Cursos de A a Z feitos pra você!

Manual ABAP

SUMARIO

CONCEITO BREVE DO SAP 1
1 INSTÂNCIA X CLIENT 1
1.1 Estrutura da Empresa 1
1.1.1 Fábricas 1
1.1.1.1 América do Norte (Estados Unidos) 1
1.2 Objetos “Client Dependent” e “Client Independent” 1
1.3 Transações 2
2 EDITOR ABAP/4 2
2.1 Tabela de Sumário de Convenção de Nomes 3
3 EXEMPLOS CURSO BÁSICO 6
3.1 EXEMPLOS 6
3.2 EXERCÍCIOS 20
3.3 RESOLUÇÕES 25
4 TEORIA BDC SESSION 31
4.1 BDC Session 31
4.2 Passos para criação de uma BDC Session 31
4.3 Identificando telas em uma transação 32
4.4 Gerando a tabela BDC 32
4.5 Enviando uma tabela BDC para o sistema 35
4.6 Processando dados com CALL TRANSACTION 35
4.7 Processando dados com BDC_INSERT 36
4.8 Função BDC_OPEN_GROUP 36
4.9 Função BDC_INSERT 36
4.10 Função BDC_CLOSE_GROUP 36
5 EXEMPLOS BDC SESSION 37
6 EXERCÍCIOS BDC SESSION 41
6.1 Exercício BDC 45
6.1.1 Resolução: 46
7 COMANDO SELECT 51
7.1 Comando SELECT 51
7.2 SELECT * FROM dbtab 51
7.3 Adições : 51
7.4 SELECT * FROM dbtab INTO TABEL itab. 52
7.5 SELECT * FROM dbtab APPENDING TABLE itab. 52
7.6 SELECT SINGLE * FROM dbtab WHERE f1 = g1 AND… AND fn + ... 52
7.7 SELECT * FROM dbtab APPENDING CORRESPONDING FILEDS OF TABLE itab. 52
7.8 SELECT * FROM dbtab FOR ALL ENTRIES in itab WHERE... 52
8 TIPOS DE SELECT EXISTENTES E MAIS UTILIZADOS 54
8.1 SELECT * FROM ... <tabela> 54
8.2 SELECT * FROM <tabela> WHERE <campo> EQ <conteúdo>. 54
8.3 SELECT * FROM <tabela> WHERE <table field> BETWEEN <field1> AND <field2> 54
8.4 SELECT * FROM <tabela> WHERE <table field> LIKE … ‘_R$’. 54
8.5 SELECT * FROM <tabela> WHERE <table field> IN (…… , …….). 54
8.6 SELECT * FROM <tabela> WHERE <table field> IN <internal table>. 54
8.7 SELECT * FROM <tabela> ORDER BY <field1> <field2> … PRIM ARY KEY 54
8.8 SELECT * FROM <tabela> BYPASSING BUFFER. 54
8.9 SELECT * FROM <tabela> APPENDING TABLE <internal table> 54
8.10 SELECT * FROM <tabela> INTO TABLE <internal table> 54
8.11 SELECT … INTO CORRESPONDING FIELDS OF TABLE <internal table>. 54
8.12 SELECT * APPENDING COREESPONDING FIELDS OF TABLE <internal table>. 55
8.13 SELECT SINGLE * FROM <tabela> WHERE <campo> EQ <conteúdo> 55
8.14 SELECT <a1> <a2> ... INTO (<f1>, <f2>, …) FROM … <tabela> WHERE … 55
8.15 SELECT MAX (campo) 55
8.16 SELECT * FROM SFLIGHT WHERE PRICE IN ITAB. 55

8.17 SELECT * FROM (<tabela>) INTO <work area>. 55
8.18 SELECT * FROM <tabela> FOR ALL ENTRIES IN <tabela interna> 55
8.19 SELECT carrid MIN(price) MAX(price) INTO (carid, minimum, maximum) 55
9 COMANDOS MAIS UTILIZADOS EM ABAP 56
10 DATA DICTIONARY 58
10.1 Objetivos 58
10.2 Conceitos de Bancos de Dados Relacionais 58
10.3 Modelo Entidade-Relacionamento 58
10.4 Restrições de Mapeamento 58
10.5 Modelo Relacional 58
10.6 Como distinguir as ocorrências umas das outras? 58
10.7 Como representar os relacionamentos entre tabelas? 58
10.8 Como definir todos esses elementos num BD? 59
11 R/3 DATA DICTIONARY 59
11.1 Introdução 59
11.2 Funções desempenhadas pelo Data Dictionary 59
11.2.1 Gerenciamento das Definições de Dados 59
11.2.2 Provisão de informações para avaliações 59
11.2.3 Suporte ao desenvolvimento 59
11.2.4 Suporte à documentação 59
11.2.5 Garantia de que as definições de dados sejam flexíveis e atualizadas 59
11.3 Elementos de Dados 60
11.4 Domínios 60
11.5 Objeto de Bloqueio 60
11.6 Macth Codes ID 61
11.7 Objetos Match Code 61
11.8 Pools / Clusters 61
11.9 Tabelas Transparentes e Cluster 62
11.10 Nomes de Tabelas - ATAB 62
11.11 Campos de Tabelas 63
11.12 Tabela de Índice 63
11.13 Tabela de Grupo 63
11.14 Grupo de Tipo 63
11.15 Estrutura 64
11.16 Views 64
12 ALV 65
12.1 Introdução 65
12.2 Relatórios tradicionais 65
12.3 Relatórios ALV 66
12.4 Barra de ferramentas 67
12.5 Opções standard 68
12.6 O botão de Detalhes 68
12.7 Os botões de Total e Subtotal 69
12.8 O botão Pré-visualz.impressão 70
12.9 O botão Microsoft Excel 70
12.10 O botão Microsoft Excel – Opção Excel Macros SAP 71
12.11 O botão Microsoft Excel – Opção Tabela 71
12.12 O botão Microsoft Excel – Opção Tabela Pivot 71
12.13 O botão Processamento de Texto 72
12.14 O botão File Local 72
12.15 O botão Destinatário de correio eletrônico 73
12.16 O botão Modificar layout 73
12.17 O botão Gravar Layout 75
12.18 O botão Selecionar Layout 75
12.19 Inserindo uma figura no cabeçalho 76
12.20 Importando uma imagem para o R/3 77
12.21 Exemplo de programa 78

12.22 Estruturas SLIS 87
13 ALV – Mais Detalhes 90
13.1 UTILIZAÇÃO 90
13.2 Conceito de Variante de Exibição 92
13.3 Tela de Seleção Relatório 94
14 FORMATAÇÃO DO RELATÓRIO EM ALV 96
14.1 Definições de dados para o layout ALV 96
14.2 Rotinas para Formatação do Layout 97
14.3 Como mudar cor de uma coluna 103
14.4 Como mudar a cor de uma linha 104
14.5 Como utilizar outros botões na tela 105
14.6 Programas Standard - Modelo 106

CONCEITO BREVE DO SAP

O SAP é um sistema que trabalha com um número muito grande de tabelas interligadas, que armazenam e manipulam os valores de controle dos processos. Essas tabelas são responsáveis pelo armazenamento dos valores do sistema e são divididas em grupos que se interligam em um todo. Assim, existem tabelas responsáveis pelas informações de FI, outras pelas informações de SD, outras ainda por MM, mas todas elas apresentam campos chaves que permitem, pelos mais diferentes e complicados caminhos, a interligação e consistência de todo o sistema. Embora a ferramenta ABAP/4 dentro do SAP seja muito poderosa e praticamente capaz de permitir qualquer customização do sistema, é muito importante manter os conceitos originais sempre em mente, e nunca tentar forçar alguma coisa que deveria ter um comportamento natural. Por exemplo, nunca tente alterar um valor de uma tabela do SAP (embora perfeitamente possível, com o comando UPDATE), sem um minucioso estudo de suas implicações anteriormente. Isso pode comprometer a integridade dos dados do sistema, se não forem atualizados todos os valores de todas as tabelas relacionadas a essa alteração.

1 INSTÂNCIA X CLIENT

1.1 Estrutura da Empresa

1.1.1 Fábricas

1.1.1.1 América do Norte (Estados Unidos)

- ✓ Alsip, Ilinois;
- ✓ Atlanta, Geórgia;
- ✓ Champaign, Ilinois;

Também é muito importante o conceito do funcionamento do ambiente do sistema durante a evolução de um projeto. Inicialmente devemos entender os conceitos de client e instância:

- ✓ **Client**: é definido como sendo uma unidade independente do R/3, em termos comerciais, organizacionais e técnicos. Isso significa que possuem sua própria configuração, dados de aplicação e dados cadastrais (master data);
- ✓ **Instância:** é definida como um ambiente do R/3 que agrupa um ou mais clients, onde se executa um determinado trabalho;

Uma instância de trabalho, geralmente possui mais de um client, onde são trabalhados simultaneamente diferentes frentes de trabalho do projeto. A intenção dessa divisão é que se possa trabalhar somando valores, sem que haja conflitos de interesse. Por exemplo, durante um projeto, o client para desenvolvimento das customizações de ABAP deve ser diferente dos outros, pois trabalha muito com testes e alterações constantes, o que inviabiliza outros tipos de serviços.

Se essa divisão muitas vezes ajuda, algumas vezes atrapalha. Geralmente as massas de dados são diferentes nos clients, e o comportamento principalmente nos testes dos produtos customizados pode ser diferente. O recomendado pela própria SAP é que exista um client só para testes, com massa de dados completa que permita "recarga" sempre que necessário, o que permitiria que as condições de teste pudessem ser repetidas. No dia a dia de um projeto isso é muito difícil, pois a manutenção desses clients pelo time de basis geralmente não é muito bem vista.

As instâncias variam também ao longo de um projeto. A medida que o sistema vai sendo refinado, geralmente se inicia uma nova instância livre dos vícios e restos de testes da anterior. Pelo menos 3 instâncias sempre existem durante o período de um projeto. A instância de desenvolvimento, a de pré-produção e finalmente a de produção. Cada vez que o sistema é migrado de uma instância para a outra, somente deve ser aproveitado o que está comprovadamente funcionando na instância anterior, de modo a diminuir os erros a cada migração.

1.2 Objetos "Client Dependent" e "Client Independent"

Todos os objetos criados em um sistema SAP, podem ser divididos em dois grupos, tratados diferentemente ao longo de um projeto. Os objetos chamados "Client Independents" são aqueles que uma vez criados podem ser utilizados por todos os clients de uma mesma instância, sem que se necessite de nenhuma articulação adicional. O simples fato de se encontrar ativo no repositório do sistema habilitado para a instância, o permite ser utilizado por todos os clients dessa instância, de maneira simultânea. O maior exemplo de objetos independentes do client são os programas de ABAP/4 normais customizados. Por exemplo, um report criado em um client de desenvolvimento, pode ser executado de um outro client de teste, existente na mesma instância, uma vez que tenha sido gerado e ativado.

Os objetos dependentes do client, ao contrário, uma vez criado, por exemplo, no client de desenvolvimento não pode ser executado de nenhum outro, a menos que seja transportado ou copiado para o seu destino. Esse é um exemplo típico dos formulários em SAPscript.

Dentro da classe dos objetos dependentes do client, existe um subgrupo que exige uma atenção maior ainda. É o caso dos logotipos utilizados dentro de um SAPscript. Esses objetos, além de não serem compartilhados pelos clients, também não geram request ao serem criados, o que impossibilita serem transportados diretamente, precisando associarmos, primeiro, esses elementos a uma Change Request através de um programa. Nesse caso, os logotipos devem ser gerados em cada um dos clientes em que se deseje utilizá-lo, através da execução de um programa do SAP. Esse tipo de problema, ao longo de um projeto, exige uma atenção especial, para evitar problemas futuros. Por exemplo, o desenhista técnico deve tomar o cuidado de anexar o arquivo .TIF do logotipo na especificação do MDM, para futura repetição do processo de geração do logotipo dentro do sistema. Caso esse cuidado não seja tomado, futuramente pode-se obter um logotipo de tamanho diferente, gerando re-work e perda de tempo.

1.3 Transações

Transação é um código alfanumérico de 20 caracteres, utilizado para iniciar um processo dentro do sistema SAP. Todo e qualquer processo ou parte dele deve ser executado dentro do sistema através de uma transação. Na customização de ABAP/4, sempre que um GAP do sistema é coberto, isso gera pelo menos uma transação, de modo que o usuário possa executar esse produto customizado de dentro do sistema.

Toda operação realizada através do menu do sistema, também corresponde a uma transação. Um método para conhecermos o código de uma transação cujo caminho pelo menu é sabido, é entrarmos na mesma, e na tela inicial desta transação, utilizarmos o menu Sistema Status, que informa o programa tela e transação executados.

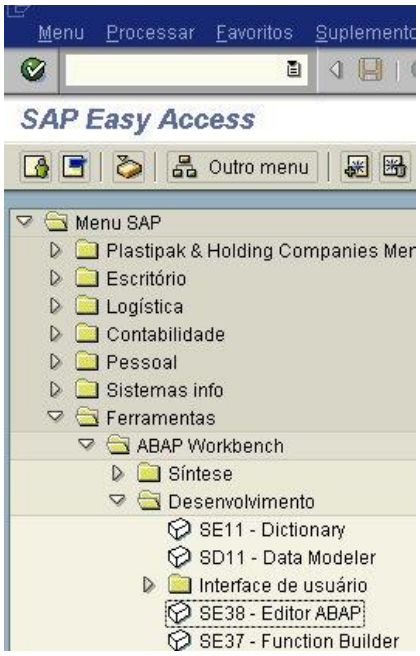
No desenvolvimento de customizações ABAP/4, as principais transações utilizadas, são:

- ✓ **SE38:** abre o editor ABAP/4;
- ✓ **SE16:** permite a visualização do conteúdo de tabelas do SAP**;**
- ✓ **SE11:** permite a visualização da arquitetura de uma tabela/estrutura do SAP;
- ✓ **SE43:** criação de menus;
- ✓ **SE93:** criação de transações customizadas;
- ✓ **SE71:** criação e manutenção de formulários SAPscripts;
- ✓ **CMOD:** criação de projetos Exits (Field Exits, User Exits, ...);

2 EDITOR ABAP/4

O editor de programação ABAP/4 do SAP pode ser encontrado através do caminho :

Menu SAP => Ferramentas => ABAP Workbench => Desenvolvimento => Editor Abap



ou pela transação SE38.

Uma tela para a entrada do nome do programa é aberta, como exemplificada abaixo. Para criar um programa novo, utilize um nome ainda não existente no repositório, e apertar o botão Criar. Para editar ou exibir um programa já existente, entrar com o nome do programa e apertar os botões respectivos.

Existe um padrão de nomenclatura que deve ser seguido, não só para nome de programas, mas para todos os desenvolvimentos no SAP R/3. Esses padrões podem variar de projeto a projeto e principalmente com a versão do SAP com a qual se está trabalhando. Em todos os casos os nomes dos desenvolvimentos começam sempre com Z ou Y.

2.1 Tabela de Sumário de Convenção de Nomes

Object	Structure / Example	Max Length	Position	Description
ABAP Programs	ZP_XX_X_$$$$$$$$$$$$$$$$$$$$	30	1 2 4-5 7 9-30	Z – Permanent P – Project Identifier Functional Descriptor (Table 1) Program Type: Like the old naming standards Free choice for Program Name Ex: ZA_MM_R_0010
Data Elements	ZP_E_XX_$$$$$$$$$$$$$$$$$$	30	1 2 4 6-7 9-30	Z – Permanent P – Project Identifier E – For Data Element Functional Descriptor (Table 1) DDIC name identifier
Domains	ZP_D_XX_$$$$$$$$$$$$$$$$$$	30	1 2 4 6-7 9-30	Z – Permanent P – Project Identifier D – For Domain Functional Descriptor (Table 1) DDIC name identifier
Match Code Objects	ZP$$	4	1 2 3-4	Z – Permanent P – Project Identifier Functional Descriptor (Table 1)
Tables	ZPTXX_$$$$$$$$$$	16	1 2 3 4-5 7-16	Z – Permanent P – Project Identifier T – For Tables Functional Descriptor (Table 1) Sequential Number
Structure	ZPSXX_$$$$$$$$$$$$$$$$$$$$	30	1 2	Z – Permanent P – Project Identifier

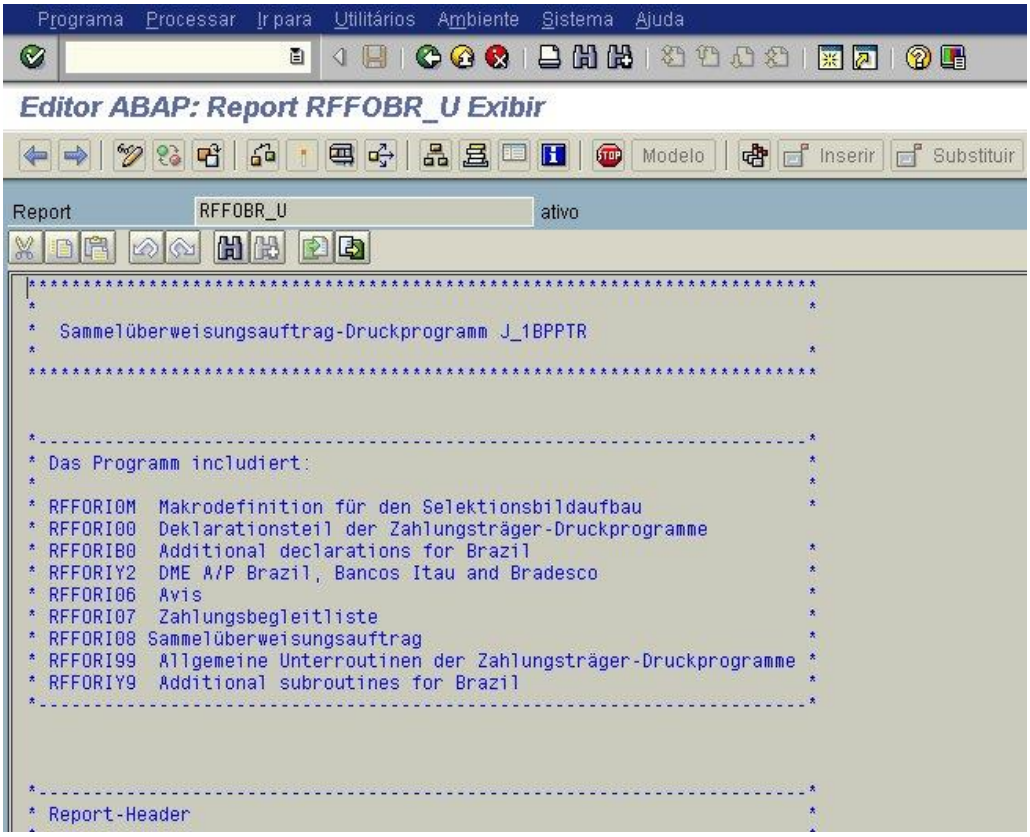
Object	Structure / Example	Max Length	Position	Description
			3 4-5 7-30	S – For Structures Functional Descriptor (Table 1) DDIC name identifier
View	ZPVXX_$$$$$$$$$$	16	1 2 3 4-5 7-16	Z – Permanent P – Project Identifier V – For View Name Functional Descriptor (Table 1) DDIC name identifier
Function Modules	ZPFXX_$$$$$$$$$$$$$$$$$$$$$$	30	1 2 3 4-5 7-30	Z – Permanent P – Project Identifier F – For Function Functional Descriptor (Table 1) DDIC name identifier
Function Groups	ZP_XX_$$$$$$$$$$$$$$$$$$$$	26	1 2 4-5 7-26	Z – Permanent P – Project Identifier Functional Descriptor (Table 1) DDIC name identifier
Message Class	ZP_XX	8	1 2 4-5	Z – Permanent P – Project Identifier Functional Descriptor (Table 1)
Transaction	ZPXX$$	6	1 2 3-4 5-6	Z – Permanent A – Project Identifier Functional Descriptor (Table 1) Sequential Number EX. ZAMM01
User exit project definition	ZPXX$$$$	8	1 2 3-4 5-8	Z – Permanent A – Project Identifier Functional Descriptor (Table 1) User exit Identifier
Sapscript Form	ZP_XX_$$$$$$$$$$	16	1 2 4-5 7- 16	Z – Permanent P – Project Identifier Functional Descriptor (Table 1) Descriptive Identifier Ex. ZA_SD_NOTAFISCAL

Table 1 – Functional Description	
AM	Asset Management
CO	Accounting
ES	Especificação do Produto
FI	Finance
GP	General Product
MM	Material Management
PP	Production Planning
SD	Sales & Distribution
WM	Warehouse Management

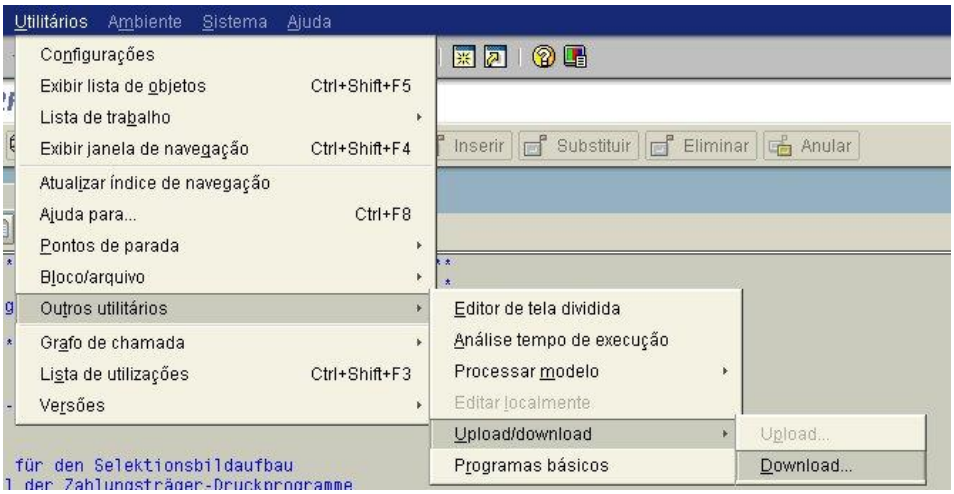
Program Type	
R	Report
O	On line
B	Batch
I	Consulte padrão para Interface

X	Include
S	Sapscript

A aparência do editor não difere muito de um editor de texto bem simples, onde se é possível escrever linhas de comando e lógica.



Um dos recursos permitidos para que se utilize um outro editor como o Note Pad do Windows, é o recurso de Download e Upload. Através do menu: Utilitários => Outros Utilitários => Upload/Download, o sistema permite que códigos escritos em outros editores possam ser carregados no editor ABAP/4 e vice-versa.



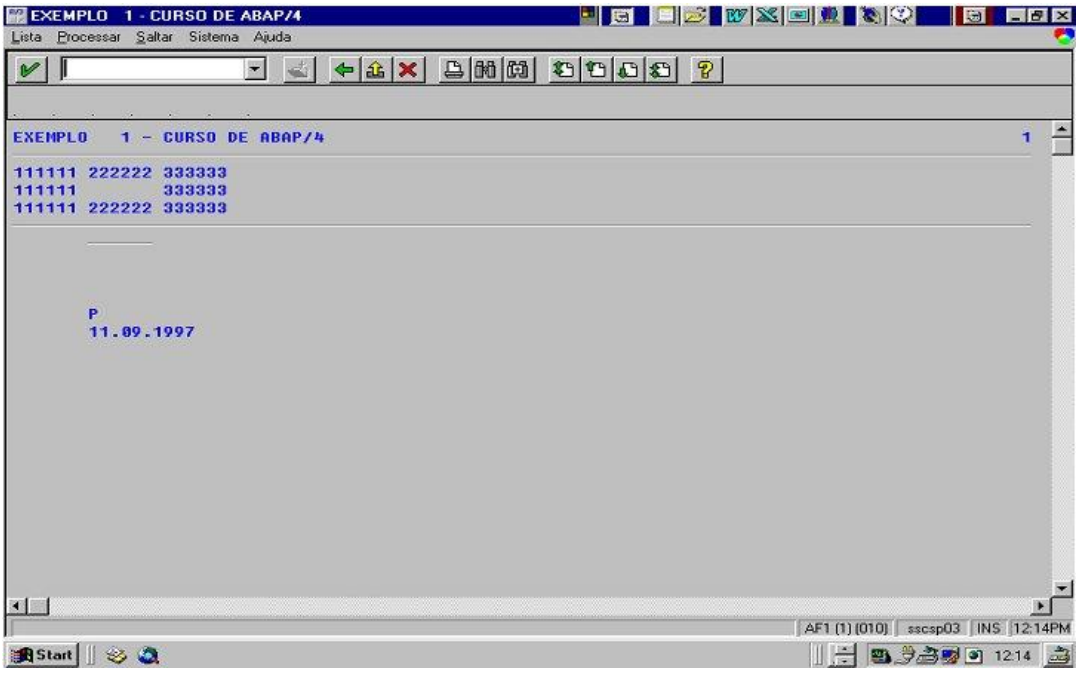
Esse recurso nos permite salvar os códigos ainda inativos ou incompletos. É extremamente útil para mantermos um controle próprio das versões quando se está codificando.

3 EXEMPLOS CURSO BÁSICO

3.1 EXEMPLOS

EXEMPLO 1

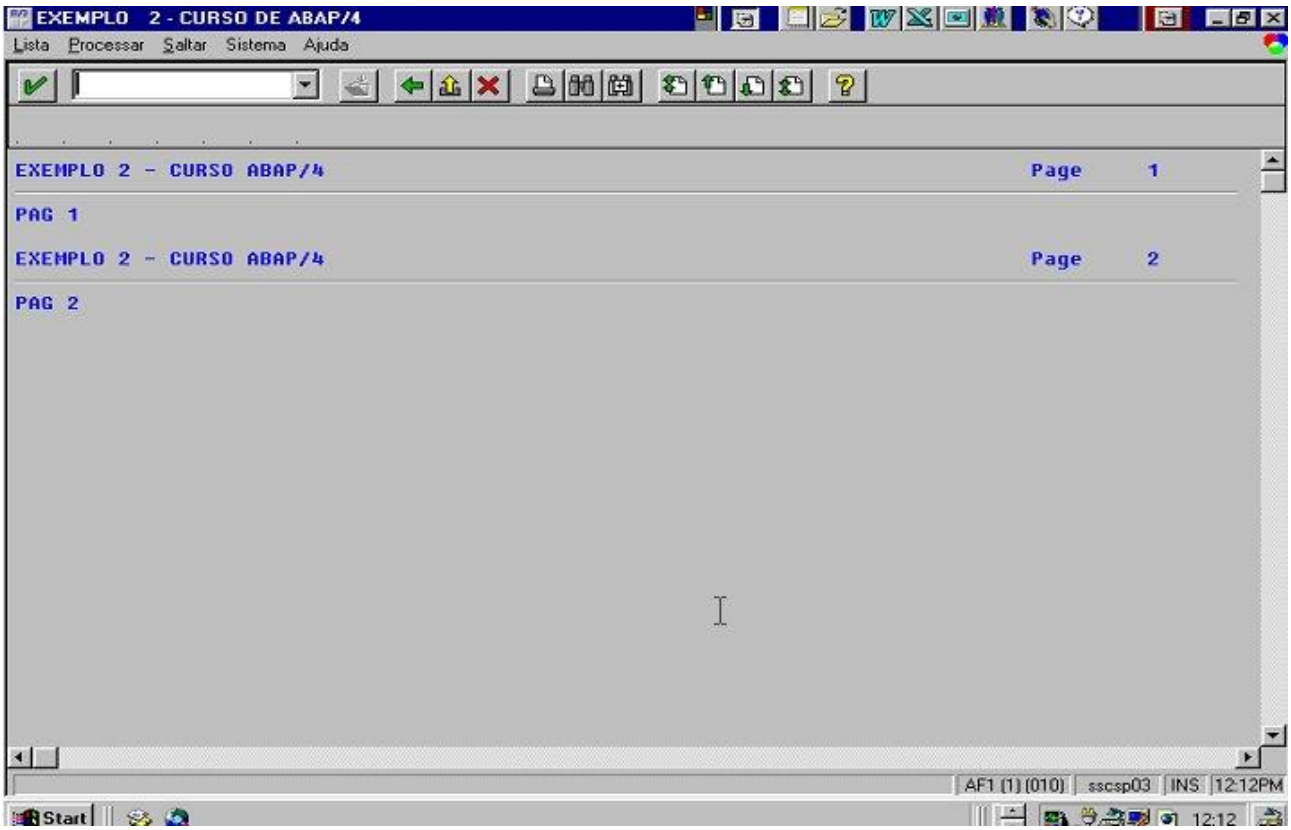
```
REPORT ZEXP0001.

WRITE: '111111'.
WRITE: '222222',
       '333333'.
WRITE: /'111111'.
WRITE: 15 '333333'.
WRITE: / TEXT-001.
ULINE.
ULINE 8(6).
SKIP.
SKIP 2.
WRITE : 8 SY-LANGU.
WRITE : / SY-DATUM UNDER SY-LANGU.
```



Existem alguns tipos de variáveis chamadas de variáveis do sistema. Elas possuem informações e dados do processamento, como o idioma de acesso (sy-langu), a data (sy-datum), a hora (sy-uzeit), etc.. Essas informações estão contidas na estrutura SYST (Campos de sistema ABAP, que pode ser abreviada para SY) e podem ser acessadas conforme o exemplo acima, o nome da estrutura mais o campo que se deseja.

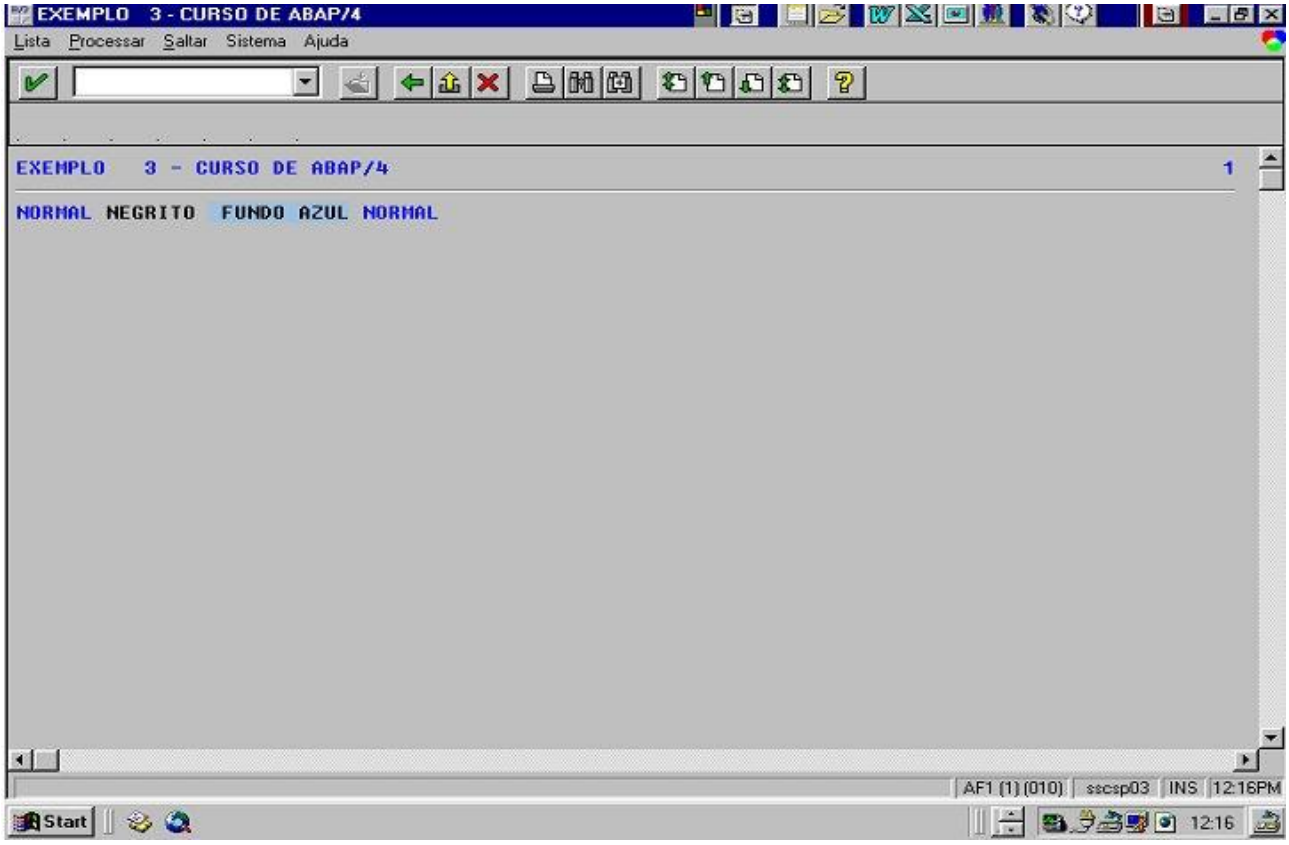
EXEMPLO 2

```
REPORT ZEXC0002 NO STANDARD PAGE HEADING.

WRITE 'PAG 1'.

NEW-PAGE.

WRITE 'PAG 2'.

TOP-OF-PAGE.

WRITE :   'EXEMPLO 2 - CURSO ABAP/4',
          80 'Page',
              SY-PAGNO.
ULINE.
```



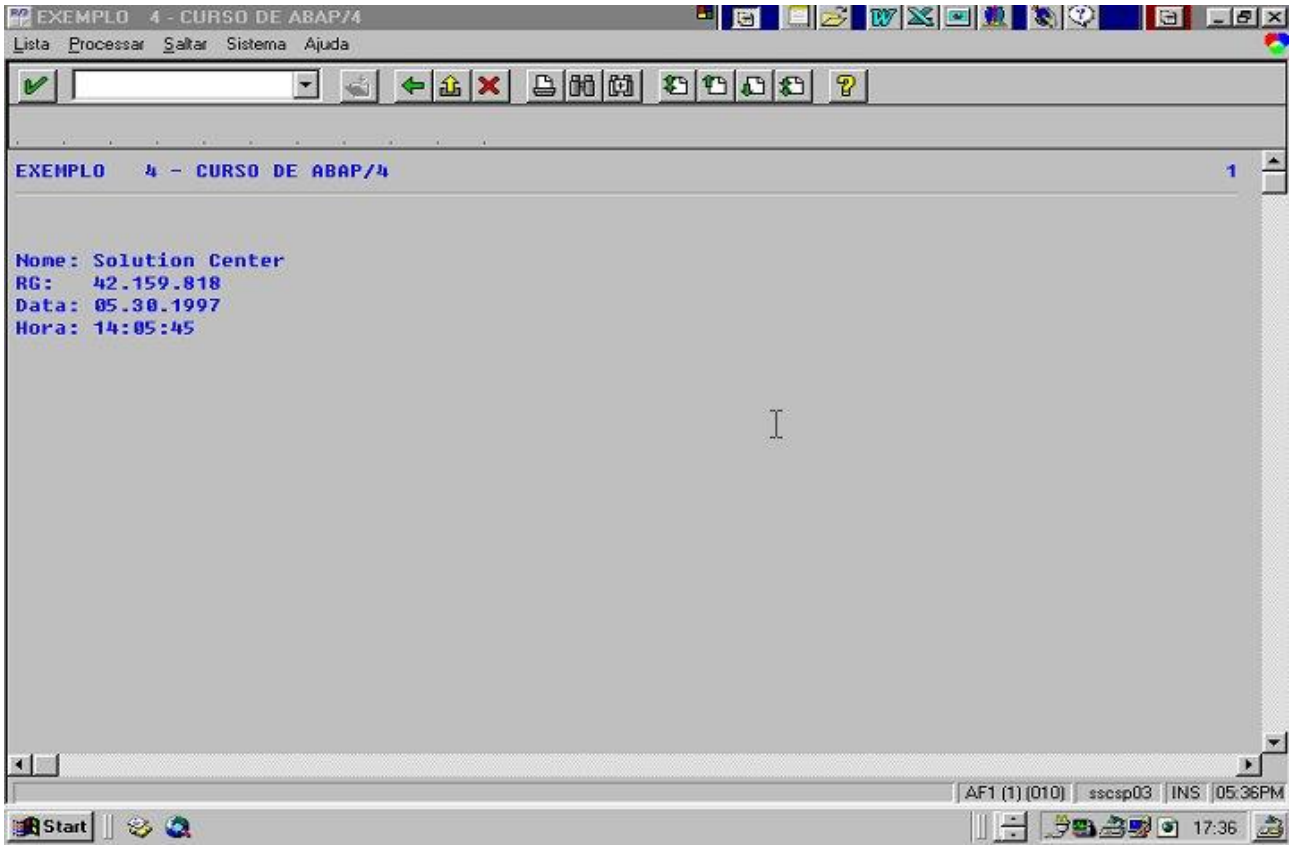
EXEMPLO 3

```
REPORT ZEXP0003.

WRITE : 'NORMAL'.

FORMAT INTENSIFIED OFF.

WRITE : 'NEGRITO '.

FORMAT COLOR 1.

WRITE : 'FUNDO AZUL'.

FORMAT COLOR OFF.

FORMAT INTENSIFIED ON.

WRITE : 'NORMAL'.
```



