

GRAFICO 3D

UDERSON LUIS FERMINO

Sumario:

Este artigo descreva a utilização de geração de gráficos em 3D, descrevendo de forma simplificada ao leitor.

Sobre o Autor:

Uderson Luis Fermino, formado em Ciências da Computação pela Faculdade de Pesquisa e Ensino IPEP, atua no mercado a 2 anos como desenvolvedor Java nas plataformas: (J2SE, J2EE e J2ME), com participação em grandes projetos envolvendo estas tecnologias. É consultor ABAP com experiências em REPORT, ALV (GRID, LIST, BLOCK, OO, TREE, HIERARQUICK), IDOC, ALE, ONLINE, SAPSCRIPT, SMARTFORM, NETWEAVER (JCO, BSP, WebDynpro).

Email:

Uderson@gmail.com

GRAFICO 3D

UDERSON LUIS FERMINO

GRAPH_MATRIX_3D

Esta função é responsável por gerar gráficos no SAP em 3D, ela contém diversos parâmetros, porém para exemplificar o uso da mesma será descrito apenas o parâmetro do exemplo.

```
CALL FUNCTION 'GRAPH_MATRIX_3D'
  EXPORTING
    col1 = 'Gasolina'
    col2 = 'Alcool'
    col3 = 'Diesel'
    titl = 'Quantidade de Combustíveis Vendidos'
  TABLES
    data = itab_data
    opts = itab_options
  EXCEPTIONS
    OTHERS = 1.
```

Geraremos um Gráfico contendo quantidades de combustíveis vendido durante um período, este exemplo foi retirado de um case, onde temos empresa distribuidora de petróleo que possui determinadas filiais onde cada filial vende matérias do tipo: gasolina, álcool e diesel.

Para gerar os valores teremos os será declarada uma tabela interna contendo a seguinte estrutura:

```
DATA: BEGIN OF itab_data OCCURS 0,
      filial      TYPE c LENGTH 4,      "Filial
      tvolzmgas   TYPE p DECIMALS 4,   "Volume em litros de Gas
      tvolzmalc   TYPE p DECIMALS 4,   "Volume em litros de Alc
      tvolzmdie   TYPE p DECIMALS 4,   "Volume em litros de Die
    END OF itab_data,
```

Esta tabela conterá os seguintes informações:

Filial será o campos chave, e os demais campos conterá os valores de cada produto.

Aqui será exemplificado o carregamento destes dados (não explicaremos a lógica), pois o que interessa são os dados que será gerados para criação e exibição do gráfico.

```
DATA: BEGIN OF itab_data OCCURS 0,
      matnr      LIKE zhistped-matnr,
      tvolzm1    TYPE p DECIMALS 4,
      tvolzm2    TYPE p DECIMALS 4,
      tvolzm3    TYPE p DECIMALS 4,
    END OF itab_data,
```

GRAFICO 3D

UDERSON LUIS FERMINO

```
BEGIN OF itab_options OCCURS 0,
    option(20),
END OF itab_options.

* Consulta de Pedidos
DATA: BEGIN OF zfat_ped OCCURS 0.
    INCLUDE STRUCTURE zhistped.
DATA:   cf      TYPE p DECIMALS 4,
       cfr     TYPE p DECIMALS 4,
       tvolzm  TYPE p DECIMALS 4.
DATA: END OF zfat_ped.

* Tabela de Produtos
DATA: BEGIN OF zprodutos OCCURS 0,
    matnr      LIKE zhistped-matnr,
    maktx      LIKE makt-maktx,
    kwmeng     TYPE p DECIMALS 3,
    kbetr      LIKE konp-kbetr,
    prc        TYPE p DECIMALS 3,
    mrg        TYPE p DECIMALS 3,
END OF zprodutos.

TYPES: BEGIN OF st_prod,
    vkorg      LIKE zhistped-vkorg,
    matnr      LIKE zhistped-matnr,
    maktx      LIKE makt-maktx,
    kwmeng     LIKE zhistped-kwmeng,
    kbetr      LIKE konp-kbetr,
    tvolzm     TYPE p DECIMALS 4,
END OF st_prod.

DATA: tp_prod TYPE st_prod.

DATA: collect_prod TYPE HASHED TABLE OF st_prod
    WITH UNIQUE KEY vkorg matnr maktx WITH HEADER LINE.

*-----
--
* Variáveis
*-----
--
DATA: r_tot_gas      TYPE p DECIMALS 3,
     r_tot_alc      TYPE p DECIMALS 3,
     r_tot_die      TYPE p DECIMALS 3.

*-----
--
* Declaração de Parâmetros
*-----
--
SELECTION-SCREEN BEGIN OF BLOCK b01 WITH FRAME.

SELECT-OPTIONS so_data FOR sy-datum DEFAULT sy-datum.

SELECTION-SCREEN END OF BLOCK b01.

*-----
--
```

