

SAP R/3

ABAP/4

Comando/Função	Descrição	Pg
ADD	Adiciona um valor a um outro campo numérico	6
APPEND	Gravar em tabela interna	6
APPEND LINES	Copia um parte de uma tabela interna	32
APPEND SORTED BY	Usado para classificar tabelas em ordem durante o APPEND	31
AT END OF	Identifica o ultimo registro de uma seqüência (quebra)	6
AT EXIT-COMMAND	Usado para definir a execução de módulos com botões	6
AT FIRST	Identifica o primeiro registro da tabela percorrida	6
AT LAST	Identifica o ultimo registro da tabela percorrida	7
AT LINE-SELECTION	Evento para abrir uma segunda tela dentro de um relatório (drew down)	7
AT NEW	Identifica o primeiro registro de uma seqüência (quebra)	7
AT SELECTION-SCREEN ON	Analisar o resultado de uma variável	7
AT USER-COMMAND	Evento para inicializar os botões criados	8
AUTHORITY-CHECK OBJECT	Para verificar a autorização de objeto	8
AS TEXT	Usado para classificar caracteres acentuados	32
ASSING f1 TO <f>	Atribuição Dinamica	31
AS SYMBOL AS ICON AS LINE	Comandos usados para formatação de relatórios.	32
CALL TRANSACTION	Comando usado para chamar transações	8
CASE .. WHEN .. ENDCASE	Comando para tomadas de desição	9
CHECK	Substitui o comando IF	9
CLEAR	Limpar o conteúdo de uma variável	9
CLOSE DATASET	Fecha um arquivo externo	9
CLOSE_FORM	Função usada para fechar um formulário SAPSCRIPT	9
COLLECT	Organiza a tabela com todas as descrições a esquerda totalizando os campos numéricos	10
COMMIT WORK	Efetiva as alterações no banco	10
CONCATENATE	Agrupa vários campos em um só	10
CONSTANTS	Criar constantes para serem usadas em todo o programa	10
CONTINUE	Volta ao inicio de um loop após a verificação de uma condição	10
CONDENSE	Remove os espaços em branco em um texto	
Copiando dados de uma tabela interna para outra tabela		32
DATA	Criar tabelas internas e variáveis	11
DELETE	Deleta um registro de uma tabela interna	11
DESCRIBE TABLE	Determina o numero de linhas de uma tabela interna	33
DO .. ENDDO	Usado para criar loop	11
DOWNLOAD	Função para download	11
END_FORM	Função usada para finalizar um formulário SAPSCRIPT	12
END-OF-PAGE	Marca o final de uma cabeçalho	12
END-OF-SELECTION	Evento para marcar o fim das seleções de dados	12
EXIT	Comando usado para sair de um laço ou sub-rotina	12

Comando e Funções em ABAP/4

FORM	Indica o início de uma sub-rotina	13
FORMAT COLOR	Usado para formatar a saída do comando WIRTE (cor)	12
FREE	Libera espaço utilizado por uma tabela interna na memória	13
FUNÇÕES MATEMATICAS	Diversas funções matemáticas e operadores	13
GRAPH_MATRIX_3D	Função pra criar gráficos	13
GUI_DOWNLOAD	Função para fazer download de arquivos	14
HIDE	Usado para definir campos para uma segunda janela em relatórios	15
ID .. FIELD ..	Objetos de autorização	15
IF .. ELSE .. ENDIF.	Comando de desição	15
IF NOT .. IS INITIAL	Verifica se uma variável ou head liner esta vazio	15
INITIALIZATION	Evento inicializar as variáveis	16
INSERT	Insere o conteúdo de uma variável em uma tabela transparente	16
INSERT LINES	Inserir linhas dentro de uma tabela interna de uma outra tabela.	33
LEAVE PROGRAM	Comando que sai do programa. Termina a execução	16
LEAVE TO SCREEN	Comando que direciona a ida de uma tela a outra	16
LINE-COUNT	Numero de linhas por pagina	16
LINE-SIZE	Numero de colunas por pagina	16
LOOP AT	Ler o conteúdo de uma tabela	16
MESSAGE	Usado para mostrar mensagens na tela	17
MESSAGE-ID	Usado para definir uma classe de mensagens	17
MODIFY	Modifica o conteúdo de uma tabela interna	17
MODULE	Usado para a criação de módulos de programa	17
MOVE	Move o conteúdo de uma tabela para outra	17
MOVE-CORRESPONDING	Move apenas os campos com o mesmo nome	17
NEW-PAGE	Inicia nova pagina em um relatório	18
NO STANDARD PAGE HEADING	Ignora o cabeçalho STD do SAP	18
OPEN DATASET	Para abrir arquivo externo (Somente para UNIX)	18
OPEN_FORM	Função para abrir um formulário SAPSCRIPT	19
OPERADOS DE STRING DE CARACTERES		33
PARAMETERS	Criar parâmetros de seleção	19
PERFOM	Usado para chamar sub-rotinas	20
POPUP_TO_CONFIRM_WITH_MESSAGE	Função para abrir uma janela de dialogo com opções de seleção	20
READ DATASET .. INTO ..	Usado para ler e importar de um arquivo externo	20
READ TABLE	Ler uma tabela interna	20
READ_TEXT	Função para ler texto de uma tabela	21
REFRESH	Limpa todo o conteúdo de uma tabela interna	21
REPLACE	Replica um texto dentro de um outro texto	21
REPORT	Inicio de relatório	21
RESERVE	Reserva linhas para quebra de pagina	21
SAPGUI_PROGRESS_INDICATOR	Função para indicar o progresso de uma tarefa	21
SEARCH	Procura um texto dentro de outro texto	22
SELECT	Usado para selecionar dados de uma tabela	22
SELECTION-OPTIONS	Para colocar as opções de seleção	23
SELECTION-SCREEN	Criar um BOX na janela	23
SET PF-STATUS	Comando usado para criar botões	23
SET TITLEBAR	Criar títulos em janelas	23

Comando e Funções em ABAP/4

SET PARAMETERS	Passa parâmetros para chamada de função	23
SKIP	Pular uma linha	23
SORT	Ordenar o conteúdo de uma tabela interna	24
SPLIT .. AT .. INTO	Comando usado para ler arquivo externo e gravar na tabela	24
STANDARD TOOLBAR	Define os valores padrões para os botões do R/3	24
START_FORM	Função para inicializar um formulário SAPSCRIPT	25
START-OF-SELECTION	Evento para marcar o início das seleções de dados	25
STRLEN	Volta o tamanho de um texto	25
SUBTRACT	Subtrai um valor numérico de outro valor	26
SUM	Totaliza um campo numérico	26
SXPG_CALL_SYSTEM	Função que executa comandos externos no servidor	27
SY-BATCH	Indica a execução em Background	27
SY-DATUM	Volta a data do sistema	27
SY-LANGU	Retorna a linguagem do logon	27
SY-LINNO	Linha corrente de um relatório	27
SY-LISEL	Linha escolhida em relatórios interativos	27
SY-MANDT	Volta o valor do mandante	27
SY-PAGNO	Faz a paginação automática (nr de paginas)	27
SY-SUBRC	Volta o resultado de uma pesquisa	27
SY-TABIX	Linha corrente de uma tabela interna	27
SY-TVAR0 .. SY-TVAR9	Elementos de textos, para títulos de relatórios	27
SY-UCOMM	Variável do sistema que retorna o nome de um botão ou evento	27
SY-UNAME	Retorna o nome do Usuário	27
SY-UZEIT	Retorna a hora do sistema	27
SY-VLINE	Serve para fazer o fechamento de bordas de um relatório	27
SY-ULINE	Imprime uma linha com n posições	27
SY-CPROG	Nome do programa principal	27
SY-TCODE	Código da transação	27
SY-DBCNT	Contador de interação para SELECT	27
TABLES	Usado para abrir tabelas	27
TOP-OF-PAGE	Marcar o início de uma cabeçalho	28
TRANSFER .. TO ..	Usado para ler e gravar em arquivo externo (UNIX)	28
TRANSLATE	Transforma texto minúsculo em maiúsculo	28
TYPES	Define tipos e tabelas interna	28
ULINE	Coloca uma linha continua na tela	28
UNPACK	Coloca zeros a frente de um numero	29
UPDATE	Atualiza uma tabela transparente	29
VARYING	Obtem os componentes de uma string de campo e sequencia.	34
WHILE .. ENDWHILE	Cria laços de execução	29
WRITE	Saída de relatório	29
WRITE_FORM	Função para escrever no formulário SAPSCRIPT	30
WS_UPLOAD	Função para fazer UPLOAD de arquivo externo	31

Exemplo de programas usando comandos e funções

Programa 1	WRITE FORMAT SKIP ULINE	35
------------	----------------------------------	----

Programa 2	TABLES SELECT DATA REFRESH CLEAR LOOP LINE-SIZE LINE-COUNT PARAMETERS SORT READ TABLE	36
Programa 3	SELECTION-SCREEN SELECTION-OPTIONS SELECT INTO TABLE SELECT INTO TABLE INNER JOIN INITIALIZATION START-OF-SELECTION END-OF-SELECTION	38
Programa 4	NO STANDARD PAGE HEADING AT SELECTION-SCREEN ON TOP-OF-PAGE END-OF-PAGE MESSAGE-ID MESSAGE SY-SYBRC CONCATENATE	40
Programa 5	APPEND AT END OF SY-DATUM SY-PAGNO AT LINE-SELECTION IF NOT IS INITIAL SET TILTEBAR SY-ULINE SY-VLINE HIDE SUM	43
Programa 6	DATA (INCLUDE STRUCTURE) CONSTANTS PARAMETERS RLGRAP-FILENAME WS_UPLOAD SY-MANDT SPLIT UNPACK INSERT MOVE-CORRESPONDING COMMIT WORK	47
Programa 7	CALL TRANSACTION	50
Programa 8	PARAMETERS RADIOBUTTON GUI_DOWNLOAD MESSAGE xxx WITH TRANSFER READ DATASET CLOSE DATASET OPEN DATASET	55
Programa 9	SXPG_CALL_SYSTEM	61
Programa 10	GRAPH_MATRIX_3D	62

Programa 11	SET PF-STATUS AT USER-COMMAND SY-UCOMM DOWNLOAD	64
Programa 12 – Modulo On-line	CASE .. WHEN LEAVE PROGRAM LEAVE TO SCREEN MODULE AT EXIT COMMAND	66
Programa 13	AUTHORITY-CHECK OBJECT ID xxx FIELD	67
Programa 14	USO DE ALV	72

Criar Transação	78
Module POOL – ON LINE	81
Criar Janelas	91
Objetos de Autorização	98
USER-EXIT	102
FIELD-EXIT	107
Criar Tabelas	112
Principais Tabelas Genéricas	118
SapScript – Etapas de criação	121

ADD var1 TO var2	Comando usado adicionar um valor a outro campo numérico.
<p>DATA wa TYPE I VALUE 3. DATA wb TYPE I VALUE 4.</p> <p>ADD wa TO wb.</p> <p>WRITE wa. " retorna 3 WRITE wb. " retorna 7</p>	
APPEND t_tabela	Comando usado para gravar em tabela interna
APPEND t_tabela.	
AT END OF campo	Em conjunto com o comando LOOP, identifica o ultimo registro de uma seqüência.
<p>SORT BY campo. LOOP AT t_tabela. AT END of campo. " antes de mudar de campo (valor do campo) é que é realizada a " verificação ENDAT. ENDDLOOP.</p>	
AT EXIT-COMMAND	Usado para definir se será executado um modulo dentro das rotinas on-line. Tratativa de botões.
<p>MODULE nome_do_modulo AT EXIT-COMMAND. (*) MODULE nome_do_modulo.</p> <p>(*) Se os botões STANDARD TOOLBAR estiverem definidos com o tipo E, será executado este modulo, antes do modulo seguinte.</p>	
AT FIRST	Em conjunto com o comando LOOP, identifica o primeiro registro de uma tabela percorrida.
<p>SORT BY campo. LOOP AT t_tabela. AT FIRST. " entra nesta rotina ao iniciar o LOOP, somente no primeiro registro. ENDAT. ENDDLOOP.</p>	

AT LAST	Em conjunto com o comando LOOP, identifica o ultimo registro de uma tabela percorrida.
<p>SORT BY campo. LOOP AT t_tabela. AT LAST. "entra nesta rotina ao finalizar o LOOP, somente no ultimo registro. ENDAT. ENDDLOOP.</p>	
AT LINE-SELECTION	Evento usado para abrir uma segunda tela em um relatório (drill down)
<p>AT LINE-SELECTION. IF NOT T_MARC IS INITIAL. SET TITLEBAR 'DET'. WRITE SY-ULINE(83). WRITE : 'Material : '. WRITE : T_MARC-MATNR,' - ' ENDIF.</p>	
AT NEW campo	Em conjunto com o comando LOOP, identifica o primeiro registro de uma seqüência (quebra)
<p>SORT BY campo. LOOP AT t_tabela. AT NEW campo. " entra nesta rotina ao mudar de campo. ENDAT. ... ENDDLOOP.</p>	
AT SELECTION-SCREEN ON variável	Verifica o resultado de uma variável de seleção (conteúdo)
<p>SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001. SELECT-OPTIONS s_material FOR T_MARC-MATNR. SELECTION-SCREEN END OF BLOCK b_janela.</p> <p>AT SELECTION-SCREEN ON s_material.</p>	

AT USER COMMAND	Evento usado para inicializar botões criados pelo programa
<p>SET PF-STATUS 'BOTA0'.</p> <p>INITIALIZATION.</p> <p>AT USER-COMMAND.</p> <p>IF sy-ucomm = 'DOWNLOAD'. (*)</p> <p> PERFORM F_DOW_NOVO.</p> <p>ENDIF.</p> <p>* variável de sistema. retorna o nome do botão (DOWNLOAD)</p>	
AUTHORITY-CHECK OBJECT objeto	Usado para verificar a autorizações de objetos
<p>AUTHORITY-CHECK OBJECT 'ZZC0307Y' (*)</p> <p>ID 'actvt' FIELD 'DUMMY'. "Objetos da autorização</p> <p>IF SY-SUBRC NE 0.</p> <p> MESSAGE E006 WITH TEXT-E01.</p> <p>ENDIF.</p> <p>(*) comando para verificar autorização de objeto, usar junto com SU21 e SU24</p>	
CALL TRANSACTION transação USING t_tabela MODE tipo UPDATE tipo2 MESSAGES INTO t_mensagens	Comando usado para chamar transações
<p>* parâmetro tipo</p> <p>* A = DISPLAY</p> <p>* E = ERROS</p> <p>* N = NÃO MOSTRAR</p> <p>* P = NÃO MSOTRAR / DEBUGAR</p> <p>* parâmetro tipo2</p> <p>* A = ASSINCRONO</p> <p>* S = SINCRONO</p> <p>* L = LOCAL UPDATE</p> <p>CALL TRANSACTION 'XK01' USING T_BDC MODE 'A' UPDATE 'S' MESSAGES INTO T_MSG.</p> <p>* simula um ENTER passando pela primeira tela</p> <p>* SET PARAMETER ID 'NN' FILED valor do campo para chamar transações passando</p> <p>* parâmetros</p> <p>SET PARAMETERS ID 'VF' FIELD t_alv-vbeln.</p> <p>CALL TRANSACTION 'VF03' AND SKIP FIRST SCREEN.</p>	

CASE .. WHEN .. ENDCASE.	Comando usado para controle de variáveis
<pre> CASE vl_okcode. WHEN 'SIM'. ... WHEN 'NAO'. ... WHEN OTHERS. ... ENDCASE.</pre>	
CHECK var	Semelhante ao comando IF. Se a condição for verdadeira continua logo após a verificação, se não aborta a execução.
<pre> FORM dados. LOOP AT t_tabela. t_tabela-campo = KUNNR. CHECK t_tabela-campo = '0001'. " se o conteúdo do campo for igual a '0001', continua t_tabela-campo = '0002'. " executando a instrução abaixo, se for diferente aban APPEND t_tabela. "dona o loop e sai do form. ENDDLOOP. ENDFORM.</pre>	
CLEAR var	Limpa o conteúdo de uma variável ou tabela
<pre> CLEAR var. CLEAR t_tabela. " limpa o header line</pre>	
CLOSE DATSET	Fecha um arquivo externo
<pre> CLOSE arquivo.</pre>	
CLOSE_FORM	Função para fechar um SAPSCRIPT
<pre> CALL FUNCTION 'CLOSE_FORM' IMPORTING RESULT = t_result " tabela interna EXCEPTIONS UNOPENED = 1 BAD_PAGEFORMAT_FOR_PRINT = 2 SEND_ERROR = 3 SPOOL_ERROR = 4 CODEPAGE = 5 OTHERS = 6. IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	

COLLECT t_tabela	Organiza a tabela interna com todas as descrições a esquerda totalizando os campos numericos
<p>COLLECT t_tabela.</p> <p>Exemplo :</p> <p>Tabela-nome = 'JOAO' . , Tabela-uf = 'SP'. , Tabela-valor = 100. COLLECT tabela.</p> <p>Tabela-nome = 'JOAO' . ,Tabela-uf = 'RJ'. , Tabela-valor = 300. COLLECT tabela.</p> <p>Tabela-nome = 'JOAO' . , Tabela-uf = 'SP'. , Tabela-valor = 500. COLLECT tabela.</p> <p>Resultado : JOAO SP 600, JOAO RJ 300.</p>	

COMMIT WORK	Efetiva as alterações no Banco de dados
COMMIT WORK.	

CONCATENATE var1 var2 INTO var3	Agrupa varias variáveis em uma única variável
CONCATENATE 'abc' 'XXX' INTO var3. " var3 = abcXXX	

CONSTANTS ct1 TYPE tp VALUE x	Criar constantes que podem ser usadas em todo o programa
CONSTANTS c_tipo TYPE C VALUE ';'.	

CONTINUE	Volta ao inicio de um loop após a verificação de uma condição.
<p>FORM dados.</p> <p>DO condição.</p> <p> t_tabela-campo = KUNNR.</p> <p> IF t_tabela-campo = '0001'.</p> <p> CONTINUE. " se o conteúdo do campo for igual a '0001', volta ao inicio</p> <p> ELSE. " do loop.</p> <p> EXIT.</p> <p> ENDIF.</p> <p>ENDDO.</p> <p>ENDFORM.</p>	

CONDENSE var	Remove os espaços em branco em um texto
<p>Var = ' Dr.'.</p> <p>CONDENSE var.</p>	
DATA var TYPE tp. DATA var LIKE tp.	Define tabelas internas e variáveis.

DATA : BEGIN OF t_tabela	
<p>* CRIAR UMA TABELA EM INTERNA COM BASE EM CAMPOS DE UMA TABELA STD DATA : BEGIN OF T_001W OCCURS 0, WERKS LIKE T001W-WERKS, NAME1 LIKE T001W-NAME1, END OF T_001W.</p> <p>* CRIAR UMA VARIÁVEL DATA wcor TYPE C.</p> <p>* CRIAR UMA TABELA INTERNA COPIANDO A ESTRUTURA DA TABELA STD DATA : BEGIN OF t_tempo OCCURS 0. INCLUDE STRUCTURE ztab1_01. DATA : END OF t_tempo.</p>	
DELETE t_tabela	Deleta um registro de uma tabela interna ou transparente
DELETE t_tabela.	
DO condição .. ENDDO.	Cria um laço e verifica a condição no início do laço
DO condição. " a condição é verificada no início do laço e se for verdadeira continua dentro " do loop ENDDO.	
DOWNLOAD	Função usada para fazer download de arquivos, nesta função podemos escolher o formato de arquivo e temos a janela de menu abrir
CALL FUNCTION 'DOWNLOAD' * EXPORTING TABLES DATA_TAB = T_001L * FIELDNAMES = EXCEPTIONS INVALID_FILESIZE = 1 INVALID_TABLE_WIDTH = 2 INVALID_TYPE = 3 NO_BATCH = 4 UNKNOWN_ERROR = 5 GUI_REFUSE_FILETRANSFER = 6 OTHERS = 7 . IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.	

