-SYMBOLS:  
 <fs\_wa>       TYPE any,  
 <fs\_field>    TYPE any  
 .

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

\*--------------------------------------------------------------------\*  
\*\*\*INCLUDE LZRLES\_008F01.  
\*--------------------------------------------------------------------\*  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  ZF\_FORM\_05  
\*&---------------------------------------------------------------------\*  
\*       A cada novo registro inserido na tabela  
\*----------------------------------------------------------------------\*  
FORM zf\_form\_05.  
  
\*--chamada da rotina passando parâmetros  
  PERFORM zf\_log\_new USING x\_header.  
  
ENDFORM.                    "ZF\_FORM\_05  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  ZF\_FORM\_18  
\*&---------------------------------------------------------------------\*  
\*       Ao modificar algum registro da tabela  
\*----------------------------------------------------------------------\*  
FORM zf\_form\_18.  
  
\*--chamada da rotina passando parâmetros  
  PERFORM zf\_log\_modify TABLES extract  
                               total  
                         USING x\_header.  
  
ENDFORM.                    "ZF\_FORM\_18

\*&---------------------------------------------------------------------\*  
\*&      Form  zf\_log\_modify  
\*&---------------------------------------------------------------------\*  
\*       Ao modificar algum registro da tabela  
\*----------------------------------------------------------------------\*  
FORM zf\_log\_modify  TABLES extract  TYPE STANDARD TABLE  
                           total    TYPE STANDARD TABLE  
                     USING x\_header TYPE vimdesc.  
\*--declaração local  
  DATA:  
   o\_dy\_table TYPE REF TO data,  
   o\_dy\_line  TYPE REF TO data,  
  
   lw\_fc      TYPE lvc\_t\_fcat,  
   lc\_change  TYPE c  
  .  
  
  FIELD-SYMBOLS:  
   <fs\_table> TYPE STANDARD TABLE,  
   <fs\_struc> TYPE any,  
  
   <fs\_field>  TYPE any.  
  
  UNASSIGN: <fs\_table>, <fs\_struc>  
  .  
  IF lw\_fc[] IS INITIAL.  
\*--cria a dinamicamente a estrutura da work area.  
    PERFORM zf\_monta\_fieldcat TABLES lw\_fc  
                               USING x\_header-viewname.  
  
\*--cria a dinamicamente a estrutura da work area.  
    PERFORM zf\_monta\_fieldcat TABLES lw\_fc  
                               USING 'VIMFLAGTAB'.  
  ENDIF.  
  
  CALL METHOD cl\_alv\_table\_create=>create\_dynamic\_table  
    EXPORTING  
      it\_fieldcatalog = lw\_fc  
    IMPORTING  
      ep\_table        = o\_dy\_table.  
  
  ASSIGN o\_dy\_table->\* TO  <fs\_table>.  
  
  IF  <fs\_table> IS ASSIGNED.  
    CREATE DATA o\_dy\_line LIKE LINE OF <fs\_table>.  
    ASSIGN o\_dy\_line->\* TO <fs\_struc>.  
  
    IF <fs\_struc> IS ASSIGNED.  
  
      LOOP AT extract INTO <fs\_struc>.  
        UNASSIGN: <fs\_field>.  
        ASSIGN COMPONENT 'ACTION' OF STRUCTURE <fs\_struc> TO <fs\_field> .  
  
        CHECK <fs\_field> IS ASSIGNED.  
        CHECK <fs\_field> = 'U'. "registro modificado  
  
        lc\_change = abap\_true.  
  
        UNASSIGN: <fs\_field>.  
        ASSIGN COMPONENT cc\_UNAME OF STRUCTURE <fs\_struc> TO <fs\_field> .  
        IF <fs\_field> IS ASSIGNED.  
          <fs\_field> = sy-uname.  
        ENDIF.  
  
        UNASSIGN: <fs\_field>.  
        ASSIGN COMPONENT cc\_UDATE OF STRUCTURE <fs\_struc> TO <fs\_field> .  
        IF <fs\_field> IS ASSIGNED.  
          <fs\_field> = sy-datum.  
        ENDIF.  
  
