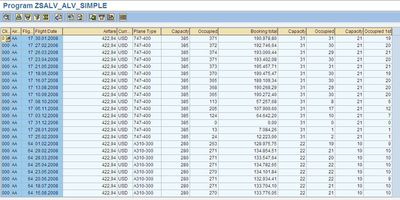
**[Relatório ALV](http://guilhermerabelosap.spaceblog.com.br/1528261/Relatorio-ALV/" \o "Relatório ALV)**

[](http://perlbal.hi-pi.com/blog-images/836792/gd/13111927651/Relatorio-ALV.jpg)

**ALV** - **A**bap **L**ist **V**iew  - Basicamente uma função Standard que pega a sua tabela interna de dados e transforma em um relatório. Existem algumas funcionalidades como gerar um arquivo Excel, gráficos, somatórias de campos, "links" para chamar uma outra transação, etc.. etc..

 Criando um Relatório ALV.

Nesse meu exemplo Irei usar somente tabelas STANDARD  
-------------------------------------------------------------------------------------------------------------------------------------------

REPORT  Zteste\_Alv MESSAGE-ID sy LINE-SIZE 140   "Coluna  
                                       LINE-COUNT 100. "Linha  
  
\*----------------------------------------------------------------------\*  
\*                       TABELAS STANDARD R/3  
\*----------------------------------------------------------------------\*  
  
TABLES: ekko,  
          ekpo,  
          makt.  
  
\*----------------------------------------------------------------------\*  
\* Types Globais  
\*----------------------------------------------------------------------\*  
  
TYPES: BEGIN OF y\_ekko,  
        ebeln  TYPE ekko-ebeln,  
END OF y\_ekko.  
  
TYPES: BEGIN OF y\_ekpo,  
         ebeln  TYPE ekpo-ebeln,  
         ebelp  TYPE ekpo-ebelp,  
         matnr  TYPE ekpo-matnr.  
TYPES: END OF y\_ekpo.  
  
TYPES: BEGIN OF y\_makt,  
        matnr  TYPE makt-matnr,  
        maktx  TYPE makt-maktx.  
TYPES: END OF y\_makt.  
  
\*Criação de type de saida de dados

TYPES : BEGIN OF y\_saida,  
         ebeln  TYPE ekpo-ebeln,  
         ebelp  TYPE ekpo-ebelp,  
         matnr   TYPE ekpo-matnr,  
         maktx   TYPE makt-maktx.  
TYPES: END OF y\_saida.  
  
\* ALV

TYPE-POOLS: slis,   "Tabela para utilização do ALV.  
              kkblo. "Tabela para utilização do ALV.  
   
\*----------------------------------------------------------------------\*  
\*                   Tabelas Globais  
\*----------------------------------------------------------------------\*  
  
DATA:  t\_ekko   TYPE TABLE OF y\_ekko,  
  t\_ekpo   TYPE TABLE OF y\_ekpo,  
  t\_makt   TYPE TABLE OF y\_makt,  
  t\_saida  TYPE TABLE OF y\_saida.  
  
\* ALV  
DATA: t\_fcat          TYPE TABLE OF slis\_fieldcat\_alv,  
       t\_sort          TYPE TABLE OF slis\_sortinfo\_alv,  
       t\_top\_of\_page  TYPE TABLE OF slis\_listheader.  
  
\*----------------------------------------------------------------------\*  
\*                       Work-Areas Globais  
\*----------------------------------------------------------------------\*  
  
DATA: wa\_ekko   TYPE y\_ekko,  
  wa\_ekpo   TYPE y\_ekpo,  
  wa\_makt   TYPE y\_makt,  
  wa\_saida  TYPE y\_saida.  
  
\* ALV

DATA: wa\_fcat         TYPE slis\_fieldcat\_alv,  
       wa\_sort           TYPE slis\_sortinfo\_alv,  
       wa\_top\_of\_page  TYPE slis\_listheader,  
       wa\_layout        TYPE slis\_layout\_alv.  
  
\*----------------------------------------------------------------------\*  
\*                       Tela de Seleção  
\*----------------------------------------------------------------------\*  
  
SELECTION-SCREEN: BEGIN OF BLOCK block1 WITH FRAME TITLE text-001. "Critério de Seleção

SELECT-OPTIONS: s\_ebeln FOR ekko-ebeln.

SELECTION-SCREEN: END OF BLOCK block1.  
  
\*----------------------------------------------------------------------\*  
\*                         Start-of-Selection  
\*----------------------------------------------------------------------\*  
  
START-OF-SELECTION.  
  
  PERFORM seleciona\_dados.

  PERFORM monta\_saida.  
  
  PERFORM imprimir\_alv.

END-OF-SELECTION.

\*----------------------------------------------------------------------\*  
\*                         End-of-Selection  
\*----------------------------------------------------------------------\*

\*----------------------------------------------------------------------\*  
\*                         Forms  
\*----------------------------------------------------------------------\*  
  
\*----------------------------------------------------------------------\*  
\* Form SELECIONA\_DADOS  
\*----------------------------------------------------------------------\*  
\* Rotina para Selecionar todos os Dados  
\*----------------------------------------------------------------------\*  
FORM seleciona\_dados .  
  
  SELECT  ebeln  
     FROM ekko  
     INTO TABLE t\_ekko  
     WHERE ebeln IN s\_ebeln.  
  
  IF sy-subrc <> 0.  
     MESSAGE s002(sy) WITH 'dados nao encontrados' DISPLAY LIKE 'E'.  
     LEAVE LIST-PROCESSING.  
  ENDIF.  
  
  IF NOT t\_ekko[] IS INITIAL.  
  