GRAFICO 3D

UDERSON LUIS FERMINO

```
* Evento: Start-of-selection
*-----
--
START-OF-SELECTION.

    PERFORM: zf_geral.

END-OF-SELECTION.

*****
*
*** Form: zf_check_data
*** Descrição: Verifica se os parâmetros obrigatórios da tela estão
*** preenchidos corretamente
*** Entrada: N/A
*** Saída: N/A
*****
*
FORM zf_check_data .

    IF so_data-low IS INITIAL.
        so_data-low = sy-datum.
    ENDIF.

    IF so_data-high IS INITIAL.
        so_data-high = so_data-low.
    ENDIF.

ENDFORM.                " zf_check_data

*****
*
*** Form: zf_pedidos_total
*** Descrição: Totalização dos Pedidos por Base, Produto e Regional
*** Entrada: N/A
*** Saída: N/A
*****
*
FORM zf_pedidos_total.

* Carrega tabela produtos
  PERFORM: soma_vol_prod.

ENDFORM.                " zf_pedidos_total

*****
*
*** Form: soma_vol_prod
*** Descrição: Carrega Tabela Produtos
*** Entrada: N/A
*** Saída: N/A
*****
*
FORM soma_vol_prod.
```

GRAFICO 3D

UDERSON LUIS FERMINO

```
LOOP AT zfat_ped.
*   Sumariza
   MOVE-CORRESPONDING zfat_ped TO tp_prod.
   COLLECT tp_prod INTO collect_prod.
ENDLOOP.

SORT collect_prod BY vkorg matnr.

LOOP AT collect_prod.

   IF collect_prod-matnr = '0000000000000000101' OR
      collect_prod-matnr = '0000000000000000102' OR
      collect_prod-matnr = '0000000000000000103'.
      r_tot_gas = r_tot_gas + collect_prod-tvolzm.
   ENDIF.

   IF collect_prod-matnr = '0000000000000000109'.
      r_tot_alc = r_tot_alc + collect_prod-tvolzm.
   ENDIF.

   IF collect_prod-matnr = '0000000000000000104' OR
      collect_prod-matnr = '0000000000000000105' OR
      collect_prod-matnr = '0000000000000000106' OR
      collect_prod-matnr = '0000000000000000107' OR
      collect_prod-matnr = '0000000000000000121' OR
      collect_prod-matnr = '0000000000000000124' OR
      collect_prod-matnr = '0000000000000000125' OR
      collect_prod-matnr = '0000000000000000126' OR
      collect_prod-matnr = '0000000000000000127' OR
      collect_prod-matnr = '0000000000000000130' OR
      collect_prod-matnr = '0000000000000000131' OR
      collect_prod-matnr = '0000000000000000132' OR
      collect_prod-matnr = '0000000000000000133'.
      r_tot_die = r_tot_die + collect_prod-tvolzm.
   ENDIF.

   AT END OF vkorg.
      itab_data-matnr = collect_prod-vkorg.
      itab_data-tvolzm1 = r_tot_gas.
      itab_data-tvolzm2 = r_tot_alc.
      itab_data-tvolzm3 = r_tot_die.
      append itab_data.

      clear: r_tot_gas, r_tot_alc, r_tot_die.

ENDAT.

ENDLOOP.

ENDFORM.                " soma_vol_prod
*&-----
-*
*&      Form  zf_geral
*&-----
-*
*      text
*-----
```