END_FORM	Função usada para finalizar um SAPSCRIPT
<pre>CALL FUNCTION 'END_FORM' IMPORTING RESULT = t_result EXCEPTIONS UNOPENED = 1 BAD_PAGEFORMAT_FOR_PRINT = 2 SPOOL_ERROR = 3 CODEPAGE = 4 OTHERS = 5 .</pre> <pre>IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	
END-OF-PAGE	Marca o final de um cabeçalho de relatório
<pre>TOP-OF-PAGE. WRITE 'NOVO CABECALHO DEFINIDO PELO PROGRAMA'. END-OF-PAGE.</pre>	
END-OF-SELECTION	Evento usado para marcar o fim das seleções de dados.
<pre>START-OF-SELECTION. SELECT marc~matnr marc~werks t001w~name1 makt~maktx INTO TABLE t_marc FROM marc INNER JOIN t001w ON marc~werks = t001w~werks INNER JOIN makt ON marc~matnr = makt~matnr and makt~spras = sy-langu WHERE t001w~werks IN S_CENTRO AND marc~matnr IN S_MATERI. END-OF-SELECTION.</pre>	
EXIT	Comando usado para sair de um laço ou sub-rotina
<pre>FORM dados. DO condição. t_tabela-campo = KUNNR. IF t_tabela-campo = '0001'. CONTINUE. ELSE. EXIT. " força a saída do loop e sai do form ENDIF. ENDDO. ENDFORM.</pre>	

FORM	Indica o início de uma sub-rotina
FORM dados. . . . ENDFORM.	

FORMAT COLOR cor	Usado para formatar a saída do comando WRITE
FORMAT COLOR COL_TOTAL. WRITE 'QUALQUER COISA'. FORMAT COLOR COL_KEY. WRITE / 'QUALQUER COISA'. FORMAT COLOR COL_BACKGROUND. " VOLTA AO PADRÃO	
FREE t_tabela	Libera espaço utilizado por uma tabela interna na memória
FREE t_tabela.	

Funções matemáticas e operadores lógicos	
** Exponenciação DIV Divisão de inteiro MOD Resto da divisão de inteiro	

GRAPH_MATRIX_3D	Função para criar gráficos
CALL FUNCTION 'GRAPH_MATRIX_3D' EXPORTING COL1 = 'TV' " nome das colunas COL2 = 'VIDEO CASSETE' COL3 = 'RADIO' COL4 = 'DVD' * COL5 = '' * COL6 = '' DIM1 = 'PRODUTOS' " eixo X,Y DIM2 = 'VENDEDOR' INFORM = '6' TITL = 'VENDAS 2003' " TITULO DO GRAFICO VALT = 'QTD' " valor do eixo WINPOS = '5' " posição inicial WINSZX = '5' WINSZY = '70' TABLES DATA = t_vendedor " tabela com o nome dos vendedores OPTS = t_tabela . " tabela de valores	

GUI_DOWNLOAD	Função para download de arquivos
<pre> CALL FUNCTION 'GUI_DOWNLOAD' EXPORTING * BIN_FILESIZE = FILENAME = V_ARQUIVO "nome do arquivo que será criado FILETYPE = 'ASC' * IMPORTING * FILELENGTH = TABLES DATA_TAB = T_MARD " tabela com os dados para exportação EXCEPTIONS FILE_WRITE_ERROR = 1 NO_BATCH = 2 GUI_REFUSE_FILETRANSFER = 3 INVALID_TYPE = 4 NO_AUTHORITY = 5 UNKNOWN_ERROR = 6 HEADER_NOT_ALLOWED = 7 SEPARATOR_NOT_ALLOWED = 8 FILESIZE_NOT_ALLOWED = 9 HEADER_TOO_LONG = 10 DP_ERROR_CREATE = 11 DP_ERROR_SEND = 12 DP_ERROR_WRITE = 13 UNKNOWN_DP_ERROR = 14 ACCESS_DENIED = 15 DP_OUT_OF_MEMORY = 16 DISK_FULL = 17 DP_TIMEOUT = 18 FILE_NOT_FOUND = 19 DATAPROVIDER_EXCEPTION = 20 CONTROL_FLUSH_ERROR = 21 OTHERS = 22 . IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF. </pre>	

HIDE	Usado para definir campos para uma segunda janela em relatórios.
<pre> SORT T_MARC BY matnr werks lgort. LOOP AT T_MARC. WRITE /5 T_MARC-MATNR. WRITE 18 T_MARC-MAKTX. WRITE 60 T_MARC-WERKS. WRITE 80 T_MARC-NAME1. WRITE 120 T_MARC-LGORT. WRITE 140 T_MARC-LFGJA. WRITE 150 T_MARC-LFMON. WRITE 160 T_MARC-LABST. * HIDE com os campos que queiramos mostrar na nova janela. Deve ser colocado depois de imprimir linha HIDE :T_MARC-MATNR,T_MARC-MAKTX,T_MARC-WERKS, T_MARC-NAME1. ENDLOOP.</pre>	
ID .. FIELD ..	Objetos de autorização. Serve para selecionarmos quais objetos dentro de uma classe de autorização que iremos tratar.
<pre> AUTHORITY-CHECK OBJECT 'ZZC0307Y' (*) ID 'actvt' FIELD 'DUMMY'. "Objetos da autorização IF SY-SUBRC NE 0. MESSAGE E006 WITH TEXT-E01. ENDIF. (*) comando para verificar autorização de objeto, usar junto com SU21 e SU24</pre>	
IF .. ELSE .. ENDIF.	Comando de desição.
<pre> IF var = 'A'. ELSE. ENDIF.</pre>	
IF NOT .. IS INITIAL.	Verifica se uma variável ou tabela interna esta vazia.
<pre> IF NOT var IS INITIAL. ENDIF.</pre>	

INITIALIZATION.	Evento para inicializar as variáveis. Deve ser o primeiro evento a ser colocado no programa. Não é um evento obrigatório.
<p>TABLES</p> <p>DATA</p> <p>.....</p> <p>INITIALIZATION.</p>	
INSERT t_tabela_transparente.	Insere o conteúdo de uma variável em uma tabela transparente
<p>INSERT tabela.</p>	
LEAVE PROGRAM.	Comando usado para sair de um programa on-line.
<p>LEAVE PROGRAM.</p>	
LEAVE TO SCREEN nnnn.	Comando usado para mudar de tela dentro de programas on-line.
<p>LEAVE TO SCREEN 9000. " 9000 , numero da tela</p>	
LINE-COUNT nnn. LINE-SIZE nnn.	Numero de linhas por pagina. Numero de colunas por pagina.
<p>* Estes comandos são usados dentro do comando REPORT no inicio do programa.</p> <p>REPORT LINE-COUNT 60 LINE-SIZE 130.</p>	
LOOP AT t_tabela.	Comando usado para ler uma tabela interna.
<p>LOOP AT t_tabela.</p> <p>.</p> <p>.</p> <p>ENDDLOOP.</p>	

MESSAGE-ID nnn.	Comando usado para definir classe de mensagens.
MESSAGE mmm.	Comando usado para chamar mensagens.

* MESSAGE-ID nnn deve ser usado junto com o comando REPORT, no início do programa
 * Existem 3 tipos de mensagens : W-> de avisos, E-> de erros, S-> de sucesso

REPORT LINE-COUNT 80 LINE-SIZE 60 MESSAGE-ID Z01.

MESSAGE W001. " mensagem com avisos (warnigs)
 MESSAGE E002. " mensagem com erros.
 MESSAGE S003. " mensagem de sucesso.

* Também podemos passar parâmetros para as mensagens.
 * Na criação da mensagem na classe de mensagens devemos colocar o símbolo & para indicar
 * a passagem de um parâmetro.

MESSAGE S003 WITH parâmetro.

MODIFY t_tabela. MODIFY t_tabela TRANSPORTING campos	Comando usado para modificar uma tabela interna.
---	--

MODIFY t_tabela. " modifica toda a linha da tabela
 MODIFY t_tabela TRANSPORTING campos " modifica os campos selecionados

MODULE	Comando usado criar módulos em programas on-line.
---------------	---

MODULE nome_modulo.

MOVE	Move todo o conteúdo de uma tabela para outra. Os campos devem ser os mesmos.
-------------	--

LOOP AT t_tabela.
 MOVE tabela.
 INSERT tabela.
 ENDLOOP.
 COMMIT WORK.

MOVE-CORRESPONDING t_tabela TO tabela	Move apenas os campos correspondente entre 2 tabelas.
--	---

LOOP AT t_tabela.
 MOVE-CORRESPONDING t_tabela TO tabela.
 INSERT tabela.
 ENDLOOP.
 COMMIT WORK.

NEW-PAGE	Inicia uma nova pagina em um relatório.
NEW-PAGE.	
NO STANDARD PAGE HEADING	Ignora o cabeçalho STD do R/3
<p>* Deve ser usado em conjunto com o comando REPORT</p> <p>REPORT LINE-COUNT 80 LINE-SIZE 60 MESSAGE-ID Z01 NO STANDARD PAGE HEADING.</p>	
OPEN DATASET arquivo FOR OUTOUT/INPUT IN TEXT MODE parametro.	Comando usado para abrir um arquivo externo. Usado somente em servidores UNIX.
<p>* para exportar o arquivo OPEN DATASET p_arq FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.</p> <p>* para importar o arquivo OPEN DATASET p_arq FOR INPUT IN TEXT MODE ENCODING DEFAULT.</p>	

OPEN_FORM	Função usada para abrir um SAPSCRIPT
<pre> CALL FUNCTION 'OPEN_FORM' EXPORTING DEVICE = 'PRINTER' DIALOG = 'X' FORM = 'Z_C01001' " nome do formulário sapscrip * LANGUAGE = SY-LANGU IMPORTING * LANGUAGE = * NEW_ARCHIVE_PARAMS = RESULT = t_RESULT " tabela interna para receber parâmetros EXCEPTIONS CANCELED = 1 DEVICE = 2 FORM = 3 OPTIONS = 4 UNCLOSED = 5 MAIL_OPTIONS = 6 ARCHIVE_ERROR = 7 INVALID_FAX_NUMBER = 8 MORE_PARAMS_NEEDED_IN_BATCH = 9 SPOOL_ERROR = 10 CODEPAGE = 11 OTHERS = 12. IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	

PARAMETERS <i>parâmetro LIKE tabela</i> PARAMETERS <i>parâmetro TYPE tipo</i> PARAMETERS <i>botão</i> RADIOBUTTON GROUP <i>grupo.</i>	Cria parâmetros de seleção para janelas.
<pre> PARAMETERS P_WERKS LIKE T_001W-WERKS. " com os mesmos atributos do campo PARAMETERS CODIGO(3) TYPE C. * para criar radio-button PARAMETERS p_bot1 RADIOBUTTON GROUP g1. PARAMETERS p_bot2 RADIOBUTTON GROUP g1. * faz com que o parâmetro seja obrigatório. PARAMETERS p_matnr LIKE T_MARA-MATNR OBLIGATORY. * RLGRAP-FILENAME valor padrão para receber nomes de arquivos a importar e exportar * arquivos PARAMETERS P_ARQ LIKE RLGRAP-FILENAME. " OBLIGATORY. * PARA CRIAR VARIANTS DE EXIBIÇÃO PARA RELATÓRIOS ALV PARAMETERS p_varia LIKE disvariant-variant. "Variante de exibição</pre>	

PERFORM nome_formulario	Chama um formulário (subprograma)
PERFORM nome_do_sub_programa.	
POPUP_TO_CONFIRM_WITH_MESSAGE	Função que exibe na tela uma janela com opções.
READ DATASET arquivo INTO t_tabela	Usado para ler e importar dados de um arquivo externo.
LOOP AT t_mard. TRANSFER texto09 TO p_arq. READ DATASET p_aqr INTO t_tabela. ENDLOOP.	
READ TABLE t_tabela WITH KEY campo = valor	Usado para ler um registro específico.
SORT t_tabela1 BY campo. LOOP AT t_tabela1. * com o comando BINARY SEARCH a pesquisa fica mais rápida. READ TABLE t_tabela2 WITH KEY campo = t_tabela1-campo BINARY SEARCH. * após o READ TABLE devemos verificar o resultado IF SY-SUBRC EQ 0. " verdadeiro (encontrou o dado) ELSE. ENDIF. ENDLOOP.	

READ_TEXT	Função usada para ler um texto qualquer
<pre> CALL FUNCTION 'READ_TEXT' EXPORTING CLIENT = SY-MANDT ID = '0001' LANGUAGE = SY-LANGU NAME = V_MATERIAL_LOGO OBJECT = 'KNMT' TABLES LINES = T_TEXTO " tabela interna que guardara as textos EXCEPTIONS ID = 1 LANGUAGE = 2 NAME = 3 NOT_FOUND = 4 OBJECT = 5 REFERENCE_CHECK = 6 WRONG_ACCESS_TO_ARCHIVE = 7 OTHERS = 8. IF SY-SUBRC <> 0. ENDIF.</pre>	
REFRESH t_tabela.	Limpa o conteúdo de uma tabela.
REFRESH t_tabela.	
REPLACE var WITH var2 INTO var3	Replica um texto dentro de outro texto.
<pre> DATA campo(4) TYPE C. Campo = 'ABCD'. REPLACE 'B' WITH 'texto' INTO campo. " novo valor de campo = 'AtextoCD'</pre>	
REPORT	Comando para iniciar um programa.
REPORT.	
RESERVE linhas	Comando usado para reservas linhas para quebra de relatórios
RESERVE 4. " reserva 4 linhas para quebra de relatórios	
SAPGUI_PROGRESS_INDICATOR	Função para indicar o progresso de uma tarefa.

SEARCH	Pesquisa um texto dentro de outro
<pre> SEARCH 'CIDADE DE DEUS' FOR 'DEUS'. IF SY-SUBRC = 0 " valor encontrado ELSE. ... ENDIF.</pre>	
SELECT tabela SELECT SINGLE SELECT campos INTO TABLE t_tabela FROM tabela WHERE condição.	Para pesquisar dados dentro de uma tabela. Retorna apenas um valor dentro da tabela. Pesquisa para tabelas internas.
<p>* opção com select lendo direto uma tabela interna, os campos devem estar na ordem da tabela interna. A tabela interna já fica preenchida.</p> <pre>SELECT werks name1 INTO TABLE t_001w FROM T001W.</pre> <p>* select com WHERE</p> <pre>SELECT matnr maktx INTO TABLE t_makt FROM MAKT WHERE SPRAS = 'PT'.</pre> <p>* OS WHERE'S DEVEM SEGUIR A SEQUENCIA DOS CAMPOS DA TABELA, quando o campo for um parâmetro com vários dados usar IN no lugar do sinal de igual</p> <pre>SELECT matnr maktx INTO TABLE t_makt FROM MAKT WHERE matnr IN S_MATERI AND SPRAS = sy-langu.</pre> <p>* USO DE INNER JOIN</p> <p>* SELECT campos_das_tabelas que contem os dados INTO TABLE nome_da_tabela que ira guardar os dados FROM tabela principal para fazer o join INNER JOIN tabela secundária para fazer o join com o from ON condição de ligação do join WHERE condições</p> <pre> SELECT marc~matnr marc~werks t001w~name1 makt~maktx INTO TABLE t_marc FROM marc INNER JOIN t001w ON marc~werks = t001w~werks INNER JOIN makt ON marc~matnr = makt~matnr and makt~spras = sy-langu WHERE t001w~werks IN S_CENTRO AND marc~matnr IN S_MATERI. SELECT SINGLE * FROM T001W WHERE WERKS = T001W-WERKS.</pre> <p>SELECT * INTO CORRESPONDING FIELDS OF tabela FROM TABLE</p> <p>SELECT * INTO CORRESPONDING FIELDS OF tabela (grava na linha de cabeçalho)</p> <p>* SELECT para retornar o ultimo numero de uma sequencia ou o maio numero</p> <pre> TABLES zprevisao. DATA z_ultimo like zprevisao-nrdoc. SELECT MAX(nrdoc) INTO z_ultimo FROM zprevisao.</pre>	

SELECTION-OPTIONS	Comando usado para colocar opções de seleção.
<p>SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001. SELECT-OPTIONS S_MATERI FOR T_MARC-MATNR. SELECT-OPTIONS S_CENTRO FOR T_MARC-WERKS. SELECTION-SCREEN END OF BLOCK b_janela.</p>	
SELECTION-SCREEN	Cria um BOX na janela
<p>* PARA CRIAR UM BOX NA JANELA * SELECTION-SCREEN BEGIN OF BLOCK nome_do_bloco WITH FRAME TITLE * nome_do_elemento_de_texto * -> DEPOIS COLOCAR OS SELECT-OPTIONS (variáveis) * FECHAR COM : SELECTION-SCREEN END OF BLOCK nome_do_bloco * clicar duas vezes sobre o nome_do_bloco para colocar o nome * que ira aparecer na tela SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001. SELECT-OPTIONS S_MATERI FOR T_MARC-MATNR. SELECT-OPTIONS S_CENTRO FOR T_MARC-WERKS. PARAMETERS P_BOT1 RADIOBUTTON GROUP b1. PARAMETERS P_BOT2 RADIOBUTTON GROUP b1. SELECTION-SCREEN END OF BLOCK b_janela.</p>	
SET PF-STATUS 'xxxx'	Comando usado para criar botões para serem exibidos em relatórios.
<p>* depois dar dublo click sobre o nome do botão para criar o objeto. SET PF-STATUS 'nome_do_botão'.</p>	
SET TITLEBAR ' xxx'	Comando usado para criar TITULOS de janelas.
<p>* depois dar dublo click sobre o nome do titulo para criar o objeto. SET TITLEBAR 'NNN'.</p>	
SET PARAMETERS ID 'xx' FIELD campo	Usado para passar parâmetros para chamada de funções.
<p>* SET PARAMETER ID 'NN' FILED valor do campo para chamar transações passando * parâmetros SET PARAMETERS ID 'VF' FIELD t_alv-vbeln. CALL TRANSACTION 'VF03' AND SKIP FIRST SCREEN.</p>	
SKIP n	Comando usado para pular linhas em relatório.
<p>WRITE 'IMPRIME UMA LINHA'. SKIP 2. WRITE 'PULOS DUAS LINHAS'.</p>	

<i>SORT t_tabela BY campon</i>	Comando usado ordenar tabelas internas
SORT t_tabela BY campo.	