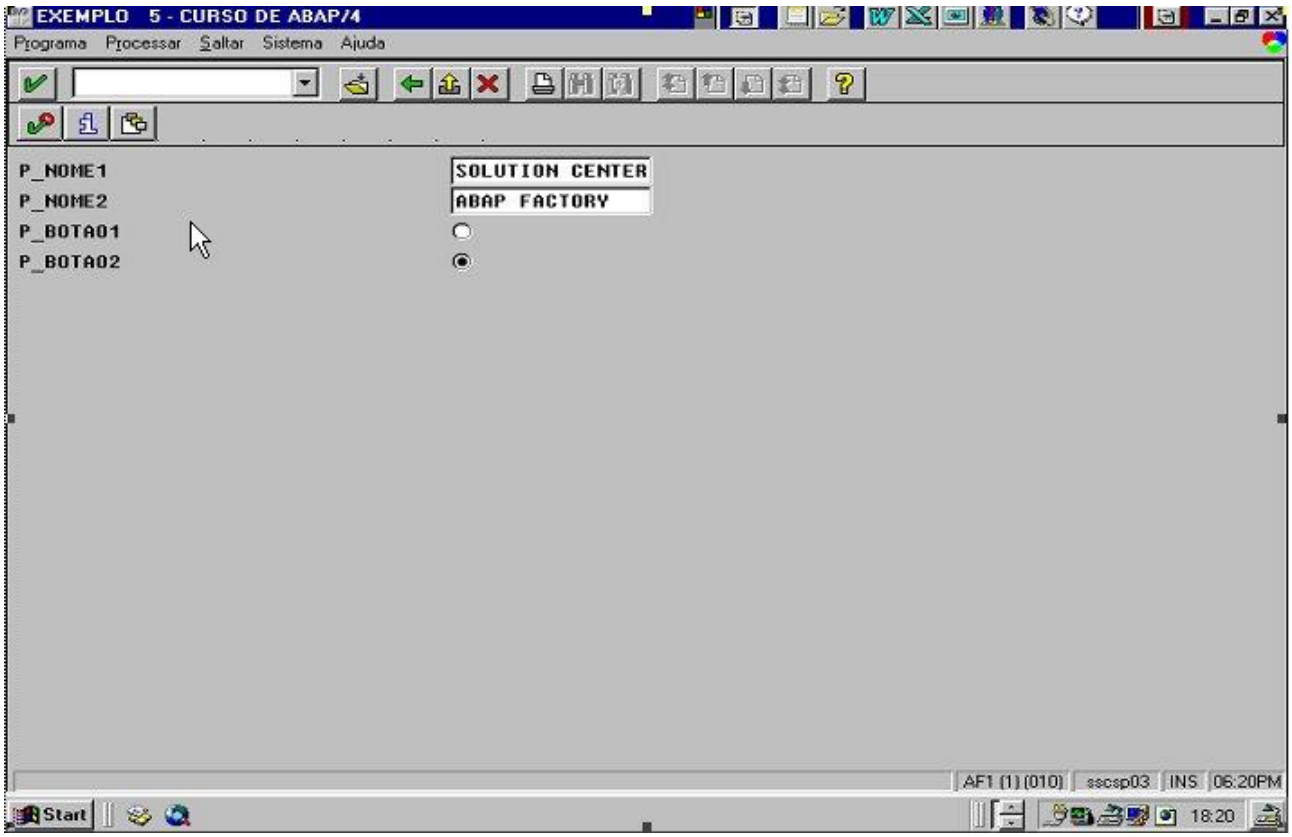
EXEMPLO 4

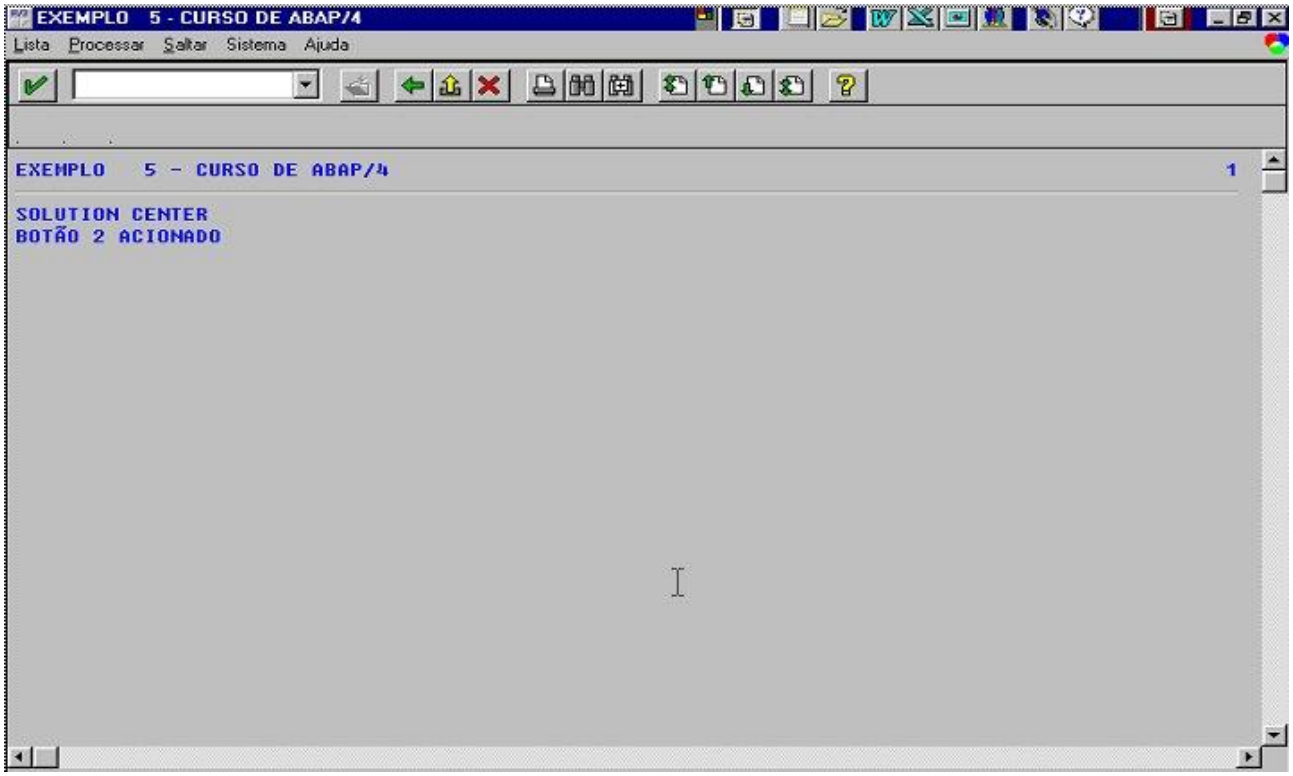
```
REPORT ZEXP0004.

DATA: NOME(20)   TYPE C,
      RG(10)     TYPE I,
      DATA       LIKE BKPF-BUDAT,
      HORA(8)    VALUE '14:05:45'.

MOVE 'Solution Center' TO NOME.
RG = 42159818.
DATA = '19973005'.
SKIP 2.
WRITE:  'Nome:',
         NOME,
       / 'RG:',
         RG UNDER NOME,
       / 'Data:',
         DATA,
       / 'Hora:',
         HORA.
```

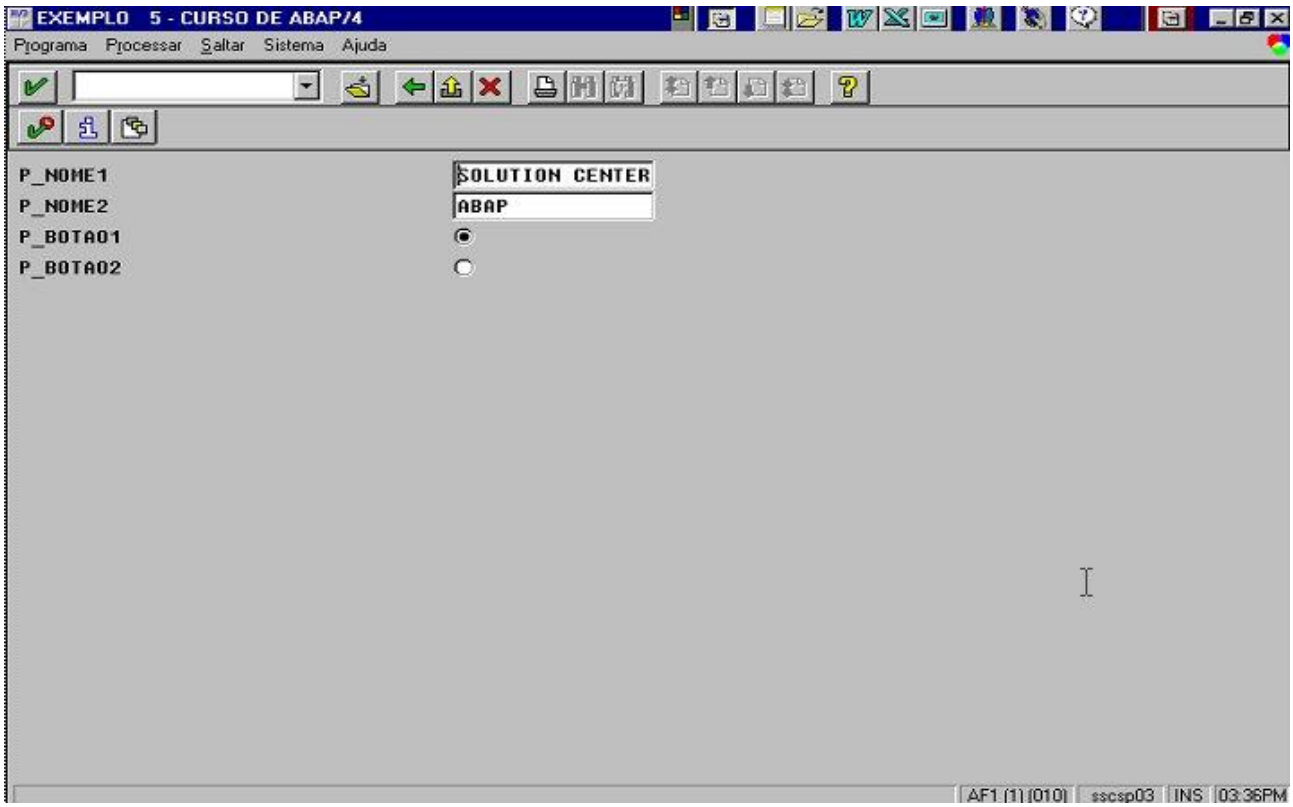


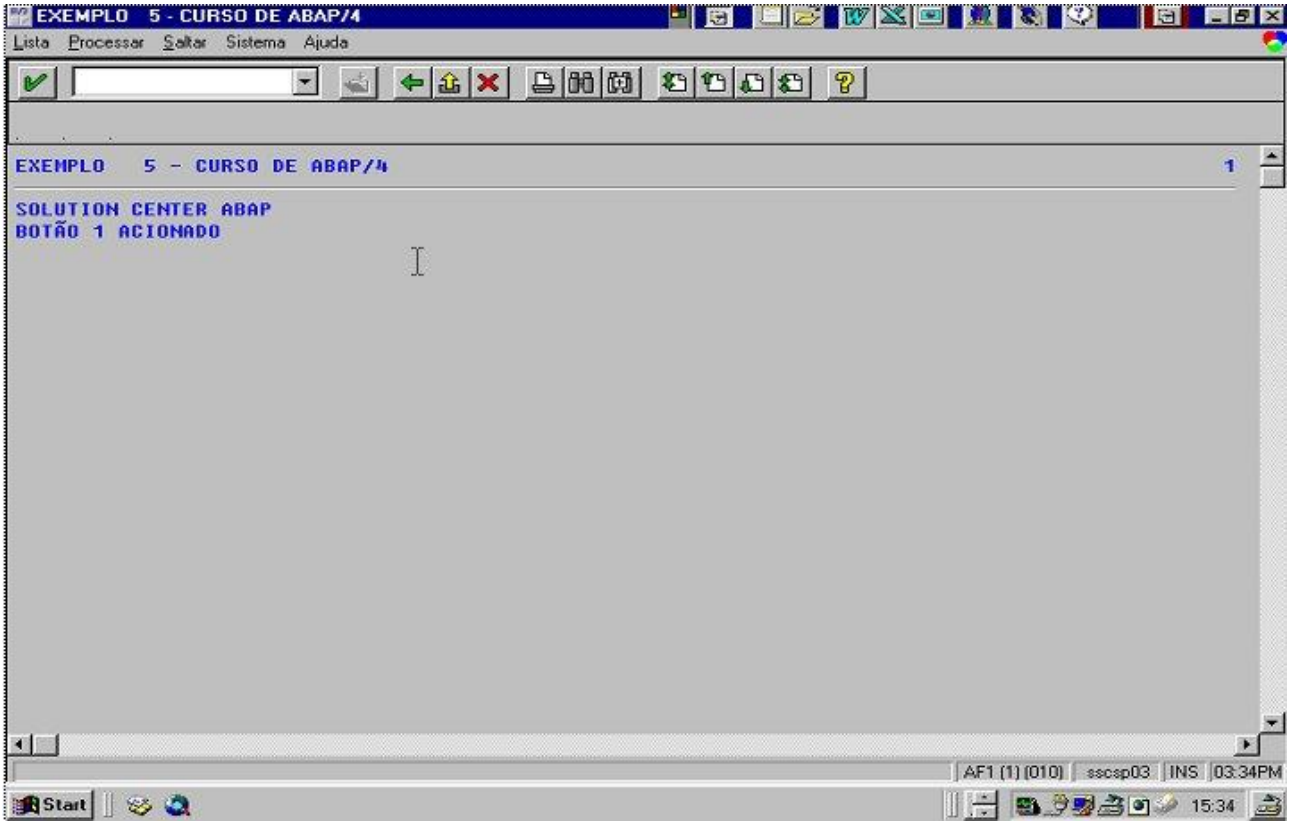
EXEMPLO 5

```
REPORT ZEXP0005.

PARAMETER: P_NOME1(15) TYPE C,
                    P_NOME2(15) TYPE C DEFAULT 'Abap Factory',
                    P_BOTAO1 RADIOBUTTON GROUP G1,
                    P_BOTAO2 RADIOBUTTON GROUP G1.

WRITE P_NOME1.

IF P_NOME2 NE 'ABAP FACTORY'.
  WRITE P_NOME2.
ENDIF.

IF P_BOTAO1 = 'X'.
  WRITE / 'BOTÃO 1 ACIONADO'.
ELSE.
  WRITE / 'BOTÃO 2 ACIONADO'.
ENDIF.
```



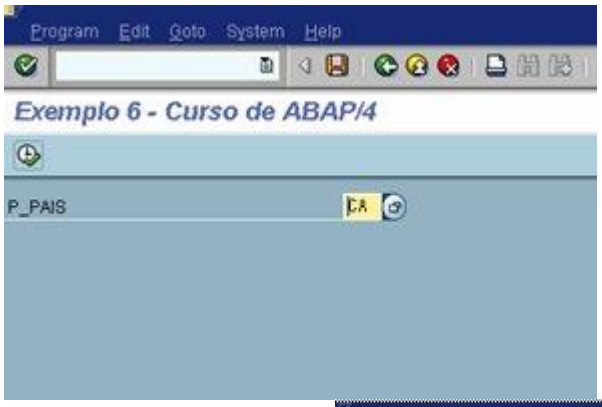

EXEMPLO 5 - CURSO DE ABAP/4
Lista Processar Saltar Sistema Ajuda
EXEMPLO 5 - CURSO DE ABAP/4 1
SOLUTION CENTER
BOTÃO 2 ACIONADO

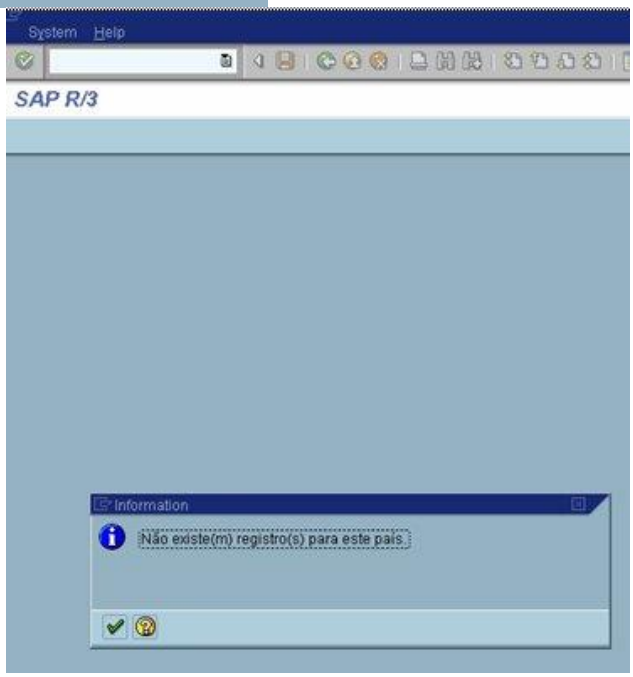

EXEMPLO 5 - CURSO DE ABAP/4
Programa Processar Saltar Sistema Ajuda
P_NOME1
SOLUTION CENTER
P_NOME2
ABAP
P_BOTAO1
P_BOTAO2
AF1 (1) [010]
sscsp03
INS
03:36PM

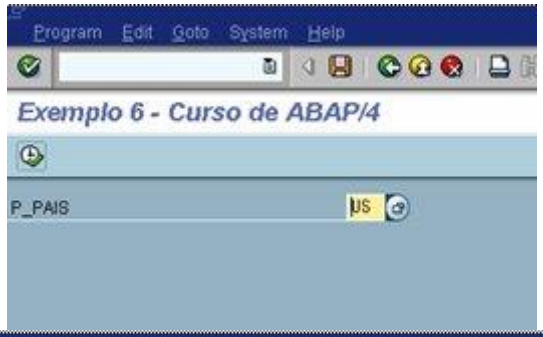

EXEMPLO 5 - CURSO DE ABAP/4
Lista Processar Saltar Sistema Ajuda
EXEMPLO 5 - CURSO DE ABAP/4 1
SOLUTION CENTER ABAP
BOTÃO 1 ACIONADO
AF1 (1) (010) sscsp03 INS 03:34PM
Start
15:34

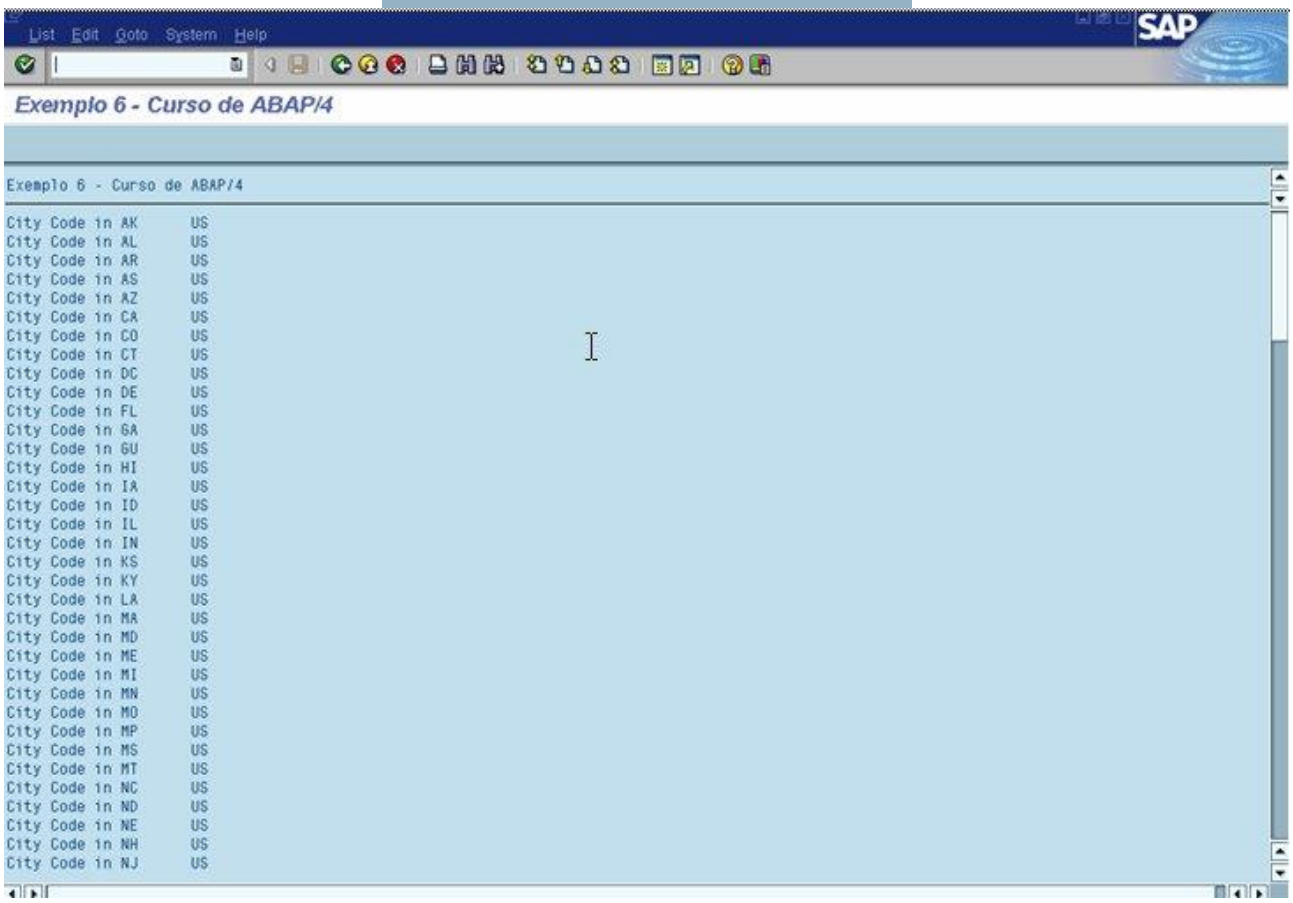
EXEMPLO 6

```
REPORT ZEXP0006 message-id za.
PARAMETER: P_PAIS LIKE T005S-LAND1.

TABLES T005H.

SELECT * FROM T005H WHERE LAND1 = P_PAIS ORDER BY CITYC.

  WRITE: / T005H-Bezei,
         T005H-LAND1.

ENDSELECT.

IF SY-SUBRC NE 0.
   MESSAGE I000.
ENDIF.
```



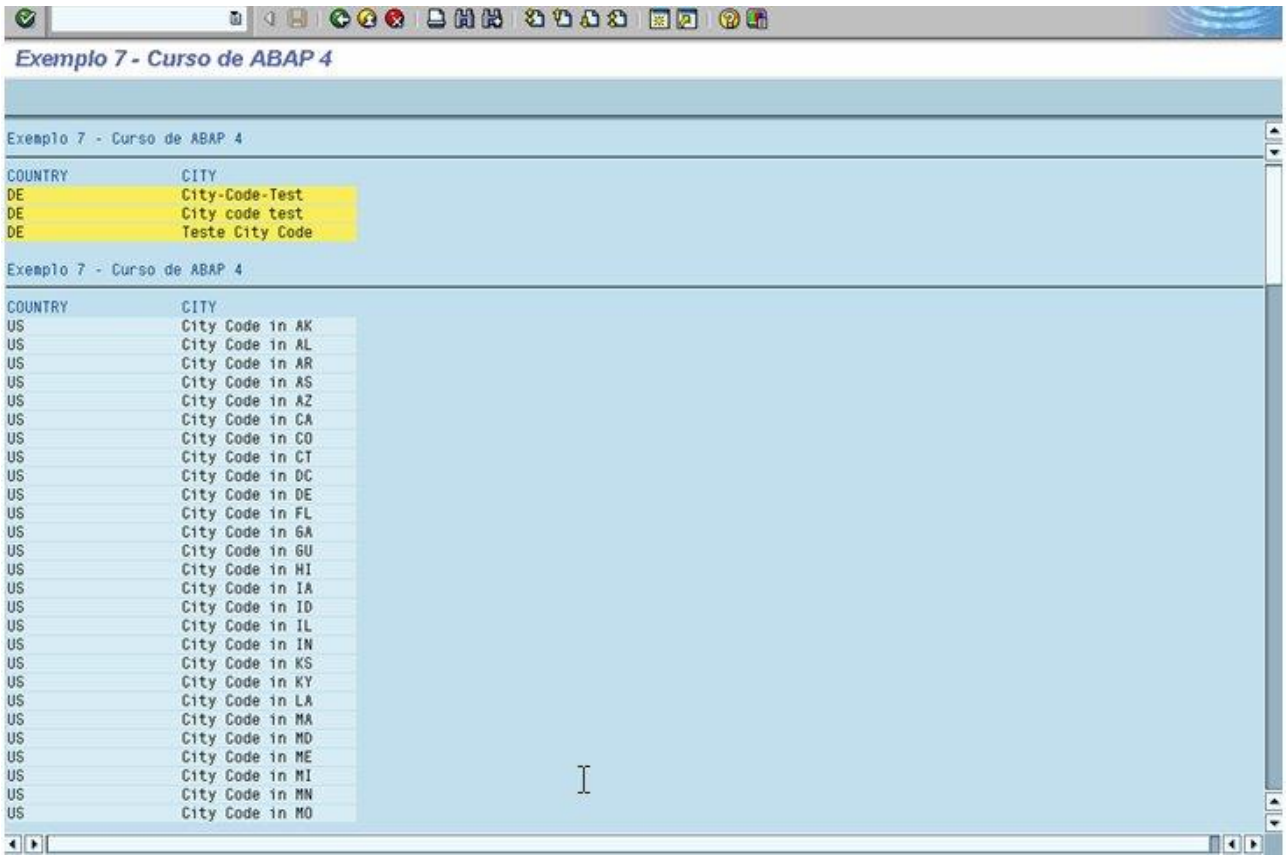



Program Edit Goto System Help
Exemplo 6 - Curso de ABAP/4
P_PAIS
US


List Edit Goto System Help
SAP
Exemplo 6 - Curso de ABAP/4
Exemplo 6 - Curso de ABAP/4
City Code in AK US
City Code in AL US
City Code in AR US
City Code in AS US
City Code in AZ US
City Code in CA US
City Code in CO US
City Code in CT US
City Code in DC US
City Code in DE US
City Code in FL US
City Code in GA US
City Code in GU US
City Code in HI US
City Code in IA US
City Code in ID US
City Code in IL US
City Code in IN US
City Code in KS US
City Code in KY US
City Code in LA US
City Code in MA US
City Code in MD US
City Code in ME US
City Code in MI US
City Code in MN US
City Code in MO US
City Code in MP US
City Code in MS US
City Code in MT US
City Code in NC US
City Code in ND US
City Code in NE US
City Code in NH US
City Code in NJ US

EXEMPLO 7

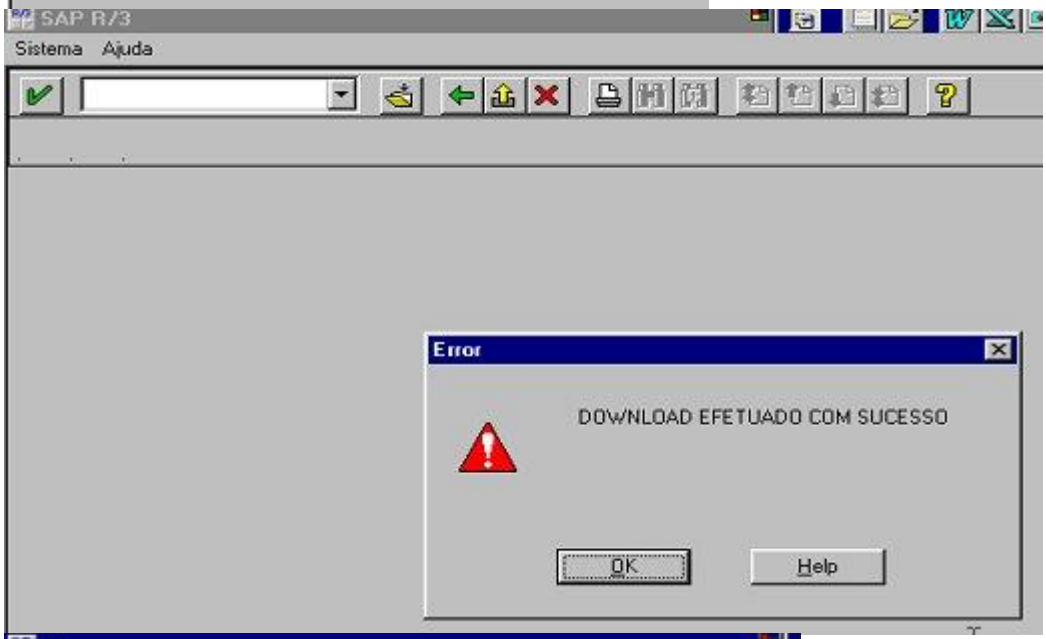
```
REPORT ZEXP0007.

TABLES: T005H..

DATA V_VAR1 VALUE '1'.

DATA: BEGIN OF T_T005H OCCURS 0,
     LAND1   LIKE T005H-LAND1,
     BEZEI   LIKE T005H-BEZEI.
DATA: END OF T_T005H.

SELECT * FROM T005H WHERE LAND1 IN ('US', 'DE') ORDER BY LAND1.

  T_T005H-LAND1 = T005H-LAND1.
  T_T005H-BEZEI = T005H-BEZEI.
  APPEND T_T005H.

ENDSELECT.

IF SY-SUBRC NE 0.
   WRITE TEXT-001.
ENDIF.

LOOP AT T_T005H.

   ON CHANGE OF T_T005H-LAND1.

        IF V_VAR1 = 0.
           NEW-PAGE.
        ENDIF.

       FORMAT COLOR OFF.
       WRITE 'COUNTRY           CITY'.

       CLEAR V_VAR1.

  ENDON.


    IF T_T005H-LAND1 = 'DE'.
       FORMAT COLOR COL_TOTAL.
    ELSE.
       FORMAT COLOR COL_NORMAL.
    ENDIF.

    WRITE : / T_T005H-LAND1,
              21 T_T005H-BEZEI.

ENDLOOP.
```

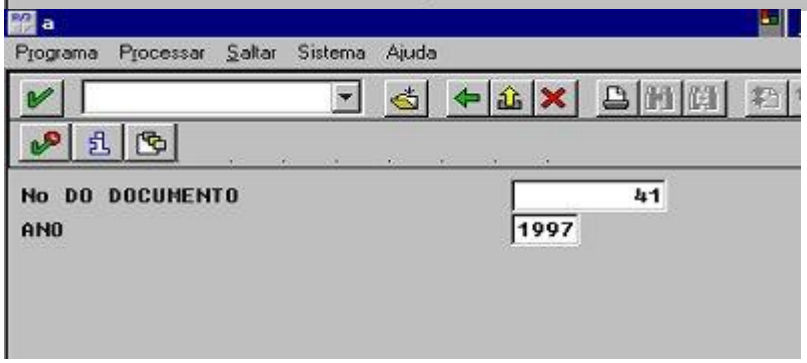

Exemplo 7 - Curso de ABAP 4
Exemplo 7 - Curso de ABAP 4
COUNTRY CITY
DE City-Code-Test
DE City code test
DE Teste City Code
Exemplo 7 - Curso de ABAP 4
COUNTRY CITY
US City Code in AK
US City Code in AL
US City Code in AR
US City Code in AS
US City Code in AZ
US City Code in CA
US City Code in CO
US City Code in CT
US City Code in DC
US City Code in DE
US City Code in FL
US City Code in GA
US City Code in GU
US City Code in HI
US City Code in IA
US City Code in ID
US City Code in IL
US City Code in IN
US City Code in KS
US City Code in KY
US City Code in LA
US City Code in MA
US City Code in MD
US City Code in ME
US City Code in MI
US City Code in MN
US City Code in MO

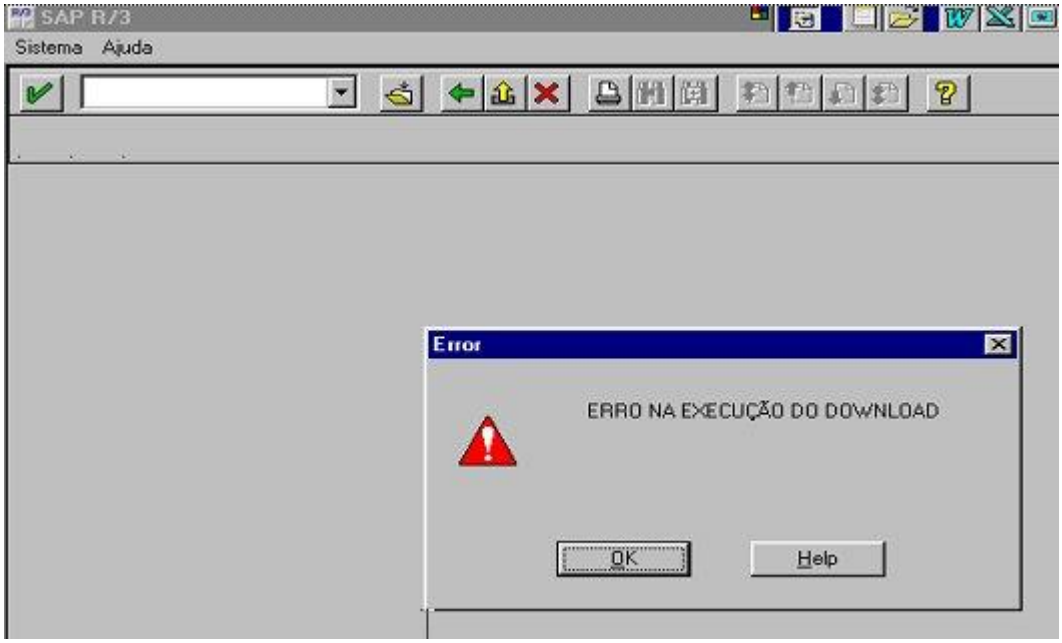
EXEMPLO 8

```
REPORT ZEXP0008 MESSAGE-ID ZA.

TABLES: BKPF.

PARAMETER: P_BELNR LIKE BKPF-BELNR DEFAULT '5000000041',
                   P_ANO   LIKE BKPF-GJAHR DEFAULT ' 2001'.

DATA: ARQ LIKE RLGRAP-FILENAME VALUE 'C:\TEMP\curso.txt'.

DATA: BEGIN OF T_ZCURSO OCCURS 0,
            ZDATA      LIKE     BKPF-BUDAT,
            SPACE1     TYPE     C VALUE ' ',
            ZBELNR     LIKE     BKPF-BELNR,
            SPACE2     TYPE     C VALUE ' ',
            ZGJAHR     LIKE     BKPF-GJAHR.
DATA: END OF T_ZCURSO.

SELECT * FROM BKPF WHERE BELNR  = P_BELNR
                         AND GJAHR = P_ANO.

  T_ZCURSO-ZDATA  = BKPF-BUDAT.
  T_ZCURSO-ZBELNR = BKPF-BELNR.
  T_ZCURSO-ZGJAHR = BKPF-GJAHR.
  APPEND T_ZCURSO.

ENDSELECT.

IF SY-SUBRC = 0.
   CALL FUNCTION 'WS_DOWNLOAD'

     EXPORTING
*       bin_filesize            = ' '
*       codepage                = ' '
        FILENAME                = ARQ
*       filetype                = ' '
*       mode                    = ' '
*       wk1_n_format            = ' '
*       WK1_N_SIZE              = ' '
*       WK1_T_FORMAT            = ' '

*       WK1_T_SIZE                    = ' '
*       col_select                    = ' '
*       col_selectmask                = ' '
*    importing
*       filelength                    =
     TABLES
        DATA_TAB                      = T_ZCURSO
*       fieldnames                    =
     EXCEPTIONS
        FILE_OPEN_ERROR               = 1
        FILE_WRITE_ERROR              = 2
        INVALID_FILESIZE              = 3
        INVALID_TABLE_WIDTH           = 4
        INVALID_TYPE                  = 5
```

```
        NO_BATCH                = 6
        UNKNOWN_ERROR           = 7
        OTHERS                  = 8.

  MESSAGE E007.
ELSE.
  MESSAGE E008.
ENDIF
```