GRAFICO 3D

UDERSON LUIS FERMINO

```
_*
FORM zf_geral.

*** Inicio - 06.05.2008 (Chamado: 0095 ) - Raphael Maia -
Sonda Procwork.

DATA: iw_vbrk_del TYPE STANDARD TABLE OF zvbrk_del_type,
      is_vbrk_del TYPE zvbrk_del_type,
      iw_vbfa     TYPE STANDARD TABLE OF vbfa_type,
      is_vbfa     TYPE vbfa_type.

*** Fim - 06.05.2008 (Chamado: 0095 ) - Raphael Maia - Sonda Procwork.

* Checagem da data ou período para consulta
PERFORM: zf_check_data.

* Função de seleção de dados

CALL FUNCTION 'YRFSD_HISTORICOPEDFAT6'
  EXPORTING
    ibegda = so_data-low
    iendda = so_data-high
  TABLES
    zfat   = zfat_ped.

DELETE ADJACENT DUPLICATES FROM zfat_ped.

*** Inicio - 06.05.2008 (Chamado: 0095 ) - Raphael Maia -
Sonda Procwork.
*** Tentando igualar o retorno do ysd015c ao ycom004u.

*** Limpa TI de zhistcomu
REFRESH iw_vbrk_del.

*** Busca registros a serem apagados.
SELECT rk~vbeln
       rk~fkdat
       rk~fkart
       fa~vbelv
       rp~vbeln
       rp~vkbur
       rp~vkgrp
INTO TABLE iw_vbrk_del
FROM vbrk AS rk
LEFT JOIN vbfa AS fa ON rk~vbeln EQ fa~vbelv
LEFT JOIN vbrp AS rp ON rk~vbeln EQ rp~vbeln
WHERE rk~fkdat   IN so_data.

*** Se seleção retornar resultados
IF NOT iw_vbrk_del IS INITIAL.

*** Apaga registros duplicados.
DELETE ADJACENT DUPLICATES FROM iw_vbrk_del.

*** Ordena tabelas.
SORT: iw_vbrk_del BY vbelnk.

*** Retirado pois afetava a ordenação por células.
```

GRAFICO 3D

UDERSON LUIS FERMINO

```
***          zfat_ped BY vbeln.

*** Limpa auxiliar da TI de historico.
CLEAR is_vbrk_del.

*** Percorre tabela de histórico apagando registros de zfat_ped.
LOOP AT iw_vbrk_del INTO is_vbrk_del.

    IF is_vbrk_del-fkart EQ 'YDVE' OR
       is_vbrk_del-fkart EQ 'YVOC' OR
       is_vbrk_del-fkart EQ 'YDVC' OR
       is_vbrk_del-fkart EQ 'YVOE' OR
       is_vbrk_del-fkart EQ 'S1'.

        DELETE zfat_ped WHERE vbeln = is_vbrk_del-vbelnk.

    ELSEIF is_vbrk_del-vkbur EQ 'SERV'.

        DELETE zfat_ped WHERE vbeln = is_vbrk_del-vbelnk.

    ELSEIF is_vbrk_del-vkgrp EQ '900'.

*** As redes estavam sendo zeradas.
*       is_vbrk_del-vkgrp EQ '901' OR
*       is_vbrk_del-vkgrp EQ '902'.

        DELETE zfat_ped WHERE vbeln = is_vbrk_del-vbelnk.

    ENDIF.

ENDLOOP.

DELETE iw_vbrk_del WHERE vbelva IS INITIAL.

IF NOT iw_vbrk_del IS INITIAL.

    REFRESH iw_vbfa.

    SELECT fa~vbelv
           fa~vbeln
           rk~fkart
    INTO TABLE iw_vbfa
    FROM vbfa AS fa
    INNER JOIN vbrk AS rk ON fa~vbeln EQ rk~vbeln
    FOR ALL ENTRIES IN iw_vbrk_del
    WHERE fa~vbelv EQ iw_vbrk_del-vbelva.

    IF NOT iw_vbfa IS INITIAL.

        SORT iw_vbfa BY vbelv vbeln.

        LOOP AT iw_vbfa INTO is_vbfa.

            IF is_vbfa-fkart EQ 'YDVE' OR
               is_vbfa-fkart EQ 'YVOC' OR
               is_vbfa-fkart EQ 'YDVC' OR
               is_vbfa-fkart EQ 'YVOE' OR
               is_vbfa-fkart EQ 'S1'.