<i>SPLIT t_tabela AT delimitador INTO t_tabela</i>	Usado para ler um arquivo externo e gravar na tabela.
<p>LOOP AT t_tabela. CLEAR t_tempo. * SPLIT tabela_interna AT delimitador_do_arquivo INTO campos que irão receber os dados na * sequência do arquivo * c_tipo é o delimitador do arquivo texto SPLIT t_tabela-texto01 AT c_tipo INTO T_TEMPO-BUKRS T_TEMPO-EKORG T_TEMPO-KTOKK T_TEMPO-NAME1 T_TEMPO-SORTL T_TEMPO-ORT01 T_TEMPO-PSTLZ T_TEMPO-LAND1 T_TEMPO-REGIO.</p> <p>APPEND t_tempo.</p> <p>ENDLOOP.</p>	

<i>STANDARD TOOLBAR</i>	Nome da barra de ferramenta com os botões padrão.
<p>Valores padrão BACK -> voltar BACK -> EXIT RW -> cancelar PRI -> imprimir %SC -> pesquisa %SC+ -> pesquisa próximo P-- -> primeira pagina P- Page UP P+ -> page down P++ -> Ultima pagina</p>	

START_FORM	Função usada para inicializar um SAPSCRIPT
<pre> CALL FUNCTION 'START_FORM' EXPORTING * ARCHIVE_INDEX = FORM = 'Z_C01001' " nome do formulário sapscrip * IMPORTING * LANGUAGE = EXCEPTIONS FORM = 1 FORMAT = 2 UNENDED = 3 UNOPENED = 4 UNUSED = 5 SPOOL_ER = 6 CODEPAGE = 7 OTHERS = 8. IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	

START-OF-SELECTION	Evento usado para marcar o inicio de um conjunto de seleções de dados.
<pre> START-OF-SELECTION. SELECT marc~matnr marc~werks t001w~name1 makt~maktx INTO TABLE t_marc FROM marc INNER JOIN t001w ON marc~werks = t001w~werks INNER JOIN makt ON marc~matnr = makt~matnr and makt~spras = sy-langu WHERE t001w~werks IN S_CENTRO AND marc~matnr IN S_MATERI. END-OF-SELECTION.</pre>	

STRLEN(texto)	Retorna a tamanho de um texto.
<pre> DATA var TYPE STRING. DATA var2 TYPE I. MOVE 'ESTE TEXTO TEM O TAMANHO DE 30' TO var. I = STRLEN(var). WRITE var2. " retorna 30</pre>	

SUBTRACT	Subtrai um valor de outro.
<p>DATA var1 TYPE I VALUE 10. DATA var2 TYPE I VALUE 4.</p> <p>SUBTRACT var2 TO var1. " retorna var1 = 6</p>	

SUM	Totaliza um campo numérico dentro de uma tabela.
<p>SORT T_MARC BY matnr werks lgort. LOOP AT T_MARC. WRITE /5 T_MARC-MATNR. WRITE 18 T_MARC-MAKTX. WRITE 60 T_MARC-WERKS. WRITE 80 T_MARC-NAME1. WRITE 120 T_MARC-LGORT. WRITE 140 T_MARC-LFGJA. WRITE 150 T_MARC-LFMON. WRITE 160 T_MARC-LABST. AT END OF werks. SUM. WRITE : /5 'TOTAL DO MATERIAL : ',t_marc-maktx. " campo numérico WRITE 160 T_MARC-LABST. ENDAT.</p> <p>ENDLOOP.</p>	

SXPG_CALL_SYSTEM	Função que executa comandos externos no servidor.
<pre> CALL FUNCTION 'SXPG_CALL_SYSTEM' EXPORTING COMMANDNAME = wvar * IMPORTING TABLES EXEC_PROTOCOL = t_btc EXCEPTIONS NO_PERMISSION = 1 COMMAND_NOT_FOUND = 2 PARAMETERS_TOO_LONG = 3 SECURITY_RISK = 4 WRONG_CHECK_CALL_INTERFACE = 5 PROGRAM_START_ERROR = 6 PROGRAM_TERMINATION_ERROR = 7 X_ERROR = 8 PARAMETER_EXPECTED = 9 TOO_MANY_PARAMETERS = 10 ILLEGAL_COMMAND = 11 OTHERS = 12. IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	

SY-BATCH SY-DATUM SY-LANGU SY-LINNO SY-LISEL SY-MANDT SY-PAGNO SY-SUBRC SY-TABIX SY-TVAR0 .. SY-TVAR9 SY-UCOMM SY-UNAME SY-UZEIT SY-VLINE SY-ULINE(n) SY-CPROG SY-TCODE SY-DBCNT	Indica a execução de um programa em background Retorna a data do sistema Retorna o idioma de logon do usuário Retorna a linha corrente de um relatório Retorna a linha selecionada em relatórios interativos Retorna o mandante do sistema Retorna a pagina atual de um relatório Retorna o valor de uma pesquisa ou condição Retorna o índice de uma tabela em loop Retorna elementos de textos ou títulos de relatórios Retorna o nome de um botão precionado Retorna o nome do usuário Retorna a hora do sistema Efetua a fechamento de bordas em um relatório Imprime uma linha com n posições Nome do programa principal Código da transação Dentro de SELECT, contém o contador de interação
Estas variáveis são do sistema e servem para controle do programa. Seus conteúdos devem ser verificados para podermos fazer o controle do programa.	

TABLES tabela	Usado para abertura de tabelas STD.
TABLES tabela. TABLES : tabela1, tabela2, tabela3.	

TOP-OF-PAGE	Marca o início de um cabeçalho de relatório
TOP-OF-PAGE. WRITE 'NOVO CABECALHO DEFINIDO PELO PROGRAMA'. END-OF-PAGE.	
TRANSFER t_tabela TO arquivo	Usado para ler e gravar em arquivo externo. (somente para UNIX)
OPEN DATASET p_arq FOR OUTPUT IN TEXT MODE ENCODING DEFAULT. LOOP AT t_mard. TRANSFER t_tabela TO p_arq. ENDLOOP. CLOSE DATASET p_arq.	
TRANSLATE var TO UPPER CASE TRANSLATE var TO LOWER CASE	Transforma um texto de letras minúsculas em maiúsculas e vice versa
TRANSLATE 'abcd' TO UPPER CASE. " retorna ABCD TRANSLATE 'ABCD' TO LOWER CASE. " retorna abcd	
TYPES tipo TYPES tabela	Define os tipos de variáveis e pode ser usado para definir dados de tabela interna.
DATA var1 TYPE I. " define uma variável do tipo inteiro * define um tipo com os campos da tabela TYPES: BEGIN OF tipo_tabela vbeln LIKE vbrk-vbeln, kdgrp LIKE vbrk-kdgrp, netwr LIKE vbrk-netwr, END OF tipo_tabela. * cria a tabela definindo com o tipo criado acima usando o header line. DATA t_tabela TYPE tipo_vbrk OCCURS 0 WITH HEADER LINE.	
ULINE	Server para colocar uma linha continua em relatórios para fazer separações.
ULINE " imprime uma linha continua	

UNPACK var	Coloca zeros na frente de um numero em formato texto
<p>DATA var(4) TYPE C. MOVE '1' TO var. WRITE var. " retorna '1' (caractere) UNPACK var TO var2. WRITE var2. " retorna '0001'</p>	
UPDATE tabela	Atualiza um tabela transparente
<p>UPDATE tabela.</p>	
WHILE condição. ENDWHILE.	Cria um laço e verifica a condição no final do laço
<p>WHILE condição. " a condição é verifica no final do laço e se for verdadeira continua dentro " do loop ENDWHILE.</p>	
WRITE 'xxxxxxx'	Comando usado para saída de impressora ou tela
<p>WRITE 'ESTA LINHA SERA IMPRESSSA'. WRITE 'ESTA LINHA SERA COLOCADA NA MESMA LINHA QUE A PRIMEIRA'. WRITE / 'COM A / FAZ UMA QUEBRA E O TEXTO SERA IMPRESSA NA LINHA DE BAIXO'. WRITE 5 'IMPRIME NA POSIÇÃO 5'.</p>	

WRITE_FORM	Função usada para escrever um SAPSCRIPT
<pre> CALL FUNCTION 'WRITE_FORM' EXPORTING ELEMENT = 'TITULO' " nome do elemento de dados da janela * FUNCTION = 'SET' * TYPE = 'BODY' WINDOW = 'TITULOS' " nome da janela * IMPORTING * PENDING_LINES = EXCEPTIONS ELEMENT = 1 FUNCTION = 2 TYPE = 3 UNOPENED = 4 UNSTARTED = 5 WINDOW = 6 BAD_PAGEFORMAT_FOR_PRINT = 7 SPOOL_ERROR = 8 CODEPAGE = 9 OTHERS = 10. IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	

WS_UPLOAD	Função usada para fazer UPLOAD de um arquivo externo.
<pre> CALL FUNCTION 'WS_UPLOAD' EXPORTING FILENAME = P_ARQ " COLOCAR O NOME DA VARIÁVEL " IRA CONTER O NOME DO ARQUIVO FILETYPE = 'ASC' " TIPO DE ARQUIVO * IMPORTING TABLES DATA_TAB = t_tabela " NOME DA TABELA INTERNA QUE IRA RECEBER " IRA RECEBER OS DADOS 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 NO_AUTHORITY = 10 OTHERS = 11. IF SY-SUBRC <> 0. MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4. ENDIF.</pre>	

ASSIGN f1 TO <F>	Atribuição dinâmica
<p>Assign f1 TO <f></p> <p>Obs : atribui o valor de f1 para o simbolo <f></p>	

APPEND tabela SORTED BY c.	Usado classificar as tabelas em ordem.
<p>APPEND tabela SORTED BY c.</p> <p>C – campo a ser classificado.</p> <p>Exemplo.</p> <p>Tabela-dados = 'xxx'.</p> <p>Tabela-valor = 1000.</p> <p>APPEND tabela SORTED BY dados.</p>	

APPEND LINES OF tabela1 FROM nf TO nt TO tabela2.	Copiando uma parte de uma tabela interna.
<p>Nf, nt = constantes, literais ou variaveis numericas.</p> <p>Nf – indice da primeira tabela1 a ser copiada (linha inicial)</p> <p>Nt – indice da ultima linha da tabela1 a ser copiada</p> <p>Obs = mais rapido que APPEND.</p>	
AS TEXT	Usado classificar os caracteres acentuados de uma tabela (em ordem binaria).
<p>SORT tabela AS TEXT.</p> <p>Não pode usar READ TABLE com BINARY SEARCH</p>	
AS SYMBOL AS ICON AS LINE	Comandos usados para formatação de relatórios.
<p>AS SYMBOL – exibe um simbolo em preto-e-branco</p> <p>AS ICON – exibe um icone de cor</p> <p>AS LINE – exibe um caracter de desenho de linha</p> <p>WRITE n1 (a) as symbol.</p> <p>WRITE n1 (c) as icon.</p> <p>WRITE n1 (b) as line.</p> <p>Devemos inserir no inicio do programa :</p> <p>INCLUDE <symbol></p> <p>ou</p> <p>INCLUDE <icon></p> <p>e</p> <p>INCLUDE <list></p>	
Copiando dados de uma tabela interna para outra	
<p>Tabela2[] = Tabela1[].</p> <p>Obs = somente se ambas forem iguais, o conteudo da tabela 2 é apagado.</p>	

DESCRIBE TABLE tabela LINES i OCCURS j.	Determinando o numero de linhas de uma tabela interna.
<p>DESCRIBE TABLE tabela LINES i OCCURS j.</p> <p>SY-TFILLI - numero de linhas SY-TLENG - comprimento de uma linha em bytes SY-TOCCU - valor atual da cláusula occurs.</p>	

INSERT LINES OF tabela1 FROM nf TO nt INTO tabela2 INDEX nb.	Inserindo linhas dentro de uma tabela interna (local especifico)
<p>INSERT LINES OF tabela1 FROM nf TO nt INTO tabela2 INDEX nb.</p> <p>Nb – ponto aonde deve ser inserido as linhas.</p>	

Operadores de String de caractere				
Operador	Significado	Verdadeiro quando	*	**
v1 CO v2	Comtem somente	v1 for composto somente de caracteres em v2	S	N
v1 CN v2	NOT v1 CO v2	v1 pode conter caracteres que não estão em v2	S	N
v1 CA v2	Comtem qualquer	v1 pode conter pelo menos um caracter em v2	S	N
v1 NA v2	NOT v1 CA v2	v1 não pode conter nenhum caracter em v2	S	N
v1 CS v2	Comtem string	v1 pode conter a string de caracteres em v2	N	S
v1 NS v2	NOT v1 CS v2	v1 não pode conter a string de carateres em c2	N	S
v1 CP v2		v1 pode conter o padrão em v2	N	S
v1 NP v2	NOT v1 CP v2	v1 não pode conter o padrão em v2	N	S

vale os operadores (coringas)

* - pode corresponder a qualquer sequencia de caratecres
+ - pode corresponder a um único caracter
- pode interpretar o proximo caracter, literalmente

Exemplo :

v1 CP 'A+C' – v1 pode conter "a" na primeira posição e "c" na terceira
v1 CP '*Ab*' – a string "ab" pode ocorrer em qualquer lugar dentro de v1
v1 CP '*#A#b*' – v1 pode conter um letra A maiuscula seguida por uma letra b minuscula

VARYING n1 NEXT n2	Obtem os componentes de uma string de campo e sequencia.
<p>Utilize a adição de varying para obter os componentes de um string de campo em sequencia. Next estabelece um distancia (em bytes) entre dois componentes por interação.</p> <p>Exemplo :</p> <pre>DATA : F1, BEGIN of s, C1 VALUE 'A', C2 VALUE 'B', C3 VALUE 'C', C4 VALUE 'D', C5 VALUE 'E', C6 VALUE 'F', END OF S.</pre> <pre>DO 6 TIMES VARYING f1 FROM S-C1 NEXT S-C2. WRITE F1. ENDDO</pre> <p>Saida : A B C D E F</p> <pre>DO 3 TIMES VARYING f1 FROM S-C1 NEXT S-C3. WRITE F1. ENDDO</pre> <p>Saida : A C E</p>	

Programa 1

```
*&-----*
*& Report  ZC00101                                *
*&                                                *
*&-----*
*&                                                *
*&                                                *
*&-----*

REPORT  ZC00101.
WRITE 'HELLO AMANTINO'.
WRITE / 'HOJE É UM BELO DIA'. " A BARRA (/) QUEBRA DE LINHA
SKIP 2. " O COMANDO SKIP PULA LINHAS NA EMISSÃO DO RELATORIO
WRITE 'OLHA NOIS AQUI TRA VEZ'.
ULINE. " COLOCA UMA LINHA RETA
FORMAT COLOR COL_TOTAL. " FORMATA A SAIDA DO COMANDO WRITE
WRITE 'QUALQUER COISA'.
FORMAT COLOR COL_KEY.
WRITE / 'QUALQUER COISA'.
FORMAT COLOR COL_BACKGROUND. " VOLTA AO PADRÃO
```

Programa 2

```

*&-----*
*& Report  ZC00103                                     *
*&                                                     *
*&-----*
*&                                                     *
*&                                                     *
*&-----*

REPORT  ZC00103 LINE-SIZE 130 LINE-COUNT 65.
* LINE-SIZE = COLUNAS
* LINE-COUNT = LINHAS POR PAGINA

TABLES : MARC, T001W.

DATA : BEGIN OF T_MARC OCCURS 0,
        MATNR LIKE MARC-MATNR,
        WERKS LIKE MARC-WERKS,
        NAME1 LIKE T001W-NAME1,
        MAKTX LIKE MAKT-MAKTX,
        END OF T_MARC.

DATA : BEGIN OF T_001W OCCURS 0,
        WERKS LIKE T001W-WERKS,
        NAME1 LIKE T001W-NAME1,
        END OF T_001W.

DATA : BEGIN OF T_MAKT OCCURS 0,
        MATNR LIKE MAKT-MATNR,
        MAKTX LIKE MAKT-MAKTX,
        END OF T_MAKT.

DATA wcor TYPE C.

* O COMANDO PARAMETERS serve para colocar um parametro de seleção PARAMETERS
nome_do_parametro LIKE tabela_interna.
* PARAMETERS P_WERKS LIKE T_001W-WERKS.
* O comando SELECT-OPTIONS funciona como o comando PARAMETERS SELECT-OPTIONS variável FOR
campo da tabela DEFAULT valor_inicial

SELECT-OPTIONS P_WERKS FOR T_001W-WERKS DEFAULT '0050'.

REFRESH T_MARC.
CLEAR T_MARC.

* SELECIONAR material e centro PARA PARAMETERS podemos usar o sinal de =
* SELECT matnr werks INTO TABLE t_marc FROM MARC WHERE werks = P_WERKS.

* SELECIONAR material e centro PARA SELECT-OPTIONS usar IN
SELECT matnr werks INTO TABLE t_marc FROM MARC WHERE werks IN P_WERKS.

* ORDENAR PRIMEIRO A TABELA PARA PODER USAR [BINARY SEARCH] EM READ TABLE
SORT t_marc BY werks.

* SELECIONAR material e a descrição
SELECT matnr maktx INTO TABLE t_makt FROM MAKT WHERE SPRAS = 'PT'.

* ORDENAR PRIMEIRO A TABELA PARA PODER USAR [BINARY SEARCH] EM READ TABLE
SORT t_makt BY matnr.

* SELECIONAR centro e nome centro
SELECT werks name1 INTO TABLE t_001w FROM T001W.

LOOP AT T_MARC.