EXEMPLO 9

```
REPORT ZEXP0009 MESSAGE-ID ZA.
TABLES: ZCURSO.

DATA: ARQ LIKE  RLGRAP-FILENAME VALUE 'C:\TEMP\CURSO.TXT',
           V_CONT    TYPE I.

DATA: BEGIN OF T_ZCURSO OCCURS 0,
            ZDATA  LIKE ZCURSO-ZDATA,
            SPACE1  TYPE C VALUE ' ',
            ZBELNR  LIKE ZCURSO-ZNUMERO,
            SPACE2  TYPE C VALUE ' ',
            ZGJAHR  LIKE ZCURSO-ZANO.
DATA: END OF T_ZCURSO.

CALL FUNCTION 'WS_UPLOAD'

    EXPORTING
*       CODEPAGE                = ' '
       FILENAME                = ARQ
*       FILETYPE                = ' '
*       HEADLEN                 = ' '
*       LINE_EXIT               = ' '
*       TRUNCLEN                = ' '
*       USER_FORM               = ' '
*       USER_PROG               = ' '
*    importing
*       filelength              =
    TABLES
       DATA_TAB                = T_ZCURSO
    EXCEPTIONS
       CONVERSION_ERROR        = 1
       FILE_OPEN_ERROR         = 2
       FILE_READ_ERROR         = 3
       INVALID_TABLE_WIDTH     = 4
       INVALID_TYPE            = 5
       NO_BATCH                = 6
       UNKNOWN_ERROR           = 7
       OTHERS                  = 8.
```

```
CLEAR V_CONT.

LOOP AT T_ZCURSO.

 ZCURSO-ZDATA   = T_ZCURSO-ZDATA.
 ZCURSO-ZNUMERO = T_ZCURSO-ZBELNR.
 ZCURSO-ZANO    = T_ZCURSO-ZGJAHR.
 INSERT ZCURSO.

 IF SY-SUBRC = 0.
   V_CONT = V_CONT + 1.
 ENDIF.

ENDLOOP.

WRITE: 'FORAM INSERIDOS ', V_CONT, 'NA TABELA ZCURSO'.
```

3.2 EXERCÍCIOS

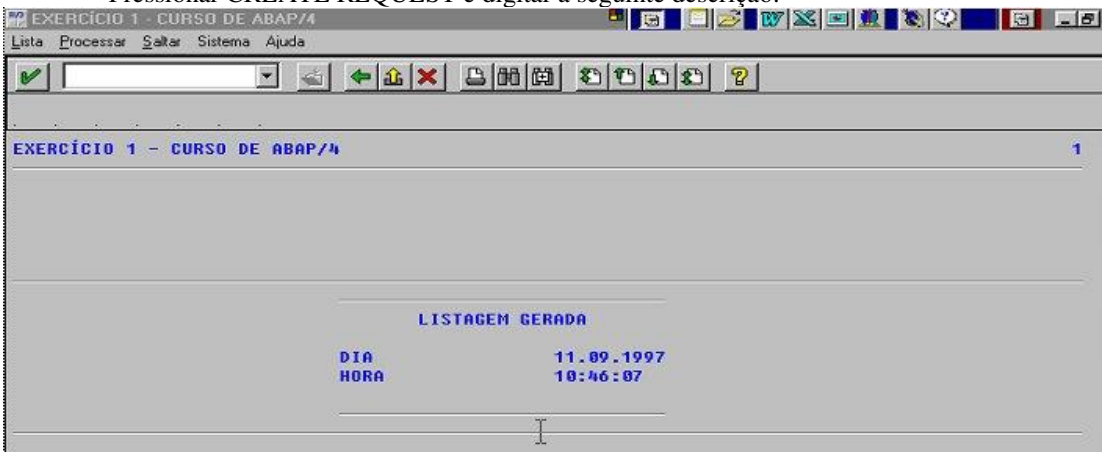
EXERCÍCIO 1

Desenvolver um programa ZEXC##01 que resulte a listagem seguinte onde :

a) ## - Número do aluno;
b) Variáveis de sistema utilizadas :
   SY-DATUM
   SY-UZEIT

OBSERVAÇÕES:

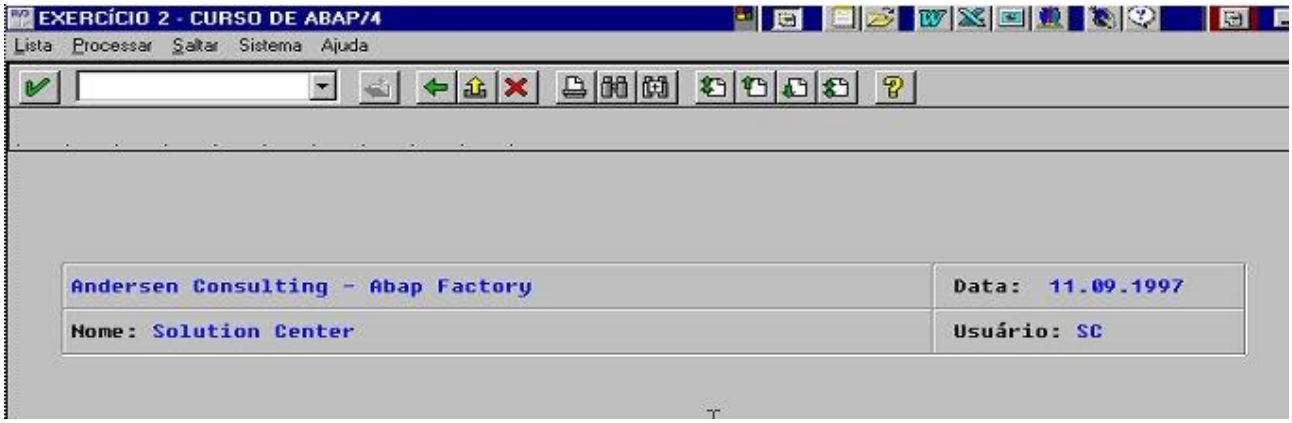
- ✓ Após indicar a Development Class: SALVAR o objeto, porém NÃO salvar como Local Object, quando o SAP perguntará por um Request Number;
- ✓ Pressionar CREATE REQUEST e digitar a seguinte descrição:



EXERCÍCIO 2

Desenvolver um programa ZEXC##02 que resulte o cabeçalho seguinte onde :

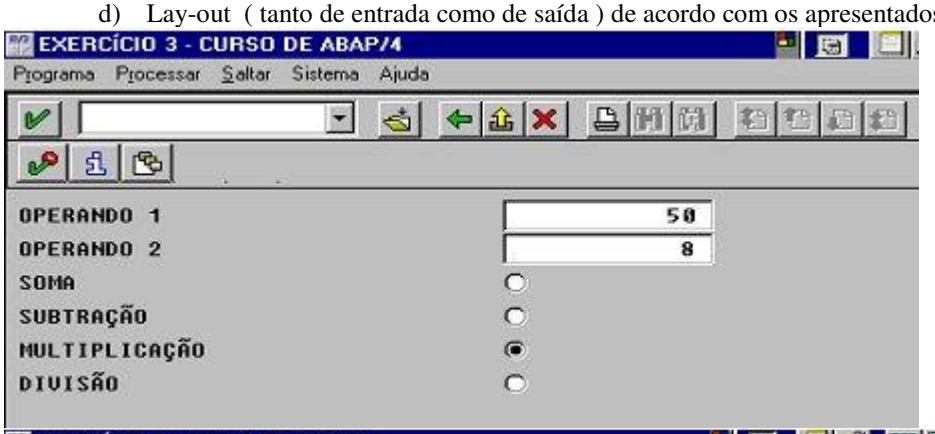
a) ## - Número do aluno
b) Variável de sistema utilizada :
   SY-UNAME
   SY-DATUM

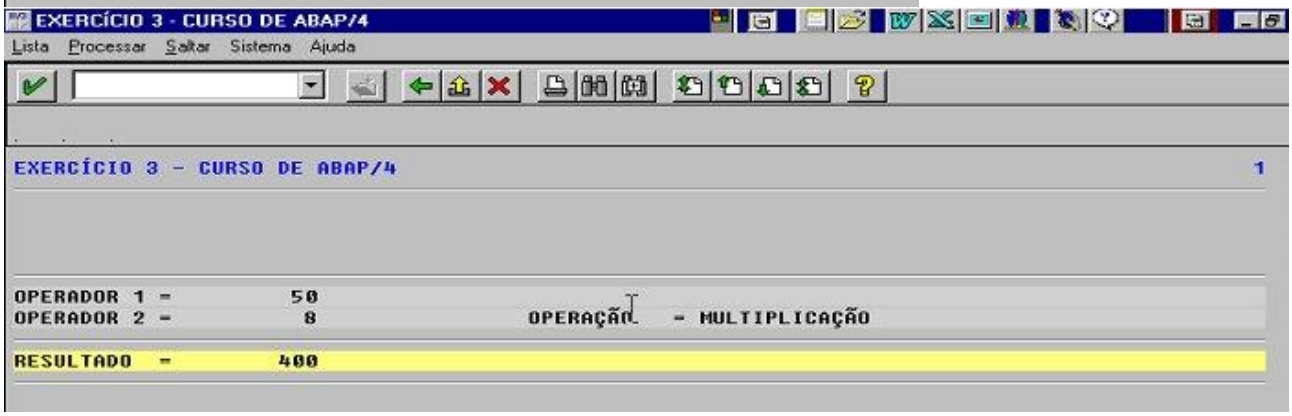


EXERCÍCIO 3

Desenvolver um programa ZEXC##03 que possua as seguintes características :

a) ## - Número do aluno
b) Simulação de uma calculadora com 4 operações (+, -, /, *) e dois campos para entrada de dados
c) Exigência quanto à entrada de todos os dados
d) Lay-out (tanto de entrada como de saída) de acordo com os apresentados a seguir



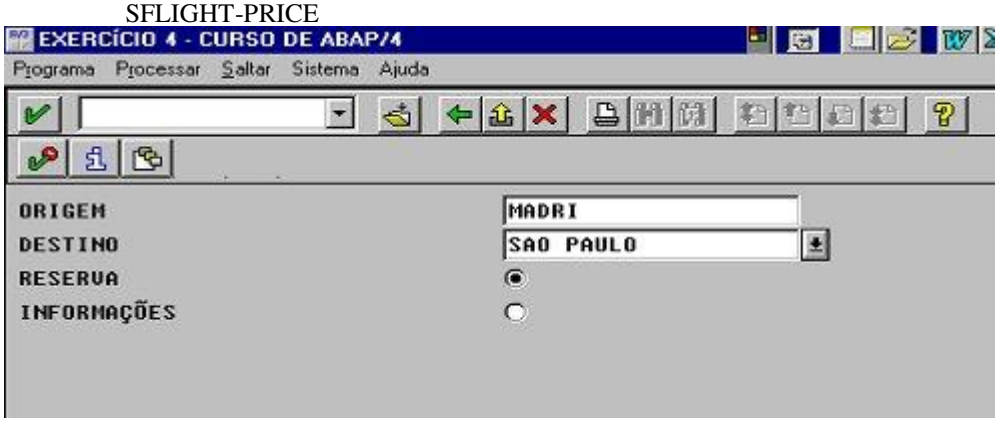


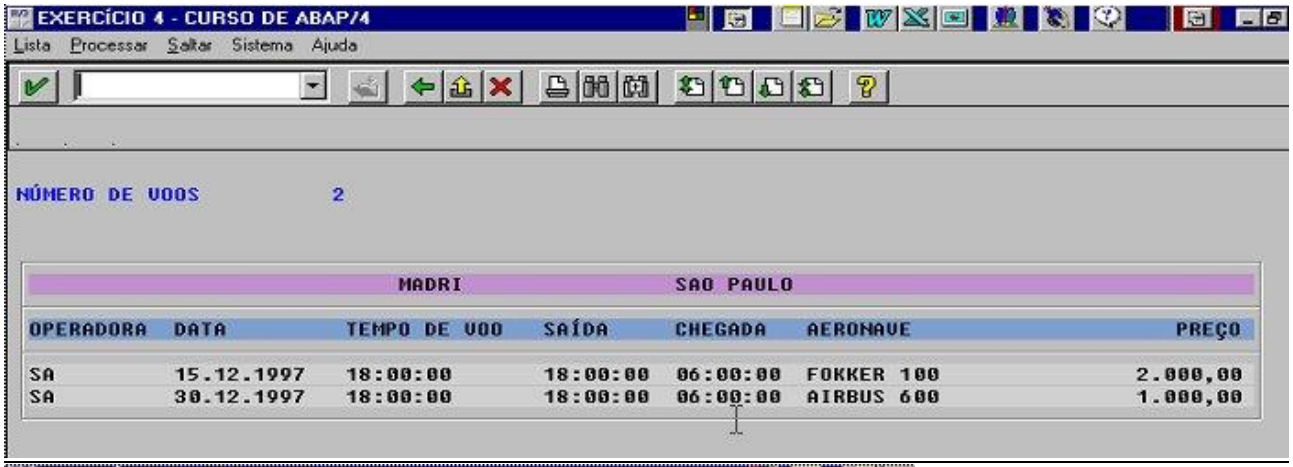
EXERCÍCIO 4

Desenvolver um programa ZEXC##04 que possua as seguintes características :

a) ## - Número do aluno
b) Que através das tabelas SPFLI e SFLIGHT resulte um relatório dos vôos existentes para um determinado percurso, sendo que a exposição da aeronave utilizada e do preço da passagem ocorram apenas quando for uma operação de reserva
c) Lay-out (tanto de entrada como de saída) de acordo com os apresentados a seguir
d) Campos utilizados :

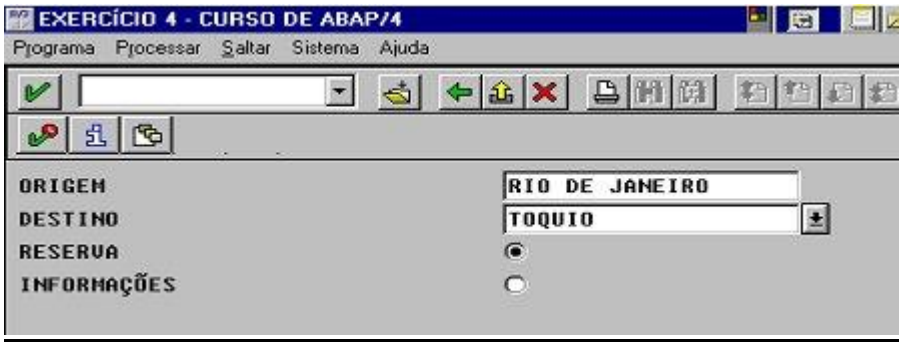
SPFLI-CARRID
SPFLI-CITYFROM
SPFLI-CITYTO
SPFLI-FLTIME
SPFLI-ARRTIME
SPFLI-DEPTIME
SFLIGHT-FLDATE
SFLIGHT-PLANETYPE
SFLIGHT-PRICE

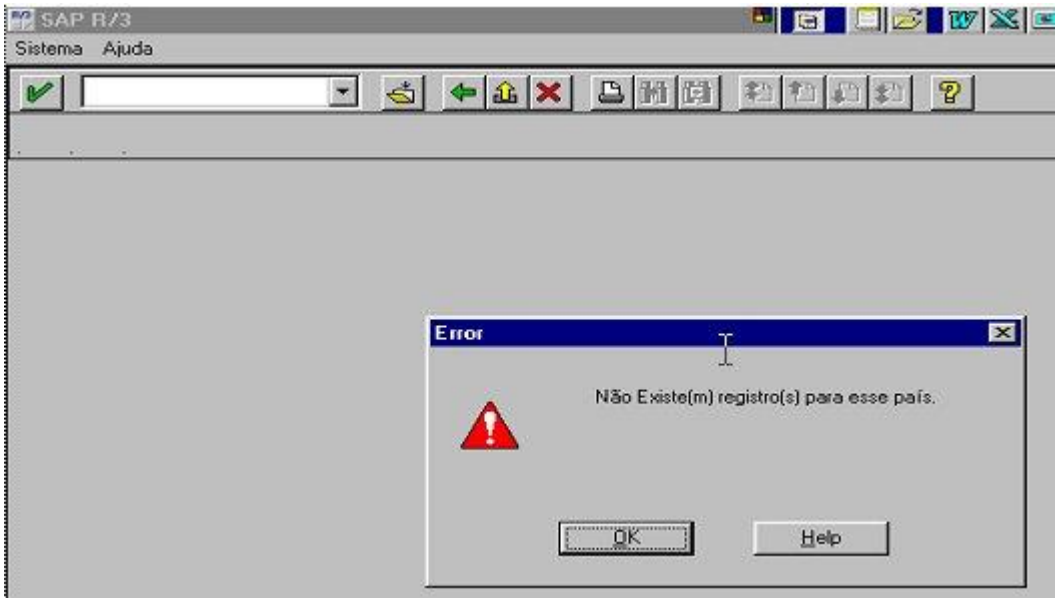



EXERCÍCIO 4 - CURSO DE ABAP/4
Lista Processar Saltar Sistema Ajuda
NÚMERO DE VOOS 2
MADRI SAO PAULO
OPERADORA DATA TEMPO DE VOO SAÍDA CHEGADA AERONAVE PREÇO
SA 15.12.1997 18:00:00 18:00:00 06:00:00 FOKKER 100 2.000,00
SA 30.12.1997 18:00:00 18:00:00 06:00:00 AIRBUS 600 1.000,00


EXERCÍCIO 4 - CURSO DE ABAP/4
Programa Processar Saltar Sistema Ajuda
ORIGEM MADRI
DESTINO SAO PAULO
RESERVA
INFORMAÇÕES


EXERCÍCIO 4 - CURSO DE ABAP/4
Lista Processar Saltar Sistema Ajuda
NÚMERO DE VOOS 2
MADRI SAO PAULO
OPERADORA DATA TEMPO DE VOO SAÍDA CHEGADA
SA 15.12.1997 18:00:00 18:00:00 06:00:00
SA 30.12.1997 18:00:00 18:00:00 06:00:00

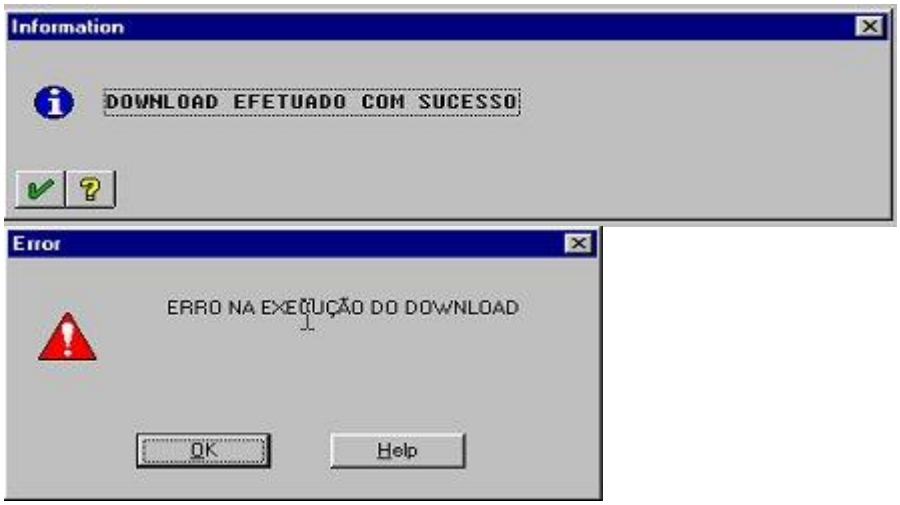

EXERCÍCIO 4 - CURSO DE ABAP/4
Programa Processar Saltar Sistema Ajuda
ORIGEM RIO DE JANEIRO
DESTINO TOQUIO
RESERVA
INFORMAÇÕES



EXERCÍCIO 5

Desenvolver um programa ZEXC##05 que a partir de um arquivo texto contendo a declaração de uma das aeronaves citadas a seguir, seja capaz de criar um outro arquivo com a declaração da aeronave bem como o nome da companhia aérea. O programa deve conter as seguintes características:

a) ## - Número do aluno
b) Exibição de mensagem tanto no caso de sucesso na criação do arquivo final como no caso da ocorrencia de algum erro.
c) Lay-out (tanto de entrada como de saída) de acordo com os apresentados a seguir
d) Campos utilizados : SFLIGHT-CARRID SFLIGHT-PLANETYPE



3.3 RESOLUÇÕES

RESOLUÇÃO 1

```
REPORT ZEXC0001.

SKIP 5.

ULINE.

ULINE 30(29).

WRITE: /37 TEXT-001.

SKIP.
```

```
WRITE: 30'DIA',
            49 SY-DATUM,
           /'HORA' UNDER 'DIA',
            SY-UZEIT UNDER SY-DATUM.

SKIP.

ULINE 30(29).

ULINE.
```

RESOLUÇÃO 2

```
REPORT ZEXC0002 NO STANDARD PAGE HEADING.

SKIP 4.

ULINE 4(88).

WRITE: /4 '|',
             5 TEXT-001,
            68 '|',
            77 SY-DATUM,
            91 '|'.

FORMAT INTENSIFIED OFF.

WRITE: 70 'Data: ',
           /1 ''.

ULINE 4(88).

WRITE: /5 'Nome:',
            70 'Usuário:'.

FORMAT INTENSIFIED ON.

WRITE: 4 '|',
           11 'Solution Center',
           68 '|',
           79  SY-UNAME,
           91 '|',
           /1 ''.

ULINE 4(88).
```

RESOLUÇÃO 3

```
REPORT ZEXC0003 MESSAGE-ID YA.

PARAMETER : P_FATOR1(5)   TYPE I,
                    P_FATOR2(5)   TYPE I,
                    P_ADD           RADIOBUTTON GROUP G1,
                    P_SUB            RADIOBUTTON GROUP G1,
                    P_MULT          RADIOBUTTON GROUP G1,
                    P_DIV            RADIOBUTTON GROUP G1.

DATA : RESULT(6)    TYPE I,
           OPERACAO(15) TYPE C.

IF P_FATOR1 = '' OR P_FATOR2 = ''.
```

```
   MESSAGE I005.
ELSE.

IF P_ADD = 'X'.
    RESULT = P_FATOR1 + P_FATOR2.
    OPERACAO = 'SOMA'.
ENDIF.

IF P_SUB = 'X'.
   RESULT = P_FATOR1 - P_FATOR2.
   OPERACAO = 'SUBTRAÇÃO'.
ENDIF.

IF P_MULT = 'X'.
   RESULT = P_FATOR1 * P_FATOR2.
   OPERACAO = 'MULTIPLICAÇÃO'.
ENDIF.

IF P_DIV = 'X'.
   RESULT = P_FATOR1 / P_FATOR2.
   OPERACAO = 'DIVISÃO'.
ENDIF.

SKIP 3.

ULINE.

FORMAT COLOR COL_NORMAL.

WRITE :   'OPERADOR 1 =',
               P_FATOR1,
         132 ' ',
            / 'OPERADOR 2 =',
                P_FATOR2,
          40 'OPERAÇÃO   =',
               OPERACAO,
         132 ' '.

ULINE.

FORMAT COLOR COL_TOTAL.

WRITE : / 'RESULTADO  =',
                RESULT,
          132 ' '.

ULINE.

ENDIF.
```

RESOLUÇÃO 4

```
REPORT ZEXC0004 MESSAGE-ID YA NO STANDARD PAGE HEADING.

PARAMETER: P_ORIGEM  LIKE SPFLI-CITYFROM,
                    P_DESTIN  LIKE SPFLI-CITYTO,
                    P_RESERV  RADIOBUTTON GROUP G1,
                    P_DEST    RADIOBUTTON GROUP G1.

TABLES: SPFLI, SFLIGHT.

DATA : V_CONTADOR TYPE I,
           V_VAR1     TYPE I.
```

```
DATA: BEGIN OF T_VOO OCCURS 0,
            CARRID    LIKE SPFLI-CARRID,
            CITYFROM  LIKE SPFLI-CITYFROM,
            CITYTO    LIKE SPFLI-CITYTO,
            FLTIME    LIKE SPFLI-FLTIME,
            ARRTIME   LIKE SPFLI-ARRTIME,
            DEPTIME   LIKE SPFLI-DEPTIME,
            FLDATE    LIKE SFLIGHT-FLDATE,
            PLANETYPE LIKE SFLIGHT-PLANETYPE,
            PRICE     LIKE SFLIGHT-PRICE.
DATA: END OF T_VOO.

SELECT * FROM SPFLI WHERE SPFLI-CITYFROM  =  P_ORIGEM
                      AND SPFLI-CITYTO       =  P_DESTIN.

  SELECT * FROM SFLIGHT WHERE SFLIGHT-CARRID   = SPFLI-CARRID
                          AND SFLIGHT-CONNID = SPFLI-CONNID.

    V_CONTADOR          = V_CONTADOR + 1.
    T_VOO-CARRID        = SPFLI-CARRID.
    T_VOO-CITYFROM    = SPFLI-CITYFROM.
    T_VOO-CITYTO        = SPFLI-CITYTO.
    T_VOO-DISTANCE    = SPFLI-DISTANCE.
    T_VOO-FLTIME        = SPFLI-FLTIME.
    T_VOO-ARRTIME     = SPFLI-ARRTIME.
    T_VOO-DEPTIME     = SPFLI-DEPTIME.
    T_VOO-DISTID        = SPFLI-DISTID.
    T_VOO-FLDATE      = SFLIGHT-FLDATE.
    T_VOO-PLANETYPE = SFLIGHT-PLANETYPE.
    T_VOO-PRICE         = SFLIGHT-PRICE.
    APPEND T_VOO.

  ENDSELECT.
ENDSELECT.

IF SY-SUBRC NE 0.
   MESSAGE E006.
ENDIF.

SKIP.

WRITE: TEXT-001,
          V_CONTADOR,
       95 ".

SKIP 2.

V_VAR1 = 1.


LOOP AT T_VOO.
  IF V_VAR1 = 1.
     ULINE.
     FORMAT COLOR COL_GROUP.
    WRITE: /   '|',
               30 T_VOO-CITYFROM,
                  T_VOO-CITYTO,
              95 '|'.

     CLEAR V_VAR1.

     ULINE.
     FORMAT COLOR COL_HEADING.
```

```
  WRITE: /  '|',
               2 'OPERADORA',
              13 'DATA',
              26 'TEMPO DE VOO',
              41 'SAÍDA',
              51 'CHEGADA'.

     IF P_RESERV = 'X'.
        WRITE: 61 'AERONAVE',
                  89 'PREÇO'.
     ENDIF.

     WRITE:   95 '|'.

     ULINE.

  ENDIF.

  FORMAT COLOR COL_NORMAL.
  WRITE:   / '|',
               2  T_VOO-CARRID,
              13  T_VOO-FLDATE,
              26  T_VOO-FLTIME,
              41  T_VOO-DEPTIME,
              51  T_VOO-ARRTIME.

  IF P_RESERV = 'X'.
     WRITE: 61 T_VOO-PLANETYPE,
               75 T_VOO-PRICE.
  ENDIF.

  WRITE:   95 '|'.

ENDLOOP.

ULINE
```

RESOLUÇÃO 5

```
REPORT ZEXC0005 MESSAGE-ID YA.

TABLES: SFLIGHT.

PARAMETER: P_ARQ  LIKE RLGRAP-FILENAME DEFAULT 'C:\TEMP\NAVE.TXT',
                   P_ARQ2 LIKE RLGRAP-FILENAME DEFAULT 'C:\TEMP\DADOS.TXT'.

DATA: BEGIN OF T_ZCURSO OCCURS 0,
             ZAERONAV LIKE SFLIGHT-PLANETYPE.
DATA: END OF T_ZCURSO.
```

```
DATA: BEGIN OF T_ZCURSO2 OCCURS 0,
              ZAERONAV LIKE SFLIGHT-PLANETYPE,
              ESPACE1 TYPE C VALUE ' ',
              ZCOMPANY LIKE SFLIGHT-CARRID.
DATA: END OF T_ZCURSO2.

CALL FUNCTION 'WS_UPLOAD'
    EXPORTING
*       CODEPAGE                = ' '
        FILENAME                = P_ARQ
*       FILETYPE                = ' '
*       HEADLEN                 = ' '
*       LINE_EXIT               = ' '
*       TRUNCLEN                = ' '
*       USER_FORM               = ' '
*       USER_PROG               = ' '
*    importing
*       filelength              =
    TABLES
        DATA_TAB                = T_ZCURSO
    EXCEPTIONS
        CONVERSION_ERROR        = 1
        FILE_OPEN_ERROR         = 2
        FILE_READ_ERROR         = 3
        INVALID_TABLE_WIDTH     = 4
        INVALID_TYPE            = 5
        NO_BATCH                = 6
        UNKNOWN_ERROR           = 7
        OTHERS                  = 8.

 SELECT * FROM SFLIGHT WHERE PLANETYPE = T_ZCURSO-ZAERONAV.

          T_ZCURSO2-ZAERONAV = SFLIGHT-PLANETYPE.
          T_ZCURSO2-ZCOMPANY = SFLIGHT-CARRID.
          APPEND T_ZCURSO2.

 ENDSELECT.

 IF SY-SUBRC = 0.
   CALL FUNCTION 'WS_DOWNLOAD'
       EXPORTING
*          BIN_FILESIZE                 = ' '
*          CODEPAGE                     = ' '
             FILENAME                   = P_ARQ2
*          FILETYPE                     = ' '
*          MODE                         = ' '
*          WK1_N_FORMAT                 = ' '
*          WK1_N_SIZE                   = ' '
*          WK1_T_FORMAT                 = ' '
*          WK1_T_SIZE                   = ' '
*          COL_SELECT                   = ' '
*          COL_SELECTMASK               = ' '
*      importing
*          filelength                   =
       TABLES
          DATA_TAB                      = T_ZCURSO2
*          FIELDNAMES                   =
       EXCEPTIONS
          FILE_OPEN_ERROR               = 1
          FILE_WRITE_ERROR              = 2
          INVALID_FILESIZE              = 3
          INVALID_TABLE_WIDTH           = 4
```

```
        INVALID_TYPE                = 5
        NO_BATCH                    = 6
        UNKNOWN_ERROR               = 7
        OTHERS                      = 8.

  MESSAGE I007.
ELSE.
  MESSAGE E008.
ENDIF.
```

4 TEORIA BDC SESSION

4.1 BDC Session

ABAP/4 tem uma técnica de programação para a colocação de dados dentro do SAP conhecida como Batch Data Communication Session ou BDC Session.

4.2 Passos para criação de uma BDC Session

- ✓ Identificar as telas que a transação processará

- ✓ Escrever o programa em ABAP para gerar a tabela de BDC que submeterá os dados na transação
- ✓ Submeter a tabela de BDC para o sistema em modo batch ou através do comando CALL TRANSACTION

4.3 Identificando telas em uma transação

Quando um usuário entra com dados no SAP utiliza transações. Cada transação tem várias telas identificadas por um nome de programa e um número de tela. As informações sobre a tela atual é obtida através no menu System, item Status.

(Tela Status)

Além de identificar o nome do programa e número da tela, deve-se também identicar o(s) campo(s) que se deseja entrar com o dados. Para conseguir saber o nome da tabela/estrutura e o nome do campo deve-se clicar sobre o campo que entraria com o dado e teclar <F1> seguido do botão "Informações Técnicas".

<Tela Informação Técnica>

Nesta tela consegue-se quase todos os dados para a sessão de BDC. O nome do programa, o número da tela e o nome do campo para a sessão de Batch input.

Além destes dados deve-se saber quais as teclas/funções de movimentação entres as telas. Por exemplo, se para passar para a próxima tela da transação deve-se teclar <Enter>, o código para a BDC é "/0".

De modo geral, deve-se pensar na transação sem a utilização do mouse. Se um botão deve ser clicado pelo mouse deve-se descobrir qual o nome da função deste botão e passar este código para a tabela BDC.

4.4 Gerando a tabela BDC

A tabela BDC é uma tabela interna com uma estrutura específica no qual é preenchida para ser enviada para a sessão batch input. Esta estrutura se chama BDCDATA e tem os seguintes campos:

Campo	Tipo	Descrição
program	Char(40)	Nome do programa da transação
dynpro	Numc(4)	Número da tela da transação
dynbegin	Char(1)	Indicador de uma nova tela
Fnam	Char(132)	Nome do campo da tela
fval	Char(132)	Valor a ser colocado no campo

Exemplo de uma tabela com estrutura BDCDATA com dados:

program	dynpro	dynbegin	fnam	fval
SAPMF02K	0100	X		
			RF02K-LIFNR	0010010
			RF02K-EKORG	CNTL
SAPMF02K	0200	X		
...	...	...	...	...

O código em ABAP para isto seria:

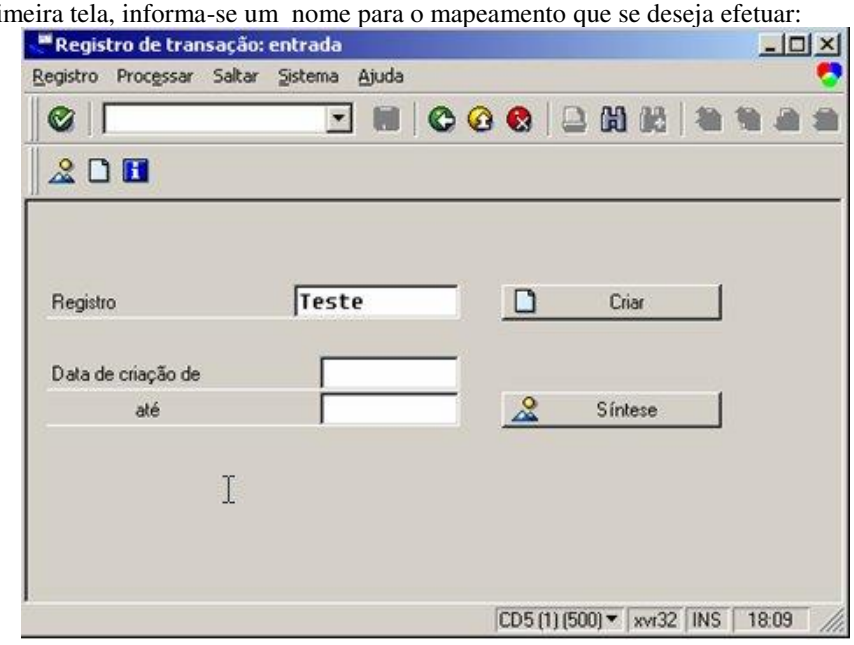
```
REPORT ZXXXXXXX.

DATA: BEGIN OF TBDC OCCURS 100.
  INCLUDE STRUCTURE BDCDATA.
DATA: END OF TBDC.

* Início do programa principal

MOVE 'SAPMF02K'    TO TBDC-PROGRAM.
MOVE '0100'        TO TBDC-DYNPRO.
MOVE 'X'           TO TBDC-DYNBEGIN.
APPEND TBDC.

MOVE 'RF02K-LIFNR' TO TBDC-FNAM.
MOVE '0010010'     TO TBDC-FVAL.
APPEND TBDC.

MOVE 'RF02K-EKORG' TO TBDC-FNAM.
MOVE 'CNTL'        TO TBDC-FVAL.
APPEND TBDC.

* E assim por diante até que a tabela esteja completa
```

É claro que existirá uma repetição muito grande de linhas para a criação de uma tabela BDC e por isso mesmo deve-se criar forms para agilizar esta movimentação.

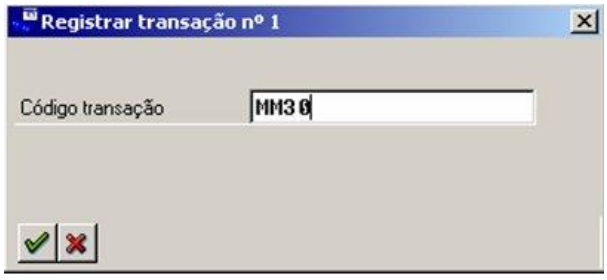
Para facilitar o trabalho de mapeamento dos campos, o SAP dispõe de uma ferramenta que faz isso para o programador. É a transação SHDB. Essa transação monitora todos os passos que o usuário faz quando utiliza uma transação e a resposta do SHDB é uma lista com os campos e telas que foram utilizados, permitindo que o programador crie as tabelas BDC. Há ainda um recurso que gera automaticamente o código do programa para a criação da tabela BDC, facilitando ainda mais o trabalho, mas a lista já é suficiente e às vezes preferível, para mantermos os mesmos padrões nos códigos.