        UNASSIGN: <fs\_field>.  
        ASSIGN COMPONENT cc\_UZEIT OF STRUCTURE <fs\_struc> TO <fs\_field> .  
        IF <fs\_field> IS ASSIGNED.  
          <fs\_field> = sy-uzeit.  
        ENDIF.  
  
        MODIFY extract FROM <fs\_struc>.  
      ENDLOOP.  
    ENDIF  
   .  
  ENDIF  
  .  
\*--somente se algum registro foi modificado.  
  IF lc\_change IS NOT INITIAL.  
    total[] = extract[].  
  ENDIF.  
ENDFORM.                    "zf\_log\_modify  
\*&---------------------------------------------------------------------\*  
\*&      Form  ZF\_LOG\_NEW  
\*&---------------------------------------------------------------------\*  
\*       A cada novo registro inserido na tabela  
\*----------------------------------------------------------------------\*  
FORM zf\_log\_new USING x\_header TYPE vimdesc.  
  
  CLEAR: gw\_name, gw\_value.  
  UNASSIGN: <fs\_field>, <fs\_wa>  
  .  
  ASSIGN (x\_header-viewname) TO <fs\_wa>.  
  
  DO 3 TIMES.  
    CASE sy-index.  
      WHEN 1.  
        gw\_name = cc\_uname.  
        gw\_value = sy-uname.  
      WHEN 2.  
        gw\_name = cc\_udate.  
        gw\_value = sy-datum.  
      WHEN OTHERS.  
        gw\_name = cc\_uzeit.  
        gw\_value = sy-uzeit.  
    ENDCASE.  
  
    ASSIGN COMPONENT gw\_name OF STRUCTURE <fs\_wa> TO <fs\_field>.  
    IF sy-subrc = 0.  
      <fs\_field> = gw\_value.  
    ENDIF.  
  ENDDO.  
  
ENDFORM.                    "ZF\_LOG\_NEW  
\*&---------------------------------------------------------------------\*  
\*&      Form  ZF\_MONTA\_FIELDCAT  
\*&---------------------------------------------------------------------\*  
FORM zf\_monta\_fieldcat  TABLES   t\_fc STRUCTURE lvc\_s\_fcat  
                         USING   p\_st TYPE any.  
\*--objetos locais  
  DATA:  
    ls\_fc        TYPE lvc\_s\_fcat,  
    lc\_tabname   TYPE tabfield-tabname,  
  
    lw\_dfies\_tab TYPE TABLE OF dfies,  
    ls\_dfies\_tab LIKE LINE OF lw\_dfies\_tab  
    .  
  CLEAR: lc\_tabname, lw\_dfies\_tab[].  
  
  lc\_tabname = p\_st.  
  CALL FUNCTION 'DDIF\_FIELDINFO\_GET'  
    EXPORTING  
      tabname        = lc\_tabname  
      langu          = sy-langu  
      all\_types      = 'X'  
    TABLES  
      dfies\_tab      = lw\_dfies\_tab  
    EXCEPTIONS  
      not\_found      = 1  
      internal\_error = 2  
      OTHERS         = 3.  
  
  IF sy-subrc <> 0.  
    RETURN.  
  ENDIF.  
  
  LOOP AT lw\_dfies\_tab INTO ls\_dfies\_tab.  
    CLEAR  ls\_fc.  
    ls\_fc-fieldname = ls\_dfies\_tab-fieldname.  
    ls\_fc-datatype  = ls\_dfies\_tab-datatype.  
    ls\_fc-inttype   = ls\_dfies\_tab-inttype.  
    ls\_fc-intlen    = ls\_dfies\_tab-leng.  
    ls\_fc-decimals  = ls\_dfies\_tab-decimals.  
    APPEND ls\_fc TO t\_fc.  
  ENDLOOP.  
  
ENDFORM.                    " ZF\_MONTA\_FIELDCAT