```

```
* PARA LER UMA TABELA INTERNA2 USAR READ TABLE
* tabela interna1 WITH KEY campo da tabela interna1
* igual ao campo da TABELA INTERNA2
* BINARY SEARCH -> DEIXA A PESQUISA MAIS RAPIDA.
  READ TABLE T_001W WITH KEY werks = t_marc-werks BINARY SEARCH.

* SY-SUBRC se voltar ZERO encontrou na Tabela
  IF sy-subrc eq 0.
    t_marc-name1 = T_001W-name1.
  ELSE.
    t_marc-name1 = '          '.
  ENDIF.

* PARA ATUALIZAR A TABELA INTERNA MODIFY tabela interna2
  MODIFY t_marc.

ENDLOOP.

LOOP AT T_MARC.

* PARA LER UMA TABELA INTERNA2 USAR READ TABLE
* tabela interna1 WITH KEY campo da tabela interna1
* igual ao campo da TABELA INTERNA2
* BINARY SEARCH -> DEIXA A PESQUISA MAIS RAPIDA.
  READ TABLE T_MAKT WITH KEY matnr = t_marc-matnr BINARY SEARCH.

* SY-SUBRC se voltar ZERO encontrou na Tabela
  IF sy-subrc eq 0.
    t_marc-maktx = T_makt-maktx.
  ELSE.
    t_marc-maktx = '          '.
  ENDIF.

* PARA ATUALIZAR A TABELA INTERNA MODIFY tabela interna2
  MODIFY t_marc.

ENDLOOP.

* PARA ORDENAR A TABELA INTERNA. SORT tabela_interna BY campo
SORT t_marc BY maktx ASCENDING.

WRITE 5 'MATERIAL'.
WRITE 20 'Descrição'.
WRITE 70 'Cód '.
WRITE 80 'CENTRO'.
ULINE.

LOOP AT T_MARC.
  IF wcor EQ 'I'.
    FORMAT COLOR COL_POSITIVE.
    wcor = 'X'.
  ELSE.
    FORMAT COLOR COL_NEGATIVE.
    wcor = 'I'.
  ENDIF.
  WRITE 5 T_MARC-MATNR.
  WRITE 20 T_MARC-MAKTX.
  WRITE 70 T_MARC-WERKS.
  WRITE 80 T_MARC-NAME1.
  SKIP 1.
ENDLOOP.
```

Programa 3

```

*&-----*
*& Report  ZC00106                                     *
*&                                                     *
*&-----*
*&                                                     *
*&                                                     *
*&-----*

REPORT  ZC00104 LINE-SIZE 130 LINE-COUNT 65.

TABLES : MARC, T001W.

DATA : BEGIN OF T_MARC OCCURS 0,
        MATNR LIKE MARC-MATNR,
        WERKS LIKE MARC-WERKS,
        NAME1 LIKE T001W-NAME1,
        MAKTX LIKE MAKT-MAKTX,
        END OF T_MARC.

DATA : BEGIN OF T_001 OCCURS 0,
        WERKS LIKE T001W-WERKS,
        END OF T_001.

REFRESH : T_MARC.
CLEAR   : T_MARC.

* PARA CRIAR UM BOX NA JANELA
* SELECTION-SCREEN BEGIN OF BLOCK nome_do_bloco WITH FRAME TITLE
* nome_do_elemento_de_texto
* -> DEPOIS COLOCAR OS SELECT-OPTIONS ( variáveis )
* FECHAR COM : SELECTION-SCREEN END OF BLOCK nome_do_bloco
* clicar duas vezes sobre o nome_do_bloco para colocar o nome
* que ira aparecer na tela
SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001.
SELECT-OPTIONS S_MATERI FOR T_MARC-MATNR.
SELECT-OPTIONS S_CENTRO FOR T_MARC-WERKS.
SELECTION-SCREEN END OF BLOCK b_janela.

* COMANDO PARA INICIALIZAÇÃO DE VARIÁVEL
INITIALIZATION.

* seleciona o CAMPO_que_iremos tratar INTO TABLE
* tabela_interna FROM tabela_de_onde_vira os dados
SELECT werks INTO TABLE t_001 FROM t001w.

* LIMPA A TABELA DE VARIÁVEL
REFRESH S_CENTRO.

* LOOP DA TABELA INTERNA
LOOP AT T_001.
* LIMPA A VARIÁVEL
clear S_CENTRO.
* iguala SIGN a I e OPTION a EQ
s_centro-sign = 'I'.
s_centro-option = 'EQ'.
* COLOCAR O VALOR do campo
s_centro-low = t_001-werks.
* atualiza a tabela.
APPEND S_CENTRO.
ENDLOOP.

* EVENTO DE EXTRAÇÃO DE DADOS - ANTES DA SELEÇÃO
START-OF-SELECTION.

```

```
* USO DE INNER JOIN
* SELECT campos_das_tabelas que contem os dados
* INTO TABLE nome_da_tabela que ira guardar os dados
* FROM tabela principal para fazer o join
* INNER JOIN tabela secundária para fazer o join com o from
* ON condição de ligação do JOIN
* WHERE condições
```

```
SELECT marc~matnr marc~werks t001w~name1 makt~maktx
      INTO TABLE t_marc
      FROM marc
        INNER JOIN t001w ON marc~werks = t001w~werks
        INNER JOIN makt ON marc~matnr = makt~matnr
                        and makt~spras = sy-langu
      WHERE t001w~werks IN S_CENTRO
            AND marc~matnr IN S_MATERI.
```

```
*WRITE 5 'MATERIAL'.
*WRITE 18 'Descrição'.
*WRITE 60 'Centro'.
*WRITE 80 'Descrição'.
*ULINE.
```

```
* EVENTO PARA EXIBIR OS DADOS
END-OF-SELECTION.
```

```
LOOP AT T_MARC.
  WRITE 5 T_MARC-MATNR.
  WRITE 18 T_MARC-MAKTX.
  WRITE 60 T_MARC-WERKS.
  WRITE 80 T_MARC-NAME1.
  SKIP 1.
ENDLOOP.
```


Programa 4

```

*&-----*
*& Report  ZC00107*
*&*
*&-----*
*&-----*

REPORT ZC00107 LINE-SIZE 130 LINE-COUNT 65 NO STANDARD PAGE HEADING
MESSAGE-ID Z01.

* REPORT nome_do_programa
* parâmetros do REPORT
* -> LINE-SIZE n_colunas
* -> LINE-COUNT n_linhas
* -> NO STANDARD PAGE HEADING ( não usar cabeçalho padrão )
* -> MESSAGE-ID z01 ( onde z01 é a classe de mensagem )

TABLES : MARC, T001W.

DATA : BEGIN OF T_MARC OCCURS 0,
        MATNR LIKE MARC-MATNR,
        WERKS LIKE MARC-WERKS,
        NAME1 LIKE T001W-NAME1,
        MAKTX LIKE MAKT-MAKTX,
        END OF T_MARC.

DATA : BEGIN OF T_001 OCCURS 0,
        WERKS LIKE T001W-WERKS,
        END OF T_001.

REFRESH : T_MARC.
CLEAR    : T_MARC.

* PARA CRIAR UM BOX NA JANELA SELECTION-SCREEN BEGIN OF BLOCK nome_do_bloco WITH FRAME
TITLE nome_do_elemento_de_texto
* -> DEPOIS COLOCAR OS SELECT-OPTIONS ( variáveis )
* FECHAR COM : SELECTION-SCREEN END OF BLOCK nome_do_bloco
* clicar duas vezes sobre o nome_do_bloco para colocar o nome
* que ira aparecer na tela
SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001.
SELECT-OPTIONS S_MATERI FOR T_MARC-MATNR.
SELECT-OPTIONS S_CENTRO FOR T_MARC-WERKS.
SELECTION-SCREEN END OF BLOCK b_janela.

* COMANDO PARA INICIALIZAÇÃO DE VARIÁVEL
INITIALIZATION.

* PARA CRIAR ROTINAS : PERFORM nome_da_rotina. NÃO USAR ROTINAS PARA CRIAR TABELAS E
ABRIR TABELAS ESTAS DEVEM ESTAR NO INICIO DO PROGRAMA FAZER SEMPRE DENTRO DOS EVENTOS

    PERFORM F_SELECIONAR.
* ANALISA O RESULTADO DA VARIÁVEL s_materi ANTES DE INICIAR LISTA DE MATERIAL
AT SELECTION-SCREEN ON s_materi.

PERFORM F_BUSCAR_MATERIAL.

TOP-OF-PAGE.

    PERFORM F_CABECA.

END-OF-PAGE.

* EVENTO DE EXTRAÇÃO DE DADOS - ANTES DA SELEÇÃO
START-OF-SELECTION.

    PERFORM F_BUSCAR_DADOS.

```

```
* EVENTO PARA EXIBIR OS DADOS
END-OF-SELECTION.
```

```
    PERFORM F_MOSTRAR_DADOS.
*&-----*
*&      Form  F_SELECIONAR
*&-----*
*      text
*-----*
*  --> p1      text
*  <-- p2      text
*-----*
FORM F_SELECIONAR .
* seleciona o CAMPO_que iremos tratar INTO TABLE tabela_interna FROM tabela_de_onde_vira
os dados
    SELECT werks INTO TABLE t_001 FROM t001w.
```

```
* LIMPA A TABELA DE VARIABEL
    REFRESH S_CENTRO.
```

```
* LOOP DA TABELA INTERNA
    LOOP AT T_001.
* LIMPA A VARIABEL
    clear S_CENTRO.
* iguala SIGN a I e OPTION a EQ
    s_centro-sign = 'I'.
    s_centro-option = 'EQ'.
* COLOCAR O VALOR do campo
    s_centro-low = t_001-werks.
* atualiza a tabela.
    APPEND S_CENTRO.
ENDLOOP.
```

```
ENDFORM.                " F_SELECIONAR
*&-----*
*&      Form  F_BUSCAR_DADOS
*&-----*
*      text
*-----*
*  --> p1      text
*  <-- p2      text
*-----*
```

```
FORM F_BUSCAR_DADOS .
* USO DE INNER JOIN
* SELECT campos_das_tabelas que contem os dados
* INTO TABLE nome_da_tabela que ira guardar os dados
* FROM tabela principal para fazer o join
* INNER JOIN tabela secundária para fazer o join com o from
* ON condição de ligação do JOIN
* WHERE condições
```

```
    SELECT marc~matnr marc~werks t001w~name1 makt~maktx
        INTO TABLE t_marc
        FROM marc
            INNER JOIN t001w ON marc~werks = t001w~werks
            INNER JOIN makt ON marc~matnr = makt~matnr
                        and makt~spras = sy-langu
        WHERE t001w~werks IN S_CENTRO
        AND marc~matnr IN S_MATERI.
```

```
* sy-subrc <> 0 não achou nada
* MESSAGE tipo_de_mensagem (E/I/W/S)+numero_seq_da_mensagem
    IF sy-subrc <> 0.
        MESSAGE S001.
    ENDIF.
```

```

ENDFORM.                  " F_BUSCAR_DADOS
*&-----*
*&      Form  F_MOSTRAR_DADOS
*&-----*
*      text
*-----*
*      --> p1      text
*      <-- p2      text
*-----*
FORM F_MOSTRAR_DADOS .
  LOOP AT T_MARC.
    WRITE 5 T_MARC-MATNR.
    WRITE 18 T_MARC-MAKTX.
    WRITE 60 T_MARC-WERKS.
    WRITE 80 T_MARC-NAME1.
    SKIP 1.
  ENDLOOP.

ENDFORM.                  " F_MOSTRAR_DADOS
*&-----*
*&      Form  F_CABECA
*&-----*
*      text
*-----*
*      --> p1      text
*      <-- p2      text
*-----*
FORM F_CABECA .

  DATA novo(60) TYPE C.

  WRITE 5 sy-datum.
  WRITE 50 'CHEMYUNION QUIMICA LTDA'.
  WRITE 120 sy-pagno.
  * WRITE AT /40 'RELATÓRIO DE MATERIAIS POR CENTRO DE CUSTO : '.
  CONCATENATE text-002 ' : ' s_centro+3(4) INTO novo.
  WRITE AT /40 novo.
  WRITE AT /5 'Material'.
  WRITE 18 'Descrição'.
  WRITE 60 'Centro'.
  WRITE 80 'Descrição'.
  ULINE.
ENDFORM.                  " F_CABECA
*&-----*
*&      Form  F_BUSCAR_MATERIAL
*&-----*
*      --> p1      text
*      <-- p2      text
*-----*
FORM F_BUSCAR_MATERIAL .
  * VERIFICA SE EXISTE O MATERIAL
  SELECT * FROM marc UP TO 1 ROWS WHERE matnr IN s_materi.
  ENDSELECT.
  * ANALISA O RESULTADO DO SELECT
  IF SY-SUBRC <> 0.
    MESSAGE W001.
  ENDIF.
ENDFORM.                  " F_BUSCAR_MATERIAL

```

Programa 5

```

*&-----*
*& Report  ZC00108                                     *
*&                                                     *
*&-----*
*&                                                     *
*&                                                     *
*&-----*

REPORT ZC00108 LINE-SIZE 200 LINE-COUNT 65 NO STANDARD PAGE HEADING
MESSAGE-ID Z01.

TABLES : MARC, MARD, T001L, T001W.
* para fazer a quebra do relatório por centro, este deve ser o primeiro
* da tabela interna.
* no SELECT ... INNER JOIN os campos devem sempre seguir a sequência da
* tabela interna.
DATA : BEGIN OF T_MARC OCCURS 0,
        WERKS LIKE MARC-WERKS,
        MATNR LIKE MARC-MATNR,
        NAME1 LIKE T001W-NAME1,
        MAKTX LIKE MAKT-MAKTX,
        LGORT LIKE MARD-LGORT,
        LFGJA LIKE MARD-LFGJA,
        LFMON LIKE MARD-LFMON,
        LABST LIKE MARD-LABST,
        SPEME LIKE MARD-SPEME,
      END OF T_MARC.

DATA : BEGIN OF T_001 OCCURS 0,
        WERKS LIKE T001W-WERKS,
      END OF T_001.

REFRESH : T_MARC.
CLEAR   : T_MARC.

SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001.
SELECT-OPTIONS S_MATERI FOR T_MARC-MATNR.
SELECT-OPTIONS S_CENTRO FOR T_MARC-WERKS.
SELECT-OPTIONS S_DEPOSI FOR T_MARC-LGORT.
SELECT-OPTIONS S_ANO    FOR T_MARC-LFGJA.
SELECT-OPTIONS S_MES     FOR T_MARC-LFMON.
SELECTION-SCREEN END OF BLOCK b_janela.

INITIALIZATION.

    PERFORM F_SELECIONAR.

AT SELECTION-SCREEN ON s_materi.

AT SELECTION-SCREEN ON s_deposi.

    PERFORM F_BUSCAR_MATERIAL.

TOP-OF-PAGE.

    PERFORM F_CABECA.

END-OF-PAGE.

START-OF-SELECTION.

    PERFORM F_BUSCAR_DADOS.

END-OF-SELECTION.

```

```

PERFORM F_MOSTRAR_DADOS.
* o comando AT LINE-SELECTION é um evento que serve para
* quando clicarmos em uma linha de relatório abrir uma outra
* janela e mostrarmos outros dados
AT LINE-SELECTION.
* IF NOT var IS INITIAL -> verifica se o header line esta vazio
* se não for vazio imprime a tela de detalhes
IF NOT T_MARC IS INITIAL.
* SET TITLEBAR 'var' mostra o titulo da nova tela
  SET TITLEBAR 'DET'.
* SY-ULINE -> imprime linha com comprimento fixo
  WRITE SY-ULINE(83).
* SY-VLINE -> serve para fazer fechamento de bordas
  WRITE / SY-VLINE.
  WRITE : 'Material      : '.
  WRITE : T_MARC-MATNR, ' - ', t_marc-maktx, ' '.
  WRITE 83 SY-VLINE.
  WRITE / SY-VLINE.
  WRITE : 'Centro        : '.
  WRITE : T_MARC-werks, ' - ', t_marc-name1, ' '.
  WRITE 83 SY-VLINE.
  WRITE /:SY-VLINE.
  WRITE : 'Deposito      : '.
  WRITE : T_MARC-lgort, ' '.
  WRITE 83 SY-VLINE.
  WRITE /:SY-VLINE.
  WRITE : 'Disponível   : '.
  WRITE : T_MARC-labst, ' '.
  WRITE 83 SY-VLINE.
  WRITE /:SY-VLINE.
  WRITE : 'Bloqueado    : '.
  WRITE : T_MARC-SPEME, ' '.
  WRITE 83 SY-VLINE.
  WRITE / SY-ULINE(83).
ENDIF.
* depois de listar a nova informação devemos limpar o header line da
* tabela
CLEAR T_MARC.

*&-----*
*&      Form  F_SELECIONAR
*&-----*
*
*      text
*-----*
*  --> p1      text
*  <-- p2      text
*-----*
FORM F_SELECIONAR .
  SELECT werks INTO TABLE t_001 FROM t001w.

  REFRESH S_CENTRO.

  LOOP AT T_001.
    clear S_CENTRO.
    s_centro-sign = 'I'.
    s_centro-option = 'EQ'.
    s_centro-low = t_001-werks.
    APPEND S_CENTRO.
  ENDLOOP.

ENDFORM.                    " F_SELECIONAR
*&-----*
*&      Form  F_BUSCAR_DADOS
*&-----*
*
*      text
*-----*
*  --> p1      text

```

```

* <-- p2      text
*-----*
FORM F_BUSCAR_DADOS .

    SELECT marc~werks marc~matnr t001w~name1 makt~maktx mard~lgort
           mard~lfgja mard~lfmon mard~labst mard~speme
    INTO TABLE t_marc
    FROM marc
    INNER JOIN t001w ON marc~werks = t001w~werks
    INNER JOIN makt ON marc~matnr = makt~matnr
                   and makt~spras = sy-langu
    INNER JOIN mard ON marc~matnr = mard~matnr
    WHERE t001w~werks IN S_CENTRO
    AND marc~matnr IN S_MATERI
    AND mard~lfgja IN S_ANO
    AND mard~lfmon IN S_MES
    AND mard~lgort IN S_DEPOSI.

    IF sy-subrc <> 0.
        MESSAGE S001.
    ENDIF.

ENDFORM.                " F_BUSCAR_DADOS
*&-----*
*&      Form  F_MOSTRAR_DADOS
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM F_MOSTRAR_DADOS .
    SORT T_MARC BY matnr werks lgort.
    * comando LOOP AT tabela serve para criar um loop até o final da tabela
    * para podermos imprimir seu conteúdo.
    LOOP AT T_MARC.
        WRITE /5 T_MARC-MATNR.
        WRITE 18 T_MARC-MAKTX.
        WRITE 60 T_MARC-WERKS.
        WRITE 80 T_MARC-NAME1.
        WRITE 120 T_MARC-LGORT.
        WRITE 140 T_MARC-LFGJA.
        WRITE 150 T_MARC-LFMON.
        WRITE 160 T_MARC-LABST.
    * HIDE com os campos que queiramos mostrar na nova janela. Deve ser
    * colocado depois de imprimir linha
        HIDE :T_MARC-MATNR,T_MARC-MAKTX,T_MARC-WERKS, T_MARC-NAME1,
T_MARC-LGORT,T_MARC-LABST, T_MARC-SPEME.
    * o comando AT END OF campo, faz com que quando houver uma quebra de
    * centro ( no caso werks ) o programa entre e execute as instruções
    * contidas dentro dele.
        AT END OF werks.
    * o comando SUM totaliza os campos numéricos
        SUM.
        WRITE : /5 'TOTAL DO MATERIAL : ',t_marc-maktx.
        WRITE 160 T_MARC-LABST.
    ENDAT.

    ENDLOOP.

ENDFORM.                " F_MOSTRAR_DADOS
*&-----*
*&      Form  F_CABECA
*&-----*
*      text
*-----*
* --> p1      text

```

```

* <-- p2      text
*-----*
FORM F_CABECA .

    DATA novo(60) TYPE C.

    WRITE 5 sy-datum.
    WRITE 50 'CHEMYUNION QUIMICA LTDA'.
    WRITE 120 sy-pagno.
    WRITE AT /40 text-002.
    WRITE AT /5 'Material'.
    WRITE 18 'Descrição'.
    WRITE 60 'Centro'.
    WRITE 80 'Descrição'.
    WRITE 120 'DEPOSTIO'.
    WRITE 140 'ANO '.
    WRITE 150 'MES '.
    WRITE 180 'QTD '.
    ULINE.
ENDFORM.                " F_CABECA
*&-----*
*&      Form  F_BUSCAR_MATERIAL
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM F_BUSCAR_MATERIAL .

    SELECT * FROM marc UP TO 1 ROWS WHERE matnr IN s_materi.
    ENDSELECT.

    IF SY-SUBRC <> 0.
        MESSAGE W001.
    ENDIF.

    SELECT * FROM t001L UP TO 1 ROWS WHERE lgort IN s_deposi.
    ENDSELECT.

    IF SY-SUBRC <> 0.
        MESSAGE W002.
    ENDIF.

ENDFORM.                " F_BUSCAR_MATERIAL