Essa transação funciona da seguinte forma:

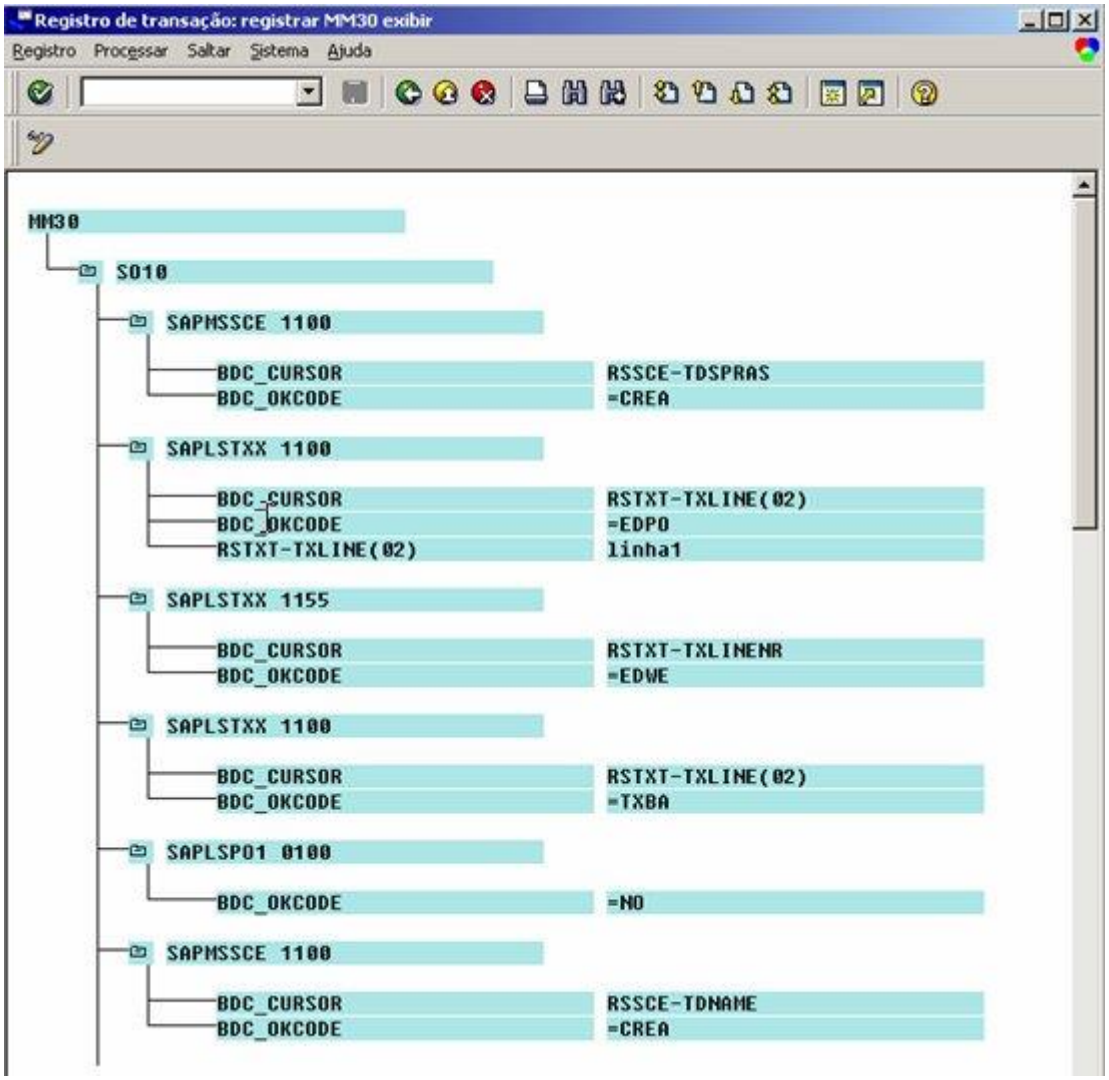
- ✓ Na primeira tela, informa-se um nome para o mapeamento que se deseja efetuar:



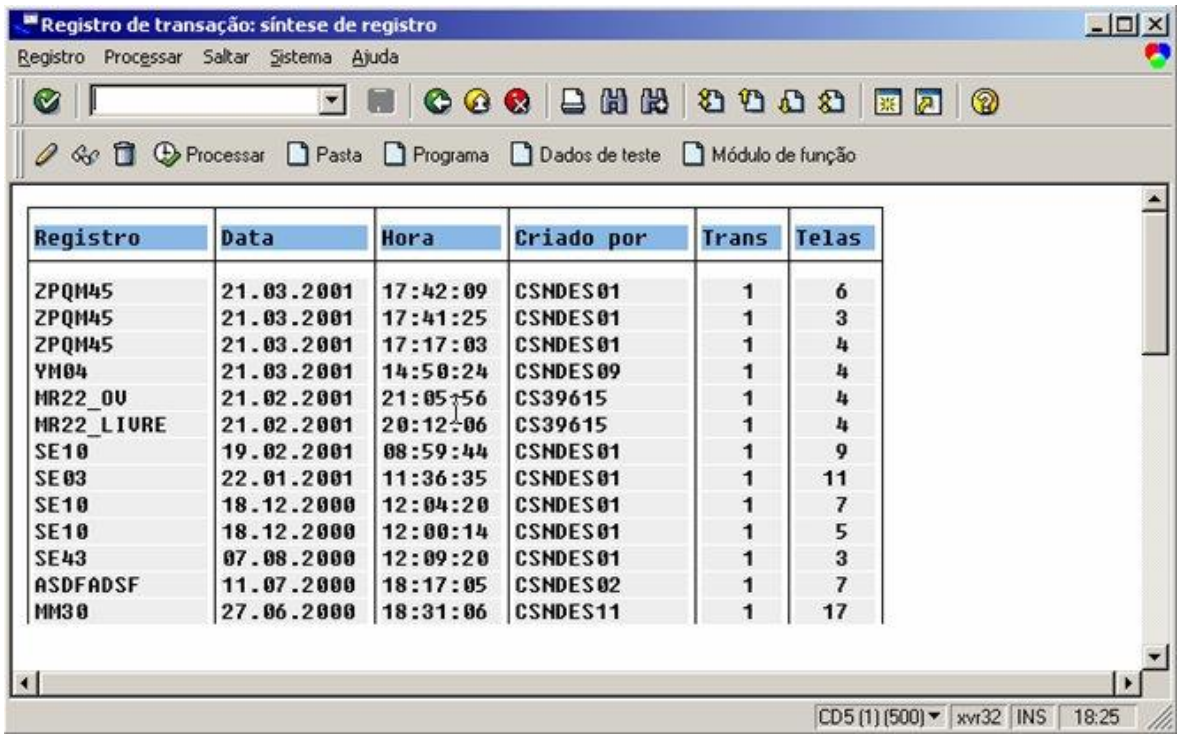
- ✓ Em seguida, informa-se a transação que deve ser mapeada.



A transação é então chamada. Deve-se fazer o processamento normal nesta transação. Quando terminar (salvar ou cancelar), o SHDB exibe a seguinte tela, com a lista dos campos e telas utilizados:



Caso esses dados satisfaçam o programador, pode-se salvá-los. Para a nova consulta, deve-se informar na primeira tela o nome com o qual o mapeamento foi salvo e clicar em síntese. Uma tela aparecerá com os mapeamentos correspondentes (pode-se criar mais de um com o mesmo nome):



Nela, teremos os seguintes botões:

Eles permitem que se crie uma pasta de Batch Input, um programa, dados de teste e ainda um módulo de função, com base no mapeamento feito.

4.5 Enviando uma tabela BDC para o sistema

Como foi citado no início, existem duas formas de se enviar uma tabela BDC para o sistema. Via CALL TRANSACTION ou via Batch Input.

4.6 Processando dados com CALL TRANSACTION

O comando CALL TRANSACTION possibilita o processamento de uma tabela BDC imediatamente pelo sistema. Os dados da tabela BDC são utilizados para executar a transação e o return code deste comando nos mostra se a transação foi executada com sucesso ou não.

A sintaxe deste comando é:

CALL TRANSACTION trans [USING bdctab MODE mode].

Os modos para executar este comando são:

A **Mostra todas as telas**
E **Mostra apenas telas com erros**
N **Não mostra as telas**

Em adição ao return code, podemos utilizar as variáveis de sistema para recuperar a mensagem que o SAP enviou ao término do processamento. As variáveis mais utilizadas são: SY-MSGID, SY-MSGV1, SY-MSGV2, SY-MSGV3 e SY-MSGV4.

4.7 Processando dados com BDC_INSERT

A segunda maneira para processar uma sessão de BDC é submetendo-a ao sistema via processamento batch. Com este método, várias transações podem ser executadas pelo SAP mas, ao contrário do CALL TRANSACTION, estas transações não serão executadas imediatamente, serão colocadas em uma pasta de Batch Input que pode ser executada na transação SM35 ou schedulada para rodar em um dia e uma hora desejada.

Existem três módulos de funções que devem ser executados para este tipo de processamento.

BDC_OPEN_GROUP
Esta função abre a sessão de BDC e necessita ser chamada antes de qualquer processamento.

BDC_INSERT
Esta função é chamada para cada transação no processamento Batch.

BDC_CLOSE_GROUP
Esta função deve ser chamada após todo processamento para que a pasta de processamento Batch seja criada.

4.8 Função BDC_OPEN_GROUP

Os seguintes parâmetros são passados para a função:

CLIENT	Cliente do SAP que será processada a sessão
GROUP	Nome para a pasta de Batch input (não precisa ser único)
HOLDDATE	Suspende o processamento da pasta de Batch input até a data especificada
KEEP	Mantem a pasta de Batch Input após o processamento
USER	Nome de usuário que executará o Batch Input.

4.9 Função BDC_INSERT

Os seguintes parâmetros são passados para a função:

TCODE	Código da transação na qual será executada a tabela BDC
DYNPROTAB	Nome da tabela interna utilizada para gerar a pasta de Batch

4.10 Função BDC_CLOSE_GROUP

Não existem parâmetros a serem passados para esta função.

5 EXEMPLOS BDC SESSION

```
* Exemplo:
* Envio de dados para a transação FD01 através de Batch Input ou Call
* Transaction (a escolha do usuário).
* Ao final do programa um relatório é emitido com a tabela de BDC
* feita.
* Ao executar o programa escolha a variante criada para que os dados
* sejam preenchidos.
*
* Se algum erro ocorrer na transação/dados via Call Transaction uma
* pasta de erro será gerada para posterior execução através da transa-
* ção SM35.
*
PROGRAM ZZBDCXX1 LINE-SIZE 255
                MESSAGE-ID ZZ.

DATA: BEGIN OF T_BDC OCCURS 0.
  INCLUDE STRUCTURE BDCDATA.
DATA: END OF T_BDC.

SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE T1.
PARAMETERS: P_BUKRS LIKE KNB1-BUKRS,              "Empresa
            P_KTOKD LIKE KNA1-KTOKD,              "Grupo de contas
            P_NAME1 LIKE KNA1-NAME1,              "Nome
            P_SORTL LIKE KNA1-SORTL,              "Termo de busca
            P_ORT01 LIKE KNA1-ORT01,              "Cidade
            P_LAND1 LIKE KNA1-LAND1,              "país
            P_PSTLZ LIKE KNA1-PSTLZ,              "Caixa postal
            P_LIFNR LIKE KNA1-LIFNR,              "Fornecedor
            P_AKONT LIKE KNB1-AKONT,              "Conta de reconciliação
            P_FDGRV LIKE KNB1-FDGRV,
            P_VZSKZ LIKE KNB1-VZSKZ,
            P_ZTERM LIKE KNB1-ZTERM,
            P_TOGRU LIKE KNB1-TOGRU,
            P_XZVER AS CHECKBOX DEFAULT 'X',
            P_XVERR AS CHECKBOX DEFAULT 'X'.

SELECTION-SCREEN END OF BLOCK B1.


SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE T2.
PARAMETERS: P_BATCH RADIOBUTTON GROUP R1,  "via Batch input
            P_CALL  RADIOBUTTON GROUP R1,              "via Call Transaction
            P_MODE(1) TYPE C DEFAULT 'N'.                        "modo call transaction
SELECTION-SCREEN END OF BLOCK B2.

INITIALIZATION.
T1 = 'Dados para a sessão BDC'.
T2 = 'Tipo processamento da sessão BDC'.

START-OF-SELECTION.

* Cria os registros na tabela de BDC
  PERFORM F_WRITE_RECORDS.

* Envia a sessão BDC
  PERFORM F_SUBMIT_BDC.

* Gera um relatório de auditoria
  PERFORM F_WRITE_LOG.
```

```
END-OF-SELECTION.

FORM F_WRITE_RECORDS.

* programa SAPMF02D - tela 105
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02D' '0105'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'RF02D-BUKRS' P_BUKRS,
                  USING 'RF02D-KTOKD' P_KTOKD,
                  USING 'BDC_OKCODE'  '/0'.

* programa SAPMF02D - tela 110
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02D' '0110'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'KNA1-NAME1' P_NAME1,
                  USING 'KNA1-SORTL' P_SORTL,
                  USING 'KNA1-SPRAS' SY-LANGU,
                  USING 'KNA1-PSTLZ' P_PSTLZ,
                  USING 'KNA1-ORT01' P_ORT01,
                  USING 'KNA1-LAND1' P_LAND1,
                  USING 'BDC_OKCODE' '/0'.

* programa SAPMF02D - tela 120
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02D' '0120'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'KNA1-LIFNR' P_LIFNR,
                  USING 'BDC_OKCODE' '/0'.

* programa SAPMF02D - tela 130
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02D' '0130'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'BDC_OKCODE' '/0'.

* programa SAPMF02D - tela 210
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02D' '0210'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'KNB1-AKONT' P_AKONT,
                  USING 'KNB1-FDGRV' P_FDGRV,
                  USING 'KNB1-VZSKZ' P_VZSKZ,
                  USING 'BDC_OKCODE' '/0'.

* programa SAPMF02D - tela 215
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02D' '0215'.

PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'KNB1-ZTERM' P_ZTERM,
                  USING 'KNB1-TOGRU' P_TOGRU,
                  USING 'KNB1-XZVER' P_XZVER,
                  USING 'KNB1-XVERR' P_XVERR,
                  USING 'BDC_OKCODE' '/11'.
```

```
ENDFORM.



FORM F_SUBMIT_BDC.

* Verificação do tipo de processamento
  IF P_BATCH EQ 'X'.

*    "Abre a pasta de Batch Input
    CALL FUNCTION 'BDC_OPEN_GROUP'
        EXPORTING
            CLIENT    = SY-MANDT
            GROUP     = 'FD01-EX01'
            KEEP      = 'X'
            USER      = SY-UNAME.

*    Insere a tabela de BDC na pasta
    CALL FUNCTION 'BDC_INSERT'
        EXPORTING
            TCODE     = 'FD01'
        TABLES
            DYNPROTAB = T_BDC.

    IF SY-SUBRC NE 0.
      MESSAGE E000 WITH 'Erro na função BDC_INSERT'.
    ENDIF.

*    Fecha a pasta de batch Input
    CALL FUNCTION 'BDC_CLOSE_GROUP'.

  ELSE.


*    Chama o método CALL TRANSACTION para inserir os dados
    CALL TRANSACTION 'FD01' USING T_BDC
                        MODE P_MODE
                        UPDATE 'S'.

    IF SY-SUBRC NE 0.
      IF P_MODE NE 'N'.
        CALL FUNCTION 'BDC_OPEN_GROUP'
          EXPORTING
              CLIENT    = SY-MANDT
              GROUP     = 'FD01-ERR01'
              KEEP      = 'X'
              USER      = SY-UNAME.

*       Insere a tabela de BDC na pasta
        CALL FUNCTION 'BDC_INSERT'
          EXPORTING
              TCODE     = 'FD01'
          TABLES
              DYNPROTAB = T_BDC.

*       Fecha a pasta de batch Input
        CALL FUNCTION 'BDC_CLOSE_GROUP'.
      ENDIF.

      MESSAGE E000 WITH 'Erro no CALL TRANSACTION'.
    ENDIF.
  ENDIF.
```

```
ENDFORM.


FORM F_BDC_SCREEN TABLES P_BDC STRUCTURE BDCDATA
              USING  P_PROGRAM P_SCREEN.

  CLEAR P_BDC.

  MOVE: P_PROGRAM TO P_BDC-PROGRAM,
        P_SCREEN  TO P_BDC-DYNPRO,
        'X'       TO P_BDC-DYNBEGIN.


APPEND P_BDC.

ENDFORM.

FORM F_BDC_FIELD TABLES P_BDC STRUCTURE BDCDATA
              USING P_NAME P_VALUE.


  CASE P_VALUE.
  WHEN SPACE.

  WHEN OTHERS.
    CLEAR P_BDC.

    MOVE: P_NAME    TO P_BDC-FNAM,
          P_VALUE   TO P_BDC-FVAL.

    APPEND P_BDC.

  ENDCASE.

ENDFORM.


FORM F_WRITE_LOG.

  ULINE.
  WRITE: /01 '|  Program | Dynpro | Dynbegin |  Field Name | Value',
         255 '|'.
  ULINE.

  LOOP AT T_BDC.

    WRITE: /01 '|', T_BDC-PROGRAM, '|'.
    IF T_BDC-DYNPRO NE '0000'.
      WRITE: 15 T_BDC-DYNPRO, ' |'.
    ELSE.
      WRITE: 15 '    ', ' |'.
    ENDIF.

WRITE: 28 T_BDC-DYNBEGIN,
          32 '|', T_BDC-FNAM,
          46 '|', T_BDC-FVAL,
          255 '|'.

  ENDLOOP.

  IF SY-SUBRC NE 0.
```

```
   MESSAGE E000 WITH 'Nenhum registro foi gravado na tabela de BDC'.
  ENDIF.

  ULINE.

ENDFORM.
```

6 EXERCÍCIOS BDC SESSION

```
* Exercício proposto:
* Desenvolver um programa em ABAP que envie dados para a transação FS01
* através de CALL TRANSACTION ou BATCH INPUT de acordo com o desejado
* pelo usuário seguinto o seguinte mapeamento da transação:
*
* Programa   Tela   Início          Campo          Valor
* SAPMF02H   0402     X
*                                   RF02H-SAKNR    '41XXCC'
*                                   RF02H-BUKRS    'AC25'
*                                   BDC_OKCODE     '/0'
* SAPMF02H   0310     X
*                                   SKAT-TXT20     '41XXCC - TESTE CC'
*                                   SKA1-GVTYP     'X '
*                                   SKA1-KTOKS     '04'
*                                   BDC_OKCODE     '/0'
* SAPMF02H   0110     X
*                                   SKB1-WAERS     'BRL '
*                                   SKB1-MWSKZ     'A0'
*                                   SKB1-FSTAG     'AC25'
*                                   BDC_OKCODE     '/11'
*
* Onde: XX é o número de sua estação
*       CC é o contador da quantidade de contas incluídas
*
* Ao final do processamento deve ser feito um relatório de auditoria
* com o nome dos campos e respectivos valores
*
* Como opção, pode-se criar um botão para fazer um download deste
* relatório para o HD Local ou para o diretório /tmp/ do servidor
* UNIX
*
* 2a. parte:
* Rodar o programa em modo CALL TRANSACTION e se algum erro ocorrer
* com a transação/dados, colocar os registros errados um uma pasta
* de batch input para que possa rodar o processo novamente sem perder
* os dados. Forçar o erro.



* Resolução:
PROGRAM ZZBDCXX2 LINE-SIZE 255
              MESSAGE-ID ZZ.

DATA: BEGIN OF T_BDC OCCURS 0.
  INCLUDE STRUCTURE BDCDATA.
DATA: END OF T_BDC.

SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE T1.
PARAMETERS: P_BUKRS LIKE RF02H-BUKRS,
          P_SAKNR LIKE RF02H-SAKNR,
          P_TXT20 LIKE SKAT-TXT20,
          P_GVTYP LIKE SKA1-GVTYP,
          P_KTOKS LIKE SKA1-KTOKS,
          P_WAERS LIKE SKB1-WAERS,
          P_MWSKZ LIKE SKB1-MWSKZ,
```

```
        P_FSTAG LIKE SKB1-FSTAG.
SELECTION-SCREEN END OF BLOCK B1.

SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE T2.
PARAMETERS: P_BATCH RADIOBUTTON GROUP R1, "via Batch input
        P_CALL  RADIOBUTTON GROUP R1. "via Call Transaction
SELECTION-SCREEN END OF BLOCK B2.

INITIALIZATION.
T1 = 'Dados para a sessão BDC'.
T2 = 'Tipo processamento da sessão BDC'.

START-OF-SELECTION.

* Cria os registros na tabela de BDC
  PERFORM F_WRITE_RECORDS.

* Envia a sessão BDC
  PERFORM F_SUBMIT_BDC.

* Gera um relatório de auditoria
  PERFORM F_WRITE_LOG.

END-OF-SELECTION.


FORM F_WRITE_RECORDS.

* programa SAPMF02H - tela 402
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02H' '0402'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'RF02H-BUKRS' P_BUKRS,
                  USING 'RF02H-SAKNR' P_SAKNR,
                  USING 'BDC_OKCODE'  '/0'.

* programa SAPMF02H - tela 310
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02H' '0310'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'SKAT-TXT20' P_TXT20,
                  USING 'SKA1-GVTYP' P_GVTYP,
                  USING 'SKA1-KTOKS' P_KTOKS,
                  USING 'BDC_OKCODE' '/0'.

* programa SAPMF02H - tela 110
  PERFORM F_BDC_SCREEN TABLES T_BDC
                  USING 'SAPMF02H' '0110'.

  PERFORM F_BDC_FIELD TABLES T_BDC:
                  USING 'SKB1-WAERS' P_WAERS,
                  USING 'SKB1-MWSKZ' P_MWSKZ,
                  USING 'SKB1-FSTAG' P_FSTAG,
                  USING 'BDC_OKCODE' '/11'.

ENDFORM.
```

```
FORM F_SUBMIT_BDC.

* Verificação do tipo de processamento
  IF P_BATCH EQ 'X'.
*    "Abre a pasta de Batch Input
    CALL FUNCTION 'BDC_OPEN_GROUP'
        EXPORTING
            CLIENT    = SY-MANDT
            GROUP     = 'FS01-RESOL'
            KEEP      = 'X'
            USER      = SY-UNAME.

*    Insere a tabela de BDC na pasta
    CALL FUNCTION 'BDC_INSERT'
        EXPORTING
            TCODE     = 'FS01'
        TABLES
            DYNPROTAB = T_BDC.

    IF SY-SUBRC NE 0.
      MESSAGE E000 WITH 'Erro na função BDC_INSERT'.
    ENDIF.

*    Fecha a pasta de batch Input
    CALL FUNCTION 'BDC_CLOSE_GROUP'.

  ELSE.

*    Chama o método CALL TRANSACTION para inserir os dados
    CALL TRANSACTION 'FS01' USING T_BDC
                        MODE 'A'
                        UPDATE 'S'.

    IF SY-SUBRC NE 0.
      MESSAGE E000 WITH 'Erro no CALL TRANSACTION'.
    ENDIF.
  ENDIF.


ENDFORM.

FORM F_BDC_SCREEN TABLES P_BDC STRUCTURE BDCDATA
                USING  P_PROGRAM P_SCREEN.

 CLEAR P_BDC.

  MOVE: P_PROGRAM TO P_BDC-PROGRAM,
        P_SCREEN  TO P_BDC-DYNPRO,
        'X'       TO P_BDC-DYNBEGIN.

  APPEND P_BDC.

ENDFORM.

FORM F_BDC_FIELD TABLES P_BDC STRUCTURE BDCDATA
```

```
              USING P_NAME P_VALUE.


  CASE P_VALUE.
  WHEN SPACE.

  WHEN OTHERS.
    CLEAR P_BDC.

    MOVE: P_NAME    TO P_BDC-FNAM,
          P_VALUE   TO P_BDC-FVAL.

    APPEND P_BDC.

  ENDCASE.

ENDFORM.

FORM F_WRITE_LOG.

  ULINE.
  WRITE: /01 '|  Program | Dynpro | Dynbegin |  Field Name | Value',
          255 '|'.
  ULINE.



  LOOP AT T_BDC.

    WRITE: /01 '|', T_BDC-PROGRAM, '|'.
    IF T_BDC-DYNPRO NE '0000'.
      WRITE: 15 T_BDC-DYNPRO, ' |'.
    ELSE.
      WRITE: 15 '    ', ' |'.
    ENDIF.
    WRITE: 28 T_BDC-DYNBEGIN,
           32 '|', T_BDC-FNAM,
           46 '|', T_BDC-FVAL,
           255 '|'.

  ENDLOOP.

  IF SY-SUBRC NE 0.
    MESSAGE E000 WITH 'Nenhum registro foi gravado na tabela de BDC'.
  ENDIF.

  ULINE.

ENDFORM.
```

6.1 Exercício BDC

Criar um programa que leia um arquivo e crie ordens de venda para os clientes de acordo com as especificações no arquivo.

A transação para criação de ordem de venda é a VA01.

Para se criar uma ordem de venda, deve-se seguir o seguinte fluxo:

- ✓ Chamar a transação VA01.
- ✓ Na primeira tela:
  - preencher Tipo da Ordem
  - <ENTER>
- ✓ Na segunda tela:
  - preencher Emissor da Ordem
  - preencher Recebedor da Mercadoria
  - preencher Data do documento do Faturamento
  - preencher Material
  - preencher Quantidade Prevista
- ✓ <SALVAR>
- ✓ <VOLTAR>

O arquivo está no seguinte formato:

- ✓ Tipo da Ordem
- ✓ Emissor da Ordem
- ✓ Recebedor da Mercadoria
- ✓ Material
- ✓ Quantidade Prevista

OBS: Os campos estão separados por ponto-e-vírgula (;)

Preencher a data do documento com a data de hj.

ARQUIVO

VEF;15;15;10010;1
VEF;14;14;10011;1
VEF;13;13;10001;3
VEF;15;14;10001;5

6.1.1 Resolução:

```
************************************************************************
*                                                                      *
*          ****************************************          *
*          *  Confidencial e Proprietário          *          *
*          *  Copyright 2001, Accenture             *          *
*          *  Todos os direitos reservados          *          *
*          ****************************************          *
*                                                                      *
*        Nome do Shell: Z_TEMPLATE_REPORT    Versão do Shell: v1.03  *
*                                                                      *
************************************************************************
* Nome do Programa   : Z_TEMPLATE_REPORT                               *
* Título do Programa : Z_EXC_BDC01_AL14_D                              *
* Programador        : Aluno do curso de ABAP                          *
* Data               : 02/04/2001                                      *
* Última atualização : 02/04/2001                                      *
*                                                                      *
* Descrição    : Este é um programa para importar dados de um arquivo  *
*              e criar ordens de venda no SAP, através da transação  *
*              VA01                                                    *
* Include      :                                                       *
*                                                                      *
* Calls        : VA01                                                  *
*                                                                      *
* Módulos de Funções:                                                  *
*                                                                      *
*----------------------------------------------------------------------*
* Lista de Modificações:                                               *
* Data       Autor       Corr. #      Descrição                        *
* 02/04/2001  ABAPDEV1     AFIK900013   Desenvolvimento inicial do prg *
*                                                                      *
************************************************************************
report Z_EXC_BDC01_AL14_D
  no standard page heading    "Não mostra o cabeçalho padrão SAP
  line-size 170               "Largura do report de 170 caracteres
  line-count 58               "Comprimento do report de 58 caracteres
  message-id zabap.                 "Classe de mensagens é ZABAP

*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
*               " SESSÃO DE DECLARAÇÃO DE DADOS "
*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
************************************************************************
* Includes                                                             *
************************************************************************

************************************************************************
* Tabelas                                                              *
************************************************************************

************************************************************************
* Dados:                                                               *
*   Constantes       (C_...)                                           *
*   Variaveis        (V_...)                                           *
*   Tabelas Internas  (I_...)                                          *
************************************************************************
*- Constantes ------------------------------------------------------*
*DATA:
*  C_..

*- Variáveis  ------------------------------------------------------*
*DATA:
*  v_..
```

```
** Variavel Com Data do Sistema Concatenada
data:   v_data(8) type n.

* atribui a data do sistema à variável no formato ddmmaaaa
concatenate sy-datum+6(2) sy-datum+4(2) sy-datum+(4) into v_data.
*- Tabela Interna ----------------------------------------------------*

** Tabela com unico campo para receber arquivo com delimitador ';'
DATA: BEGIN OF i_reg OCCURS 100,
     lreg(200) type c .
DATA: END OF I_reg.

** Tabela com campos provenientes do arquivo texto .

data: begin of i_parm OCCURS 100,
      auart  like vbak-auart ,
      kunnr  like vbak-kunnr ,
      kunnra like kuwev-kunnr ,
      mabnr  like rv45a-mabnr ,
      zmeng(16) type c.
data: end of i_parm.

* Tabela com os dados do BDC
data:   bdcdata like bdcdata    occurs 100 with header line.

************************************************************************
* Parâmetros de entrada :                                              *
*   Select Options (S_...)                                             *
*   Parameters     (P_...)                                             *
************************************************************************

selection-screen begin of block b1 with frame title t1.
  parameter: p_arq1 like rlgrap-filename . "** Nome do Arquivo
selection-screen end of block b1.

"<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
                          " SESSÃO DE PROCESSAMENTO "
"<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
*---------------------------------------------------------------------*
initialization.                  "Logica para defaults "inteligentes"
*---------------------------------------------------------------------*

*---------------------------------------------------------------------*
  t1 = 'Interface Ordem Vendas'.

*---------------------------------------------------------------------*
at selection-screen. "Tratamento de Erros e lógica para tela de seleção
*---------------------------------------------------------------------*
at selection-screen on value-request for p_arq1.
  perform f_selec_arqdos using p_arq1.

*---------------------------------------------------------------------*
start-of-selection.                "Processamento principal
*---------------------------------------------------------------------*
  perform f_select_data.
  perform f_gerabdc.

*---------------------------------------------------------------------*
end-of-selection.                  "Fim do processamento principal
*---------------------------------------------------------------------*

*---------------------------------------------------------------------*
```

```
top-of-page.        "Cabeçalho (Aparece sempre no topo da tela.)
*---------------------------------------------------------------------*
*---------------------------------------------------------------------*
end-of-page.                    "Rodapé
*---------------------------------------------------------------------*

"<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
                          " SESSÃO DE FORMS "
"<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"

*&---------------------------------------------------------------------*
*&      Form  F_SELECT_DATA
*&---------------------------------------------------------------------*
* Seleciona dados e armazena-os na tabela interna                     *
*---------------------------------------------------------------------*
form f_select_data.

* Upload do arquivo texto .

  CALL FUNCTION 'WS_UPLOAD'
      EXPORTING
          FILENAME                = p_arq1
          FILETYPE                = 'ASC'
      TABLES
          DATA_TAB                = i_reg
      EXCEPTIONS
          CONVERSION_ERROR    = 1
          FILE_OPEN_ERROR         = 2
          FILE_READ_ERROR         = 3
          INVALID_TYPE            = 4
          NO_BATCH                = 5
          UNKNOWN_ERROR       = 6
          INVALID_TABLE_WIDTH         = 7
          GUI_REFUSE_FILETRANSFER     = 8
          CUSTOMER_ERROR              = 9
          OTHERS                      = 10.


  IF SY-SUBRC <> 0.
    MESSAGE e005.
  ENDIF.

** Cria tabela interna com campos separados .
  loop at i_reg.

    split i_reg at ';' into i_parm-auart
                            i_parm-kunnr
                            i_parm-kunnra
                            i_parm-mabnr
                            i_parm-zmeng.

    append i_parm.

  endloop.

endform.                    " F_SELECT_DATA

*&---------------------------------------------------------------------*
*&      Form  F_geraBDC
*&---------------------------------------------------------------------*
* Gera Registros no BDC  *
*---------------------------------------------------------------------*
form f_gerabdc.
```

```
** Atualiza tabela DBCDATA com dados da tabela interna
  loop at i_parm .

    perform bdc_dynpro      using 'SAPMV45A' '0101'.
    perform bdc_field       using 'BDC_CURSOR'
                            'VBAK-AUART'.
    perform bdc_field       using 'BDC_OKCODE'
                            '=ENT2'.
    perform bdc_field       using 'VBAK-AUART'
                            i_parm-AUART.
    perform bdc_dynpro      using 'SAPMV45A' '4001'.
    perform bdc_field       using 'BDC_OKCODE'
                            'SICH'.
    perform bdc_field       using 'KUAGV-KUNNR'
                            i_parm-KUNNR.
    perform bdc_field       using 'KUWEV-KUNNR'
                            i_parm-KUNNRa.
    perform bdc_field       using 'RV45A-MABNR(01)'
                            i_parm-MABNR.
    perform bdc_field       using 'VBAP-ZMENG(01)'
                            i_parm-ZMENG.
    perform bdc_field       using 'VBKD-FKDAT'
                            v_data.

** Chama Funcao para gerar ordem de vendas

    call transaction c_transacao using bdcdata
            mode c_mode
            update c_update.

** Limpa dados da ordem gerada
    clear bdcdata.
    refresh bdcdata.

  endloop .

endform.                              " F_GERABDC

*---------------------------------------------------------------------*
*       Start new screen                                              *
*---------------------------------------------------------------------*
form bdc_dynpro using program dynpro.
  clear bdcdata.
  bdcdata-program  = program.
  bdcdata-dynpro   = dynpro.
  bdcdata-dynbegin = 'X'.
  append bdcdata.
endform.

*---------------------------------------------------------------------*
*       Insert field                                                  *
*---------------------------------------------------------------------*
form bdc_field using fnam fval.
  if fval <> ''.
    clear bdcdata.
    bdcdata-fnam = fnam.
    bdcdata-fval = fval.
    append bdcdata.
  endif.
endform.

*---------------------------------------------------------------------*
```

```
* FORM F_SELEC_ARQDOS                                          *
*---------------------------------------------------------------------*
* Permite seleção de arquivo no DOS.                           *
*---------------------------------------------------------------------*
form f_selec_arqdos using p_arq1 like rlgrap-filename.

* Captura o arquivo de entrada via DOS.
  call function 'WS_FILENAME_GET'
    exporting
      def_filename     = ' '
      def_path         = 'C:\'
      mask             = ',Textos,*.txt,doc,*.doc,Todos,*.*.'
    importing
      filename         = p_arq1
    exceptions
      inv_winsys       = 1
      no_batch         = 2
      selection_cancel = 3
      selection_error  = 4
      others           = 5
      .

endform.                              "Fim do 'Form f_selec_arqdos'
```

7 COMANDO SELECT

7.1 Comando SELECT

O comando select é usado para acessar e selecionar dados de tabelas internas do SAP. Por existirem diversas variações do mesmo comando, é fundamental que o programador saiba o mecanismo de funcionamento de cada uma delas pois só assim poderá dar ao programa uma performance satisfatória.

Variações:

7.2 SELECT * FROM dbtab.

```
…
ENDSELECT.
```

Seleciona dados de uma tabela SAP num processo de "loop" que começa no select e termina no endselect. A cada passagem pelo "loop" temos um elemento lido e selecionado.
É necessário que se coloque após o ENDESELCT uma condição de checagem de dados selecionados:

```
If sy-subrc ne 0.
  Write: 'Nenhum dado foi selecionado'.
Endif.
```

Se sy-subrc = 0 : pelo menos um dado foi selecionado

Se sy-subrc = 4 : nenhum dados foi lido

Exemplo:

```
SELECT * FROM BSEG.
…
ENDSELECT.
If sy-subrc ne 0.
  Write: 'Não tem dado na tabela BSEG'.
Endif.
```

7.3 Adições :

... WHERE Condition

Seleciona apenas os dados que satisfazem a condição especificada.
Exemplo:

```
SELECT * FROM T001 WHERE BUKRS EQ '0001'.
…
ENDSELECT.
```

… ORDER BY f'1…fn… ORDER BY PRIMARY KEY

Organiza os dados em ordem ascendente de acordo com os campos especificados (f1…fn).