    SELECT ebeln ebelp matnr  
       FROM ekpo  
       INTO TABLE t\_ekpo  
       FOR ALL ENTRIES IN t\_ekko  
       WHERE ebeln = t\_ekko-ebeln.  
  
  ENDIF.  
  
  
  IF NOT t\_ekpo[] IS INITIAL.  
  
    SELECT matnr maktx  
       FROM makt  
       INTO TABLE t\_makt  
       FOR ALL ENTRIES IN t\_ekpo  
       WHERE matnr = t\_ekpo-matnr.  
  
  ENDIF.  
  
ENDFORM.                    " SELECIONA\_DADOS  
  
\*----------------------------------------------------------------------\*  
\* Form MONTA\_SAIDA  
\*----------------------------------------------------------------------\*  
\* Rotina para montar a tabela T\_SAIDA  
\*----------------------------------------------------------------------\*  
FORM monta\_saida.  
  
  LOOP AT t\_ekpo INTO wa\_ekpo.  
  
     CLEAR: wa\_ekko.  
     READ TABLE t\_ekko INTO wa\_ekko WITH KEY ebeln = wa\_ekpo-ebeln.  
  
     CLEAR: wa\_makt.  
     READ TABLE t\_makt INTO wa\_makt WITH KEY matnr = wa\_ekpo-matnr.  
  
     IF sy-subrc = 0.  
       MOVE wa\_ekko-ebeln TO wa\_saida-ebeln.  
       MOVE wa\_ekpo-ebelp TO wa\_saida-ebelp.  
       MOVE wa\_makt-matnr TO wa\_saida-matnr.  
       MOVE wa\_makt-maktx TO wa\_saida-maktx.  
  
       APPEND wa\_saida TO t\_saida.  
       CLEAR: wa\_saida.  
     ENDIF.  
  
  ENDLOOP.  
  
ENDFORM.                    " MONTA\_SAIDA  
  
  
\*----------------------------------------------------------------------\*  
\* Form IMPRIMIR\_ALV  
\*----------------------------------------------------------------------\*  
\* Rotina para imprimir o relatório em formato ALV  
\*----------------------------------------------------------------------\*  
FORM imprimir\_alv.  
  
  IF NOT t\_saida[] IS INITIAL.  
  
   SORT t\_saida BY ebeln ebelp ASCENDING.

\* ("ASCENDING  - Crecente) ou  ("DESCENDING – Decrecente)  
  
      PERFORM zf\_monta\_layout.  
      PERFORM zf\_monta\_fieldcat.  
     PERFORM zf\_mostra\_alv.  
  
  ENDIF.  
  
ENDFORM.                    " IMPRIMIR\_ALV  
  
\*----------------------------------------------------------------------\*  
\* Form zf\_monta\_layout  
\*----------------------------------------------------------------------\*  
\* Rotina para montar o Layout de saida.  
\*----------------------------------------------------------------------\*  
FORM zf\_monta\_layout.  
  
\*Imprimir linha clara e escura alternadamente  
  wa\_layout-zebra             = 'X'.  
  
\*Abre a lista com as colunas do tamanho necessário para mostragem  
  wa\_layout-colwidth\_optimize = 'X'.  
  
ENDFORM.                    " zf\_monta\_layout  
  
\*----------------------------------------------------------------------\*  
\* Form zf\_monta\_fieldcat  
\*----------------------------------------------------------------------\*  
\* Rotina para Montar a Tabela Fieldcat.  
\*----------------------------------------------------------------------\*  
FORM zf\_monta\_fieldcat.  
  
  PERFORM zf\_fieldcat USING: 'T\_SAIDA' 'EBELN' 'Doc. Compra',  
                              'T\_SAIDA' 'EBELP'  'Item',  
                              'T\_SAIDA' 'MATNR' 'Material',  
                              'T\_SAIDA' 'MAKTX' 'Descrição Material'.  
  
ENDFORM.                    " zf\_monta\_fieldcat  
  
\*----------------------------------------------------------------------\*  
\* Form zf\_fieldcat  
\*----------------------------------------------------------------------\*  
\* Paramêtros que será usados na estrutura fielcat.  
\*----------------------------------------------------------------------\*  
\*      --> P\_TABELA:Paramêtro com o nome da tabela.  
\*      --> P\_CAMPO: Paramêtro com o nome do campo.  
\*      --> P\_TEXTO:  Paramêtro com o nome do texto.  
\*----------------------------------------------------------------------\*  
FORM zf\_fieldcat

 USING p\_tabela  TYPE  any  
             p\_campo   TYPE  any  
             p\_texto   TYPE  any.  
  
   wa\_fcat-tabname        = p\_tabela.  
   wa\_fcat-fieldname      = p\_campo.  
   wa\_fcat-reptext\_ddic  = p\_texto.  
  
    APPEND wa\_fcat TO t\_fcat.  
   CLEAR  wa\_fcat.  
ENDFORM.                    " zf\_fieldcat  
  
\*----------------------------------------------------------------------\*  
\* Form zf\_mostra\_alv  
\*----------------------------------------------------------------------\*  
\* Função para mostrar o Relatório ALV  
\*----------------------------------------------------------------------\*  
FORM zf\_mostra\_alv.  
  
  DATA: vl\_repid TYPE sy-repid.  
  
  vl\_repid = sy-repid.  
  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
    EXPORTING  
      i\_callback\_program = vl\_repid  
      is\_layout           = wa\_layout  
      it\_fieldcat         = t\_fcat  
      i\_default           = 'X'  
    TABLES  
      t\_outtab            = t\_saida  
    EXCEPTIONS  
      program\_error       = 1  
      OTHERS              = 2.  
  
  IF NOT sy-subrc IS INITIAL.  
     MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  
ENDFORM.                    " zf\_mostra\_alv