```

Programa 6

```

*&-----*
*& Report  ZC00109                                     *
*&                                                     *
*&-----*
*& PROGRAMA PARA LER ARQUIVO EXTERNO E IMPORTAR PARA TABELA *
*&                                                     *
*&-----*

REPORT  ZC00109

* TABELA QUE IRA CONTER OS DADOS IMPORTADOS
TABLES : ZTAB1_01.

* TABELA INTERNA PARA MANIPULAR OS DADOS
DATA : BEGIN OF t_tabela OCCURS 0,
        texto01(100) TYPE C,
      END OF t_tabela.

* COMO CRIAR UMA TABELA INTERNA COPIANDO A ESTRUTURA DA TABELA INTERNA.
DATA : BEGIN OF t_tempo OCCURS 0.
        INCLUDE STRUCTURE ztab1_01.
DATA : END OF t_tempo.

* CRIA UMA CONSTANTE caractere com o valor ';'
CONSTANTS c_tipo TYPE C VALUE ';'.

SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001.
* nome do parametro para importar o arquivo -> RLGRAP-FILENAME
PARAMETERS P_ARQ LIKE RLGRAP-FILENAME.
SELECTION-SCREEN END OF BLOCK b_janela.

START-OF-SELECTION.

PERFORM F_CARREGAR_DADOS.

PERFORM F_DADOS_P_TAB_INTERNA.

PERFORM F_ATUALIZAR_DADOS.

*&-----*
*&      Form  F_CARREGAR_DADOS                         *
*&-----*
*      text
*-----*
*  -->  p1      text
*  <--  p2      text
*-----*

FORM F_CARREGAR_DADOS .

* limpar a tabela de entrada e o read line da tabela
REFRESH t_tabela.
CLEAR   t_tabela.

* FUNÇÃO PARA IMPORTAR ARQUIVO
* chamar a função usar o botão MODELO
CALL FUNCTION 'WS_UPLOAD'
  EXPORTING
    *  CODEPAGE          = ' '
    *  FILENAME          = P_ARQ " COLOCAR O NOME DA VARIÁVEL
    *                   " IRA CONTER O NOME DO ARQUIVO
    *  FILETYPE          = 'ASC' " TIPO DE ARQUIVO
    *  HEADLEN           = ' '
    *  LINE_EXIT         = ' '
    *  TRUNCLEN          = ' '
    *  USER_FORM         = ' '

```



```

*   USER_PROG                = ' '
*   DAT_D_FORMAT             = ' '
* IMPORTING
*   FILELENGTH               =
TABLES
  DATA_TAB                  = t_tabela " NOME DA TABELA INTERNA QUE
                                   " IRA RECEBER OS DADOS

* CODIGOS DE ERROS DE RETONO

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
  NO_AUTHORITY              = 10
  OTHERS                    = 11.

* tratamento dos erros.
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

ENDFORM.                  " F_CARREGAR_DADOS

*&-----*
*&      Form  F_DADOS_P_TAB_INTERNA
*&-----*
*      text
*-----*
*      --> p1      text
*      <-- p2      text
*-----*
FORM F_DADOS_P_TAB_INTERNA .
* LIMPAR A TABELA INTERNA
REFRESH t_tempo.
CLEAR t_tempo.

LOOP AT t_tabela.
* limpar o header line da tabela
  CLEAR t_tempo.
* sy-mandt -> volta o mandante do sistema.
  t_tempo-mandt = sy-mandt.
* SPLIT tabela_interna AT delimitador_do_arquivo INTO
* campos que irão receber os dados na sequência do
* arquivo
  SPLIT t_tabela-texto01 AT c_tipo INTO t_tempo-codigo
                                         t_tempo-nome.
* APPEND nome da tabela que ira guardar os dados importados
* gravar na tabela

* coloca zeros na frente do numero
UNPACK t_tempo-codigo TO t_tempo-codigo.
APPEND t_tempo.

ENDLOOP.

ENDFORM.                  " F_DADOS_P_TAB_INTERNA
*&-----*
*&      Form  F_ATUALIZAR_DADOS
*&-----*
*      text

```

```
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM F_ATUALIZAR_DADOS .

LOOP AT t_tempo.

* move o conteúdo de uma tabela para outra tabela
* apenas os campos com o mesmo nome ( variáveis )
  MOVE-CORRESPONDING t_tempo TO ztab1_01.

* INSERE O CONTEUDO DA VARIÁVEL PARA A TABELA ( GRAVAR )
  INSERT ztab1_01.
* UPDATE tabela interna
* DELETE tabela interna

ENDLOOP.
* EFETIVA AS ALTERAÇÕES NO BANCO
COMMIT WORK.

ENDFORM.                " F_ATUALIZAR_DADOS
```

Programa 7

```

*&-----*
*& Report  ZC00110                                     *
*&-----*
*& PROGRAMA PARA BATCH IMPUT                           *
*&-----*

REPORT  ZC00110.
* Tabela interna para receber dados importados
DATA : BEGIN OF t_tabela OCCURS 0,
        texto01(1000) TYPE C,
        END OF t_tabela.

DATA: BEGIN OF T_TEMPO OCCURS 0,
        BUKRS TYPE RF02K-BUKRS,
        EKORG TYPE RF02K-EKORG,
        KTOKK TYPE RF02K-KTOKK,
        NAME1 TYPE LFA1-NAME1,
        SORTL TYPE LFA1-SORTL,
        ORT01 TYPE LFA1-ORT01,
        PSTLZ TYPE LFA1-PSTLZ,
        LAND1 TYPE LFA1-LAND1,
        REGIO TYPE LFA1-REGIO,
        SPRAS TYPE LFA1-SPRAS,
        STCD1 TYPE LFA1-STCD1,
        TXJCD TYPE LFA1-TXJCD,
        BANKS TYPE LFBK-BANKS,
        BANKL TYPE LFBK-BANKL,
        BANKN TYPE LFBK-BANKN,
        KOINH TYPE LFBK-KOINH,
        AKONT TYPE LFB1-AKONT,
        FDGRV TYPE LFB1-FDGRV,
        REPRF TYPE LFB1-REPRF,
        ZWELS TYPE LFB1-ZWELS,
        WAERS TYPE LFM1-WAERS,
        ZTERM TYPE LFM1-ZTERM,
        WEBRE TYPE LFM1-WEBRE,
        END OF T_TEMPO.

* COMO CRIAR UMA TABELA INTERNA COPIANDO A ESTRUTURA DA TABELA INTERNA.
* para BATCH IMPUT devemos usar a tabela BDCDATA
DATA : BEGIN OF t_bdc OCCURS 0.
        INCLUDE STRUCTURE bdcdata.
DATA : END OF t_bdc.

* para BATCH IMPUT devemos usar a tabela BDCMSGCOLL para
* guardar as mensagens de erro
DATA : BEGIN OF t_msg OCCURS 0.
        INCLUDE STRUCTURE BDCMSGCOLL.
DATA : END OF t_msg.

* delimitador do arquivo txt
CONSTANTS c_tipo TYPE C VALUE ';' .

SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001.
* nome do parametro para importar o arquivo -> RLGRAP-FILENAME
PARAMETERS P_ARQ LIKE RLGRAP-FILENAME.
SELECTION-SCREEN END OF BLOCK b_janela.

START-OF-SELECTION.

        PERFORM F_IMPORTAR_DADOS.

        PERFORM F_CARREGAR_DADOS.

        PERFORM F_BATCH_IMPUT.

```

```

PERFORM F_TRANSACAO_XK01.

*&-----*
*&      Form  Z_IMPORTAR_DADOS
*&-----*
*      text
*-----*
FORM F_IMPORTAR_DADOS .

* limpar a tabela de entrada e o header line da tabela
REFRESH t_tabela.
CLEAR   t_tabela.

* FUNÇÃO PARA IMPORTAR ARQUIVO
* chamar a função usar o botão MODELO
CALL FUNCTION 'WS_UPLOAD'
EXPORTING
*   CODEPAGE                = ' '
   FILENAME                = P_ARQ  " COLOCAR O NOME DA VARIÁVEL
                                " IRA CONTER O NOME DO ARQUIVO
   FILETYPE                = 'ASC'  " TIPO DE ARQUIVO
*   HEADLEN                = ' '
*   LINE_EXIT              = ' '
*   TRUNCLEN              = ' '
*   USER_FORM             = ' '
*   USER_PROG             = ' '
*   DAT_D_FORMAT          = ' '
* IMPORTING
*   FILELENGTH            =
   TABLES
   DATA_TAB              = t_tabela " NOME DA TABELA INTERNA QUE
                                " IRA RECEBER OS DADOS

* CODIGOS DE ERROS DE RETONO

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
  NO_AUTHORITY            = 10
  OTHERS                  = 11.

* tratamento dos erros.
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

ENDFORM.                " F_IMPORTAR_DADOS

*&-----*
*&      Form  F_CARREGAR_DADOS
*&-----*
*      text
*-----*
FORM F_CARREGAR_DADOS .
* LIMPAR A TABELA INTERNA
REFRESH t_tempo.
CLEAR t_tempo.

LOOP AT t_tabela.

```

```

* limpar o header line da tabela
  CLEAR t_tempo.
* SPLIT tabela_interna AT delimitador_do_arquivo INTO
* campos que irão receber os dados na sequência do
* arquivo
  SPLIT t_tabela-texto01 AT c_tipo INTO T_TEMPO-BUKRS
                                         T_TEMPO-EKORG
                                         T_TEMPO-KTOKK
                                         T_TEMPO-NAME1
                                         T_TEMPO-SORTL
                                         T_TEMPO-ORT01
                                         T_TEMPO-PSTLZ
                                         T_TEMPO-LAND1
                                         T_TEMPO-REGIO
                                         T_TEMPO-SPRAS
                                         T_TEMPO-STCD1
                                         T_TEMPO-TXJCD
                                         T_TEMPO-BANKS
                                         T_TEMPO-BANKL
                                         T_TEMPO-BANKN
                                         T_TEMPO-KOINH
                                         T_TEMPO-AKONT
                                         T_TEMPO-FDGRV
                                         T_TEMPO-REPRF
                                         T_TEMPO-ZWELS
                                         T_TEMPO-WAERS
                                         T_TEMPO-ZTERM
                                         T_TEMPO-WEBRE.

* APPEND nome da tabela que ira guardar os dados importados gravar na tabela

  APPEND t_tempo.

  ENDLLOOP.

ENDFORM.                " F_CARREGAR_DADOS

*&-----*
*&      Form  F_BATCH_INPUT
*&-----*
*      text
*-----*
FORM F_BATCH_INPUT .

LOOP AT t_tempo.

* PERFORM PARA PREENCHER A tabela BDC
*   PERFORM F_BDC_TELA USING programa tela.
*   PERFORM F_BDC_CAMPO USING campo valor.
* devemos passar os parâmetros obtidos na transação SHDB

  PERFORM F_BDC_TELA USING 'SAPMF02K' '100'.
  PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'RF02K-REF_LIFNR'.
  PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
  PERFORM F_BDC_CAMPO USING 'RF02K-BUKRS' T_TEMPO-BUKRS.
  PERFORM F_BDC_CAMPO USING 'RF02K-EKORG' T_TEMPO-EKORG.
  PERFORM F_BDC_CAMPO USING 'RF02K-KTOKK' T_TEMPO-KTOKK.

  PERFORM F_BDC_TELA USING 'SAPMF02K' '110'.
  PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFA1-TELX1'.
  PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
  PERFORM F_BDC_CAMPO USING 'LFA1-NAME1' T_TEMPO-NAME1.
  PERFORM F_BDC_CAMPO USING 'LFA1-SORTL' T_TEMPO-SORTL.
  PERFORM F_BDC_CAMPO USING 'LFA1-ORT01' T_TEMPO-ORT01.
  PERFORM F_BDC_CAMPO USING 'LFA1-PSTLZ' T_TEMPO-PSTLZ.
  PERFORM F_BDC_CAMPO USING 'LFA1-LAND1' T_TEMPO-LAND1.
  PERFORM F_BDC_CAMPO USING 'LFA1-REGIO' T_TEMPO-REGIO.

```

```

PERFORM F_BDC_CAMPO USING 'LFA1-SPRAS' T_TEMPO-SPRAS.

PERFORM F_BDC_TELA USING 'SAPMF02K' '120'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFA1-STCD3'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
PERFORM F_BDC_CAMPO USING 'LFA1-STCD1' T_TEMPO-STCD1.
PERFORM F_BDC_CAMPO USING 'LFA1-TXJCD' T_TEMPO-TXJCD.

PERFORM F_BDC_TELA USING 'SAPMF02K' '130'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFBK-KOINH(01)'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
PERFORM F_BDC_CAMPO USING 'LFBK-BANKS(01)' T_TEMPO-BANKS.
PERFORM F_BDC_CAMPO USING 'LFBK-BANKL(01)' T_TEMPO-BANKL.
PERFORM F_BDC_CAMPO USING 'LFBK-BANKN(01)' T_TEMPO-BANKN.
PERFORM F_BDC_CAMPO USING 'LFBK-KOINH(01)' T_TEMPO-KOINH.

PERFORM F_BDC_TELA USING 'SAPMF02K' '130'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFBK-BANKS(01)'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '=ENTR'.

PERFORM F_BDC_TELA USING 'SAPMF02K' '210'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFB1-FDGRV'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
PERFORM F_BDC_CAMPO USING 'LFB1-AKONT' T_TEMPO-AKONT.
PERFORM F_BDC_CAMPO USING 'LFB1-FDGRV' T_TEMPO-FDGRV.

PERFORM F_BDC_TELA USING 'SAPMF02K' '215'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFB1-REPRF'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
PERFORM F_BDC_CAMPO USING 'LFB1-REPRF' T_TEMPO-REPRF.
PERFORM F_BDC_CAMPO USING 'LFB1-ZWELS' T_TEMPO-ZWELS.

PERFORM F_BDC_TELA USING 'SAPMF02K' '220'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFB5-LFRMA'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.

PERFORM F_BDC_TELA USING 'SAPMF02K' '610'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'RF02K-LIFNR'.

PERFORM F_BDC_TELA USING 'SAPMF02K' '310'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'LFM1-WEBRE'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '/00'.
PERFORM F_BDC_CAMPO USING 'LFM1-WAERS' T_TEMPO-WAERS.
PERFORM F_BDC_CAMPO USING 'LFM1-ZTERM' T_TEMPO-ZTERM.
PERFORM F_BDC_CAMPO USING 'LFM1-WEBRE' T_TEMPO-WEBRE.

PERFORM F_BDC_TELA USING 'SAPMF02K' '320'.
PERFORM F_BDC_CAMPO USING 'BDC_CURSOR' 'RF02K-LIFNR'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '=ENTR'.

PERFORM F_BDC_TELA USING 'SAPLSP01' '300'.
PERFORM F_BDC_CAMPO USING 'BDC_OKCODE' '=YES'.

```

ENDLOOP.

* EFETIVA AS ALTERAÇÕES NO BANCO

COMMIT WORK.

ENDFORM. " F_BATCH_INPUT

```

*&-----*
*&      Form  F_BDC_TELA
*&-----*
*      text
*-----*
FORM F_BDC_TELA USING P_PROGRAMA
                     P_TELA.

```

```

* limpa o head line
CLEAR t_bdc.

* preencher os campos da tabela
t_bdc-program = P_PROGRAMA.
t_bdc-dynpro  = P_TELA.
t_bdc-dynbegin = 'X'.

APPEND t_bdc.

ENDFORM.                " F_BDC_TELA
*&-----*
*&      Form  F_BDC_CAMPO
*&-----*
*      text
*-----*
FORM F_BDC_CAMPO USING   P_CAMPO
                       P_VALOR.

* limpa o read liner
CLEAR t_bdc.

t_bdc-fnam = P_campo.
t_bdc-fval = P_valor.

APPEND t_bdc.

ENDFORM.                " F_BDC_CAMPO
*&-----*
*&      Form  F_TRANSACAO_XK01
*&-----*
*      text
*-----*
FORM F_TRANSACAO_XK01 .

REFRESH T_MSG.
CLEAR T_MSG.

* COMANDO PARA CHAMAR TRANSAÇÃO
* CALL TRANSACTION 'transação' USING tabela_interna_com_os_dados MODE
* 'tp' UPDATE 'tp2' MESSAGES INTO tabela_mensagens
* tp -> A = DISPLAY
* E = ERROS
* N = NÃO MOSTRAR
* P = NÃO MOSTRAR / DEBUGAR
* tp2 = A -> ASSINCRONO
* S = SINCRONO
* L = LOCAL UPDATE
CALL TRANSACTION 'XK01' USING T_BDC MODE 'A'
                       UPDATE 'S' MESSAGES INTO T_MSG.

ENDFORM.                " F_TRANSACAO_XK01

```

Programa 8

```

*&-----*
*& Report  ZC00111                                     *
*&                                                     *
*&-----*
*& programa para fazer download de tabelas para arquivo txt *
*&                                                     *
*&-----*

REPORT  ZC00111 LINE-SIZE 200 LINE-COUNT 65 NO STANDARD PAGE HEADING
MESSAGE-ID Z01.

TABLES : MARD, MAKT, T001W, T001L.

DATA : BEGIN OF T_MARD OCCURS 0,
        MATNR LIKE MARD-MATNR,
        WERKS LIKE MARD-WERKS,
        NAME1 LIKE T001W-NAME1,
        MAKTX LIKE MAKT-MAKTX,
        LGORT LIKE MARD-LGORT,
        LGOBE LIKE T001L-LGOBE,
        LABST LIKE MARD-LABST,
        SPEME LIKE MARD-SPEME,
      END OF T_MARD.

DATA : BEGIN OF T_001 OCCURS 0,
        WERKS LIKE T001W-WERKS,
      END OF T_001.

* PARA COLOCARMOS UM DELIMITAR TIPO (;) DEVEMOS CRIAR UMA TABELA INTERNA
* COM OS CAMPOS QUE IREMOS EXPORTAR ( TODOS TXT ) E MOVER O CONTEUDO DA
* TABELA INTERNA PARA A TABELA DE EXPORTAÇÃO

* VARIAVEL PARA CONVERTER O PARAMETRO p_arq EM string
* para passarmos na função gui_download.
DATA v_arquivo TYPE STRING.

REFRESH : T_MARD, T_001.
CLEAR    : T_MARD, T_001.

SELECTION-SCREEN BEGIN OF BLOCK b_janela WITH FRAME TITLE text-001.
  SELECT-OPTIONS S_MATERI FOR T_MARD-MATNR.
  SELECT-OPTIONS S_CENTRO FOR T_MARD-WERKS.
  SELECT-OPTIONS S_DEPOSI FOR T_MARD-LGORT.
SELECTION-SCREEN END OF BLOCK b_janela.

* PARA CRIAR BOTÕES DE SELEÇÃO
*
SELECTION-SCREEN BEGIN OF BLOCK b_janela3 WITH FRAME TITLE text-003.
  PARAMETERS P_BOT1 RADIOBUTTON GROUP b1.
  PARAMETERS P_BOT2 RADIOBUTTON GROUP b1.
SELECTION-SCREEN END OF BLOCK b_janela3.

SELECTION-SCREEN BEGIN OF BLOCK b_janela2 WITH FRAME TITLE text-002.
* OBLIGATORY faz com que o parametro seja obrigatório
PARAMETERS P_ARQ LIKE RLGRAP-FILENAME. " OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b_janela2.

INITIALIZATION.

  PERFORM F_SELECIONAR.

AT SELECTION-SCREEN ON s_materi.

AT SELECTION-SCREEN ON s_deposi.