Exemplo:

```
SELECT * FORM T001 ORDER BY MSGNR DESCENDING ARBGB ASCENDING.
…
ENDSELECT.
```

... UP TO n ROWS.

Seleciona um número máximo de dados.

Exemplo:

```
SELECT * UP TO 100 ROWS FROM T001 WHERE...
…
ENDSELECT.
```

7.4 SELECT * FROM dbtab INTO TABEL itab.

Os dados são selecionados e colocados na tabela interna itab de uma só vez. Não há mais o processo de loop e portanto não há mais ENDSELECT. Os dados novos da tabela interna são gravados por cima dos antigos.

É importante ressaltar que o * pode ser substituído pelos nomes dos campos da tabela, agilizando assim o processo e melhorando a performance.

Exemplo:

```
SELECT * FROM T001 INTO TABLE TAUX.
```

Adições:

a) WHERE
b) ORDER BY
c) UP TO n ROWS

7.5 SELECT * FROM dbtab APPENDING TABLE itab.

Mesmo processo do item 2, só que os dados novos são inseridos sem apagar os antigos.

Exemplo:

```
SELECT * FORM T100 APPENDING TABLE TAUX .
```

Adições:

a) WHERE
b) ORDER BY
c) UP TO n ROWS

7.6 SELECT SINGLE * FROM dbtab WHERE f1 = g1 AND… AND fn +

Seleciona apenas um único dado que satisfaças condições do where.

OBS.: Neste caso todas as chaves (índices da tabela) devem ser satisfeitos.

Exemplo:

```
SELECT SINGLE * FORM T100 WHERE BUKRS = ‘02’.
```

7.7 SELECT * FROM dbtab APPENDING CORRESPONDING FILEDS OF TABLE itab.

Mecanismo semelhante ao item 2, e deve ser usado quando a sintaxe do item 2 não puder ser usada.

Há diferenças de performance.

Exemplo:

```
SELECT * FORM T100 APPENDING CORRESPONDING FIELDS OF TABLE TAUX .
```

Onde TAUX recebeu a estrutura da tabela T100 (commando INCLUDE STRUCTURE).

7.8 SELECT * FROM dbtab FOR ALL ENTRIES in itab WHERE...

Usado quando selecionamos dados de uma tabela e precisamos de dados de outra tabela para compor as condições do where.

Exemplo:

```
SELECT * FORM BSEG FOR ALL ENTRIES IN T_BKPF
        WHERE BUKRS        =  T_BKPF-BUKRS  AND
                BELNR =  T_BKPF-BELNR .
```

Onde T_BKPF é uma tabela interna que recebeu a tabela BKPF.

Este tipo de comando é utilizado entre tabelas internas.

8 TIPOS DE SELECT EXISTENTES E MAIS UTILIZADOS

8.1 SELECT * FROM ... <tabela>

Quando não se impõe nenhum tipo de restrição, ocorre uma varredura seqüencial dos registros da tabela. Quando se utiliza grandes tabelas, isso obviamente afeta o runtime.

Performance: Select * seleciona todas as colunas de uma tabela. É melhor sempre especificar as colunas, pois em caso de tabelas com muitas colunas, prejudicará a performance.

8.2 SELECT * FROM <tabela> WHERE <campo> EQ <conteúdo>.

Lê todos os registros da tabela especificada onde o campo é igual ao conteúdo especificado.

Performance: Select * Where seleciona todas as colunas de uma tabela de acordo com a condição de where. É melhor sempre especificar as colunas, pois em caso de tabelas com muitas colunas, prejudicará a performance.

8.3 SELECT * FROM <tabela> WHERE <table field> BETWEEN <field1> AND <field2>.

Exemplo: field1 = 100 e field2 = 500. pega inclusive 100 e 500. Você trabalha com o range.

8.4 SELECT * FROM <tabela> WHERE <table field> LIKE … '_R$'.

_ = a primeira letra não imprta o que virá.
R a segunda deverá ser R (eu defini)
% não importa a seqüência de caracteres que virá

8.5 SELECT * FROM <tabela> WHERE <table field> IN (…… , …….).

Exemplo: select * from <table> where campo1 in (123,1000) – podem ser valores ou literais.
É como perguntar se campo1 é 123 ou 1000.

8.6 SELECT * FROM <tabela> WHERE <table field> IN <internal table>.

Exemplo:

```
DATA:  begin of ITAB occurs 10,
          Sign(1), options(2), low like sflight-price, high like sflight-proce,
       End of ITAB.
* Ranges: ITAB for sflight-table
Move:  'I' to itab-sign, 'bt' to itab-option, '500' to itab-low, '1000' to itab-high.
Append itab.
Move:  'I' to itab-sign, 'bt' to itab-option, '440' to itab-low.
Append itab.
```

8.7 SELECT * FROM <tabela> ORDER BY <field1> <field2> … PRIM ARY KEY.

Obs.: Classifica a tabela interna numa área auxiliar, sem afetar a tabela original. Evitar o uso de sorts dentro de um select. Consome mais tempo que descarregar os dados em uma tabela interna e classificá-los.

8.8 SELECT * FROM <tabela> BYPASSING BUFFER.

Usado para ler diretamente da tabela original e não do buffer.

OBS.: Select single * sempre com chave completa especificada. Particularmente do ABAP/4
Select * - procurar evitar. Informar as colunas que serão necessárias, apenas.
Uso do comando extract (insert header, details) – para relatórios.

8.9 SELECT * FROM <tabela> APPENDING TABLE <internal table>.

Lê os registros e os Inclui – não sobrepõe – em uma tabela interna.

8.10 SELECT * FROM <tabela> INTO TABLE <internal table>.

A estrutura da tabela interna deve corresponder à estrutura da tabela que está sendo acessada. O sistema lê os registros em conjunto, não individualmente através de um LOOP e ir gravando os registros, uma a um.

8.11 SELECT … INTO CORRESPONDING FIELDS OF TABLE <internal table>.

Neste caso a estrutura da tabela interna não precisa corresponder à estrutura da tabela que está sendo acessada. Movimentará os registros para as colunas definidas na tabela interna que possuam nome igual ao da tabela acessada.

Obs.: Corresponding ou appending não exisgem o endselect.

8.12 SELECT * APPENDING COREESPONDING FIELDS OF TABLE <internal table>.

Lê e grava (não sobrepõe) os dados em uma tabela interna que possua nomes idênticos aos nomes da tabela que está sendo lida.

8.13 SELECT SINGLE * FROM <tabela> WHERE <campo> EQ <conteúdo>.

Toda vez que se usa select single * a chave primária completa deve ser especificada. Se a chave especificada não é qualificada, você receberá uma mensagem de warning e a performance ficará prejudicada.

No caso de haver a necessidade de acessar um único registro via select, as opções são:

Select * ... seguido do comando EXIT

ou

select * ... UP TO 1 ROW.

Neste caso não é necessário especificar a chave completa.

8.14 SELECT <a1> <a2> ... INTO (<f1>, <f2>, …) FROM … <tabela> WHERE … .

Lê as colunas especificadas (a1, a2). Após INTO deverão ser especificadas as áreas de trabalho auxiliares (f1, f2). O número de colunas lidas deverá ser igual ao número de work-aeras especificadas.

8.15 SELECT MAX (campo)

MIN (campo)
AVG (campo)
COUNT (*) FROM <tabela> INTO (... , … , … , …)
WHERE … .

AVG e SUM somente para campos numéricos.
Não se usa endselect.
Mais rápido fazer uma rotina “a mão” que utilizar esse comando.

8.16 SELECT * FROM SFLIGHT WHERE PRICE IN ITAB.

8.17 SELECT * FROM (<tabela>) INTO <work area>.

Exemplo:

```
Data:   begin of WA,
          Line(1000,
        end of WA.
```

Parameters: tabname(10) default ‘SPFLI’.

*** especificando o nome da tabela em tempo dinamicamente no comando select sempre consome mais tempo de CPU que especificando estatiamente no programa.

```
Select * from  (tabname) into WA.
  Write: ….
Endselect.
```

8.18 SELECT * FROM <tabela> FOR ALL ENTRIES IN <tabela interna>

WHERE campo1 = conteúdo AND
WHERE campo2 = conteúdo .

Defino uma tabela interna. Alimento os campos desta tabela interna. (move e append).
No meu select campo1 e campo2 serão os campos definidos e ealimentados na tabela interna.

8.19 SELECT carrid MIN(price) MAX(price) INTO (carid, minimum, maximum)

FROM sflight
GROUP BY carrid.

Todos os campos que eu quero que apareçam na minha lista eu preciso especificar após a cláusula GROUP BY.

(carrid, maximum e minimum são campos auxiliares.
Se o nome do databease não é conhecido até o runtime, não se pode especificar a cláusula GROUP BY.

9 COMANDOS MAIS UTILIZADOS EM ABAP

Write: para escrever.
Uline: aciona um linha horizontal.
Skip: saltar linha.
Under: em baixo "a baixo"
Occurs: controle de registro na tabela. Delimita as linhas da tabela, para melhorar a performance do programa.
E000: esta expressão é usada para mensagem de erro.
S000: esta expressão é usada para mensagem de rodapé.
I000: esta expressão é usada para mensagem de informação.
W000: esta expressão é usada para mensagem de aviso, congela a imagem, aguarda retorno do usuário.
Vline: vertical " insere uma linha vertical".
" | ": igual ao V_line.
Clear V_Var1: "Limpa" zera uma variavel.
[ ]: compara o conteúdo total da tabela.
Select: Selecionar. " Varre o banco de dados"
New-Page: abrir um nova pagina.
Standard Page Heading: desabilita cabeçalho padrão.
Top - Of – Page: para desenvolver um novo cabeçalho.
End-of-page: Encerra a página "Rodapé".
Format Color 1: Insere uma nova cor que é selecionada pelo numero.
Format Intensified off: desabilita o formato padrão da letra.
Format Intensifiel on: ativa o formato padrão.
Data: Dados.
Like: como.
Move: mover.
Move Correspond: Move apenas os dados que correspondam a seleção.
Is Initial: comparação do vazio.... testa se a tabela interna esta vazia...no IF
Refresh: Limpa todo o conteúdo da tabela interna.
Clear: apaga o cabeçalho (limpa) de uma tabela interna.. Ex: clear i_tapp .
Parameter: "são os texts boxes" --> entrada de dados.
Default: Padrão.
At First: Primeiro registro da tabela.
At New: Primeiro registro da quebrar.
At Last: Ultimo registro da quebrar.
At And Of: Ultimo registro da quebra.
Order: Ordenar.
Append: Gravar um registro após a execução " Select * From". " Nunca deixando-o subescrever".
**** Gravar o cabeçabelho de uma tabela interna ***
ON CHANGE:
Describe: mostra o total do registro na tabela.
Condense: Comando que serve para juntar as palavras de forma ordenada. Ex: Impressão.
Using: usar.
OkCode: Como se fosse Enter - "Isto um codigo interno do abap".
Ws – Download: Envia "Arquivo".
Upload: Importa "Arquivo".
Loop: comando execução. " Varre o banco de dados" é utilizado para varre a tabela interna.
Exceptions:
Headlen:
Trunclen:
At selection screen:
At line selection: Comando utilizado no relatório, quando uma linha é selecionada.
Quando você cria o at line selection ele cria o botão com o nome de PICK
At user command: Controla os botões que forem adicionados ao código. Novo
set pf-stautus:
Form: Subrotinas ...
Perform: Com o comando perform executamos uma rotina que esta fora do fluxo normal do programa.
* É um comando que faz um acesso na tabela e retorna.
Call Transaction: possibilita o processamento de uma tabela BDC,
* Bach Input --> método de entrada de dados.
Initialization: "Antes da seleção. Novo

at selection-screen: ”Durante a tela Novo
start-of-selection: Evento que finaliza a inicialization ou chama outro inicialization.Começo do código ABAP, onde você começa a seleção de dados.
end-of-selection: Opcional. encerrado quando vc chama outra tela ou mensagem. Novo
top-of-page: ”Cabeçalho do relatório. Novo
At line-selection: "processos após seleção de linha. (cria o botão PICK) Novo

10 DATA DICTIONARY

10.1 Objetivos

- ✓ Apresentar conceitos de Bancos de Dados Relacionais
- ✓ SAP vs Modelo Relacional
- ✓ Conhecer as ferramentas básicas do Dicionário de Dados do R/3
- ✓ Criar objetivos através do Dicionário de Dados

10.2 Conceitos de Bancos de Dados Relacionais

10.3 Modelo Entidade-Relacionamento

Modelo desenvolvido para facilitar o projeto de banco de dados, permitindo a especificação de um esquema que represente a estrutura lógica global de um banco de dados.

- ✓ **Entidade**: É um objeto que existe e é distinguível de outros objetos, ou seja, identifica o agrupamento de objetos do mesmo tipo. Ex.: Clientes, Bancos, Agências, Contas-corrente.
- ✓ **Atributos**: São os qualificadores de uma entidade, isto é, representam no modelo o que uma entidade pretende ser. Ex.: Nome, RG, CPF, Endereço, Nro Conta, Nro Agência, Nro Banco.
- ✓ **Domínio**: Conjunto de valores permissíveis para um atributo. Ex.: Estado Civil, Seco, Cor, Meses do Ano.
- ✓ **Relacionameto**: É a associação entre duas entidades, ou seja, representa a maneira como duas entidades são relacionadas ou ligadas. Ex.: Conta-corrente de um Cliente, Agência de um Banco, Contas-corrente de uma Agência.

10.4 Restrições de Mapeamento

Representam o modo como as diferentes entidades de um modelo se relacionam. Determinadas pela cardinalidade dos relacionamentos entre as entidades.

- ✓ **Um-para-Um**: Uma ocorrência da Entidade A está relacionada com uma e apenas uma ocorrência da Entidade B.
- ✓ **Um-para-N**: Uma ocorrência da Entidade A está relacionada com uma ou várias ocorrências da Entidade B.
- ✓ **N-para-Um**: Várias ocorrências da Entidade A estão relacionadas com apenas uma ocorrência da Entidade B.
- ✓ **N-para-N**: Várias ocorrências da Entidade A está relacionada com várias ocorrências da Entidade B.

10.5 Modelo Relacional

Um banco de dados relacional é a implementação física do Modelo Entidade-Relacionamento e traduz concretamente o que o modelo conceitual procura representar. Consiste em uma coleção de tabelas cada uma das quais associada a um nome único e que possuem relacionamentos entre si. Tabelas representam fisicamente as Entidades.

Cada tabela possui uma estrutura similar àquilo que pretende a representar, isto é, tabelas são formadas de linhas que por sua vez são formadas por colunas. Colunas representam fisicamente os Atributos.

A cada linha da tabela chamamos de Ocorrência e o conjunto de ocorrências pode ou não estar relacionado com ocorrências de outras tabelas.

10.6 Como distinguir as ocorrências umas das outras?

Utilizando o conceito de Chave Primária!!!

Chave Primária: Conjunto de atributos que garante a unicidade de cada ocorrência da tabela. Exemplos: RG, CPF, Nro Chassis.

- ✓ Normalização: processo de reconhecimento da chave primária.

10.7 Como representar os relacionamentos entre tabelas?

Transferindo a chave primária de uma tabela para outra!!!

Chave Estrangeira: Quando a chave primária de uma tabela é um atributo em outra(s) tabela(s).

10.8 Como definir todos esses elementos num BD?

Utilizando linguagens especiais para cada Sistema Gerenciador de BD!!!

- ✓ **SQL**: Structured Query Language – Linguagem desenvolvida nos anos 70 para definição e manipulação de dados em sistemas de bancos de dados relacionais. Conjunto reduzido de comandos, sem recursos de lógica, única e exclusivamente para criar objetos no banco de dados e permitir que os dados possam ser mantidos, de acordo com as necessidades funcionais. Nos gerenciadores de BD, ferramentas específicas de cada fabricante oferecem recursos adicionais para a construção de lógica. Outras linguagens foram adaptadas para trabalharem em conjunto com o SQL, como o COBOL, C e ABAP/4. O SQL utilizado no R/3 é proprietário e segue um mínimo da regulamentação internacional para esta, estando longe de possuir os mesmos mecanismos de funcionamento. Seus comandos podem ser divididos em:
- ✓ **DDL**: Data Definition language – Comandos específicos para definição de objetos do banco de dados. Ex.: Create Table, Create View, Create TableSpace, Drop Table, etc.
- ✓ **DML**: Data Manipulation Language – Comandos específicos para tratamento dos dados armazenados nos objetos do banco. Ex.: Select, Insert, Delete, Update, etc.

11 R/3 DATA DICTIONARY

11.1 Introdução

O dicionário de dados do ABAP/4 (DD) é uma fonte central de informações provenientes do sistema gerenciador de dados do SAP. Sua principal função é suportar a criação e o gerenciamento das definições de dados, também conhecidas como “metadados”.

Neste curso, o DD será utilizado em exercícios práticos com o seguinte objetivo:

- ✓ Visualizar objetos do SAP (tabelas, estruturas, elementos de dados, domínios)
- ✓ Observar a estrutura e atributos dos dados armazenados na base de dados
- ✓ Elaborar consultas (queries simples) de dados
- ✓ Descobrir os relacionamentos entre diferentes objetos
- ✓ Criar objetos

O Data Dictionary é parte integrante do ABAP/4 Workbench e está numa camada intermediária entre o sistema aplicativo e o gerenciador de banco de dados.

11.2 Funções desempenhadas pelo Data Dictionary

11.2.1 Gerenciamento das Definições de Dados

Criação e manutenção das definições de dados num repositório central

11.2.2 Provisão de informações para avaliações

Permite obter informações sobre o modo como os objetos estão relacionados

11.2.3 Suporte ao desenvolvimento

Diferentemente de outros dicionários de dados, está integrado ao ambiente de desenvolvimento de modo que alterações ou criação de novos objetos promovem automaticamente a geração dos outros objetos dependentes, sejam eles parte do dicionário ou programas aplicativos.

11.2.4 Suporte à documentação

Permite obter documentação atualizada

11.2.5 Garantia de que as definições de dados sejam flexíveis e atualizadas

Geração de objetos de runtime garantindo performance

11.3 Elementos de Dados

Requisitos do SAP R/3

Os nomes dos Elementos de Dados devem iniciar com Z ou Y, podem ter um máximo de 10 caracteres de comprimento e devem ser únicos na instância da base de dados.

Padrão (standard)

Os Elementos de Dados precisam ser ativados pelo Administrador do Dicionário de Dados para melhor controle. Quando possível, use o mesmo nome como domínio associado a esse Elemento de Dados. Se houver um conflito, o nome do Elemento de Dados é o principal motivo. A SAP não cria Elemento de Dados que começam com um Z, como hábito, Elementos de Dados poderiam começar com ZZ.

Os nomes de Data Elementos poderiam ter um máximo de 8 caracteres. A SAP gerou programas ou funções que podem utilizar essas definições de objetos para SELECT-OPTIONS e PARAMETERS, o qual podem ter no máximo 8 caracteres de comprimento.

Exemplo:	ZZOBJTNM	
	ZZ	Sempre ZZ
	OBJTNM	Um nome significativo que descreva o elemento de dados.

11.4 Domínios

Requisitos do SAP R/3

Os nomes dos Domínios devem iniciar com Z ou Y, podem ter um máximo de 10 caracteres de comprimento e devem ser únicos na instância da base de dados.

Padrão (standard)

Os Domínios precisam ser ativados pelo Administrador do Dicionário de Dados para melhor controle. Quando possível, use um Domínio existente. Se não for possível, nomeie o Domínio com um nome significativo. A SAP não cria Domínios que começam com um Z, como hábito, Domínios poderiam começar com ZZ.

Os nomes de Domínios poderiam ter um máximo de 8 caracteres. A SAP gerou programas ou funções que podem utilizar essas definições de objetos para SELECT-OPTIONS e PARAMETERS e o tamanho permitido para as variáveis é de 8 caracteres de comprimento.

Exemplo:	ZZOBJTNM	
	ZZ	Sempre ZZ
	OBJTNM	Um nome significativo que descreva o elemento de dados.

11.5 Objeto de Bloqueio

Requisitos do SAP R/3

Quando criar um objeto de bloqueio, o nome do arquivo deve começar com EY ou EZ e pode ter um máximo de 10 caracteres de comprimento.

Formato Padrão (standard)

EZ_ZA100

EZ	Sempre EZ ou YZ
_ZA1000	O nome da tabela primária entrada no primeiro campo da tabela da tela de criação de Objeto de Bloqueio

11.6 Macth Codes ID

Códigos de Match Codes e Objetos devem ser verificados e ativados pelo Administrador do Dicionário de Dados.

Requisitos do SAP R/3

É hábito que os códigos de Match Codes tenham 1 caracter e não têm convenção de nomes.

Formato Padrão (standard)

Atualmente, a SAP não usa match codes 0-9 para seus objetos match code. Ultimamente códigos de Match Codes definidos para objetos SAP deveriam restringir eles mesmos de 0-9. Códigos de Match Codes que são definidos para novos objetos podem ser qualquer caracter alfanumérico (0-Z).

Exemplo: 1
Match Code 1

11.7 Objetos Match Code

Requisitos do SAP R/3

É hábito que os objetos Match Codes devam começar com um Z ou Y e podem ter 4 caracteres.

Formato Padrão (standard)

Use o Objeto SAP Match Code quando possível, caso contrário crie um objeto de acordo com suas necessidades. Iniciando o Objeto de Match Code com um Z o diferencia de outros objetos e então incluindo uma descrição segura que o Match Code será único e não interferirá com um Match Code existente.

Exemplo: Z001
Match Code para Centro de Custo

Posição	Descrição	Valores	Significado
1	Tipo	Z	Desenvolvimento
2	Funcional	A	Asset Management
		B	Basis
		F	Financial Accounting
		G	Special Leadger
		H	Human Resources Planning
		I	Maintenance
		K	Cost Accounting
		L	Warehouse Management
		M	Materials Management
		P	Production Planning
		R	Payroll
		S	Sales and Distribution
		U	General System / Utility programs
3-4	Identificador Único	00-ZZ	Única identificação de objeto Match Code

11.8 Pools / Clusters

Requisitos do SAP R/3

Um nome Pool ou Cluster deve começar com Z ou Y e pode ter um máximo de 10 caracteres de comprimento.

Formato Padrão (standard)

O nome padrão garante que o pool / cluster será único e não interferirá com nenhum pool / cluester definido pela SAP.

Exemplo: ZKIT
Pool customizado para o objeto matchcode ZKIT

11.9 Tabelas Transparentes e Cluster

Requisitos do SAP R/3

Precisam iniciar com Z, Y ou T9 e podem ter um máximo de 10 caracteres de comprimento.

Formato Padrão (standard)

Cada tabela customizada poderia começar com um Z seguido pelo código da aplicação para ela da principal finalidade.

Nomes de tabela não deveria exceder 7 caracteres. Se todos os 10 caracteres são usados, programas SAP gerados terão dificuldades devido a geração de campos e índices. Necessitando usar da transação SM31 só podem ser utilizados 5 caracteres de comprimento.

Exemplo: ZF100
Tabela customizada que pode ser visualizada on-line.

Posição	Descrição	Valores	Significado
1	Tipo	Z	Desenvolvimento
2	Funcional	A	Asset Management
		B	Basis
		F	Financial Accounting
		G	Special Leadger
		H	Human Resources Planning
		I	Maintenance
		K	Cost Accounting
		L	Warehouse Management
		M	Materials Management
		P	Production Planning
		R	Payroll
		S	Sales and Distribution
		U	General System / Utility programs
3-10	Identificador Único		Descrição alfanumérica

11.10 Nomes de Tabelas - ATAB

Requisitos do SAP R/3

Precisam iniciar com Z, Y ou T9 e podem ter um máximo de 10 caracteres de comprimento.

Formato Padrão (standard)

Cada tabela customizada para ATAB poderia começar com um T9 seguido pelo identificador único .
Tabelas que necessitam usar a transação SM31 são limitadas num máximo de 5 caracteres de comprimento.

Exemplo: Z9100
Tabela customizada definida para ATAB.

11.11 Campos de Tabelas

Requisitos do SAP R/3

Campos de usuários podem ter até 10 caracteres de comprimento e devem ser únicos na definição da tabela.

Formato Padrão (standard)

Quando possível, use o mesmo nome do elemento de dados associado com este campo. Se diversos campos na tabela usam o mesmo elemento de dados, nomeie estes campos de maneira significativa.
Quando possível, use um elemento de dados existente do SAP.
Não use espaços e caracteres especiais no nome do campo da tabela.

Exemplo: X(10) LOANNUM
Este poderia marcar um campo de número de empréstimo.

11.12 Tabela de Índice

Formato Padrão (standard)

Começa com um Z e tem um máximo de 3 caracteres de comprimento.
Exemplo: Z01

11.13 Tabela de Grupo

Formato Padrão (standard)

Uma tabela de grupo deve começar com um Z e pode ter um máximo de 8 caracteres de comprimento.

11.14 Grupo de Tipo

Requisitos do SAP R/3

Type definido pelo usuário pode ter até 5 caracteres de comprimento e deve começar com um Z.

Formato Padrão (standard)

Type Pools são definidos com a transação SE11, começando com um Z e um máximo de 5 caracteres. TYPE-POOL poderia ser associado com uma área funcional em particular.

Exemplo: ZF001
Tipo definido para TYPEs financeiros

Posição	Descrição	Valores	Significado
1	Tipo	Z	Desenvolvimento
2	Funcional	A	Asset Management
		B	Basis
		F	Financial Accounting
		G	Special Leadger
		H	Human Resources Planning
		I	Maintenance
		K	Cost Accounting

		L	Warehouse Management
		M	Materials Management
		P	Production Planning
		R	Payroll
		S	Sales and Distribution
		U	General System / Utility programs
3-4	Identificador Único	0-Z	Único Identificador

11.15 Estrutura

Requisitos do SAP R/3

Um nome de estrutura pode ter até 10 caracteres e deve iniciar com Z ou Y.

Formato Padrão (standard)

Exemplo: ZEADR
Estrutura customizada para endereço expandido do escritório principal

11.16 Views

Requisitos do SAP R/3

Uma view deve começar com Z ou Y e pode ter no máximo 10 caracteres.

Formato Padrão (standard)

Views customizadas poderia começar com Z seguido pelo tipo da view e um separador. O restante do campo poderia ser usado para identificar o tabela primária da view.

12 ALV

12.1 Introdução

Desenvolver relatórios em ABAP (Advanced Business Application Programing) com um bom visual e recursos avançados não é nada trivial.

Imagine desenvolver um relatório com cores, cabeçalho, linha de totais, label de colunas e separadores de colunas. Para piorar um pouco, que permita classificar por qualquer campo, aumentar ou diminuir o tamanho de colunas, gostaria também de poder trocar a posição das colunas, omitir ou exibir campos, totalizar, agrupar, exportar para Excel, etc.

Totalmente possível e igualmente inviável sem o uso de funções ALV.

O ALV padroniza e simplifica a exibição e operação de listas e relatórios no sistema R/3. Fornece interfaces e formatos padronizados para todas as listas e relatórios.

Na apostila vamos ver como criar programas utilizando uma função ALV. Parece pouco, mas todas trabalham de maneira similar.

12.2 Relatórios tradicionais

Doc.fat.	Grupo clientes	Valor líquido	Mont.imposto
0090000193	03	1.074,60	5,40
0090000194	02	468,64	2,36
0090000195	02	286,56	1,44
0090000196	03	468,64	2,36
0090000197	03	429,84	2,16
0090000198	03	1.312,21	6,59
0090000199	03	1.074,60	5,40
0090000200	03	1.312,21	6,59
0090000201	03	1.074,60	5,40
0090000202	03	749,83	3,77
0090000203	03	716,40	3,60
0090000204	02	749,83	3,77
0090000205	03	2.999,33	15,07
0090000206	03	2.507,40	12,60
0090000207	03	21.436,72	1.128,25
0090000208	03	4.892,50	257,50
0090000209	03	5.974,55	314,45
0090000210	02	4.825,40	253,97
0090000211	02	1.923,92	101,26
0090000212	03	3.875,29	9,71

Um relatório tradicional em ABAP não tem nenhum recurso ou formatação padrão. Tudo deve ser programado via código.

Um programa para listar um relatório como no exemplo acima, já exige muita codificação. Veja que não tem nada de complexo.

Depois de pronto, uma simples alteração no posicionamento dos campos ou no tamanho do papel, já demanda um novo processo de modificação. O usuário não tem os recursos necessários para resolver o problema.

Qualquer ação no relatório apresentado, diferente das opções do menu standard do R/3 para esse tipo de listagem, deverá ser programado. Uma tarefa nada simples, visto que o programador deverá fazer o processo de ída e volta, ou seja, o relatório deve ficar dinâmico ao ponto do usuário voltar na situação inicial, após alguma modificação.

12.3 Relatórios ALV

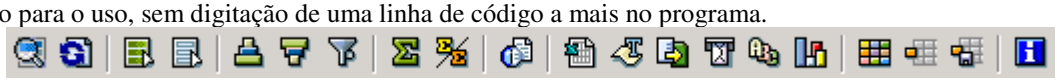
Listagens ALV são dinâmicas por definição. O programador vai escolher qual ou quais recursos irá disponibilizar em seu relatório.

Doc.fat.	GrCl	Val.líq.	Mont.imposto
90000193	03	1.074,60	5,40
90000194	02	468,64	2,36
90000195	02	286,56	1,44
90000196	03	468,64	2,36
90000197	03	429,84	2,16
90000198	03	1.312,21	6,59
90000199	03	1.074,60	5,40
90000200	03	1.312,21	6,59
90000201	03	1.074,60	5,40
90000202	03	749,83	3,77
90000203	03	716,40	3,60
90000204	02	749,83	3,77
90000205	03	2.999,33	15,07
90000206	03	2.507,40	12,60
90000207	03	21.436,72	1.128,25
90000208	03	4.892,50	257,50
90000209	03	5.974,55	314,45
90000210	02	4.825,40	253,97
90000211	02	1.923,92	101,26
90000212	03	3.875,29	9,71

Muito similar a uma planilha do Microsoft Excel, cada coluna é perfeitamente ajustável, podem ser trocadas entre si, as linhas da grade e cores são automáticas. Recursos simples que já eliminam um grande esforço de programação, principalmente em alterações.

12.4 Barra de ferramentas

Todos os demais recursos estão concentrados em uma barra de ferramentas que a função disponiliza junto a barra standard do R/3 (Na parte superior do relatório). Alguns ícones são bem comuns e com funções simples, mas está tudo pronto para o uso, sem digitação de uma linha de código a mais no programa.



Primeiramente, vamos apenas ver uma breve descrição de suas funções:

Detalhes (Ctrl+Shift+F3)	Selecione uma linha e clique nesse botão. A linha será destacada em forma de coluna.
Renovar (F8)	Reapresenta do relatório.
Marcar tudo (F5)	Marca todas as linhas.
Desmarcar tudo (F6)	Desmarca todas as linhas.
Ordenação crescente (Ctrl+F4)	Selecionando uma coluna e clicando nesse botão, todo o relatório ficará classificado na ordem crescente por essa coluna.
Ordenação decrescente (Ctrl+Shift+F4)	Idem ao anterior, mas a classificação é na ordem decrescente.
Definir filtro (Ctrl+F5)	Você poderá filtrar o seu relatório baseando-se em valores de campos.
Total (Ctrl+F6)	Totaliza a coluna selecionada. A coluna deve conter um valor. Colunas de caracteres não podem ser totalizadas.
Subtotais... (Ctrl+Shift+F6)	Apresenta subtotais de um total geral, para a coluna selecionada. Alguma coluna já deve estar totalizada, senão não haverá mudança.
Pré-visualiz.impressão (Ctrl+Shift+F10)	Mostra como irá ficar a impressão do relatório.
Microsoft Excel (Ctrl+Shift+F7)	Exporta o relatório para o Microsoft Excel.
Processamento de texto... (Ctrl+Shift+F8)	Exporta o relatório para o Microsoft Word.
File local... (Ctrl+Shift+F9)	Grava o relatório em arquivo.

Destinatário de correio eletrônico (Ctrl+F7)	Envia o relatório via e-mail através do SAP Office.
Análise ABC (Ctrl+F1)	Ranking em curva ABC. É obrigatório selecionar uma coluna de valores. O R/3 solicita algumas informações e já apresenta o resultado
Gráfico (Ctrl+Shift+F11)	Mostra o resultado em um gráfico.
Modificar layout... (Ctrl+F8)	Permite alterar o modo de apresentação do relatório.
Selecionar layout... (Ctrl+F9)	Recupera algum layout alterado e o aplica no relatório.
Gravar layout... (Ctrl+F10)	Permite gravar um layout alterado.
Informação (Ctrl+F12)	Informações como número de registros retornados, filtros sendo utilizados, campos sumarizados, etc. É apresentado por esse botão.

12.5 Opções standard

Essa barra de ferramentas pode ser configurada, para que o programador tenha possibilidade de criar ou retirar botões. Para tanto basta copiar, por exemplo, o Status-GUI (Que é a barra de ferramentas) STANDARD_FULLSCREEN do grupo de função SLVC_FULLSCREEN para o seu programa e alterar a vontade.

Não altere os códigos de retorno (ok-code) dos botões que você não irá modificar, deixe como está, pois é através desses códigos que a função ALV sabe o que o usuário escolheu.

Utilize para cópia a transação SE80, veja na figura abaixo os códigos de retorno standard:

Nem todas as funções utilizam o mesmo Status-GUI. A figura está bem completa. Você pode até criar sua própria barra de ferramentas, mas qualquer código diferente dos apresentados acima deverão ser tratados no programa. Veremos como tratar uma opção criada/alterada no exemplo de programa em anexo.

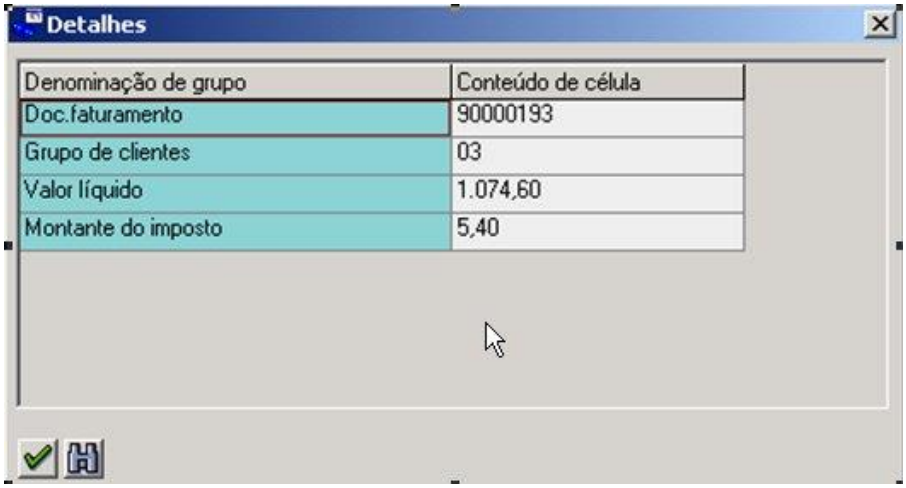
O funcionamento para alguns dos recursos da barra de ferramentas é o que veremos a seguir. Telas, procedimentos e saídas para alguns dos botões.

12.6 O botão de Detalhes

Primeiro selecione a linha desejada através do seletor de campos, e depois clique no ícone.

Doc.fat.	GrCl	Val.líq.	Mont.imposto
90000193	03	1.074,60	5,40
90000194	02	468,64	2,36
90000195	02	286,56	1,44
90000196	03	468,64	2,36
90000197	03	429,84	2,16

Veja na figura abaixo que a linha foi destacada em coluna. Isso é muito utilizado quando é permitido a edição dos campos no relatório.



12.7 Os botões de Total e Subtotal

Selecione a coluna com valor numérico, e escolha Total. Para subtotal, escolha uma coluna qualquer.

Doc.fat.	GrCl	Val.líq.	Mont.imposto
90000204	02	749,83	3,77
90000210		4.825,40	253,97
90000211		1.923,92	101,26
	02	**8.254,35**	**362,80**
90000193	03	1.074,60	5,40
90000196		468,64	2,36
90000197		429,84	2,16
90000198		1.312,21	6,59
90000199		1.074,60	5,40
90000200		1.312,21	6,59
90000201		1.074,60	5,40
90000202		749,83	3,77
90000203		716,40	3,60
90000205		2.999,33	15,07
90000206		2.507,40	12,60
90000207		21.436,72	1.128,25
90000208		4.892,50	257,50
90000209		5.974,55	314,45
90000212		3.875,29	9,71
	03	**49.898,72**	**1.778,85**
		58.153,07	**2.141,65**

Sempre primeiro é necessário totalizar uma coluna com o botão de somatória e só então definir os subtotais.