```

GRAFICO 3D

UDERSON LUIS FERMINO

```
DELETE zfat_ped WHERE vbeln = is_vbfa-vbelv.

ENDIF.

ENDLOOP.

ENDIF.

ENDIF.

ENDIF.

DATA: wcf TYPE p DECIMALS 4.
LOOP AT zfat_ped.
    zfat_ped-tvolzm = zfat_ped-kwmeng * zfat_ped-zmrp.
    wcf = ( zfat_ped-zpvs * ( zfat_ped-zada / 10 ) ).
    zfat_ped-cf = ( ( wcf / 100 ) * zfat_ped-kwmeng ).
    MODIFY zfat_ped TRANSPORTING cf tvolzm.
ENDLOOP.
* Totalização de pedidos por base, produto e regional.
PERFORM: zf_pedidos_total,
        zf_grafico.

ENDFORM.                                " zf_geral

*&-----
-*
*&      Form  zf_geral
*&-----
-*
*      text
*-----
-*
FORM zf_grafico.

CALL FUNCTION 'GRAPH_MATRIX_3D'
EXPORTING
    col1 = 'Gasolina'
    col2 = 'Alcool'
    col3 = 'Diesel'
    titl = 'Quantidade de Combustivéis Vendidos'
TABLES
    data = itab_data
    opts = itab_options
EXCEPTIONS
    OTHERS = 1.

ENDFORM.                                " zf_geral

Vamos endender o perform soma_vol_prod:
```

GRAFICO 3D

UDERSON LUIS FERMINO

Este perform depois de coletar todos os pedidos (SD) faturados (vendidos) ele separa os materiais somando os valores vendidos por filial e material, esta lógica é simples usando o comando collect:

```
LOOP AT zfat_ped.  
*   Sumariza  
    MOVE-CORRESPONDING zfat_ped TO tp_prod.  
    COLLECT tp_prod INTO collect_prod.  
ENDLOOP.
```

Para que o comando collect possa efetuar a sumarização correta é necessário declarar as chaves da tabela:

```
TYPES: BEGIN OF st_prod,  
        vkorg    LIKE zhistped-vkorg,  
        matnr    LIKE zhistped-matnr, "Nº do Produto  
        maktx    LIKE makt-maktx,    "Descrição  
        kwmeng   LIKE zhistped-kwmeng, "Volume  
        kbetr    LIKE konp-kbetr,    "Porcentagem  
        tvolzm   TYPE p DECIMALS 4,  
END OF st_prod.  
  
DATA: tp_prod TYPE st_prod.  
  
DATA: collect_prod TYPE HASHED TABLE OF st_prod  
        WITH UNIQUE KEY vkorg matnr maktx WITH HEADER LINE.
```

GRAFICO 3D

UDERSON LUIS FERMINO

Após sumarizar os valores, é verificado o tipo de material e somado os valores para cada filial.

```
LOOP AT collect_prod.

  IF collect_prod-matnr = '0000000000000000101' OR
     collect_prod-matnr = '0000000000000000102' OR
     collect_prod-matnr = '0000000000000000103'.
     r_tot_gas = r_tot_gas + collect_prod-tvolzm.
  ENDIF.

  IF collect_prod-matnr = '0000000000000000109'.
     r_tot_alc = r_tot_alc + collect_prod-tvolzm.
  ENDIF.

  IF collect_prod-matnr = '0000000000000000104' OR
     collect_prod-matnr = '0000000000000000105' OR
     collect_prod-matnr = '0000000000000000106' OR
     collect_prod-matnr = '0000000000000000107' OR
     collect_prod-matnr = '0000000000000000121' OR
     collect_prod-matnr = '0000000000000000124' OR
     collect_prod-matnr = '0000000000000000125' OR
     collect_prod-matnr = '0000000000000000126' OR
     collect_prod-matnr = '0000000000000000127' OR
     collect_prod-matnr = '0000000000000000130' OR
     collect_prod-matnr = '0000000000000000131' OR
     collect_prod-matnr = '0000000000000000132' OR
     collect_prod-matnr = '0000000000000000133'.
     r_tot_die = r_tot_die + collect_prod-tvolzm.
  ENDIF.

  AT END OF vkorg.
    itab_data-matnr = collect_prod-vkorg.
    itab_data-tvolzm1 = r_tot_gas.
    itab_data-tvolzm2 = r_tot_alc.
    itab_data-tvolzm3 = r_tot_die.
    append itab_data.

  clear: r_tot_gas, r_tot_alc, r_tot_die.

ENDAT.
```

Quando alterar a filial é inseridos os valores totais na tabela ficando +-:

GRAFICO 3D UDERSON LUIS FERMINO

FILIAL	VOLUME GAS	VOLUME ALC	VOLUME DIE
FI01	200.000,00	125.000,00	152.223,00
FI02	252.000,00	342.000,00	87.223,00

```
CALL FUNCTION 'GRAPH_MATRIX_3D'
  EXPORTING
    coll1 = 'Gasolina'
    coll2 = 'Alcool'
    coll3 = 'Diesel'
    titl  = 'Quantidade de Combustivéis Vendidos'
  TABLES
    data = itab_data
    opts = itab_options
  EXCEPTIONS
    OTHERS = 1.
```

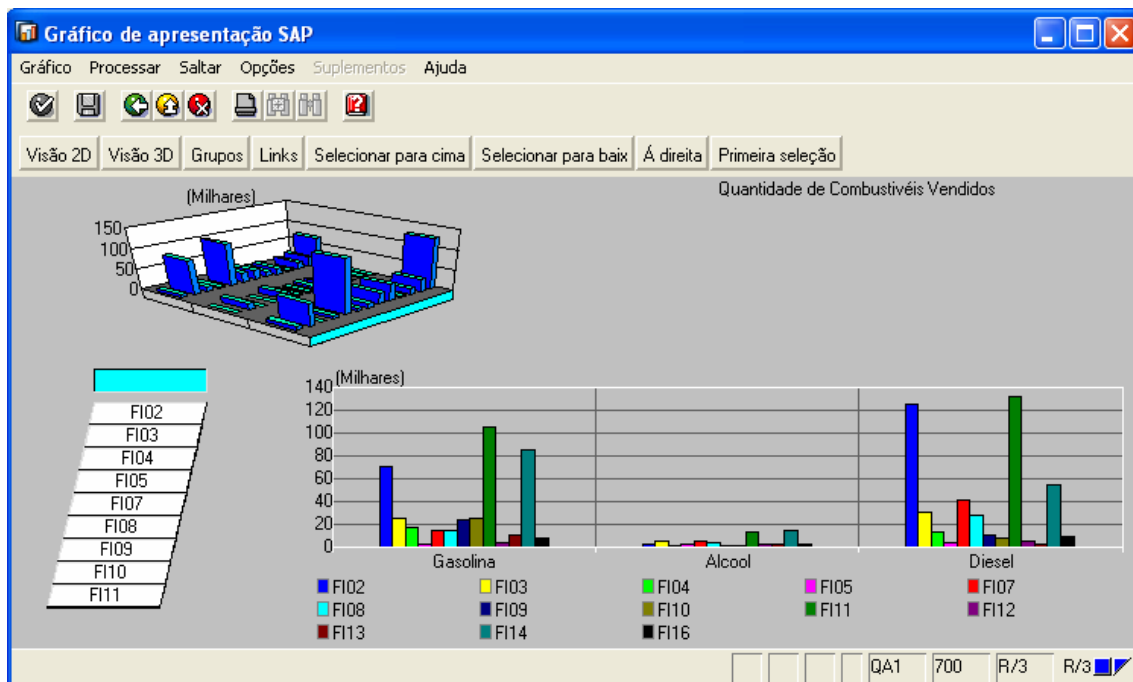
COL1 = Titulo da coluna 1

COL2 = Titulo da coluna 2

COL3 = Titulo da coluna 3

TITL = Titulo do gráfico

Ao executar teremos:



Observe o gráfico gerou os dados conformes as chaves, onde neste exemplo tenho 16 filiais onde todos vendem os 3 materiais, para cada matéria contém os valores (volume) de combustíveis vendidos por cada filial, o próprio sap (gráficos) fornece visões diferenciadas:

GRAFICO 3D

UDERSON LUIS FERMINO

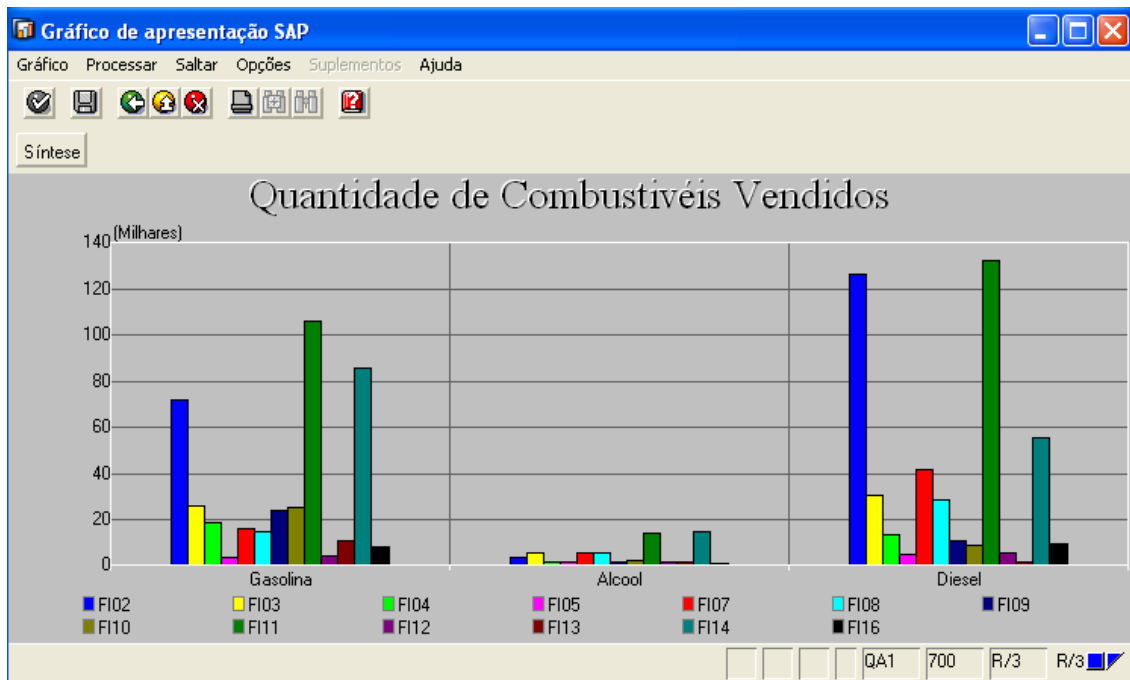


GRAFICO 3D

UDERSON LUIS FERMINO

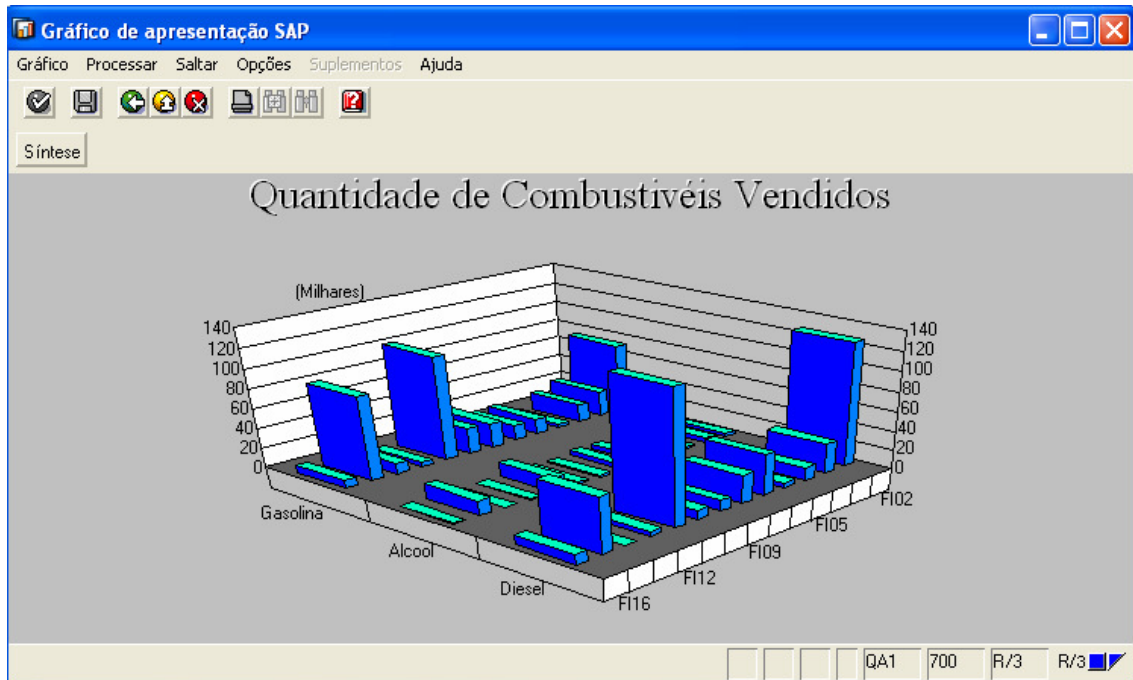