```



```

    PERFORM F_BUSCAR_MATERIAL.

TOP-OF-PAGE.

    PERFORM F_CABECA.

END-OF-PAGE.

START-OF-SELECTION.

    PERFORM F_BUSCAR_DADOS.

END-OF-SELECTION.

* VERIFICA SE A VRIAVEL P_ARQ ESTA LIMPA, SE SIM MOSTRA NA TELA SE NÃO GERA ARQUIVO
IF P_ARQ IS NOT INITIAL.

    v_arquivo = p_arq.

    IF p_bot1 = 'X'.

        PERFORM F_DOWNLOAD_ARQ.
    ELSE.
* somente para UNIX NÃO FUNCIONA EM WINDOWS
        PERFORM F_DOWN_UNIX.
    ENDIF.
ELSE.

    PERFORM F_MOSTRAR_DADOS.
ENDIF.

* o comando AT LINE-SELECTION é um evento que serve para
* quando clicarmos em uma linha de relatório abrir uma outra
* janela e mostrarmos outros dados
AT LINE-SELECTION.
* IF NOT var IS INITIAL -> verifica se o header liner esta vazio
* se não for vazio imprime a tela de detalhes
IF NOT T_MARD IS INITIAL.
* SET TITLEBAR 'var' mostra o titulo da nova tela
    SET TITLEBAR 'DET'.
* SY-ULINE -> imprime linha com comprimento fixo
    WRITE SY-ULINE(83).
* SY-VLINE -> serve para fazer fechamento de bordas
    WRITE / SY-VLINE.
    WRITE : 'Material      : '.
    WRITE : T_MARD-MATNR, ' - ', t_marc-maktx, ' '.
    WRITE 83 SY-VLINE.
    WRITE / SY-VLINE.
    WRITE : 'Centro        : '.
    WRITE : t_marc-werks, ' - ', t_mard-name1, ' '.
    WRITE 83 SY-VLINE.
    WRITE /:SY-VLINE.
    WRITE : 'Deposito      : '.
    WRITE : T_MARD-lgort.
    WRITE 83 SY-VLINE.
    WRITE /:SY-VLINE.
    WRITE : 'Disponível : '.
    WRITE : T_MARD-labst.
    WRITE 83 SY-VLINE.
    WRITE /:SY-VLINE.
    WRITE : 'Bloqueado : '.
    WRITE : T_MARD-SPEME.
    WRITE 83 SY-VLINE.
    WRITE / SY-ULINE(83).
ENDIF.
* depois de listar a nova informação devemos limpar o header liner da tabela
CLEAR T_MARD.

```

```

*&-----*
*&      Form  F_SELECIONAR
*&-----*
*-----*
FORM F_SELECIONAR .
  SELECT werks INTO TABLE t_001 FROM t001w.

  REFRESH S_CENTRO.

  LOOP AT T_001.
    clear S_CENTRO.
    s_centro-sign = 'I'.
    s_centro-option = 'EQ'.
    s_centro-low = t_001-werks.
    APPEND S_CENTRO.
  ENDLOOP.

ENDFORM.                    " F_SELECIONAR
*&-----*
*&      Form  F_BUSCAR_DADOS
*&-----*
*      text
*-----*
FORM F_BUSCAR_DADOS .

  SELECT mard~matnr mard~werks t001w~name1 makt~maktx mard~lgort
         T001L~lgobe mard~labst mard~speme
         INTO TABLE t_mard
         FROM mard
         INNER JOIN t001w ON mard~werks = t001w~werks
         INNER JOIN makt ON mard~matnr = makt~matnr
                        and makt~spras = sy-langu
         INNER JOIN t001L ON mard~werks = t001L~werks
                        AND mard~lgort = t001L~lgort
         WHERE t001w~werks IN S_CENTRO
         AND mard~matnr IN S_MATERI
         AND mard~lgort IN S_DEPOSI.

  IF sy-subrc <> 0.
    MESSAGE S001.
  ENDIF.

ENDFORM.                    " F_BUSCAR_DADOS
*&-----*
*&      Form  F_MOSTRAR_DADOS
*&-----*
*      text
*-----*
FORM F_MOSTRAR_DADOS .
  SORT T_MARD BY matnr werks lgort.
  * comando LOOP AT tabela serve para criar um loop até o final da tabela
  * para podermos imprimir seu conteúdo.
  LOOP AT T_MARD.
    WRITE / T_MARD-MATNR.
    WRITE 13 T_MARD-MAKTX.
    WRITE 55 T_MARD-WERKS.
    WRITE 65 T_MARD-NAME1(20).
    WRITE 89 T_MARD-LGORT.
    WRITE 95 T_MARD-LGOBE(10).
    WRITE 105 T_MARD-LABST.
    WRITE 122 T_MARD-SPEME.
  * HIDE com os campos que queiramos mostrar na nova janela. Deve ser
  * colocado depois de imprimir linha
    HIDE :T_MARD-MATNR,T_MARD-MAKTX,T_MARD-WERKS, T_MARD-NAME1,
T_MARD-LGORT,T_MARD-LABST, T_MARD-SPEME.
  * o comando AT END OF campo, faz com que quando houver uma quebra de

```

```

* centro ( no caso werks ) o programa entre e execute as instruções
* contidas dentro dele.
  AT END OF werks.
* o comando SUM totaliza os campos numéricos
  SUM.
  WRITE : / 'TOTAL DO MATERIAL '.
  WRITE '.....:'.
  WRITE 105 T_MARD-LABST.
  WRITE 122 T_MARD-SPEME.
  ENDAT.

  ENDLLOOP.

ENDFORM.                  " F_MOSTRAR_DADOS
*&-----*
*&      Form  F_CABECA
*&-----*
*      text
*-----*
FORM F_CABECA .

  DATA novo(60) TYPE C.
  WRITE 5 sy-datum.
  WRITE 50 'CHEMYUNION QUIMICA LTDA'.
  WRITE 120 sy-pagno.
  WRITE AT / 'Material'.
  WRITE 13 'Descrição'.
  WRITE 55 'Centro'.
  WRITE 65 'Descrição'.
  WRITE 89 'DEPOSITO.'.
  WRITE 115 'Estoque'.
  WRITE 130 'Bloqueado'.
  ULINE.
ENDFORM.                  " F_CABECA
*&-----*
*&      Form  F_BUSCAR_MATERIAL
*&-----*
*      text
*-----*
FORM F_BUSCAR_MATERIAL .

  SELECT * FROM mard UP TO 1 ROWS WHERE matnr IN s_materi.
  ENDSELECT.
  IF SY-SUBRC <> 0.
    MESSAGE W001.
  ENDIF.
  SELECT * FROM t001L UP TO 1 ROWS WHERE lgort IN s_deposi.
  ENDSELECT.

  IF SY-SUBRC <> 0.
    MESSAGE W002.
  ENDIF.

ENDFORM.                  " F_BUSCAR_MATERIAL
*&-----*
*&      Form  F_DOWNLOAD_ARQ
*&-----*
*      text
*-----*
FORM F_DOWNLOAD_ARQ .
* FUNÇÃO USADO PARA EXPORTAR ARQUIVOS
CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
*   BIN_FILESIZE          =
  FILENAME                = V_ARQUIVO
  FILETYPE                 = 'ASC'

```

```

* APPEND                                = ' '
* WRITE_FIELD_SEPARATOR                 = ' '
* HEADER                                = '00'
* TRUNC_TRAILING_BLANKS                 = ' '
* WRITE_LF                              = 'X'
* COL_SELECT                            = ' '
* COL_SELECT_MASK                       = ' '
* DAT_MODE                              = ' '
* CONFIRM_OVERWRITE                     = ' '
* NO_AUTH_CHECK                         = ' '
* IMPORTING
* FILELENGTH                            =
TABLES
  DATA_TAB                             = T_MARD
EXCEPTIONS
  FILE_WRITE_ERROR                      = 1
  NO_BATCH                              = 2
  GUI_REFUSE_FILETRANSFER               = 3
  INVALID_TYPE                          = 4
  NO_AUTHORITY                          = 5
  UNKNOWN_ERROR                         = 6
  HEADER_NOT_ALLOWED                    = 7
  SEPARATOR_NOT_ALLOWED                 = 8
  FILESIZE_NOT_ALLOWED                  = 9
  HEADER_TOO_LONG                       = 10
  DP_ERROR_CREATE                       = 11
  DP_ERROR_SEND                         = 12
  DP_ERROR_WRITE                        = 13
  UNKNOWN_DP_ERROR                     = 14
  ACCESS_DENIED                         = 15
  DP_OUT_OF_MEMORY                     = 16
  DISK_FULL                            = 17
  DP_TIMEOUT                           = 18
  FILE_NOT_FOUND                       = 19
  DATAPROVIDER_EXCEPTION                = 20
  CONTROL_FLUSH_ERROR                   = 21
  OTHERS                               = 22
.
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ELSE.
* MESSAGE xxx WITH yyy
* xxx numero da mensagem
* yyy parametro que será mostrado na mensagem
* na criação da mensagem devemos colocar o símbolo & para
* indicar que será passado um parametro.
* EXEMPLO : Arquivo & criado com sucesso.
  MESSAGE S003 WITH P_ARQ.
ENDIF.

ENDFORM.                                " F_DOWNLOAD_ARQ
*&-----*
*&      Form  F_DOWN_UNIX
*&-----*
*      text
*-----*
FORM F_DOWN_UNIX .
* abre o arquivo para gravar em txt
* OPEN DATASET parametro FOR OUTPUT IN TEXT MODE.
* ou INPUT para ler o arquivo ( importar )
OPEN DATASET p_arq FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.

IF sy-subrc = 0.
  MESSAGE S004.
ELSE.
  MESSAGE W005.

```

```
ENDIF.

DATA texto01(20)  type C.
DATA texto02(40)  type C.
DATA texto03(04)  type C.
DATA texto04(20)  type C.
DATA texto05(04)  type C.
DATA texto06(20)  type C.
DATA texto07(18)  type C.
DATA texto08(18)  type C.
DATA texto09(200) type C.

* é necessário fazer loop da tabela interna
LOOP AT t_mard.

* TRANSFER tabela T0 parametro arquivo
* serve para ler e gravar o arquivo externo
    texto01 = t_mard-matnr.
    texto02 = t_mard-maktx.
    texto03 = t_mard-werks.
    texto04 = t_mard-name1.
    texto05 = t_mard-lgort.
    texto06 = t_mard-lgobe.
    texto07 = t_mard-labst.
    texto08 = t_mard-speme.
    CONCATENATE texto01 texto02 texto03
                texto04 texto05 texto06
                texto07 texto08 INTO texto09.

    TRANSFER texto09 TO p_arq.

* READ DATASET parametro INTO tabela interna
* serve para ler e importar tabela interna
* READ DATASET p_arq INTO arquivo ou variável.

ENDLOOP.

* para fechar o arquivo externo
* CLOSE DATASET parametro
CLOSE DATASET p_arq.

ENDFORM.                " F_DOWN_UNIX
```

Programa 9

```

*&-----*
*& Report   ZC00114                                     *
*&-----*
*& PROGRAMA PARA EXECUTAR COMANDOS EXTERNOS             *
*& EXECUTA O COMANDO NO SERVIDOR                       *
*&-----*
REPORT   ZC00114
* DEFINIR TABELA INTERNA COM A STRUCTURA DA TABELA BTCXPM
DATA : BEGIN OF t_btc OCCURS 0.
      INCLUDE STRUCTURE   BTCXPM.
DATA : END OF t_btc.
* DECLARAR UMA VARIÁVEL COM O TIPO SXPGCOLIST-NAME
DATA wvar LIKE   SXPGCOLIST-NAME.
* IGUALAR ESTA VARIÁVEL AO COMANDO CRIADO NA SM69 PASSAR COMO PARAMETRO
wvar = 'ZCOMPLEX'.
INITIALIZATION.
START-OF-SELECTION.
  PERFORM F_COMANDO_EXTERNO.
  PERFORM F_MOSTRAR_TABELA.
END-OF-SELECTION.
*&-----*
*&       Form   F_COMANDO_EXTERNO                       *
*&-----*
FORM F_COMANDO_EXTERNO .
* FUNÇÃO QUE IRA EXECUTAR O PARAMETRO
CALL FUNCTION 'SXPG_CALL_SYSTEM'
  EXPORTING
    COMMANDNAME                = wvar
*   ADDITIONAL_PARAMETERS      = ' '
* IMPORTING
*   STATUS                     =
*   EXITCODE                   =
  TABLES
    EXEC_PROTOCOL              = t_btc
EXCEPTIONS
  NO_PERMISSION                = 1
  COMMAND_NOT_FOUND            = 2
  PARAMETERS_TOO_LONG          = 3
  SECURITY_RISK                 = 4
  WRONG_CHECK_CALL_INTERFACE    = 5
  PROGRAM_START_ERROR           = 6
  PROGRAM_TERMINATION_ERROR     = 7
  X_ERROR                      = 8
  PARAMETER_EXPECTED            = 9
  TOO_MANY_PARAMETERS           = 10
  ILLEGAL_COMMAND               = 11
  OTHERS                       = 12
.
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
ENDFORM.              " F_COMANDO_EXTERNO
*&-----*
*&       Form   F_MOSTRAR_TABELA                         *
*&-----*
FORM F_MOSTRAR_TABELA .
  CLEAR T_BTC.
  LOOP AT T_BTC.
    WRITE / T_BTC.
  ENDLOOP.
ENDFORM.              " F_MOSTRAR_TABELA

```

Programa 10

```

*&-----*
*& Include          ZC00115                      *
*&-----*

REPORT ZC00115.

DATA : BEGIN OF T_tabela OCCURS 0,
        campo(20) TYPE C,
      END OF T_tabela.
DATA : BEGIN OF T_vendedor OCCURS 0,
        nome(10) TYPE C,
        tv      TYPE I,
        video   TYPE I,
        radio   TYPE I,
        dvd     TYPE I,
      END OF T_vendedor.

INITialization.

PERFORM F_CARREGA_DADOS.

START-OF-SELECTION.
  PERFORM F_GRAFICO.
END-OF-SELECTION.

*&-----*
*&      Form  F_CARREGA_DADOS                     *
*&-----*
*      text
*-----*

FORM F_CARREGA_DADOS .
REFRESH t_tabela.
CLEAR t_tabela.

t_tabela-campo = 'FIFRST=3D'.
APPEND t_tabela.
t_tabela-campo = 'P3TYPE=T0'.
APPEND t_tabela.
t_tabela-campo = 'P3CTYP=R0'.
APPEND t_tabela.
t_tabela-campo = 'TISIZE=2'.
APPEND t_tabela.
t_tabela-campo = 'CLBACK=X'.
APPEND t_tabela.
COMMIT WORK.

REFRESH t_vendedor.
CLEAR t_vendedor.

t_vendedor-nome = 'ZE'.
t_vendedor-tv   = 98.
t_vendedor-VIDEO = 83.
t_vendedor-radio = 45.
t_vendedor-dvd  = 65.
APPEND t_vendedor.
t_vendedor-nome = 'MANE'.
t_vendedor-tv   = 52.
t_vendedor-VIDEO = 38.
t_vendedor-radio = 23.
t_vendedor-dvd  = 5.
APPEND t_vendedor.
t_vendedor-nome = 'JOAO'.
t_vendedor-tv   = 73.
t_vendedor-VIDEO = 54.
t_vendedor-radio = 35.

```

```

t_vendedor-dvd      = 49.
APPEND t_vendedor.
COMMIT WORK.

ENDFORM.                  " F_CARREGA_DADOS
*&-----*
*&      Form  F_GRAFICO
*&-----*
*      text
*-----*
FORM F_GRAFICO .
* FUNÇÃO PARA CRIAR GRAFICOS
CALL FUNCTION 'GRAPH_MATRIX_3D'
EXPORTING
*   AUTO_CMD_1      = ' '
*   AUTO_CMD_2      = ' '
*   COL1            = 'TV'
*   COL2            = 'VIDEO CASSETE'
*   COL3            = 'RADIO'
*   COL4            = 'DVD'
*   COL5            = ' '
*   COL6            = ' '
*   DIM1            = 'PRODUTOS'
*   DIM2            = 'VENDEDOR'
*   INBUF           = ' '
*   INFORM          = '6'
*   MAIL_ALLOW      = ' '
*   PWDID           = ' '
*   SET_FOCUS       = 'x'
*   SMFONT          = ' '
*   SO_CONTENTS     = ' '
*   SO_RECEIVER     = ' '
*   SO_SEND         = ' '
*   SO_TITLE        = ' '
*   STAT            = ' '
*   SUPER           = ' '
*   TIMER           = ' '
*   TITL            = 'VENDAS 2003'
*   VALT            = 'QTD'
*   WDID            = ' '
*   WINID           = ' '
*   WINPOS          = '5'
*   WINSZX          = '5'
*   WINSZY          = '70'
*   X_OPT           = ' '
*   NOTIFY          = ' '
* IMPORTING
*   B_KEY           =
*   B_TYP           =
*   CUA_ID          =
*   MOD_COL         =
*   MOD_ROW         =
*   MOD_VAL         =
*   M_TYP           =
*   RBUFF           =
*   RWNID           =
TABLES
*   DATA           = t_vendedor
*   OPTS            = t_tabela.

ENDFORM.                  " F_GRAFICO

```


Programa 11

```

*&-----*
*& Report  ZC00116                                     *
*&-----*

REPORT  ZC00116 LINE-SIZE 80 LINE-COUNT 65 NO STANDARD PAGE HEADING.

TABLES T001L.
DATA: BEGIN OF T_001L OCCURS 0,
      WERKS TYPE T001L-WERKS,
      LGORT TYPE T001L-LGORT,
      LGOBE TYPE T001L-LGOBE,
      END OF T_001L.

REFRESH t_001l.
CLEAR t_001l.

SELECT werks lgort lgobe INTO TABLE t_001l FROM T001L.

* COMANDO PARA CRIAR BOTAO NO RELATORIO
* SET PF-STATUS 'nome_botao'
* DAR DUPLIO CLICK NO nome_do_botão PARA CRIAR O OBJETO

SET PF-STATUS 'BOTA0'.

INITIALIZATION.

TOP-OF-PAGE.
  PERFORM F_CABECALHO.
END-OF-PAGE.

START-OF-SELECTION.

PERFORM f_imprimir_dados.

END-OF-SELECTION.

* EVENTO PARA O BOTAO
AT USER-COMMAND.
  * variável de sistema. retorna o nome do botão (DOWNLOAD)
  IF sy-ucomm = 'DOWNLOAD'.
    PERFORM F_DOW_NOVO.

  ENDIF.

*&-----*
*&      Form  F_CABECALHO                               *
*&-----*
FORM F_CABECALHO .
  WRITE 5 sy-datum.
  WRITE 30 'CHEMYUNION QUIMICA LTDA'.
  WRITE 75 sy-pagno.
  WRITE /2 'Centro'.
  WRITE 10 'Deposito'.
  WRITE 20 'Descrição'.

  ULINE.
ENDFORM.              " F_CABECALHO

*&-----*
*&      Form  f_imprimir_dados                           *
*&-----*
FORM f_imprimir_dados .
  * O FORM QUE VAI SER CHAMADO POR OUTRO PROGRAMA deve ter todas as infor
  * mações para gerar o form ( ou seja neste caso montar a tabela )