No exemplo, o relatório possui os totais dos campos “Valor Líquido” e “Montante do Imposto”. Foi escolhido o campo “Grupo de cliente” para que o relatório apresentasse os subtotais.

Por default, o relatório fica na ordem crescente do campo que foi escolhido como subtotal. É possível totalizar e subtotallizar vários campos.

Clicando nos ícones em destaque, você pode apresentar apenas as linhas de totais, as linhas de subtotais, apenas o total geral ou ainda mostrar as linhas de detalhe de algum “Grupo de cliente” e de outro não.

12.8 O botão Pré-visualz.impressão

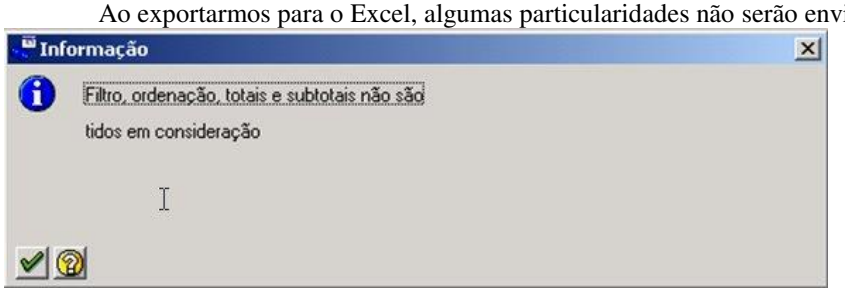
O Relatório ALV não é impresso como se vê na tela do computador. Clique nesse botão para visualizar a impressão no modo que irá ser a saída realmente.

Doc.fat.	GrCl	Val.líq.	Mont.imposto
90000193	03	1.074,60	5,40
90000194	02	468,64	2,36
90000195	02	286,56	1,44
90000196	03	468,64	2,36
90000197	03	429,84	2,16
90000198	03	1.312,21	6,59
90000199	03	1.074,60	5,40
90000200	03	1.312,21	6,59
90000201	03	1.074,60	5,40
90000202	03	749,83	3,77
90000203	03	716,40	3,60
90000204	02	749,83	3,77
90000205	03	2.999,33	15,07
90000206	03	2.507,40	12,60
90000207	03	21.436,72	1.128,25
90000208	03	4.892,50	257,50
90000209	03	5.974,55	314,45
90000210	02	4.825,40	253,97
90000211	02	1.923,92	101,26
90000212	03	3.875,29	9,71

No exemplo acima não temos os totais ou subtotais. Mas essas linhas são mantidas. O que perdemos é apenas o formato de grid (Microsoft Excel). Passamos a ver um relatório tradicional.

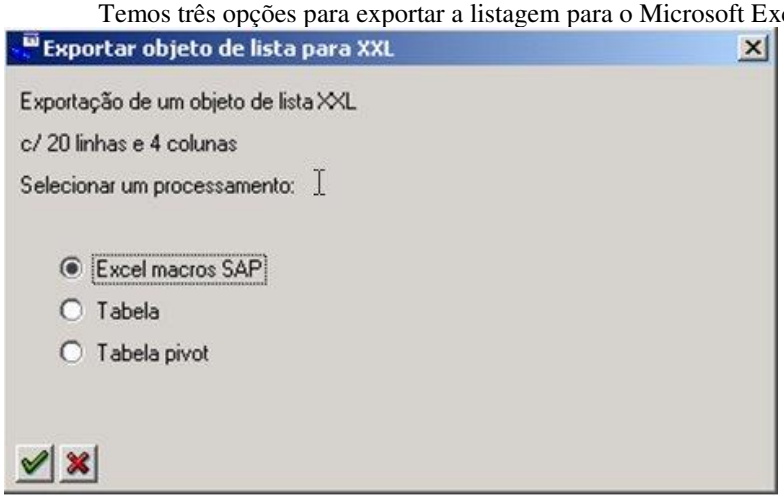
12.9 O botão Microsoft Excel

Ao exportarmos para o Excel, algumas particularidades não serão enviadas:



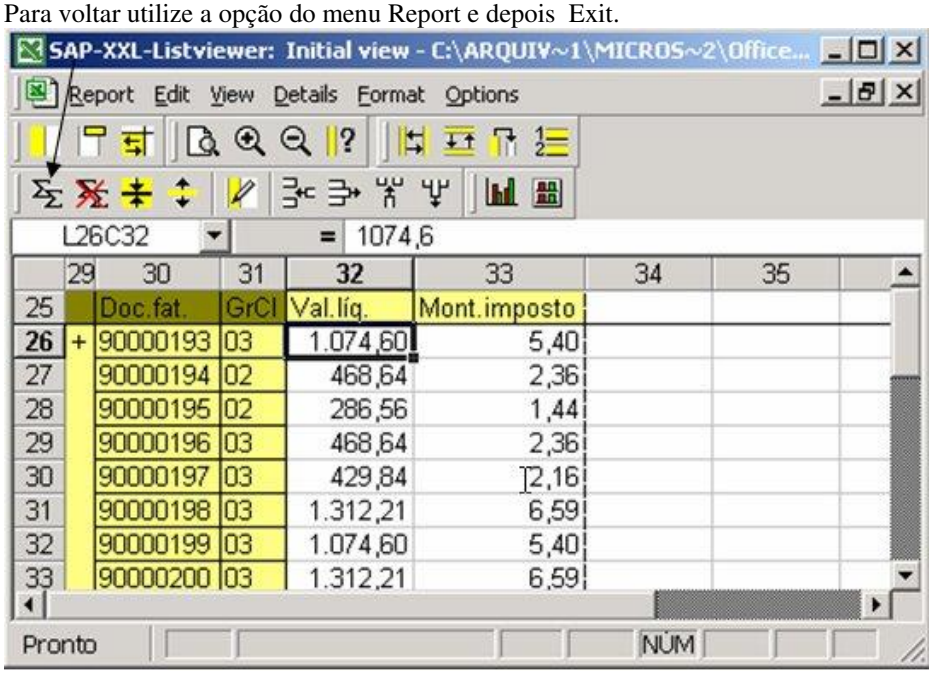
Assim, se a intensão e gerar o relatório para envia-lo ao Excel, não perca tempo com formatações. Apenas o básico é transferido.

Temos três opções para exportar a listagem para o Microsoft Excel, vamos ver a saída de cada uma delas.



12.10 O botão Microsoft Excel – Opção Excel Macros SAP

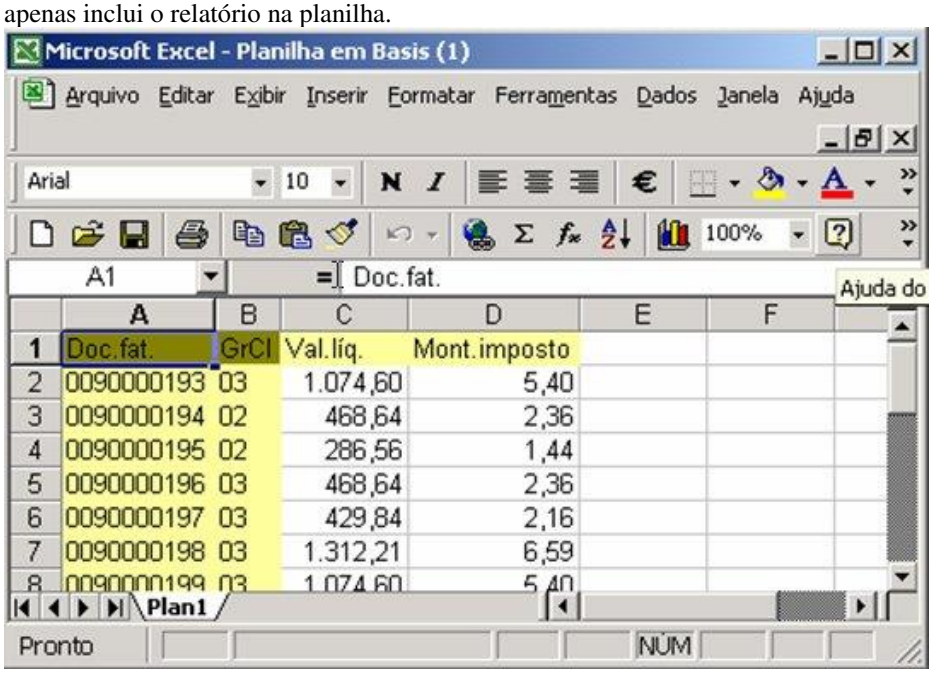
Com essa opção, o R/3 envia o relatório para o Excel e já apresenta várias barras de ferramentas próprias para que você tenha as mesmas facilidades. Repare na figura, que as opções de ferramentas do próprio Excel não aparecem. Para voltar utilize a opção do menu Report e depois Exit.



	29	30	31	32	33	34	35
25		Doc.fat.	GrCl	Val.líq.	Mont.imposto		
26	+	90000193	03	1.074,60	5,40		
27		90000194	02	468,64	2,36		
28		90000195	02	286,56	1,44		
29		90000196	03	468,64	2,36		
30		90000197	03	429,84	2,16		
31		90000198	03	1.312,21	6,59		
32		90000199	03	1.074,60	5,40		
33		90000200	03	1.312,21	6,59		

12.11 O botão Microsoft Excel – Opção Tabela

É a forma mais simples de exportação. Veja que o R/3 mantem as barras de ferramentas do Excel e apenas inclui o relatório na planilha.

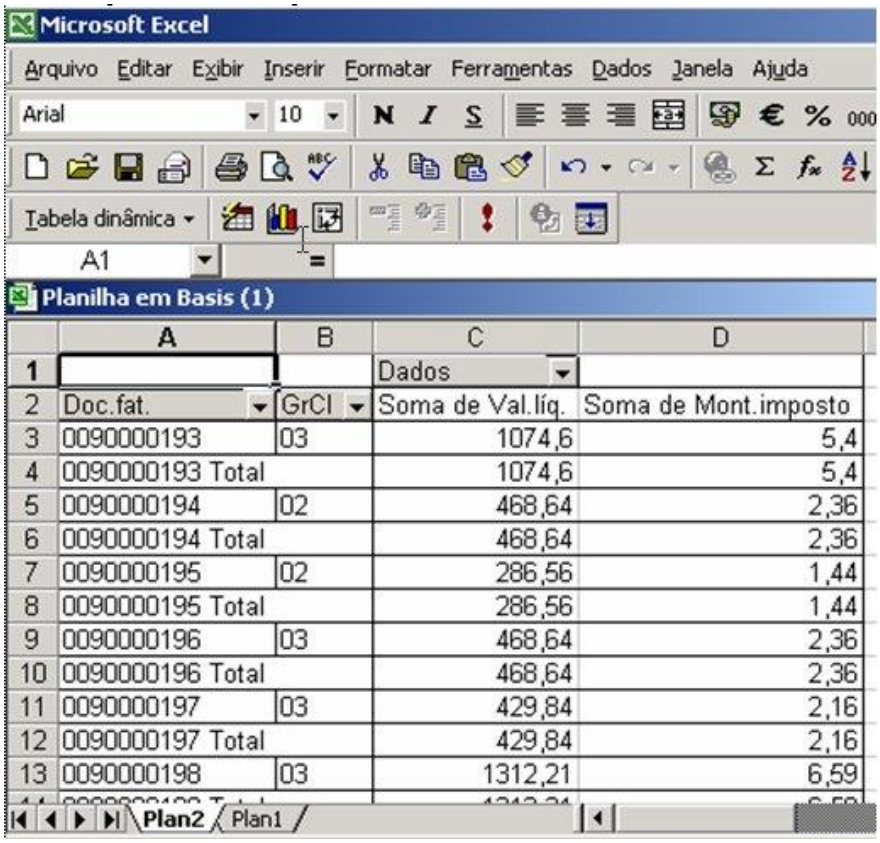


	A	B	C	D	E	F
1	Doc.fat.	GrCl	Val.líq.	Mont.imposto		
2	0090000193	03	1.074,60	5,40		
3	0090000194	02	468,64	2,36		
4	0090000195	02	286,56	1,44		
5	0090000196	03	468,64	2,36		
6	0090000197	03	429,84	2,16		
7	0090000198	03	1.312,21	6,59		
8	0090000199	03	1.074,60	5,40		

12.12 O botão Microsoft Excel – Opção Tabela Pivot

Nessa opção o R/3 abre o Excel e monta o relatório que permanece com algumas funções.

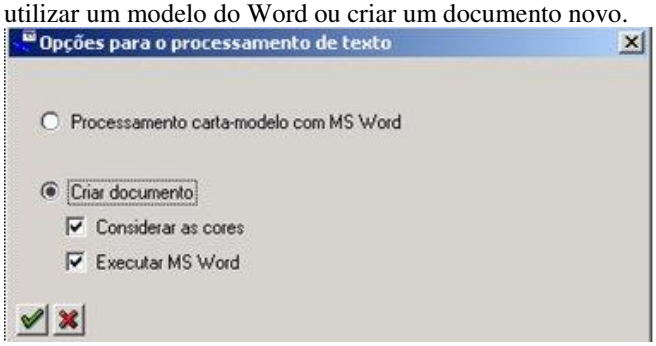
Mesmo que o seu Excel não esteja exibindo a barra de ferramentas “Tabela Dinâmica”, a comunicação R/3 – Excel vai passar a exibi-la, para o acesso aos recursos do relatório.



	A	B	C	D
1			Dados	
2	Doc.fat.	GrCl	Soma de Val.líq.	Soma de Mont.imposto
3	0090000193	03	1074,6	5,4
4	0090000193 Total		1074,6	5,4
5	0090000194	02	468,64	2,36
6	0090000194 Total		468,64	2,36
7	0090000195	02	286,56	1,44
8	0090000195 Total		286,56	1,44
9	0090000196	03	468,64	2,36
10	0090000196 Total		468,64	2,36
11	0090000197	03	429,84	2,16
12	0090000197 Total		429,84	2,16
13	0090000198	03	1312,21	6,59

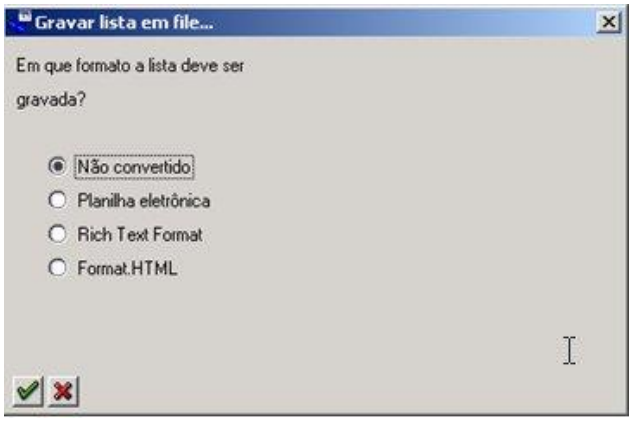
12.13 O botão Processamento de Texto

Na exportação do relatório para o Microsoft Word não temos tantas opções, é bastante simples. Você pode utilizar um modelo do Word ou criar um documento novo.



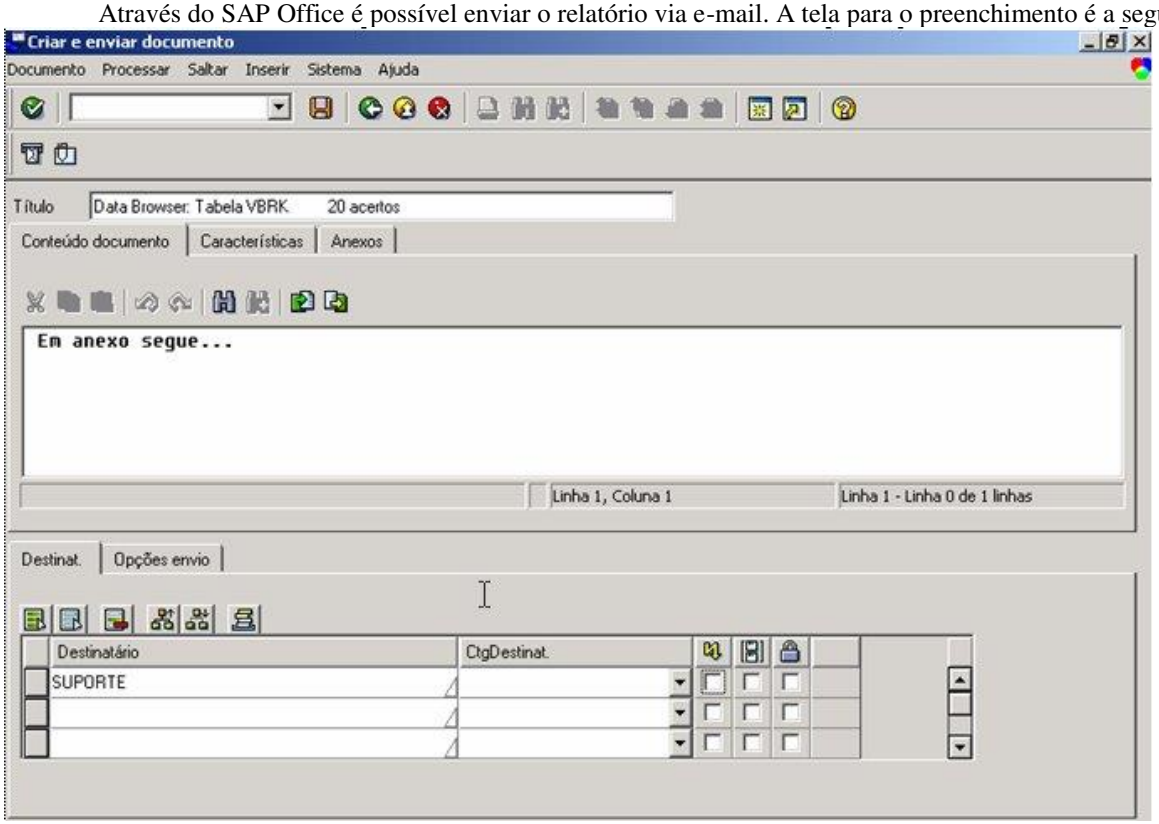
12.14 O botão File Local

Ao salvar o relatório em arquivo local, temos algumas opções. Escolha e logo em seguida o sistema irá solicitar o caminho e nome do arquivo.



12.15 O botão Destinatário de correio eletrônico

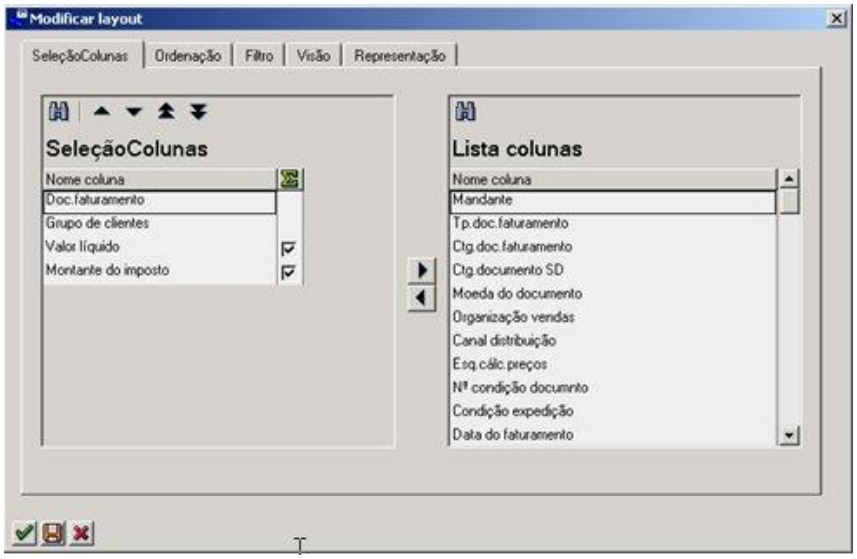
Através do SAP Office é possível enviar o relatório via e-mail. A tela para o preenchimento é a seguinte:



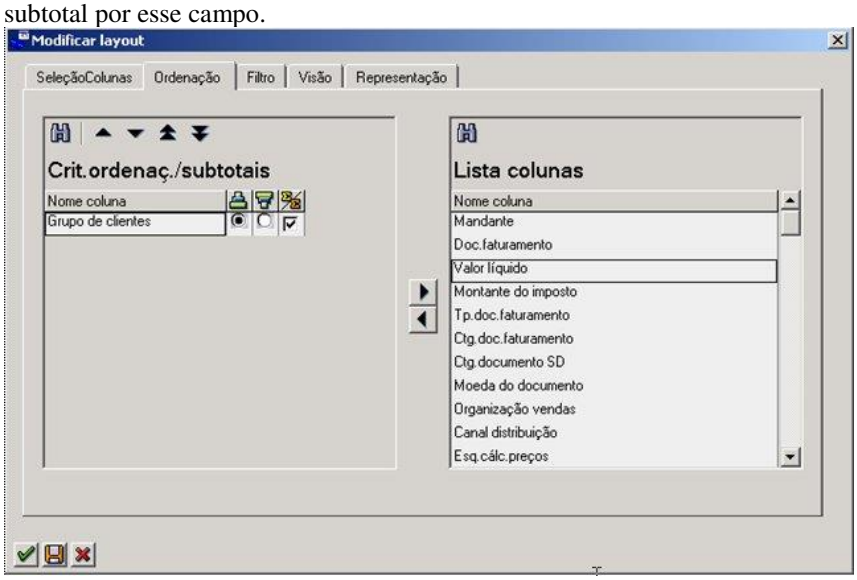
12.16 O botão Modificar layout

Essa opção engloba as funções do Filtro, Totalizar, Subtotalizar, Ordenar, Selecionar colunas, Ocultar colunas, etc. Vejamos cada guia de opção.

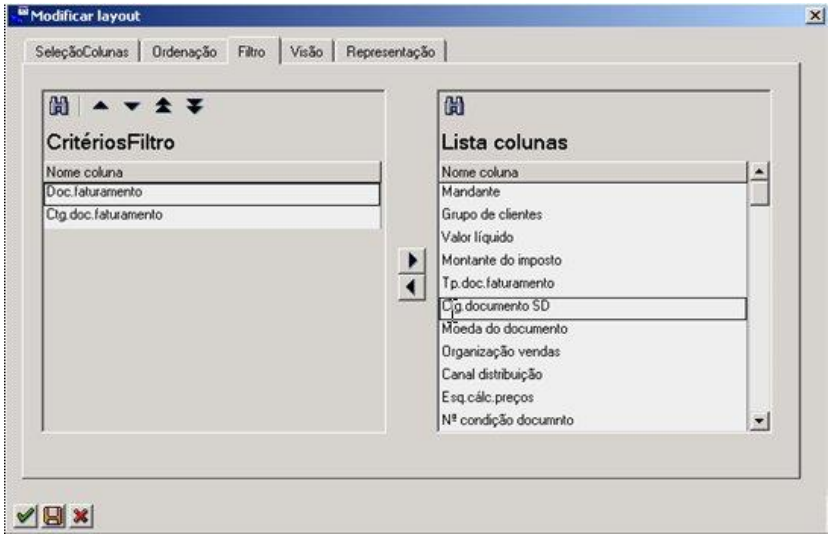
Na figura abaixo, na parte da esquerda, temos as colunas que estão visíveis no relatório. Na parte direita, temos todas as disponíveis. Veja que já temos aqui, a possibilidade de informar quais campos pretendemos totalizar.



Na guia ordenação, informamos quais campos queremos classificar, em qual ordem e ainda, se desejamos um subtotal por esse campo.



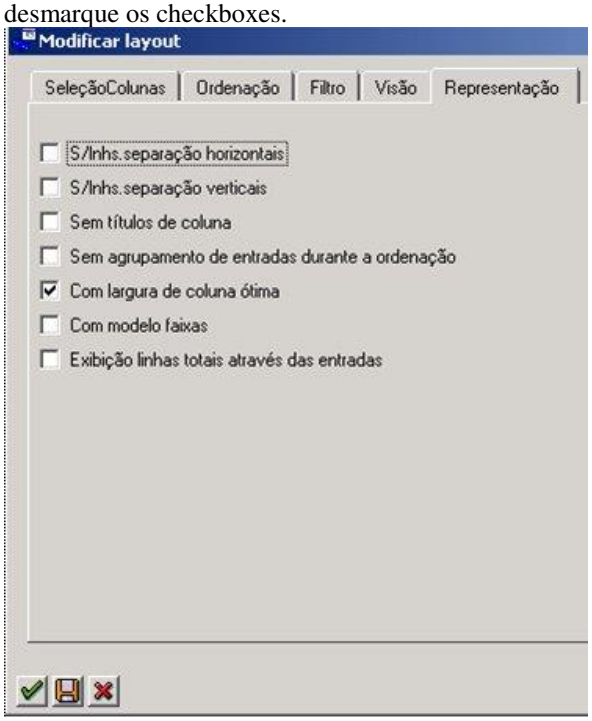
Na guia filtro definimos por qual campo (ou quais) gostaríamos de delimitar o resultado. No caso abaixo, foi escolhido o número do documento de faturamento e sua categoria. A função solicita essa range de valores para gerar o relatório.



A opção Visão raramente é utilizada. Não se cria visões do relatório. É muito mais utilizado e simples as opções de layout. Na guia Visão você pode utilizar templates do Excel ou do Crystal Reports. Também há pouca documentação sobre o assunto.

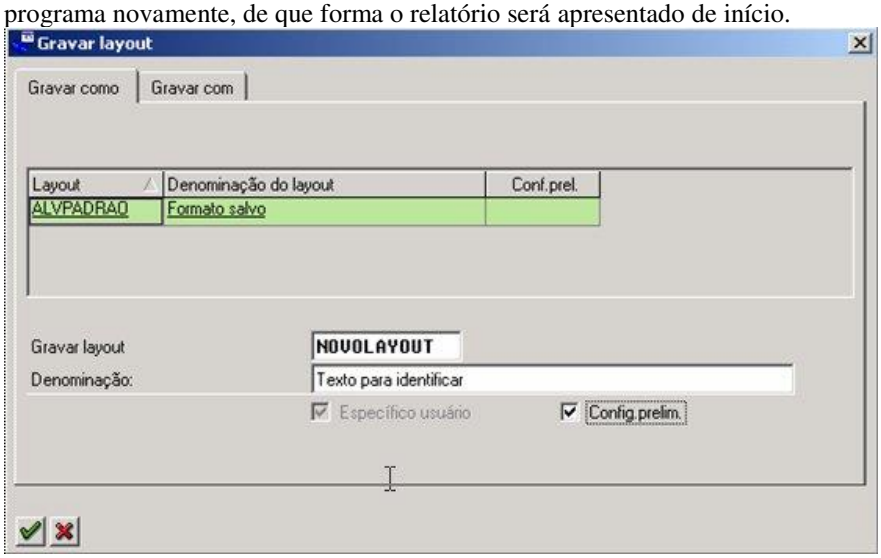
Os layouts que você cria, não deixam de ser visões diferentes de um mesmo relatório.

Na guia de Representação você pode alterar algumas opções no formato de seu relatório. Marque ou desmarque os checkboxes.



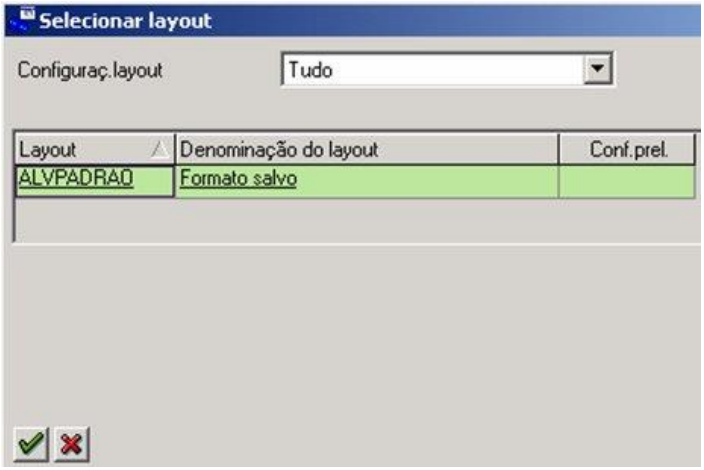
12.17 O botão Gravar Layout

O formato que seu relatório estiver no momento (Que foi configurado) pode ser salvo. Vários layouts diferentes para um mesmo relatório. O flag "Config.prelim" vai indicar qual o layout default, ou seja, ao executar o programa novamente, de que forma o relatório será apresentado de início.



12.18 O botão Selecionar Layout

Da mesma forma é possível mudar o formato de seu relatório a qualquer momento. Basta selecionar outro layout salvo. Na figura abaixo, temos apenas um.

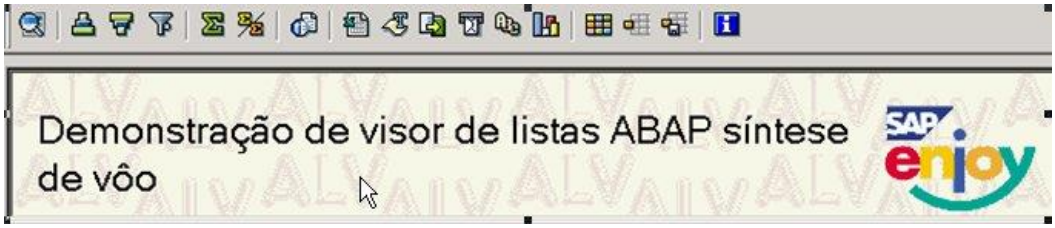


No código em anexo, você verá que é possível, na tela de seleção do programa permitir ao usuário selecionar um layout, antes do relatório ser apresentado com o layout default (Caso exista algum).

Sem nenhum layout, o relatório é apresentado conforme foi codificado no programa.

12.19 Inserindo uma figura no cabeçalho

Que tal o cabeçalho do seu relatório no formato da figura abiaixo (por exemplo), e logo depois as linhas de detalhe:



Isso melhora bem a aparência dos relatórios. Veja que temos até a possibilidade de inserir figura em background, como na Web. O texto também é configurável, você pode inserir várias linhas e alterar o tamanho da fonte.

No exemplo acima foi utilizado como background a imagem ALV_BACKGROUND e o nome do logotipo é ENJOYSAP_LOGO.

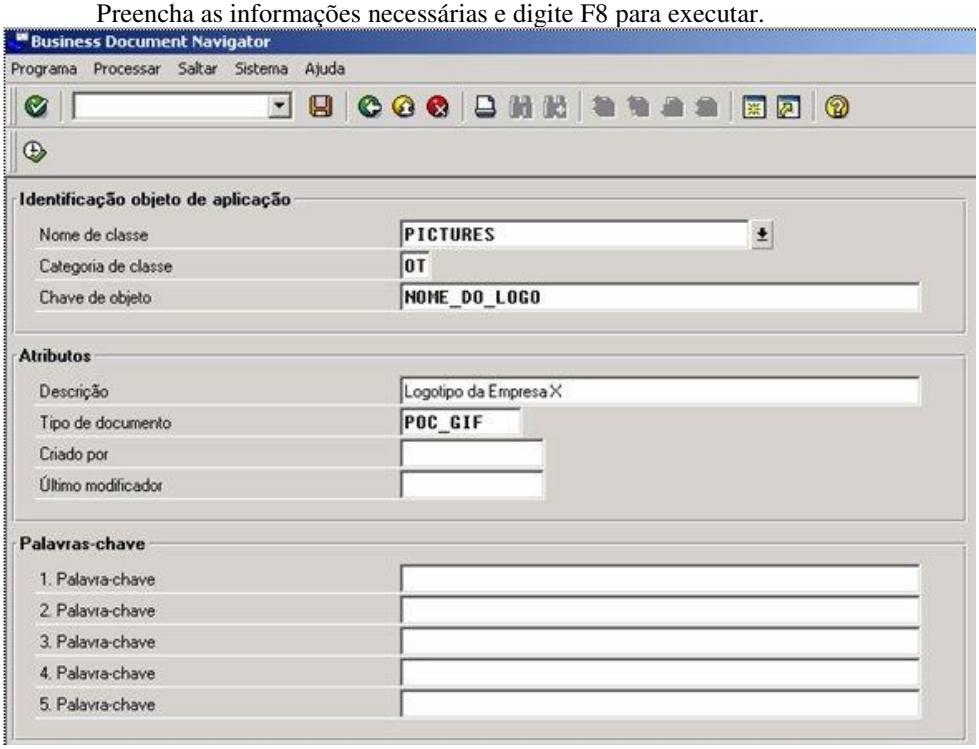
A tabela com todas as figuras disponíveis é a BDS_CONN05, CLASS = PICTURE.

12.20 Importando uma imagem para o R/3

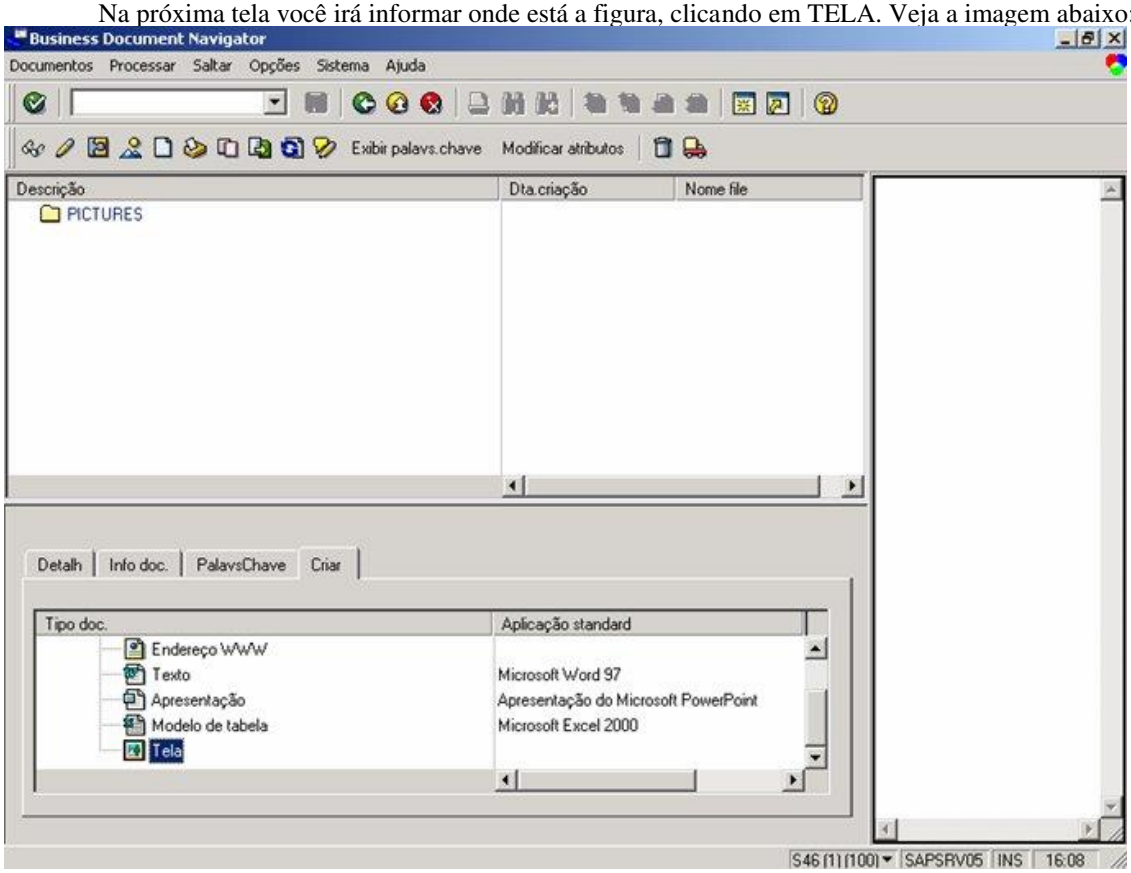
O R/3 permite que você importe figuras, assim não ficamos limitados apenas, as que são standard. Por exemplo importar o logotipo da empresa e inserir no cabeçalho do relatório ALV. Vejamos como se faz.

Após criar a imagem (.GIF, .BMP, .JPG, etc) em algum aplicativo (MS Paint, Photoshop, Paint Shop Pro, etc) o que precisamos, é utilizar a transação 0FPM002 (Import Logo for Reporting) e importar a imagem para o R/3. Nem todas as versões do R/3 possuem essa transação.