GRAFICO 3D UDERSON LUIS FERMINO

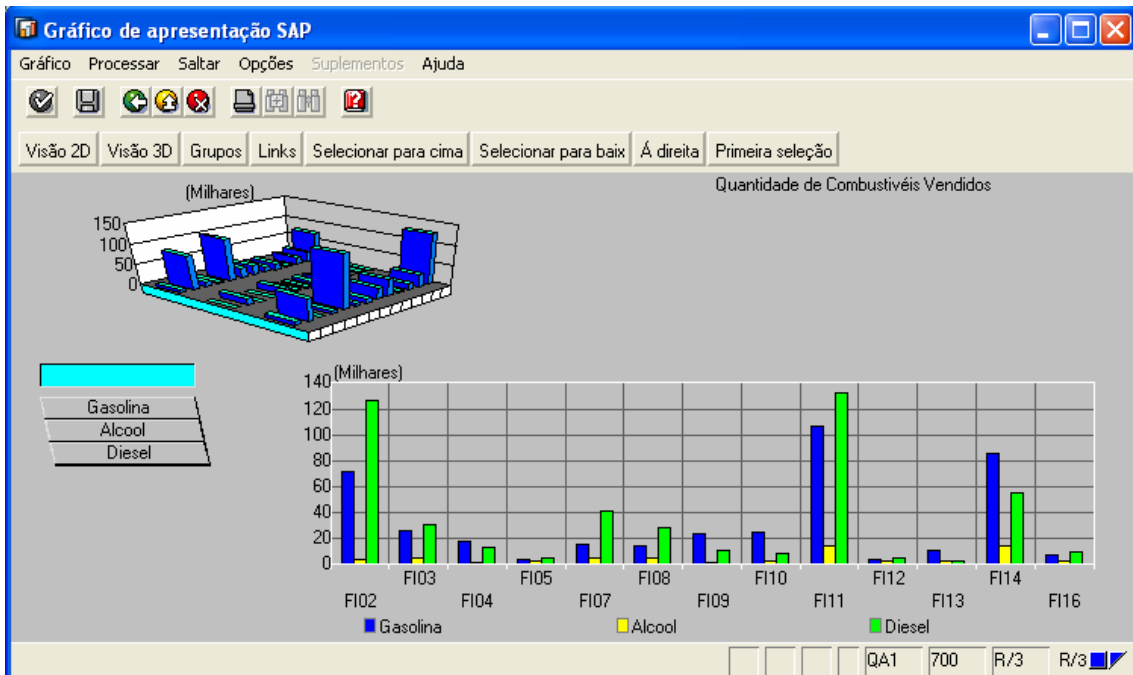
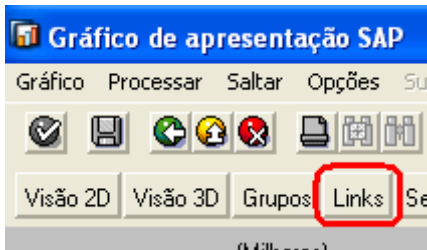
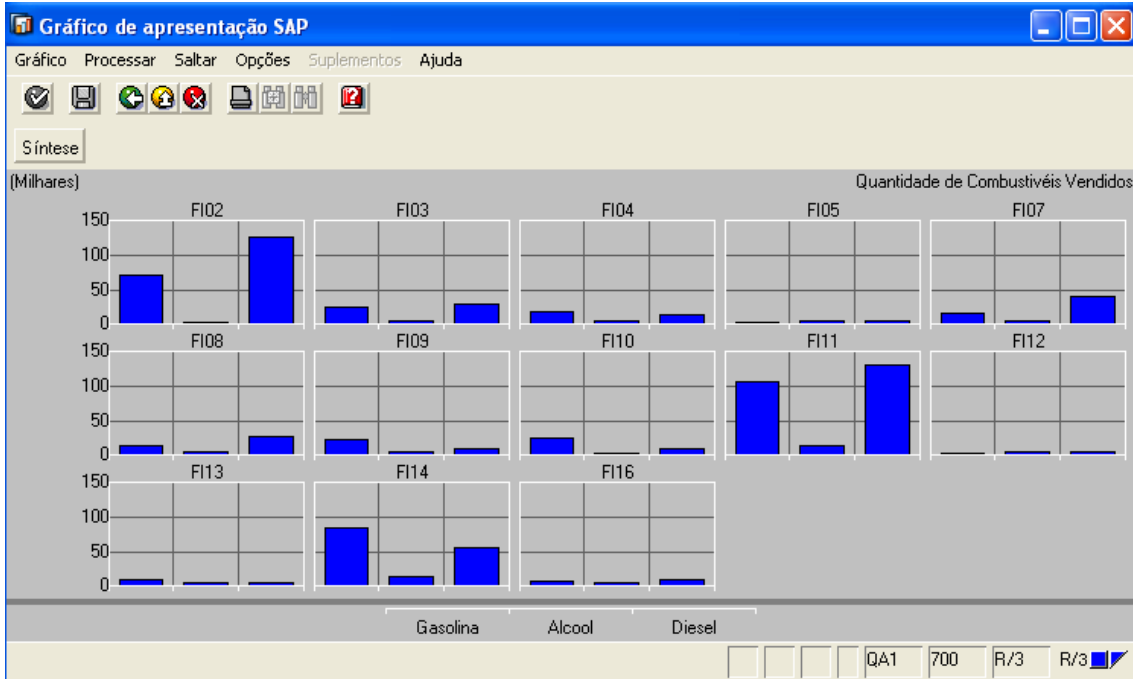