```

```

LOOP AT t_0011.
  WRITE /2 t_0011-werks.
  WRITE 10 t_0011-lgort.
  WRITE 20 t_0011-lgobe.
ENDLOOP.
ENDFORM.                  " f_imprimir_dados
*&-----*
*&      Form  F_DOW_NOVO
*&-----*
*  --> p1      text
*  <-- p2      text
*-----*
FORM F_DOW_NOVO .
*  tambem temos a função UPLOAD

*  ESTA FUNÇÃO VEM COM TELA DE ESCOLHAR ( MENU ABRIR )
*  podemos escolher o formato de arquivo.
CALL FUNCTION 'DOWNLOAD'
*  EXPORTING
*    BIN_FILESIZE           = ' '
*    CODEPAGE               = ' '
*    FILENAME               = ' '
*    FILETYPE               = ' '
*    ITEM                   = ' '
*    MODE                   = ' '
*    WK1_N_FORMAT           = ' '
*    WK1_N_SIZE             = ' '
*    WK1_T_FORMAT           = ' '
*    WK1_T_SIZE             = ' '
*    FILEMASK_MASK          = ' '
*    FILEMASK_TEXT          = ' '
*    FILETYPE_NO_CHANGE     = ' '
*    FILEMASK_ALL           = ' '
*    FILETYPE_NO_SHOW       = ' '
*    SILENT                 = 'S'
*    COL_SELECT             = ' '
*    COL_SELECTMASK         = ' '
*    NO_AUTH_CHECK          = ' '
*  IMPORTING
*    ACT_FILENAME           =
*    ACT_FILETYPE           =
*    FILESIZE               =
*    CANCEL                 =
*  TABLES
*    DATA_TAB              = T_001L
*    FIELDNAMES             =
*  EXCEPTIONS
*    INVALID_FILESIZE       = 1
*    INVALID_TABLE_WIDTH    = 2
*    INVALID_TYPE           = 3
*    NO_BATCH               = 4
*    UNKNOWN_ERROR          = 5
*    GUI_REFUSE_FILETRANSFER = 6
*    OTHERS                 = 7
* .
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

ENDFORM.                  " F_DOW_NOVO

```

Programa 12 – Modulo On-line

```

*-----*
***INCLUDE MZTXT01_PAI_9020 .
*-----*
*&-----*
*&      Module  USER_COMMAND_EXIT_9020  INPUT
*&-----*
*      text
*-----*
MODULE USER_COMMAND_EXIT_9020 INPUT.
  PERFORM USER_COMMAND_EXIT_9020.
ENDMODULE.
" USER_COMMAND_EXIT_9020  INPUT
*&-----*
*&      Form  USER_COMMAND_EXIT_9020
*&-----*
*      text
*-----*
*      -->  p1      text
*      <--  p2      text
*-----*
FORM USER_COMMAND_EXIT_9020 .
  DATA v1_okcode LIKE sy-ucomm.
  v1_okcode = vg_okcode_9020.
  CLEAR vg_okcode_9020.

  CASE v1_okcode.
    WHEN 'BACK' OR 'RW'.
* VOLTAR PARA A TELA 9000
      LEAVE TO SCREEN 9010.
    WHEN 'END'.
* SAIR DO PROGRAMA
      LEAVE PROGRAM.
* LEAVE TO SCREEN 9010.
      LEAVE TO SCREEN 9010.
    ENDCASE.

ENDFORM.
" USER_COMMAND_EXIT_9020

```

COMANDO EXECUTADOS DENTRO DA TELA (nome de tela)

```

PROCESS BEFORE OUTPUT.
  MODULE STATUS_9000.
*
PROCESS AFTER INPUT.
* MODULO PARA TRATAR OS BOTAO DA PRIMEIRA TELA
* SO ENTRAR SE OS BOTAO TEM O PARAMETRO "E"
* AT EXIT-COMMAND.
* DEVE ESTAR EM PRIMEIRO
MODULE USER_COMMAND_EXIT_9000 AT EXIT-COMMAND.
* MODULO QUE VAI TRARTAR O BOTÃO CRIADO
MODULE USER_COMMAND_9000.

```

Programa 13

```

*&-----*
*& Report  ZC0307                                     *
*&                                                     *
*&-----*
*& Criar objetos de autorização                         *
*&                                                     *
*&-----*

REPORT  ZC0307 message-id Z001.

TABLES: ZTAB1_03.

DATA: BEGIN OF T_ZTAB103 OCCURS 0,
      TEXT0(100) TYPE C,
END OF T_ZTAB103.

DATA: BEGIN OF T_ZTAB1_03 OCCURS 0.
      INCLUDE STRUCTURE ZTAB1_03.
DATA END OF T_ZTAB1_03.

* CRIANDO UMA CONSTANTE QUE INFORMA DELIMITADOR DO
* ARQUIVOS .TXT QUE SERÁ LIDO PARA O BATCH INPUT

CONSTANTS: C_DELIMITADOR TYPE C VALUE';',
           C_NOME(8)      TYPE C VALUE 'ABAP'.

SELECTION-SCREEN BEGIN OF BLOCK B_001 WITH FRAME TITLE TEXT-001.
  PARAMETERS P_TEXTO LIKE RLGRAP-FILENAME.
SELECTION-SCREEN END OF BLOCK B_001.

START-OF-SELECTION.
  PERFORM F_CARDADOS.
END-OF-SELECTION.

*&-----*
*&      Form  F_CARDADOS                               *
*&-----*
FORM F_CARDADOS .
* data: vl_nome(8) type c.

* vl_nome = sy-uname.

* if vl_nome ne C_NOME.
*   MESSAGE E006 WITH TEXT-E01.
* endif.

* comando para verificar autorização de objeto
* usar junto com su21 e su24

authority-check object 'ZZC0307Y'

* objetos da autorização
* ID 'nome_do_objeto' FIELD 'campo'

* id 'TCD' field 'DUMMY'.
id 'ACTVT' field 'DUMMY'.

IF SY-SUBRC NE 0.
  MESSAGE E006 WITH TEXT-E01.
ENDIF.
ENDFORM.                    " F_CARDADOS

```

Programa 14

```

*&-----*
*& Report  ZEXEMPLO_ALV1                                     *
*&                                                         *
*&-----*
*&                                                         *
*&                                                         *
*&-----*

* -----
*  Descrição : Programa de exemplo para criação de relatório ALV
* -----
report zexemplo_alv1.

* -----
*  Tabelas transparentes
* -----
tables:
  vbrk,           "Documentos de faturamento
  vbrp.           "Documento de faturamento: dados de item

* -----
*  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. - sempre ter
* -----
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,

  begin of y_vbrp,
    posnr      like vbrp-posnr,      "Item do doc.de faturamento
    matnr      like vbrp-matnr,      "Nº do material
    fking      like vbrp-fking,      "Qde.faturada
    vrkme      like vbrp-vrkme,      "Unidade de venda
  end of y_vbrp.

* -----
*  Tabelas internas ALV
* -----
*  As estruturas aqui utilizadas (SLIS) estão explicadas com as opções
*  mais importantes no final da apostila
* -----
*  sempre ter estas tabelas
data:
  t_listheader  type slis_t_listheader,
* para primeira alv
  t_fieldcat    type slis_t_fieldcat_alv      with header line,
* segunda alv

```

```

t_fieldcatvbrp type slis_t_fieldcat_alv          with header line,
* faz parte da 1 alv. indica o campo que vai ser index e totalizado
t_sort         type slis_sortinfo_alv          occurs 0 with header line.

* estruturas std - obrigatorio
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,
  t_vbrp        type y_vbrp          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: end   of t_alv.

* -----
* Variáveis de uso geral
* -----
data:
  v_tabix       like sy-tabix,
  v_repid       like sy-repid,    " nome do programa
  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_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.
* recupera o nome do programa
  v_repid = sy-repid.
  clear v_variante.
  v_variante-report = v_repid.
* função para recuperar todas as variante de exibição
  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.                  "zf_init_alv

* -----
* 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.
* lista todas as variantes
  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.                                "zf_recupera_layouts_salvos

* -----
*      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.                                "zf_selecao_dados
* -----
*      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.
*  colunas dos relatorios alv

  clear t_fieldcat.
  t_fieldcat-fieldname   = 'MARK'.      " nome do campo
  t_fieldcat-tabname     = 'T_ALV'.      " tabela interna
  t_fieldcat-reptext_ddic = 'S'.         " título da coluna
  t_fieldcat-inttype     = 'C'.         " tipo
  t_fieldcat-outputlen   = 1.           " tamanho
  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.
  t_fieldcat-hotspot     = 'X'.         " aparecer a mão
  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'.         " indica campo totalizado
  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'.           " campos ocultos
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.           "zf_monta_tabela_alv
```

```
*-----
*      Form  zf_sort_subtotal
*-----
*      Classificação e item de subtotalização
*-----
```

```
form zf_sort_subtotal.
```

```
clear t_sort[].
t_sort-spos = 1.           "
t_sort-fieldname = 'KDGRP'. " nome do campo
t_sort-tabname   = 'T_ALV'.
t_sort-up       = 'X'.     " org. crescente
t_sort-subtot   = 'X'.     " totalizado
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.           "zf_sort_subtotal
```

```

*-----
*      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
v_layout-zebra            = 'X'.  "Impressão zebra

* 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      " nome do programa
    i_background_id         = 'ALV_BACKGROUND' " pano de fundo
    i_callback_top_of_page  = 'ZF_TOP_OF_PAGE' " cabeçalho
    i_callback_pf_status_set = 'ZF_STATUS'    " rotina dos botões
    i_callback_user_command = 'ZF_USER_COMMAND' " define as ações bt
    it_fieldcat             = t_fieldcat[]   " tabela das colunas
    is_layout               = v_layout       " layout
    it_sort                 = t_sort[]       " totalização
    i_default               = 'X'
    i_save                  = 'A'
    is_variant              = v_variante     " varian layout
    is_print                = v_print
  tables
    t_outtab               = t_alv          " tabela dos dados
  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

endform.                  "zf_executa_funcao_alv
*-----
*      Form  zf_top_of_page
*-----
*      Cabeçalho do relatório
*-----
form zf_top_of_page.

* 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 = '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.

* Apresenta o cabeçalho.

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

endform.                "zf_top_of_page

* -----
*           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_BOT0ES.

set pf-status 'ZALV_BOT0ES'.

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

endform.                "zf_status
* -----
*           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)
* SELFELD:  é uma estrutura com dados que nos permite identificar
*           o que foi selecionado. Essa estrutura também está
*           explicada no anexo ao final da apostila

* manter o relatório na linha selecionada antes do drill down.
selfield-row_stable = 'X'.
* manter o relatório na coluna selecionada antes do drill down.
selfield-col_stable = 'X'.

case ucomm.
*** Visualizar Documento de Faturamento - Transação VF03
when 'DOCF'.
*** Pesquisar o registro selecionado pelo usuário
read table t_alv index selfield-tabindex.
* SET ,,,, IF id do campo ( F1/F9 ) FILED tab com o valor do campo
set parameter id 'VF' field t_alv-vbeln.
* AND SKIP FIRST SCREEN -> PULA A PRIMEIRA TELA
call transaction 'VF03' and skip first screen.

when '&IC1'.
*** Pesquisar o registro selecionado pelo usuário

```

```

        read table t_alv index selfield-tabindex.
*** Verificar itens do Documento de Faturamento
        perform zf_carregar_t_vbrp.
        perform zf_monta_tabela_alv_vbrp.      "Preenche o catálogo
        perform zf_executa_funcao_alv_vbrp.    "Gera o relatório

        when 'ATUA'.

            loop at t_alv where mark = 'X'.
                v_tabix = sy-tabix.
*** Atualizar a tabela transparente
                update vbrk set netwr = t_alv-netwr
                    where vbeln = t_alv-vbeln.

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

            endloop.

        endcase.

endform.                                "zf_user_command
*&-----*
*&      Form  zf_carregar_t_vbrp
*&-----*
form zf_carregar_t_vbrp .

    clear t_vbrp.
    free  t_vbrp.

    select posnr matnr
           fking vrkme
    into table t_vbrp
    from vbrp
    where vbeln eq t_alv-vbeln.

endform.                                " zf_carregar_t_vbrp
*&-----*
*&      Form  zf_monta_tabela_alv_vbrp
*&-----*
form zf_monta_tabela_alv_vbrp .

    free  t_fieldcatvbrp.

    clear t_fieldcatvbrp.
    t_fieldcatvbrp-fieldname      = 'POSNR'.
    t_fieldcatvbrp-tabname        = 'T_VBRP'.
    t_fieldcatvbrp-reptext_ddic   = 'Item'.
    t_fieldcatvbrp-inttype        = 'N'.
    t_fieldcatvbrp-outputlen      = 6.
    append t_fieldcatvbrp.

    clear t_fieldcatvbrp.
    t_fieldcatvbrp-fieldname      = 'MATNR'.
    t_fieldcatvbrp-tabname        = 'T_VBRP'.
    t_fieldcatvbrp-reptext_ddic   = 'Nº do material'.
    t_fieldcatvbrp-inttype        = 'C'.
    t_fieldcatvbrp-outputlen      = 10.
    append t_fieldcatvbrp.

    clear t_fieldcatvbrp.
    t_fieldcatvbrp-fieldname      = 'FKIMG'.
    t_fieldcatvbrp-tabname        = 'T_VBRP'.
    t_fieldcatvbrp-reptext_ddic   = 'Qtde.faturada'.
    t_fieldcatvbrp-inttype        = 'P'.
    t_fieldcatvbrp-outputlen      = 20.

```

```

append t_fielddcatvbrp.

clear t_fielddcatvbrp.
t_fielddcatvbrp-fieldname      = 'VRKME'.
t_fielddcatvbrp-tabname       = 'T_VBRP'.
t_fielddcatvbrp-reptext_ddic  = 'Unidade'.
t_fielddcatvbrp-inttype       = 'C'.
t_fielddcatvbrp-outputlen     = 3.
append t_fielddcatvbrp.

endform.                " zf_monta_tabela_alv_vbrp
*&-----*
*&      Form  zf_executa_funcao_alv_vbrp
*&-----*
form zf_executa_funcao_alv_vbrp .

* 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_VBRP'
    it_fielddcat            = t_fielddcatvbrp[]
    is_layout               = v_layout
    i_default               = 'X'
    i_save                  = 'A'
    is_print                = v_print
*   i_screen_start_column   = 60      ABRE OUTRA JANELA POPUP UP
*   i_screen_start_line    = 5        COORDENADAS
*   i_screen_end_column    = 120
*   i_screen_end_line      = 20
  tables
    t_outtab                = t_vbrp
  exceptions
    program_error           = 1
    others                  = 2.

endform.                " zf_executa_funcao_alv_vbrp
*&-----*
*      Form  zf_top_of_page_vbrp
*&-----*
*      Cabeçalho do relatório
*&-----*
form zf_top_of_page_vbrp.

* 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 = 'Itens do Documento de Faturamento'.
append v_listheader to t_listheader.

```

* Definição do Projeto

```
clear v_listheader.  
v_listheader-typ = 'H'.
```

* TYP = S, outro tipo de fonte

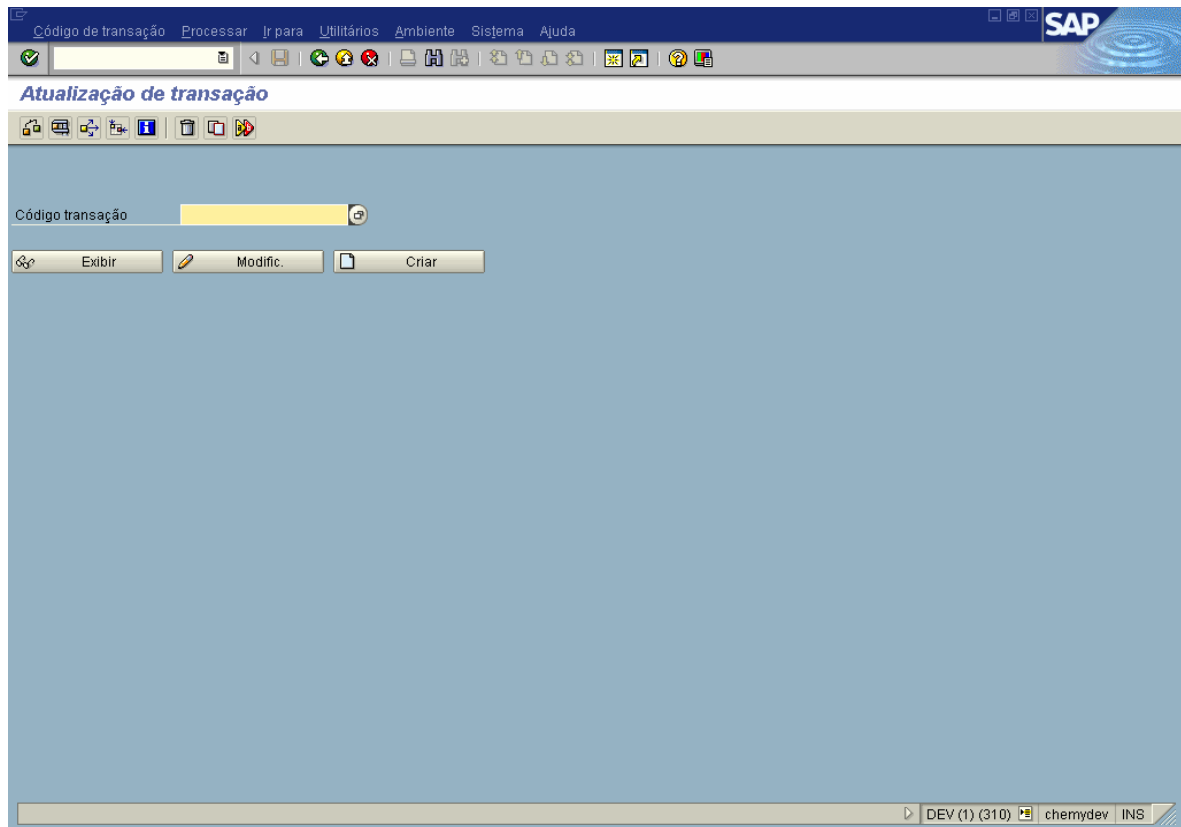
```
v_listheader-info = t_alv-vbeln.  
append v_listheader to t_listheader.
```

* Apresenta o cabeçalho.

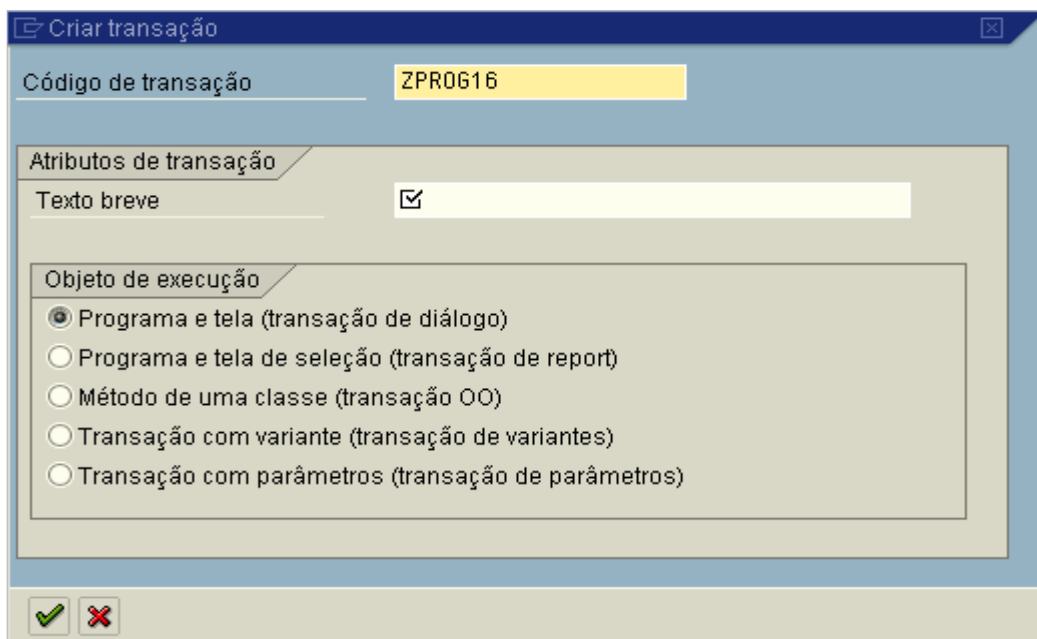
```
call function 'REUSE_ALV_COMMENTARY_WRITE'  
  exporting  
    i_logo          = 'ENJOYSAP_LOGO'  
    it_list_commentary = t_listheader.
```

```
endform.                "zf_top_of_page_vbrp
```

CRIAR TRANSAÇÕES – TRANSAÇÃO SE93



COLOCAR O NOME DA TRANSAÇÃO E DAR CRIAR.