Preencha as informações necessárias e digite F8 para executar.



Na próxima tela você irá informar onde está a figura, clicando em TELA. Veja a imagem abaixo:



Com duplo clique no item TELA, a transação apresenta a tela padrão para abrir um arquivo. Localize a figura e pronto.

12.21 Exemplo de programa

```
*---------------------------------------------------------------------
*  ProcWork Informática - ASPEN - Unidade de Negócios SAP
*  Descrição : Programa de exemplo para criação de relatório ALV
*  Dezembro, 2002
*---------------------------------------------------------------------
report zexemplo_alv.
*---------------------------------------------------------------------
* Tabelas transparentes
*---------------------------------------------------------------------
tables:
  vbrk.                                "Documentos de faturamento
*---------------------------------------------------------------------
* Tipos standard
*
* O conjunto de tipos VRM e SLIS são utilizados por funções ALV. Defina
* sempre no início. O mais importante mesmo é o SLIS.
*---------------------------------------------------------------------
type-pools:
  vrm,                                  "Necessário para uso de ALV
  slis.                                "Tipos globais para ALV
*---------------------------------------------------------------------
* Tipos do usuário
*---------------------------------------------------------------------
types:
  begin of y_vbrk,
    vbeln       like vbrk-vbeln,           "Número documento
    kdgrp       like vbrk-kdgrp,           "Grupo de clientes
    netwr       like vbrk-netwr,           "Valor líquido
    mwsbk       like vbrk-mwsbk,           "Montante do imposto
    fkart       like vbrk-fkart,           "Tipo documento faturamento
    vtweg       like vbrk-vtweg,           "Canal de distribuição
    kunag       like vbrk-kunrg,           "Emissor da ordem
    xblnr       like vbrk-xblnr,           "Nota fiscal
    mark        type c,                    "Marcar alterações
  end of y_vbrk.

*---------------------------------------------------------------------
* Tabelas internas ALV
*
* As estruturas aqui utilizadas (SLIS) estão explicadas com as opções
* mais importantes no final da apostila
*---------------------------------------------------------------------
data:
  t_linecolor    type slis_specialcol_alv   occurs 0 with header line,
  t_listheader   type slis_t_listheader,
  t_fieldcat     type slis_t_fieldcat_alv           with header line,
  t_sort         type slis_sortinfo_alv     occurs 0 with header line.

data:
  v_listheader   type slis_listheader,    "Cabeçalho
  v_layout       type slis_layout_alv,    "layout para saída
  v_print        type slis_print_alv,     "Ctrl de impressão
  v_variante     like disvariant.         "Variante de exibição
*---------------------------------------------------------------------
* Tabelas internas
*---------------------------------------------------------------------
data:
  t_vbrk         type y_vbrk      occurs 0 with header line.
```

```
* A próxima tabela é necessário porque não é possível um select em
* tabelas que possuem campos como outras tabelas.
* No caso foi necessário incluir a SLIS_T_SPECIALCOL_ALV
data: begin of t_alv occurs 0.
  include structure t_vbrk.
data:
  color    type slis_t_specialcol_alv. "Definir a cor
data:  end    of t_alv.

*---------------------------------------------------------------------
* Variáveis de uso geral
*---------------------------------------------------------------------
data:
  v_tabix        like sy-tabix,
  v_repid        like sy-repid,
  v_flag.

*---------------------------------------------------------------------
* Tela de seleção
*---------------------------------------------------------------------
selection-screen begin of block one.

select-options:
  s_vbeln for  vbrk-vbeln.              "Documento de faturamento

selection-screen skip.

parameters:
  p_varia like disvariant-variant.      "Variante de exibição

*---------------------------------------------------------------------
* O usuário terá a opção de iniciar a apresentação do relatório com
* algum layout salvo anteriormente.
* Essa escolha será armazenada em P_VARIA. Utilizamos uma função que
* retorna todos os layout possíveis.
*---------------------------------------------------------------------

selection-screen end    of block one.

*---------------------------------------------------------------------
* Eventos
*---------------------------------------------------------------------
initialization.
  perform zf_init_alv.

at selection-screen on value-request for p_varia.
  perform zf_recupera_layouts_salvos.

*---------------------------------------------------------------------
* Principal
*---------------------------------------------------------------------
start-of-selection.

  perform:
    zf_selecao_dados,       "Seleciona a VBRK
    zf_altera_cores,        "Alterando as cores dos campos
    zf_monta_tabela_alv,    "Preenche o catálogo
    zf_sort_subtotal,       "Ordenação dos campos e subtotais
```

```
    zf_executa_funcao_alv.  "Gera o relatório

end-of-selection.
*---------------------------------------------------------------------
* Rotinas
*---------------------------------------------------------------------

*---------------------------------------------------------------------
*       Form  zf_init_alv
*---------------------------------------------------------------------
*       Busca layout de exibição default para o relatório. Se houver
*       algum formato padrão para o relatório, essa função busca e já
*       apresenta o relatório nesse formato.
*       Um layout fica como default quando marcamos "Config.Prelim." Um
*       flag que pode ser marcado na opção "Gravar layout" na barra de
*       ferramentas do ALV
*---------------------------------------------------------------------
form zf_init_alv.
  v_repid = sy-repid.
  clear v_variante.
  v_variante-report = v_repid.
  call function 'REUSE_ALV_VARIANT_DEFAULT_GET'
       EXPORTING
            i_save     = 'A'
       CHANGING
            cs_variant = v_variante
       EXCEPTIONS
            not_found  = 2.
  if sy-subrc = 0.
    p_varia = v_variante.
  endif.
endform.

*---------------------------------------------------------------------
*       Form  zf_recupera_layouts_salvos
*---------------------------------------------------------------------
*       Abre um search help com os layouts já gravados. Se o usuário
*       escolher algum aqui, o programa vai iniciar a apresentação do
*       relatório com esse layout, e não o que é default, retornado na
*       função REUSE_ALV_VARIANT_DEFAULT_GET em ZF_INIT_ALV (Acima)
*---------------------------------------------------------------------
form zf_recupera_layouts_salvos.

  v_variante-report = v_repid.
  call function 'REUSE_ALV_VARIANT_F4'
       EXPORTING
            is_variant = v_variante
            i_save     = 'A'
       IMPORTING
            es_variant = v_variante
       EXCEPTIONS
            not_found  = 2.
  if sy-subrc = 2.
    message id sy-msgid type 'S' number sy-msgno
            with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  else.
    p_varia = v_variante-variant.
  endif.

endform.
```

```
*---------------------------------------------------------------------
*       Form  zf_selecao_dados
*---------------------------------------------------------------------
*       Seleção dos dados
*---------------------------------------------------------------------
form zf_selecao_dados.

  select vbeln kdgrp netwr mwsbk
         fkart vtweg kunag xblnr
    from vbrk
    into table t_vbrk
   where vbeln in s_vbeln.

  loop at t_vbrk.
    move-corresponding t_vbrk to t_alv.
    append t_alv.
  endloop.

endform.
*---------------------------------------------------------------------*
*       Form  zf_altera_cores
*---------------------------------------------------------------------*
*       Permite informar a cor que vc deseja para a coluna, inclusive
*       pelo valor da variável
*---------------------------------------------------------------------*
FORM zf_altera_cores.

*  Veja algumas cores
*  Azul     = 1
*  Verde    = 5
*  Normal   = 2
*  Vermelha = 6

*  Mas vai depender de como estão as cores do seu SAP-Gui

  loop at t_alv.

    refresh t_linecolor.
    refresh t_alv-color.

    t_linecolor-fieldname = 'VBELN'.
    t_linecolor-color-col = '6'.
    t_linecolor-color-inv = '1'.      "Inverso, 1 liga e 0 desliga

    append t_linecolor.

*     Temos a opção INV, que é invertido, mesmo esquema do INT
*     1 liga e 0 desliga


    if t_vbrk-netwr > 10000.
      t_linecolor-fieldname = 'NETWR'.
      t_linecolor-color-col = '1'.
      t_linecolor-color-int = '1'.  "Negrito (1-ligado, 0-Desligado)
    else.
      t_linecolor-fieldname = 'NETWR'.
```

```
        t_linecolor-color-col = '6'.
        t_linecolor-color-int = '1'.  "Negrigo (1-ligado, 0-Desligado)
      endif.

      append t_linecolor.

      t_alv-color[] = t_linecolor[].
      modify t_alv.

  endloop.

ENDFORM.


*---------------------------------------------------------------------
*       Form  zf_monta_tabela_alv
*---------------------------------------------------------------------
*       Monta tabela para apresentação do relatório. Aqui montamos um
*       catálogo com as informações dos campos.
*       Veja que não estamos preenchendo todas as opções do catálogo,
*       não é necessário. No anexo você poderá encontrar os principais
*---------------------------------------------------------------------
form zf_monta_tabela_alv.

  clear t_fieldcat.
  t_fieldcat-fieldname     = 'MARK'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'S'.
  t_fieldcat-inttype       = 'C'.
  t_fieldcat-outputlen     = 1.
  t_fieldcat-checkbox      = 'X'.
  append t_fieldcat.


  clear t_fieldcat.
  t_fieldcat-fieldname        = 'VBELN'.
  t_fieldcat-tabname          = 'T_ALV'.
  t_fieldcat-reptext_ddic     = 'Doc. Fatura'.
  t_fieldcat-inttype          = 'C'.
  t_fieldcat-outputlen        = 10.
  append t_fieldcat.

  clear t_fieldcat.
  t_fieldcat-fieldname     = 'KDGRP'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'Grupo de Clientes'.
  t_fieldcat-inttype       = 'C'.
  t_fieldcat-outputlen     = 2.
  append t_fieldcat.

* Para o campo NETWR, o relatório já vai mostrar linha de total
  clear t_fieldcat.
  t_fieldcat-fieldname     = 'NETWR'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'Valor líquido'.
  t_fieldcat-inttype       = 'P'.
  t_fieldcat-outputlen     = 15.
  t_fieldcat-do_sum        = 'X'.
  append t_fieldcat.
```

```
  clear t_fieldcat.
  t_fieldcat-fieldname     = 'MWSBK'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'Montante do Imposto'.
  t_fieldcat-inttype       = 'P'.
  t_fieldcat-outputlen     = 15.
  append t_fieldcat.


* Os campos abaixo não irão aparecer no relatório, apenas quando
* o usuário modificar o layout e inserir esses campos nas colunas
* a serem apresentadas

  clear t_fieldcat.
  t_fieldcat-fieldname     = 'FKART'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'Tipo do documento'.
  t_fieldcat-inttype       = 'C'.
  t_fieldcat-outputlen     = 4.
  t_fieldcat-no_out        = 'X'.
  append t_fieldcat.

  clear t_fieldcat.
  t_fieldcat-fieldname     = 'VTWEG'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'Canal de Distribuição'.
  t_fieldcat-inttype       = 'C'.
  t_fieldcat-outputlen     = 2.
  t_fieldcat-no_out        = 'X'.
  append t_fieldcat.

  clear t_fieldcat.
  t_fieldcat-fieldname     = 'KUNAG'.
  t_fieldcat-tabname       = 'T_ALV'.
  t_fieldcat-reptext_ddic  = 'Emissor da Ordem'.
  t_fieldcat-inttype       = 'C'.
  t_fieldcat-outputlen     = 10.
  t_fieldcat-no_out        = 'X'.
  append t_fieldcat.

* Para o campo XBLNR, não vamos preencher nada. Nem disponível
* na modificação do layout ele vai estar.
* Não é necessário atribuir todos os campos, não ocorre
* nenhum erro.

endform.

*---------------------------------------------------------------------
*       Form  zf_sort_subtotal
*---------------------------------------------------------------------
*       Classificação e item de subtotalização
*---------------------------------------------------------------------
form zf_sort_subtotal.

  clear t_sort[].
  t_sort-spos      = 1.
  t_sort-fieldname = 'KDGRP'.
  t_sort-tabname   = 'T_ALV'.
  t_sort-up        = 'X'.
  t_sort-subtot    = 'X'.
```

```
  append t_sort.

* Com isso o relatório vai sair classificado em ordem crescente de Grupo
* de cliente e ainda irá aparecer um subtotal por esse campo.

endform.

*---------------------------------------------------------------------
*       Form  zf_executa_funcao_alv
*---------------------------------------------------------------------
*       Apresenta relatório
*---------------------------------------------------------------------
form zf_executa_funcao_alv.

* Preenchendo algumas opções de impressão (Não é obrigatório)
  v_layout-expand_all        = 'X'.  "Abrir subitens
  v_layout-colwidth_optimize = 'X'.  "Largura melhor possível da coluna
  v_layout-edit              = 'X'.  "Permitir a edição

* Indicando para função qual o layout que deve ser apresentado
* primeiro

  v_variante-variant         = p_varia.
  v_print-no_print_listinfos = 'X'.


  call function 'REUSE_ALV_GRID_DISPLAY'
       EXPORTING
            i_callback_program       = v_repid
            i_background_id          = 'ALV_BACKGROUND'
            i_callback_top_of_page   = 'ZF_TOP_OF_PAGE'
*           i_callback_pf_status_set = 'ZF_STATUS'
            i_callback_user_command  = 'ZF_USER_COMMAND'
            it_fieldcat              = t_fieldcat[]
            is_layout                = v_layout
            it_sort                  = t_sort[]
            i_default                = 'X'
            i_save                   = 'A'
            is_variant               = v_variante
            is_print                 = v_print
       TABLES
            t_outtab                 = t_alv
       EXCEPTIONS
            program_error            = 1
            others                   = 2.

* As funções que geram relatórios ALV possuem vários parâmetros de
* I_CALLBACK. Os que mais são utilizados, são os que estão
* na chamada acima. Para ver os demais use a transação SE37. Esses
* parâmetros são preenchidos com nomes de FORMS do programa

* i_callback_program       = Qual programa que executou a função
* i_callback_top_of_page   = Rotina de cabeçalho
* i_callback_pf_status_set = Qual barra de tarefas a função vai usar
* i_callback_user_command  = Tratamento dos botões alterados ou criados

endform.
```

```
*---------------------------------------------------------------------
*       Form zf_user_command
*---------------------------------------------------------------------
*       Tratamento das opções do usuário. Por exemplo um Drill-down ou
*       algum botão que você inseriu ou alterou. O importante é conhecer
*       os parâmetros que o form recebe
*---------------------------------------------------------------------
form zf_user_command using ucomm    like sy-ucomm
                           selfield type slis_selfield.
* UCOMM:    é o sy-ucomm (Ok-code)
* SELFIELD: é uma estrutura com dados que nós permite identifcar
*           o que foi selecionado. Essa estrutura também está
*           explicada no anexo ao final da apostila

*    Salva a posição do relatório (Linha escolhida)
     selfield-row_stable = 'X'.

*    Uma das questões foi como alterar o conteúdo de uma tabela
*    transparente com as alterações feitas no relatório ALV
*    Segue um exemplo de como pode ser feito:
*    Em nossa barra de ferramentas criamos o botão com código
*    ZATU

     if ucomm = 'ZATU'.

*       Vamos ler a tabela T_VBRK onde mark = X. A idéia é que
*       o usuário mark com X os registros alterados
        loop at t_vbrk where mark = 'X'.
            v_tabix = sy-tabix.
*           Atualiza a tabela transparente
            " update ztabela ....

*           Então voltamos a T_VBRK sem marcação alguma
            clear t_vbrk-mark.
            modify t_vbrk index v_tabix.

*           Veja que esse tipo de esquema pode ser feito para
*           excluir registros também
        endloop.

     endif.


*    Para testar o código do botão
     if ucomm = 'ZLOG'.
        " perform ...
        " call transaction...
     endif.

*    Para um drill down a partir de um registro
     if not selfield-tabindex = 0.
        read table t_vbrk index selfield-tabindex.
```

```
        "perform ...
        "call transaction ...
      else.
*       Clicou em linha inválida, linha de total, cabeçalho, etc
      endif.

endform.


*---------------------------------------------------------------------
*       Form  zf_top_of_page
*---------------------------------------------------------------------
*       Cabeçalho do relatório
*---------------------------------------------------------------------
form zf_top_of_page.

* Uma dica, em relatórios ALV com utilização de drill down, na volta ao
* relatório principal ele vai executar novamente o cabeçalho. Isso faz
* com que ele fique duplicado. Utilize um flag que após apresentar o
* cabeçalho uma vez, fique marcado com um 'X', por exemplo.

* V_FLAG na primeira vez é branco.

  check v_flag is initial.


* Monta as linhas de cabeçalho

  clear t_listheader[].
  clear v_listheader.
  v_listheader-typ  = 'H'.

* TYP = H, faz com que a fonte fique maior

  v_listheader-info = 'Apenas um Exemplo em ALV'.
  append v_listheader to t_listheader.

* Definição do Projeto
  clear v_listheader.
  v_listheader-typ  = 'A'.

* TYP = S, outro tipo de fonte

  v_listheader-info = 'Segunda linha do cabeçalho'.
  append v_listheader to t_listheader.

* O campo INFO, pode ter no máximo 60 caracteres

* Apresenta o cabeçalho.
* Veja que já é uma outra função, essa apresenta a figura, já
* a função REUSE_ALV_GRID_DISPLAY (Principal) tem o parâmetro
* que você indica a imagem que será apresentada em background,
* como na WEB.

  call function 'REUSE_ALV_COMMENTARY_WRITE'
       EXPORTING
            i_logo             = 'ENJOYSAP_LOGO'
            it_list_commentary = t_listheader.

* Para não apresentar mais o cabeçalho no refresh
```

```
  v_flag = 'X'.

endform.

*---------------------------------------------------------------------
*       Form zf_status
*---------------------------------------------------------------------
*       Status com botão de log (Item a mais na barra ALV)
*---------------------------------------------------------------------
form zf_status using rt_extab type slis_t_extab.
* Aqui estamos informando a função que ela deverá utilizar a barra de
* ferramentas ZALV_BOTOES.

  "set pf-status 'ZALV_BOTOES'.

* Também é possível excluir funções
  "if sy-uname = ...
     "EXCLUDING ...
  "endif.

endform.
```

12.22 Estruturas SLIS

SLIS_FIELDCAT_ALV – Tabela que é o catálogo de campos. Vai conter toda informação necessária sobre cada campo do relatório. É formada por um grupo de tipos, aqui descrevemos os mais utilizados. Obrigatório (Ou no mínimo) preencha os campos que estão sublinhados.		
FIELDNAME	Tam. 30, tipo caracter	Nome do campo, que vai ser uma coluna do seu relatório.
TABNAME	Tam. 30, tipo caracter	Nome da tabela que possui o campo definido acima. Essa tabela deve os registros (linhas) do seu relatório.
ICON	Tam. 1, tipo caracter	Preencha com X, caso o campo represente um ícone.
SYMBOL	Tam. 1, tipo caracter	Preencha com X, caso o campo represente um símbolo.
CHECKBOX	Tam. 1, tipo caracter	Preencha com X, caso deseje apresentar um checkbox na coluna. Utilizado para campos do tipo flag.
JUST	Tam. 1, tipo caracter	Justificar a texto do campo. Use R, C ou L (Direita, Esquerda ou Centro).
LZERO	Tam. 1, tipo caracter	Preencher com X, para eliminar zeros à esquerda.
NO_SIGN	Tam. 1, tipo caracter	Preencher com X, para não apresentar sinal em valores.
NO_ZERO	Tam. 1, tipo caracter	Preencher com X, para não exibir valores zerados.
DO_SUM	Tam. 1, tipo caracter	Se a coluna é um valor, preencha com X, caso queira a sumarização já na apresentação do relatório.
NO_OUT	Tam. 1, tipo caracter	Preencher com X, se deseja que a coluna não apareça.
OUTPUTLEN	Tam. 6, tipo numérico	Aqui você indica o tamanho do seu campo.
INTTYPE	Tam. 1, tipo caracter	Tipo do campo, veja os tipos principais: C-Cadeia de caracteres N-Cadeia de caracteres (Só número) D-Data (data: AAAAMMDD) T-Momento (hora:HHMMSS) X-Seqüência de byte (hexadecimal) I-Nº inteiro (4 byte c/sinal) P-Compactado F-Ponto flutuante
REPTEXT_DDIC	Tam. 30, tipo caracter	Label da coluna do relatório.
HOTSPOT	Tam. 1, tipo caracter	Preencher com X, caso deseje que ao passar o mouse por cima do campo, apareça o ícone de uma "mão". Indicando que existe alguma ação ao clicar sobre esse campo.

SLIS_T_LISTHEADER – Tabela para criação do cabeçalho de seu relatório. O preenchimento de todos os campos, inclusive a própria utilização dessa tabela não é obrigatória.		
TYP	H = Header, S = Selection, A = Action	Dependendo do tipo (H,S ou A) o tipo de letra modifica.
KEY	Tam. 20, tipo caracter	Não precisa preencher.
INFO	Tam. 60, tipo caracter	É o texto que você quer que apareça no cabeçalho. Você pode utilizar n linhas. Veja no programa exemplo.

SLIS_SORTINFO_ALV – Tabela que vai indicar para função como é a classificação dos campos do relatório. O preenchimento de todos os campos, inclusive a própria utilização dessa tabela não é obrigatória.		
FIELDNAME	Tam. 30, tipo caracter	Nome do campo, que vai ser uma coluna do seu relatório
TABNAME	Tam. 30, tipo caracter	Nome da tabela que possui o campo definido acima.
UP	Tam. 1, tipo caracter	Marque esse campo com X para classificar em ordem crescente.
DOWN	Tam. 1, tipo caracter	Marque esse campo com X para classificar em ordem decrescente.
SUBTOT	Tam. 1, tipo caracter	Marque com X para que no relatório apareça um subtotal por esse campo.
SPOS	Tam. 2, tipo numérico	Seqüência de ordenação dos campos

DISVARIANT - Variant.Exibição (Layouts salvos)		
REPORT	Nome do programa ABAP	Preencher com o nome do programa. Utilize a variável do sistema SY-REPID
HANDLE	ID controle p/chamadas múltiplas a partir do mesmo programa	Uso interno
LOG_GROUP	Conceito lógico de grupo	Uso interno
USERNAME	Nome do usuário para gravação específica do usuário	Uso interno
VARIANT	Layout	Uso interno. Aqui vai retornar o nome do layout que você salvou
TEXT	Denominação layout	Uso interno. Aqui retorna a descrição que você deu para o seu layout
DEPENDVARS	Vetor para entradas de variante dependentes	Uso interno

SLIS_LAYOUT_ALV – Define o formato de saída do relatório. O preenchimento de todos os campos, inclusive a própria utilização dessa estrutura não é obrigatória.		
NO_COLHEAD	Tam. 1, tipo caracter	Preencha com X para que o seu relatório não tenha label das colunas.
ZEBRA	Tam. 1, tipo caracter	Preencher com X para que sua listagem apareça zebrada.
NO_VLINE	Tam. 1, tipo caracter	Preencher com X para que as colunas do relatório não tenham divisões.
NUMC_SUM	Tam. 1, tipo caracter	Preencher com X, para que o relatório permita totalização de campos to tipo N (Caracteres numéricos).
EDIT	Tam. 1, tipo caracter	Preencher com X para que o usuário possa editar o valor do campo no relatório.
NO_INPUT	Tam. 1, tipo caracter	Preencher com X se o campo for apenas de saída.
COLWIDTH_OPTIMIZE	Tam. 1, tipo caracter	Marcar com X para que a largura da coluna fique de acordo com o maior tamanho: Label ou Detalhe
NO_TOTALLINE	Tam. 1, tipo caracter	Preenchendo esse campo com X, em seu relatório não irá aparecer linha de total.
TOTALS_BEFORE_ITEMS	Tam. 1, tipo caracter	Preencher com X para que os totais apareçam

		antes das linhas de detalhe.
TOTALS_ONLY	Tam. 1, tipo caracter	Preencher com X para que em seu relatório apareçam apenas os totais.
TOTALS_TEXT	Tam. 60, tipo caracter	Texto para as linhas de total
SUBTOTALS_TEXT	Tam. 60, tipo caracter	Texto para as linhas de subtotal

SLIS_PRINT_ALV – Informação para a impressão. O preenchimento de todos os campos, inclusive a própria utilização dessa estrutura não é obrigatória.		
PRINT	Tam. 1, tipo caracter	Preencher com X para permitir a impressão.
PRNT_TITLE	Tam. 1, tipo caracter	Preencher com X para permitir a impressão do título do relatório.
NO_PRINT_LISTINFOS	Tam. 1, tipo caracter	Preencher com X para que na impressão não apareçam as informações da listagem.

SLIS_SELFIELD – Informação sobre o registro selecionado. As informações são retiradas do catálogo.		
TABNAME	Tam. 30, tipo caracter	Nome da tabela que dá origem aos dados.
TABINDEX	Sy-tabix	Vai indicar a posição do registro dentro da tabela. Se precisar recuperar os valores utilize esse índice em um comando read table ... index slis_selfield-tabindex.
COL_STABLE	Tam. 1, tipo caracter	Marque com X, para manter o relatório na coluna em que estava antes do drill down.
ROW_STABLE	Tam. 1, tipo caracter	Marque com X, para manter o relatório na linha selecionada antes do drill down.
REFRESH	Tam. 1, tipo caracter	Para atualizar o relatório, preencher com X.

Problemas e correções por favor enviar para: marcelo.bueno@procwork.com.br

13 ALV – Mais Detalhes

O ALV Grid é uma ferramenta flexível para exibição de relatórios ou árvore.
São disponibilizados botões que permitem ao usuário manipular os dados (classificar, filtrar, somar).
Além dos botões standards do sistema, é possível criar novos botões conforme a necessidade do usuário. Isto pode eliminar certas etapas no processo de gerenciamento de eventos para controles.

ATENÇÃO: O ALV não permite processamento em Background

13.1 UTILIZAÇÃO

O Abap List Viewer padroniza e simplifica o uso de listas e relatórios no sistema R/3.
Pode-se especificar os campos a serem exibidos no relatório e modificar a seqüência em que esses campos são exibidos. Além disso, pode-se ajustar a largura das colunas individuais para atender a requisitos específicos.

O ALV permite :

Usar variantes de exibição standard predefinidas pela SAP

 Modificar layout

 Selecionar Layout

 Gravar layout

Ordenar os dados: Ordenar as linhas de acordo com os valores das colunas, em seqüência crescente e decrescente

 Ordenar Crescente

 Ordenar Decrescente

Definir um filtro : Exibir somente os campos desejados

 Definir Seleção de dados

Formar totais e subtotais : Em uma lista, é possível calcular totais e subtotais de uma ou mais colunas selecionadas.

 Total

 Sub-Total

Exibição de informações detalhadas : Pode-se acessar informações detalhada de linhas individuais da lista

Detalhe

Pesquisa : É possível pesquisar informações específicas

Pesquisar

Impressão de Lista e pré-visualização : Pode-se imprimir as listas e chamar uma pré-exibição antes de imprimir

Impressão

Pré-Visualização

Exportação de dados : Pode-se copiar as listas, por exemplo, para uma planilha ou grava-las como arquivo local

Arquivo Local

Excel

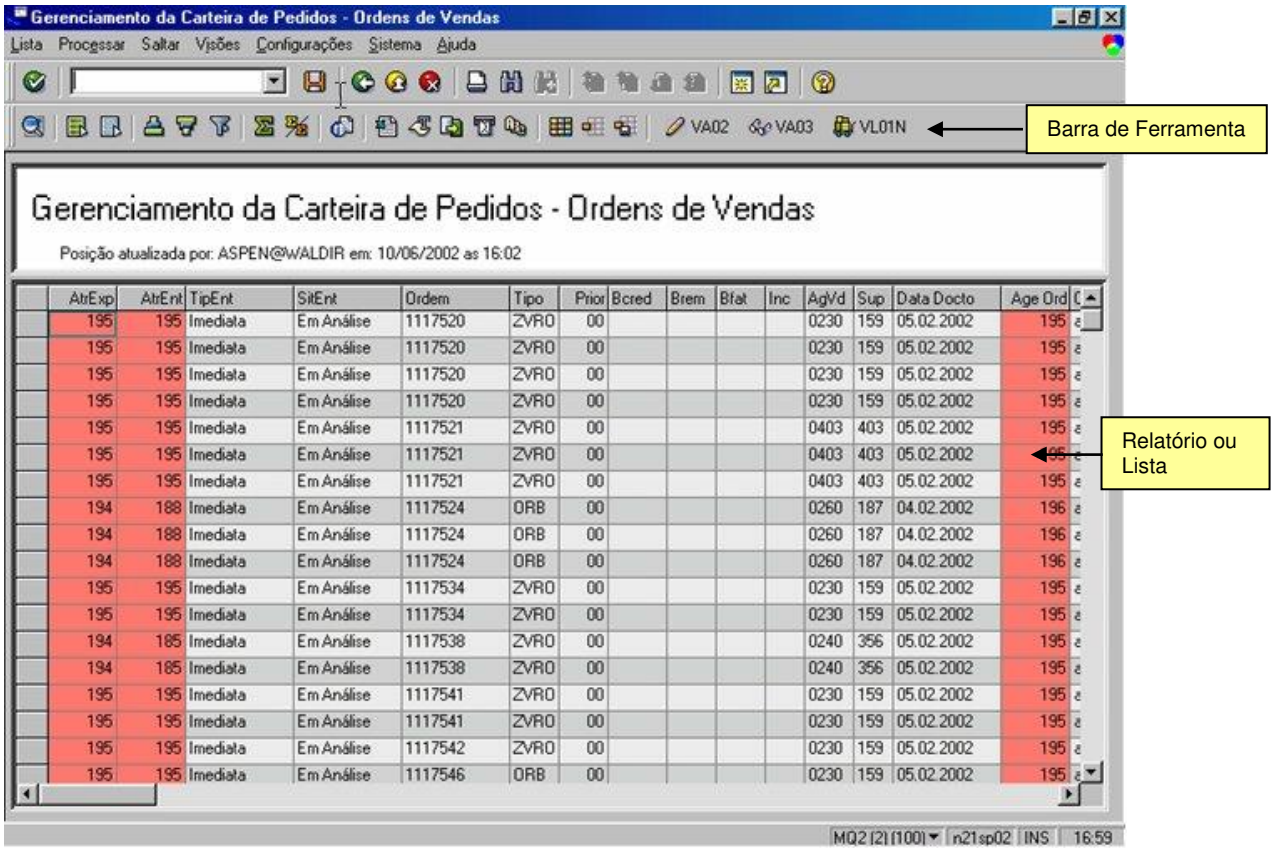
Processamento de Texto

Destinatário de Correio Eletrônico – Enviar relatório via correio SAP

Deslocar colunas: Mudar coluna de lugar

O ALV Grid é formado basicamente por :

- ✓ Uma barra de ferramenta
- ✓ Um título
- ✓ Uma lista de saída



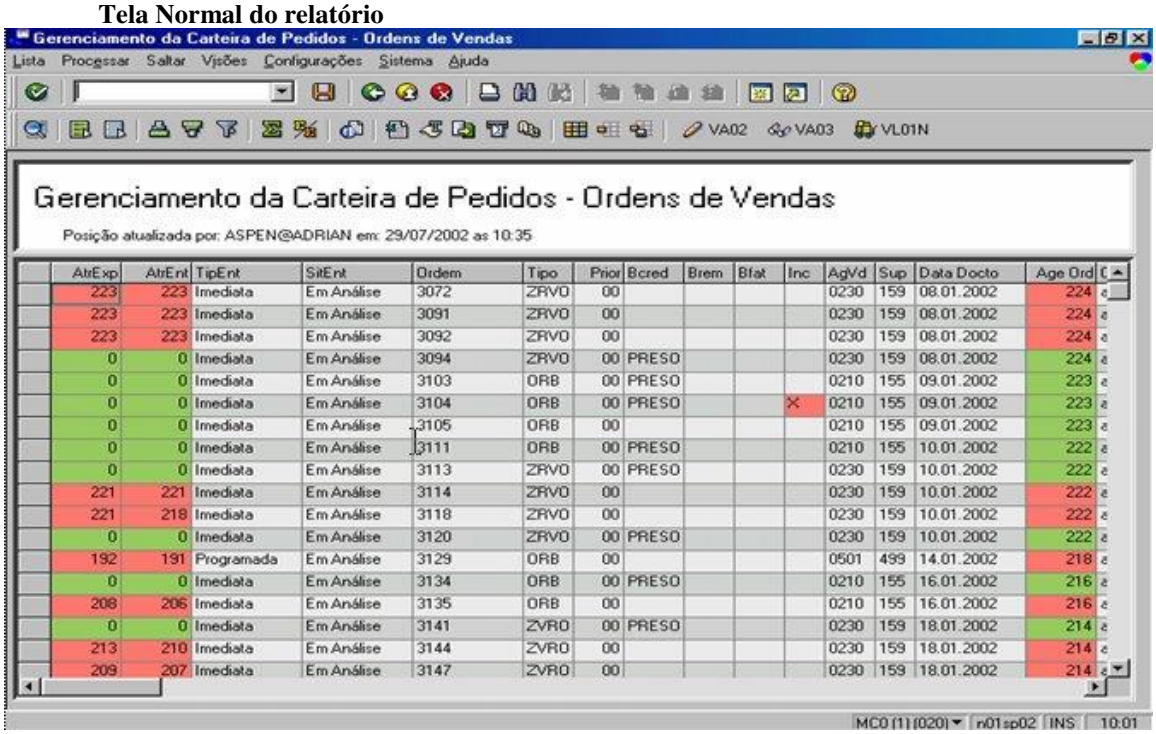
13.2 Conceito de Variante de Exibição

A variante de exibição é utilizada para exibir o relatório em vários formatos diferente. Cada usuário pode criar a sua própria variante e utiliza-la para visualizar o relatório.

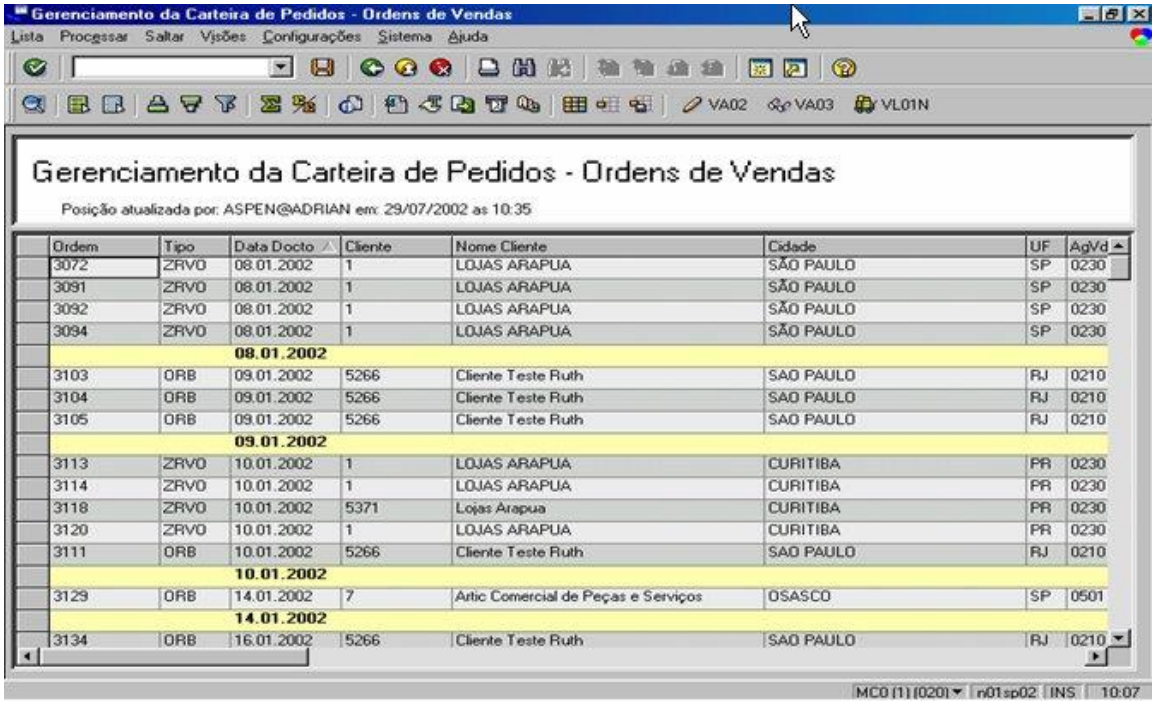
Por exemplo : Um relatório que tenha 20 campos exibidos na tela. Você pode eliminar os campos que não irá utilizar, acrescentar totalizadores, ordenação, mudar os campos de posição, etc.