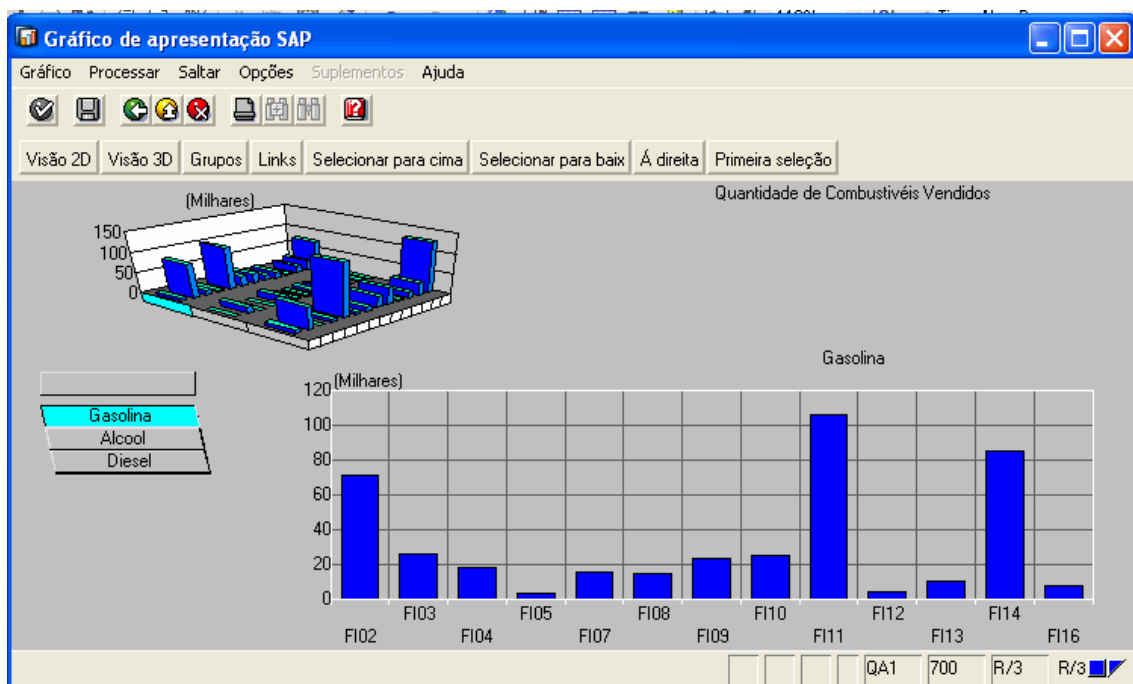


GRAFICO 3D

UDERSON LUIS FERMINO



Os gráficos são gerados e manipulados facilmente após ter os dados gerados e organizados.