EXEMPLO:

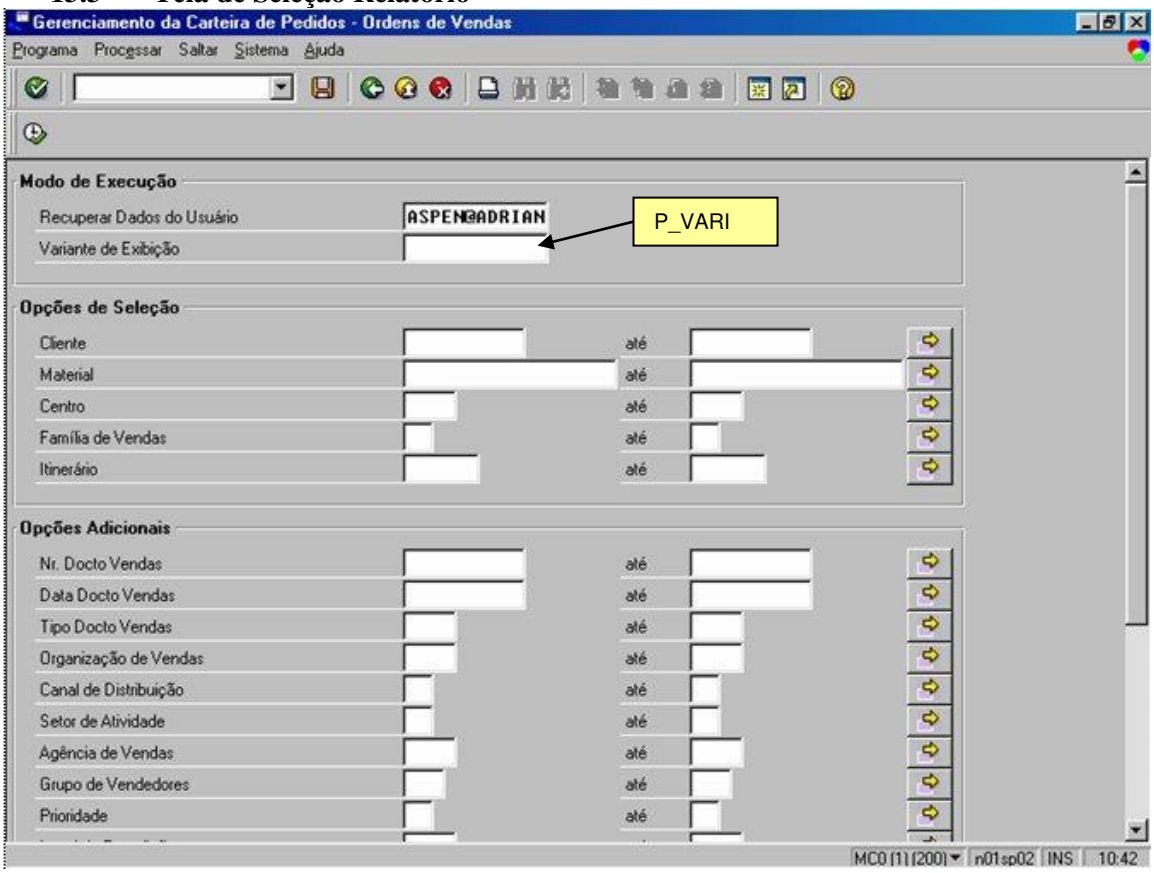
Tela Normal do relatório



Eliminando Campos, Ordenando pela Data Documento e Criando Totalizadores


Gerenciamento da Carteira de Pedidos - Ordens de Vendas
Lista Processar Saltar Visões Configurações Sistema Ajuda
VA02 VA03 VL01N
Gerenciamento da Carteira de Pedidos - Ordens de Vendas
Posição atualizada por: ASPEN@ADRIAN em: 29/07/2002 as 10:35
Ordem | Tipo | Data Docto | Cliente | Nome Cliente | Cidade | UF | AgVd
3072 | ZRVO | 08.01.2002 | 1 | LOJAS ARAPUA | SÃO PAULO | SP | 0230
3091 | ZRVO | 08.01.2002 | 1 | LOJAS ARAPUA | SÃO PAULO | SP | 0230
3092 | ZRVO | 08.01.2002 | 1 | LOJAS ARAPUA | SÃO PAULO | SP | 0230
3094 | ZRVO | 08.01.2002 | 1 | LOJAS ARAPUA | SÃO PAULO | SP | 0230
08.01.2002
3103 | ORB | 09.01.2002 | 5266 | Cliente Teste Ruth | SAO PAULO | RJ | 0210
3104 | ORB | 09.01.2002 | 5266 | Cliente Teste Ruth | SAO PAULO | RJ | 0210
3105 | ORB | 09.01.2002 | 5266 | Cliente Teste Ruth | SAO PAULO | RJ | 0210
09.01.2002
3113 | ZRVO | 10.01.2002 | 1 | LOJAS ARAPUA | CURITIBA | PR | 0230
3114 | ZRVO | 10.01.2002 | 1 | LOJAS ARAPUA | CURITIBA | PR | 0230
3118 | ZRVO | 10.01.2002 | 5371 | Lojas Arapua | CURITIBA | PR | 0230
3120 | ZRVO | 10.01.2002 | 1 | LOJAS ARAPUA | CURITIBA | PR | 0230
3111 | ORB | 10.01.2002 | 5266 | Cliente Teste Ruth | SAO PAULO | RJ | 0210
10.01.2002
3129 | ORB | 14.01.2002 | 7 | Artic Comercial de Peças e Serviços | OSASCO | SP | 0501
14.01.2002
3134 | ORB | 16.01.2002 | 5266 | Cliente Teste Ruth | SAO PAULO | RJ | 0210
MC0 (1) (020) n01sp02 INS 10:07

13.3 Tela de Seleção Relatório



EVENTO – INITIALIZATION

Neste evento você verificará se existe uma variante definida como default para o relatório através da função REUSE_ALV_VARIANT_DEFAULT_GET

Variável :

```
def_variante     LIKE disvariant.
```

Parâmetro da tela de seleção :

```
PARAMETERS : p_vari LIKE disvariant-variant.
```

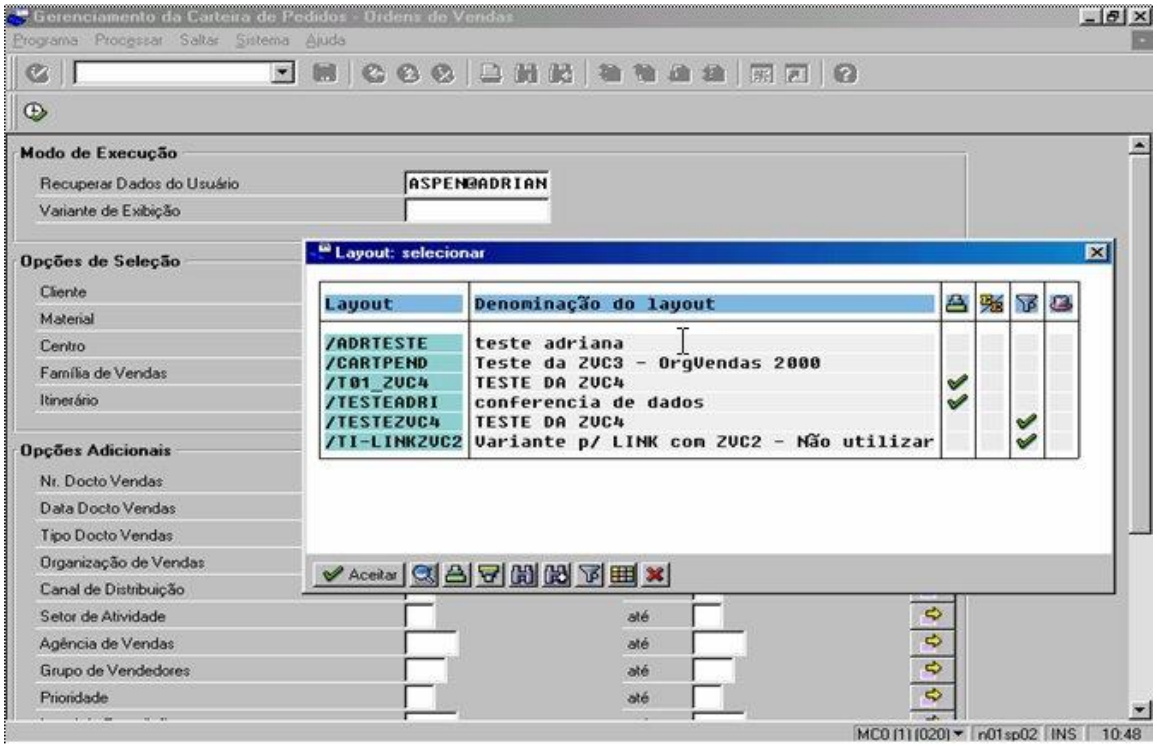
Função :

```
CALL FUNCTION 'REUSE_ALV_VARIANT_DEFAULT_GET'
    EXPORTING
        i_save     = 'A'
    CHANGING
        cs_variant = def_variante
    EXCEPTIONS
        not_found  = 2.
 IF sy-subrc = 0.
       p_vari = def_variante-variant.
 ENDIF.
```

EVENTO – AT SELECTION-SCREEN ON VALUE-REQUEST .....

Neste evento você deverá criar uma rotina para exibir as variantes já existentes para o relatório em questão.

```
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_vari.
    PERFORM f_f4_variant.
```



Variáveis :

```
variant_exit(01) TYPE c,
variante        LIKE disvariant,
```

Função :

```
CALL FUNCTION 'REUSE_ALV_VARIANT_F4'
   EXPORTING
      is_variant          = variante
      i_save     = 'A'
   IMPORTING
      e_exit              = variant_exit
      es_variant          = def_variante
   EXCEPTIONS
      not_found           = 2.
IF sy-subrc = 2.
        MESSAGE ID sy-msgid TYPE 'S' NUMBER sy-msgno
                    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ELSE.
      IF variant_exit = space.
            p_vari = def_variante-variant.
      ENDIF.
ENDIF.
```

EVENTO – AT SELECTION-SCREEN

Neste evento você irá tratar a variante informada, ou seja, verificar a sua existência, pois o usuário pode ter digitado um nome qualquer ao invés de utilizar o F4.

```
AT SELECTION-SCREEN.
        PERFORM f_variante.

Verificar se existe valor no parâmetro da tela
IF NOT p_vari IS INITIAL.
```

Mover valor em branco para a estrutura def_variante e o nome da variante para o campo def_variante-variant

```
    MOVE variante TO def_variante.
    MOVE p_vari TO def_variante-variant.
```

Executar a função

```
        CALL FUNCTION 'REUSE_ALV_VARIANT_EXISTENCE'
            EXPORTING
              i_save     = 'A'
            CHANGING
              cs_variant = def_variante.
```

Mover estrutura def_variante para variante

```
        variante = def_variante.

    ELSE.
```

Se não existir valor no parâmetro inicializar a estrutura Variante

```
        CLEAR variante.
        variante-report = v_repid.
    ENDIF.
```

14 FORMATAÇÃO DO RELATÓRIO EM ALV

Neste tópico iremos mostrar as rotinas necessárias para formatação de um relatório em ALV. Formatação geral, de campos, botões e etc.

14.1 Definições de dados para o layout ALV

Definicao de Tipos

Grupos de tipo

```
TYPE-POOLS: slis.
TYPE-POOLS: kkblo.
```

Workareas

```
DATA:   w_layout    TYPE slis_layout_alv,
        w_print     TYPE slis_print_alv
        w_event     TYPE slis_alv_event,
        w_line      TYPE slis_listheader.

DATA:   colinfo TYPE kkblo_specialcol.
```

Tabela Interna

```
DATA:   l_sort              TYPE slis_t_sortinfo_alv,
        l_event     TYPE slis_t_event,
        l_top_of_page       TYPE slis_t_listheader,
        t_campos            TYPE slis_t_fieldcat_alv WITH HEADER LINE.
```

Tabela Interna para Impressão do Relatório

```
DATA : BEGIN OF t_relat OCCURS 0,
          Campo1
          Campo2
          Campo3
          .........
          .........
          .........
         box(1)     TYPE c,                     " Campo Selecao Linha
         colinfo    TYPE kkblo_t_specialcol,    " Formatação de Colunas
END OF t_relat.
```

14.2 Rotinas para Formatação do Layout

PERFORM f_evento_lista. " Eventos da Lista

O evento mais comum a ser definido é o TOP_OF_PAGE para impressão do cabeçalho na lista

Executar a função para selecionar todos os eventos possíveis de serem tratados

```
CALL FUNCTION 'REUSE_ALV_EVENTS_GET'
     EXPORTING
        I_LIST_TYPE = 0
     IMPORTING
        ET_EVENTS   = t_event.
```

Ler a tabela para o evento TOP_OF_PAGE

```
READ TABLE t_event WITH KEY NAME =  SLIS_EV_TOP_OF_PAGE
                           INTO w_event.
```

Se o evento foi encontrado então cadastrar o nome do form do seu programa (que trata o cabeçalho) no campo FORM

```
IF SY-SUBRC = 0.
   MOVE 'F_CABECALHO' TO w_event-form.
   APPEND w_event TO t_event.
ENDIF.
```

Criar o FORM F_CABECALHO

Criar o FORM f_cabeçalho e inserir o código abaixo.

```
CALL FUNCTION 'REUSE_ALV_COMMENTARY_WRITE'
   EXPORTING
    it_list_commentary       = l_top_of_page.
```

Outros eventos que retornarão da função e podem ser tratados :

CALLER_EXIT
USER_COMMAND
TOP_OF_PAGE
TOP_OF_COVERPAGE
END_OF_COVERPAGE
FOREIGN_TOP_OF_PAGE
FOREIGN_END_OF_PAGE
PF_STATUS_SET
LIST_MODIFY
TOP_OF_LIST
END_OF_PAGE
END_OF_LIST
AFTER_LINE_OUTPUT
BEFORE_LINE_OUTPUT
REPREP_SEL_MODIFY
SUBTOTAL_TEXT

Estes eventos estão gravados dentro do Tipo SLIS

PERFORM f_cabec_lista. " Cabecalho da Lista

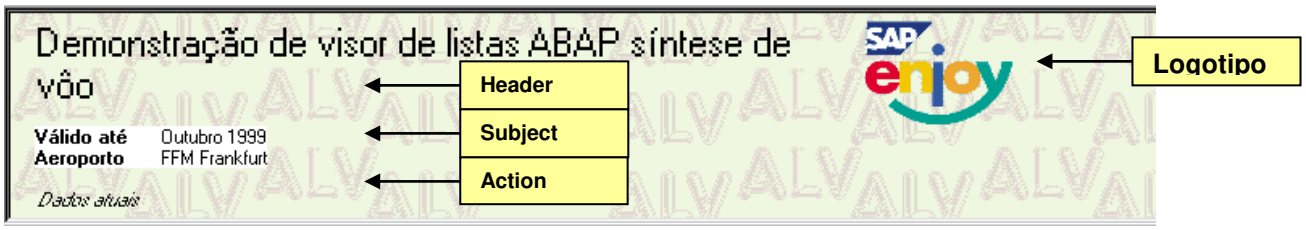
Neste FORM iremos criar os textos que deverão sair no cabeçalho do relatório

```
* Texto Principal - Header
 CLEAR w_line.
 w_line-typ  = 'H'.
 w_line-info = text-h01.
 APPEND w_line TO t_top_of_page.
```

No campo TYP deverá ser informado :

H – Header
S – Selection
A – Action

Estes tipos irão colocar os textos em formatos diferentes (letra, negrito ou itálico). Estas informações serão gravadas na tabela interna T_TOP_OF_PAGE

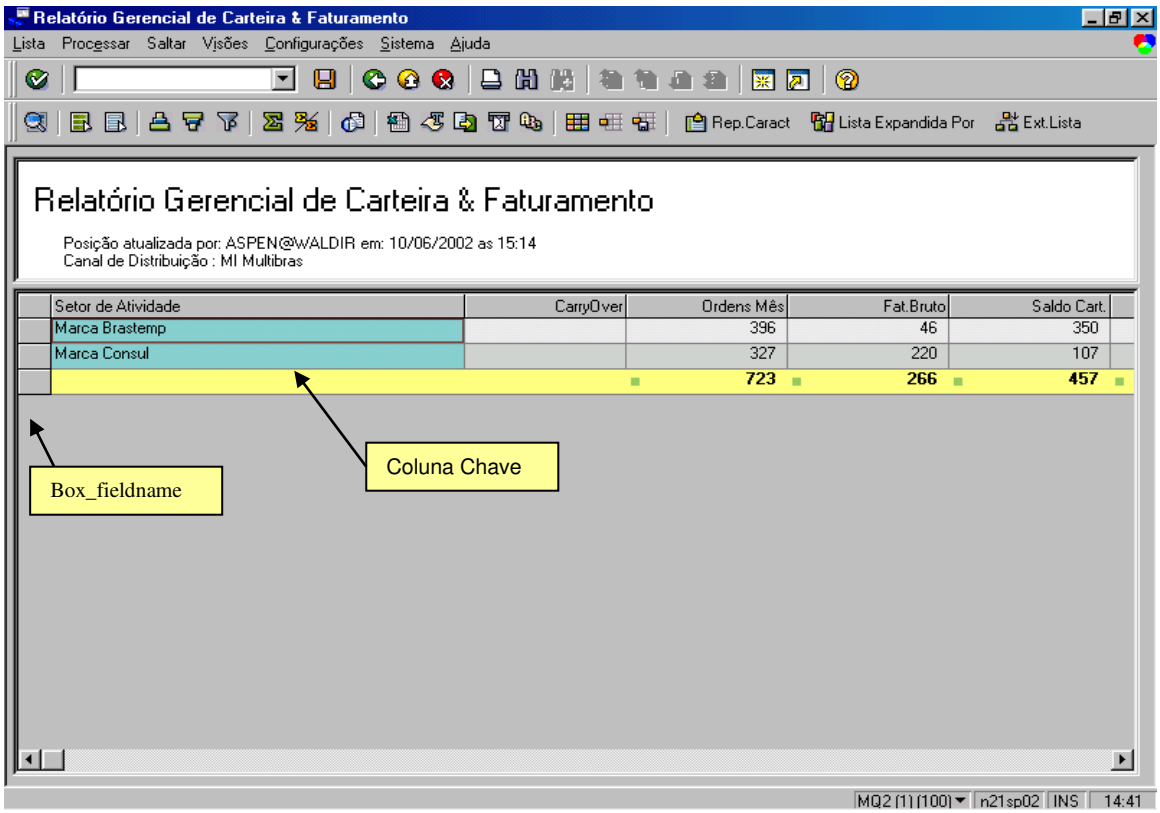


PERFORM f_layout. " Layout Geral da Lista

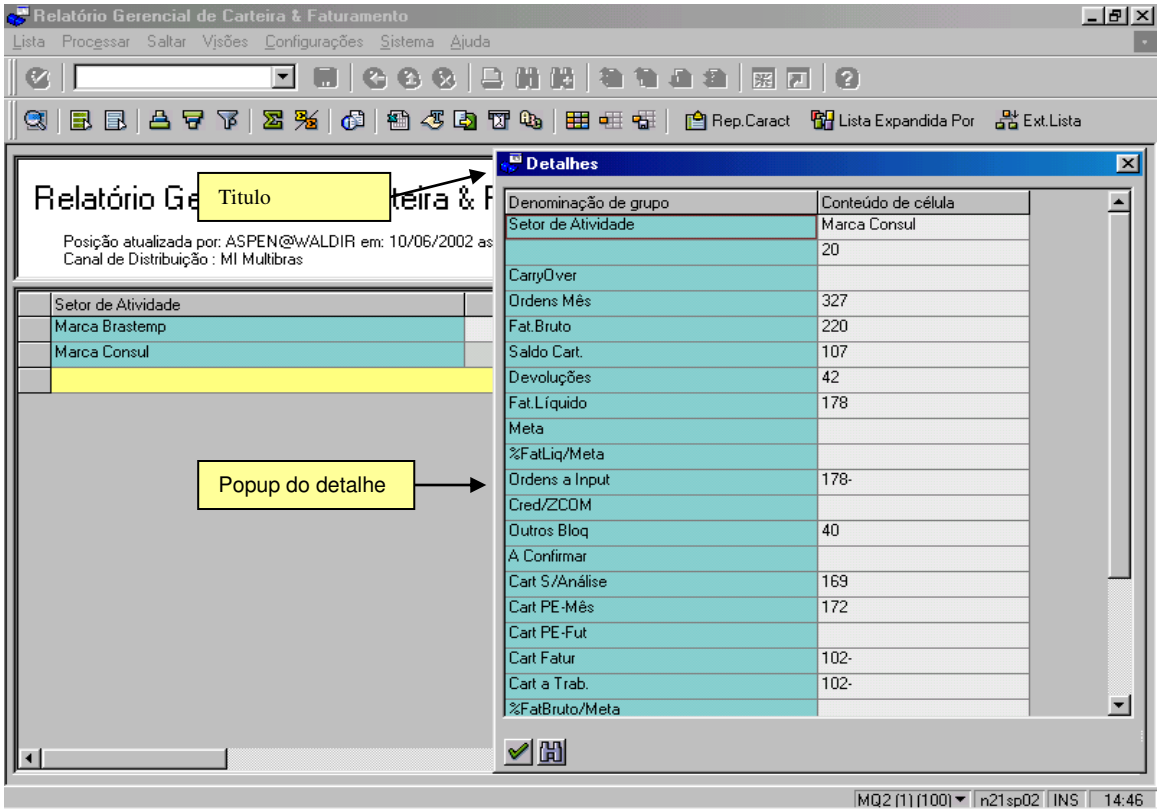
Neste FORM iremos formatar o relatório nos parâmetros gerais, ou seja, a aparëncia do relatório.
Veja abaixo os campos que podem ser formatados :

Campo	Descrição
Parâmetros Gerais	
no_colhead	Sem Títulos (Cabeçalho)
no_hotspot	Títulos sem Hotspot
no_vline	Colunas separadas por espaços
Zebra	Listrado (Uma linha clara outra escura)
cell_merge	Não suprimir a replicação de campo
Edit	Edição somente para o grid todo
edit_mode	Edição somente para o grid todo
numc_sum	Total para campos numéricos
no_input	Somente exibição de campos
f2code	
Reprep	
no_keyfix	Não fixar coluna chave
expand_all	Expandir todas as posições
no_author	Nenhuma verificação padrão da autoridade
PF-status	
def_status	Status Default
item_text	
Opções de Display	
colwidth_optimize	
no_min_linesize	Tamanho da linha = tamanho da lista
min_linesize	Default 80
max_linesize	Default 250
Window_titlebar	
no_uline_hs	
Exceções	
lights_fieldname	Nome do campo para exceção
lights_tabname	Nome da tabela para exceção
lights_rollname	
lights_condense	
Somatórios	
no_sumchoice	Sem escolha para Somar para cima
no_totalline	Sem Total Linha
no_subchoice	
no_subtotals	Sem Sub-Total
no_unit_splitting	
totals_before_items	Mostrar total antes dos itens
totals_only	Mostrar somente os totais
totals_text	Texto para a 1ª. coluna na linha de total
subtotals_text	Texto para a 1ª. coluna na linha de Sub-total
Interações	

box_fieldname	Nome do Campo para Checkbox
box_tabname	
box_rollname	
expand_fieldname	
hotspot_fieldname	Nome do Campo para Hotspot
confirmation_prompt	Confirmar Saída da lista
key_hotspot	keys as hotspot " K_KEYHOT
flexible_key	Mover as colunas chaves
group_buttons	Grupo de Botões
get_selinfos	Ler tela de seleção
group_change_edit	Settings by user for new group
no_scrolling	Sem movimentar tela
Detalhes Tela	
detail_popup	Mostrar detalhes em nova janela
detail_initial_lines	Mostrar somente as linhas iniciais
detail_titlebar	Título para tela de detalhes
Mostar Variantes	
header_text	Texto para o botão
default_item	
Cores	
info_fieldname	
Coltab_fieldname	Nome do campo que conterá as cores das colunas
Outros	
list_append	Sem chamada de tela
xifunckey	Extended interaction(SAPQuery)
xidirect	Extended INTeraction(SAPQuery)
dtc_layout	Configuração de layout para Tabstrip



TELA DE DETALHE



PERFORM f_sort. '' Ordenação/SubTotal

Neste FORM você deve informar os campos pelos quais a lista deve ser ordenada inicialmente, bem como se deve-se gerar sub-total ou total para esta quebra.

Campo Centro

```
t_sort-spos      = '1'.
t_sort-fieldname = 'WERKS'.
t_sort-tabname   = 'T_RELAT'.
t_sort-up        = 'X'.
t_sort-subtot    = 'X'.
APPEND t_sort.
CLEAR t_sort.
```

Segue abaixo os campos que podem ser formatados para cada campo de ordenação.

Campo	Descrição
Spos	Seqüência de Ordenação
fieldname	Nome do campo
Tabname	Nome da tabela a qual pertence o campo
Up	Ordenação Menor para Maior
Down	Ordenação Maior para Menor
Group	
Subtot	Gerar Sub Total
Comp	
Expa	
Obligatory	

PERFORM f_info_campos.

Neste FORM você irá definir o layout de cada coluna do relatório.
Veja abaixo os campos que podem ser formatados :

Campo	Descrição
row_pos	Saída na linha (1,2,3,...)
col_pos	Posição da coluna
fieldname	Nome do campo
tabname	Nome da tabela interna
currency	Moeda
cfieldname	Campo com Unidade de Moeda
ctabname	Tabela referência para Unidade Moeda
ifieldname	initial column
quantity	Unidade de Medida
qfieldname	Campo com Unidade Medida
qtabname	Tabela com Unidade Medida
round	Arredondar Campo
exponent	Expoente para floats
key	Campo como Chave
icon	Campo como Ícone
symbol	Campo como Símbolo
checkbox	Campo como Checkbox
just	Alinhamento – R (Direita) L (Esquerda) C (Centralizado)
lzero	leading zero
no_sign	Sem sinal
no_zero	Não imprimir campos zerados
no_convext	
edit_mask	Máscara de Edição
emphasize	Campo em destaque
fix_column	Fixar Coluna
do_sum	Totalizar Coluna
no_out	Não exibir o campo
tech	
outputlen	Tamanho do campo para saída dos dados
Offset	
seltext_l	Descrição Longa do Campo
seltext_m	Descrição Média do Campo
seltext_s	Descrição Curta do Campo
Ddictxt	Indicação de Qual descrição utilizar - (S)hort (M)iddle (L)ong
rollname	
datatype	Tipo do Campo
inttype	Tipo do Campo
intlen	Tamanho do Campo
lowercase	Letra Minúscula
ref_fieldname	Campo referência
ref_tabname	Tabela referência
roundfieldname	Nome campo para Arredondamento
roundtabname	Nome tabela
decimalsfieldname	Nome campo para Decimais
decimalstabname	Nome tabela
decimals_out	Número de decimais para escrever um número
text_fieldname	

reptext_ddic	Texto para a coluna
ddic_outputlen	Tamanho do Campo
key_sel	field not obligatory
no_sum	Não totalizar o campo
sp_group	Grupo
Reprep	
Input	Campo como input de dados
Edit	Uso interno
Hotspot	Campo como hotspot

PERFORM f_print.

Neste FORM você irá definir dados de impressão.

Campo	Descrição
prnt_info	Informações de Impressão
print	
prnt_title	
no_coverpage	
no_new_page	
reserve_lines	Linhas reservadas para o final da página
no_print_listinfos	Não imprimir página com número de registros selecionados
no_change_print_params	Não alterar tamanho de linha

Exibir Relatório

Neste FORM você irá exibir o relatório.

```
 v_repid = sy-repid.
 CALL FUNCTION 'K_KKB_SAVE_MODE_GET'
     IMPORTING
         e_save = v_save.

 CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
   EXPORTING
              I_CALLBACK_PROGRAM          = v_repid
              I_CALLBACK_PF_STATUS_SET  = 'F_SET_STATUS'
              I_CALLBACK_USER_COMMAND  = 'F_USER_COMMAND'
              IS_LAYOUT                   = w_layout
              IS_LAYOUT                        = w_print
              IT_FIELDCAT                 = t_campos[]
              IT_SORT                          = t_sort[]

              I_DEFAULT                        = 'X'
              I_SAVE                           = v_save
              IS_VARIANT                  = variante
              IT_EVENTS                   = t_event[]
   TABLES
              t_outtab                         = t_relat
   EXCEPTIONS
              PROGRAM_ERROR                    = 1
              OTHERS                           = 2.

 IF sy-subrc <> 0.
   MESSAGE ID sy-msgid TYPE 'I' NUMBER sy-msgno
         WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
   STOP.
 ENDIF.
```

14.3 Como mudar cor de uma coluna

Para mudar a cor de uma coluna, você deve definir na sua tabela interna final (que será utilizada na função de exibição do relatório) o campo COLINFO (ou o nome que você desejar) que seja do tipo kkblo_t_specialcol.

Este campo será uma tabela interna onde você definirá para cada linha do relatório a cor de cada coluna. Isso deve ser feito no momento em que você estiver gerando a tabela final para a impressão.

Definir o campo na tabela interna final

```
DATA : BEGIN OF t_relat OCCURS 0,
                    Campo1
                    Campo2
                    Campo3
                    ......
                    ......
                    COLINFO TYPE kkblo_t_specialcol
       END OF t_relat.
```

Definir uma workarea auxiliar : DATA: colinfo TYPE kkblo_specialcol.

```
CLEAR colinfo.
```

Nome da coluna a ser colorida

```
colinfo-fieldname = 'ATREXP'.
```

De acordo com o valor do campo a cor será diferente

```
IF t_relat-atrexp < 0.
   colinfo-color-col = '3'.    " Amarelo
ELSEIF t_relat-atrexp = 0.
   colinfo-color-col = '5'.    " Verde
ELSE.
  colinfo-color-col = '6'.    " Vermelho
ENDIF.

colinfo-color-int = '1'.       " Intenso
APPEND colinfo TO t_relat-colinfo.
```

Os campos que devem ser formatados são :

- FIELDNAME = Nome do campo na tabela interna
- COLOR-COR = Número da cor
- COLOR-INT = Intensidade da cor
  - 1 : Intenso
  - Branco : cor normal

14.4 Como mudar a cor de uma linha

Para mudar a cor da linha toda e não somente de uma coluna você deve seguir o mesmo procedimento da coluna porém fazer para todas as colunas do relatório.

Por exemplo sua tabela tem 5 campos então :

Definir a tabela interna final

```
DATA : BEGIN OF t_relat OCCURS 0,
                    Campo1
                    Campo2
                    Campo3
                    Campo4
                    Campo5
                    COLINFO TYPE kkblo_t_specialcol
       END OF t_relat.
```

Definir uma workarea auxiliar : DATA: colinfo TYPE kkblo_specialcol.

CLEAR colinfo.

Todas as colunas deverão ter a mesma cor e intensidade

```
colinfo-color-col = '3'.    " Amarelo
colinfo-color-int = '1'.    " Intenso
```

1a. Coluna

```
colinfo-fieldname = 'CAMPO1'.
APPEND colinfo TO t_relat-colinfo.
```

2ª. Coluna

```
colinfo-fieldname = 'CAMPO2'.
APPEND colinfo TO t_relat-colinfo.
```

3ª. Coluna

```
colinfo-fieldname = 'CAMPO3.
APPEND colinfo TO t_relat-colinfo.
```

4ª. Coluna

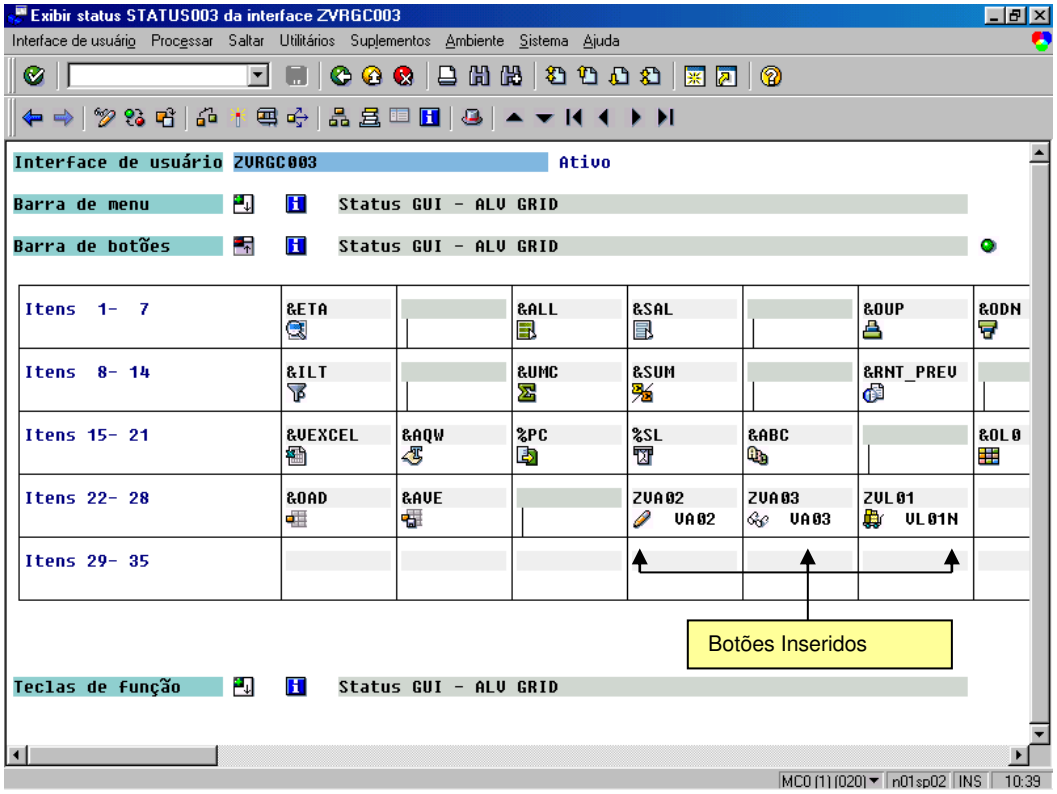
```
colinfo-fieldname = 'CAMPO4.
APPEND colinfo TO t_relat-colinfo.
```

5ª. Coluna

```
colinfo-fieldname = 'CAMPO5.
APPEND colinfo TO t_relat-colinfo.
```

14.5 Como utilizar outros botões na tela

Para inserir novos botões na tela, você deve copiar o Status Gui Standard STANDARD_FULLSCREEN para outro a ser utilizado no seu relatório. Após a cópia você deve retirar botões que não serão utilizados e inserir os seus novos botões :



O tratamento destes novos botões deverá ser no FORM F_USER_COMMAND especificado na chamada da função de exibição do relatório :

```
        CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
form f_user_command USING p_ucomm LIKE sy-ucomm
                    p_selfield TYPE slis_selfield.

          CASE p_ucomm.
                * Modificar Documento
                  WHEN 'ZVA02'.
                              CHECK p_selfield-fieldname = 'VBELN'.
                              SET PARAMETER ID 'AUN' FIELD p_selfield-value.
                              CALL TRANSACTION 'VA02' AND SKIP FIRST SCREEN.
                * Exibir Documento
                  WHEN 'ZVA03'.
                              CHECK p_selfield-fieldname = 'VBELN'.
                              SET PARAMETER ID 'AUN' FIELD p_selfield-value.
                              CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.
                * Exibir Remessa
                  WHEN 'ZVL01'.
                               CHECK p_selfield-fieldname = 'VBELN'.
                               READ TABLE t_relat INDEX p_selfield-tabindex.
                               IF sy-subrc EQ 0.
                                 SET PARAMETER ID 'AUF'   FIELD t_relat-vbeln.
                                 SET PARAMETER ID 'VST'   FIELD t_relat-vstel.
                                 SET PARAMETER ID 'LEDAT' FIELD t_relat-mbdat.
                                 CALL TRANSACTION 'VL01N' AND SKIP FIRST SCREEN.
                               ENDIF.
                  WHEN OTHERS.
          ENDCASE.

endform.                  " f_user_command
```

14.6 Programas Standard - Modelo

BALVST02_GRID – Programa teste visor de listas ABAP: lista simples modelo vôo.
BALVST03_GRID - Programa teste visor de listas ABAP: lista simples modelo vôo.
BALVHT01 - Programa de teste ALV: lista seqüencial hierárquica modelo de vôo.

Na versão 4.6 todos os programas BCALV*