COLOCAR TEXTO BREVE E ESCOLHER OBJETO DE EXECUÇÃO.

Para report -> 2 item (**Programa e tela de seleção-transação de report**)

The screenshot shows the SAP 'Criar Transação reports' (Create Transaction reports) dialog box. The title bar includes 'Código de transação', 'Processar', 'Ir para', 'Utilitários', 'Ambiente', 'Sistema', and 'Ajuda'. The SAP logo is in the top right corner. The main area contains the following fields and options:

- Código de transação:** ZPR0616
- Pacote:** (empty)
- Texto da transação:** Transação do programa ZC00116
- Programa:** ZC00116
- Tela de seleção:** 1000
- Início com variante:** (empty)
- Objeto de autorização:** (empty) with a 'Valor' button.
- Classificação:**
 - Classificação de transação:**
 - ☒ Transação Professional User
 - ☐ Transação Easy Web
 - ☐ Ativo globalmente
 - Capacidade GUI:**
 - ☐ SAP GUI para HTML
 - ☐ SAP GUI para Java
 - ☒ SAP GUI para Windows

The status bar at the bottom shows 'DEV (1) (310)', 'chemydev', and 'INS'.

PARA CRIAR TRANSAÇÃO DE VISÃO.

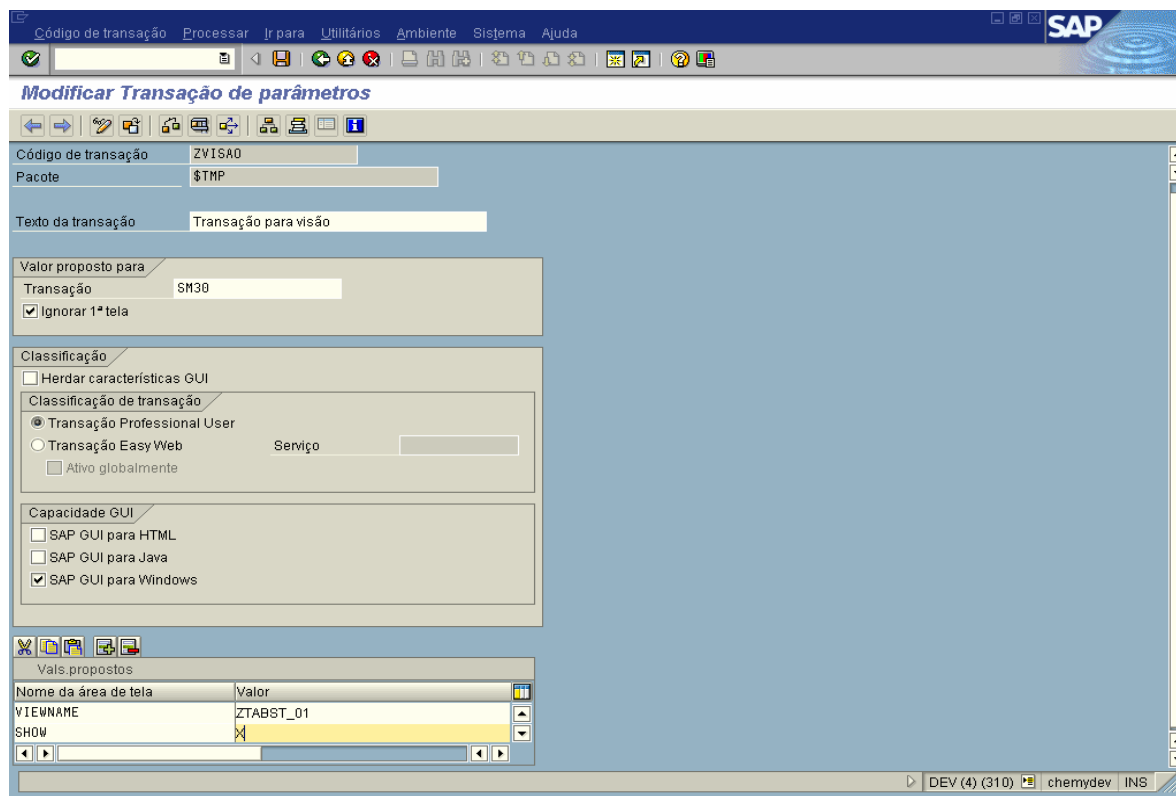
A tela abaixo deve estar conforme abaixo.

The screenshot shows the SAP 'Criar transação' (Create Transaction) dialog box. The title bar includes 'Criar transação' and a close button. The main area contains the following fields and options:

- Código de transação:** ZVISA0
- Atributos de transação:**
 - Texto breve:** Transação para visão
 - Objeto de execução:**
 - ☐ Programa e tela (transação de diálogo)
 - ☐ Programa e tela de seleção (transação de report)
 - ☐ Método de uma classe (transação OO)
 - ☐ Transação com variante (transação de variantes)
 - ☒ Transação com parâmetros (transação de parâmetros)

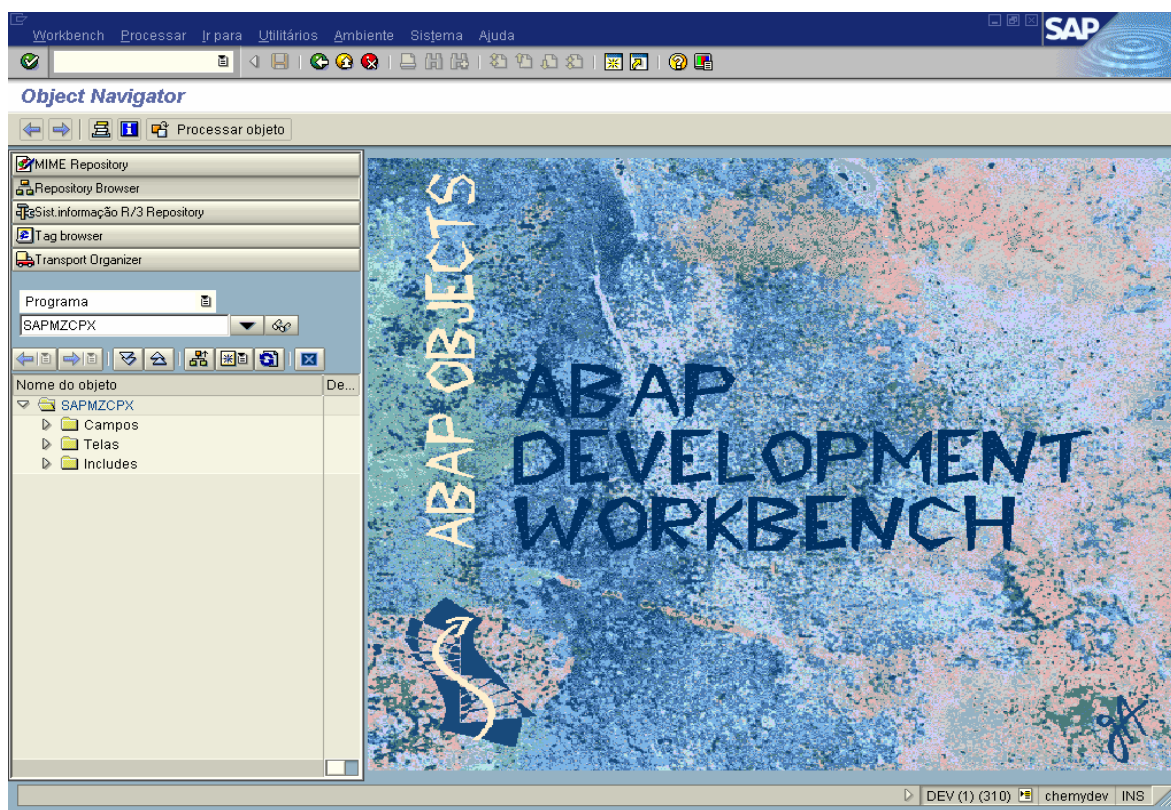
The status bar at the bottom shows a green checkmark icon and a red X icon.

COLOCAR O NOME DA TRANSAÇÃO QUE VAI SER USADA E DAR ENTER.
MARCAR IGNORAR 1ª TELA.



PREENCHER NOME DA AREA DE TELA : VIEWNAME e como valor O NOME DA VISÃO E SHOW para abrir a transação com EXIBIR e valor X, se quiser que entre no modo de EDITAR UPDATE e valor X

NOÇÕES DE MODULE POOL (SE80) – programas “ON LINE”



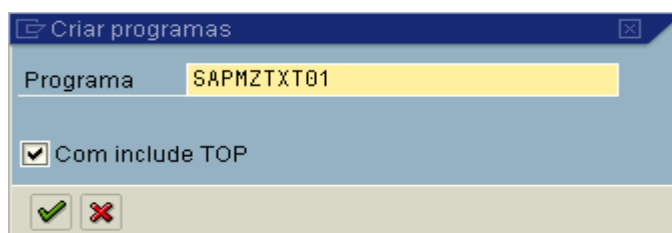
MODULO POOL : programa SAPMZ.... (**z**) sempre começar com **SAPMZ**
O MODULO POOL CONTROLA DIVERSAS TELAS
POR PADRÃO AS TELAS CLIENTE DEVEM COMEÇAR COM 9000, 9010, 9020,



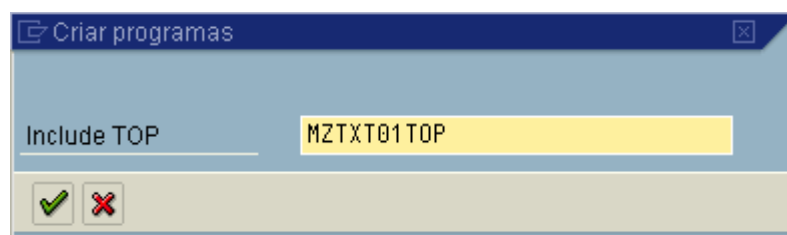
CODIGO DAS TELAS DIVIDIDOS EM 2 PARTES :
PBO (processamento antes dos dado aparecer na tela)
PAI (processamento depois do usuário)

DENTRO DE CADA PROCESSOS (PBO , PAI) EXISTEM MÓDULOS, É DENTRO DO MODULO É QUE VAI O ABAP.

PARA CRIAR O PROGRAMA DIGITAR O NOME NO BOX E CLICAR NO OCULOS:



CONFIRMAR

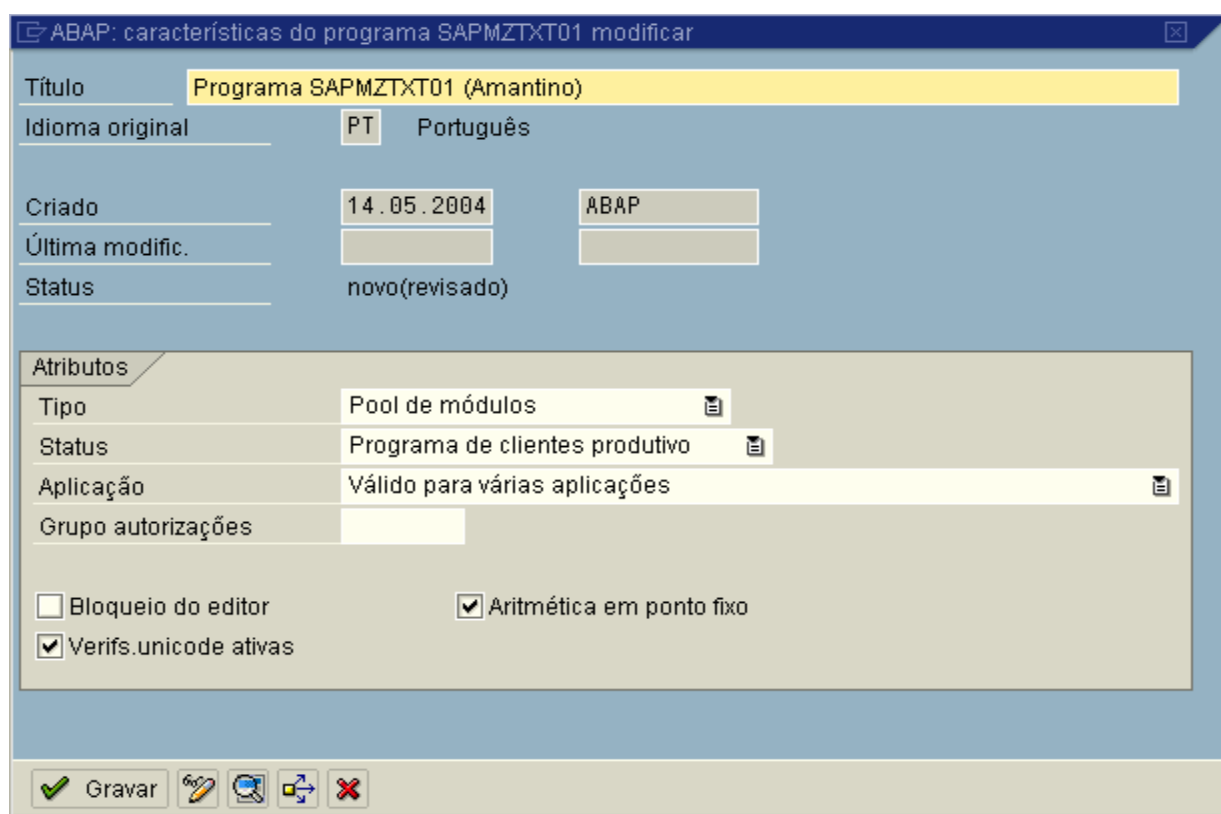


Criar programas

Include TOP MZTXT01TOP

✓ ✗

CONFIRMAR E PREENCHER TELA ABAIXO



ABAP: características do programa SAPMZTXT01 modificar

Título Programa SAPMZTXT01 (Amantino)

Idioma original PT Português

Criado 14.05.2004 ABAP

Última modific.

Status novo(revisado)

Atributos

Tipo Pool de módulos

Status Programa de clientes produtivo

Aplicação Válido para várias aplicações

Grupo autorizações

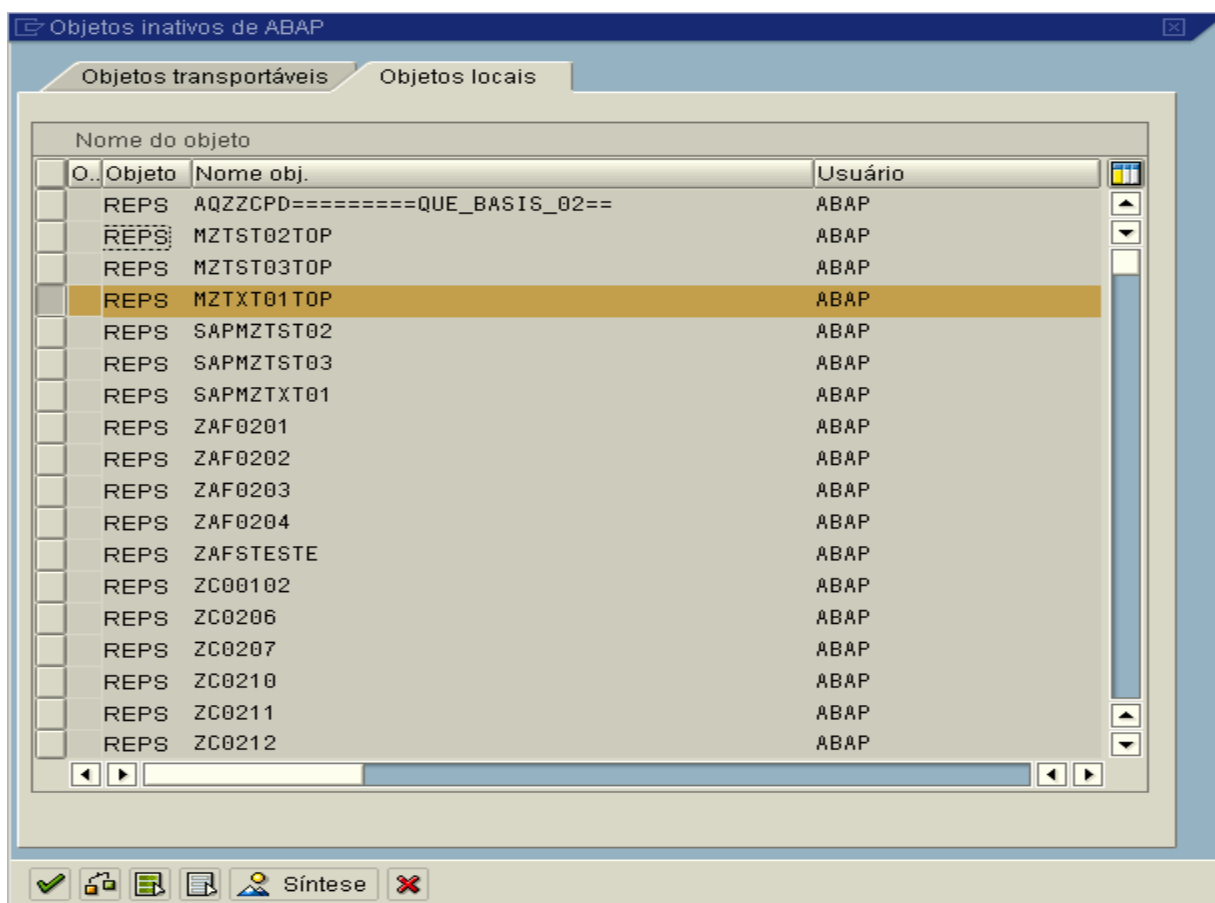
☐ Bloqueio do editor ☒ Aritmética em ponto fixo

☒ Verifs.unicode ativas

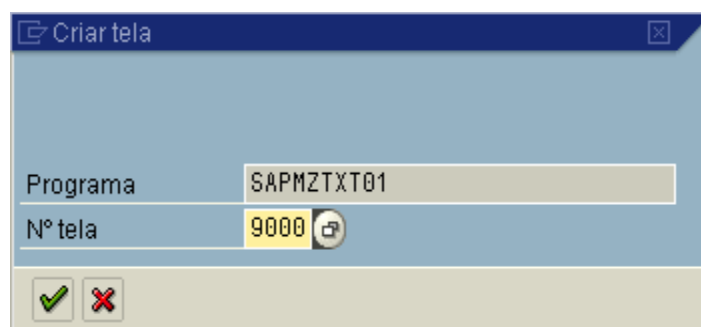
✓ Gravar ✎ ↻ ↺ ✗

CLICAR EM TOP E TIRAR O ASTERISTO (*) DO MODULO **TOP (global Data)**

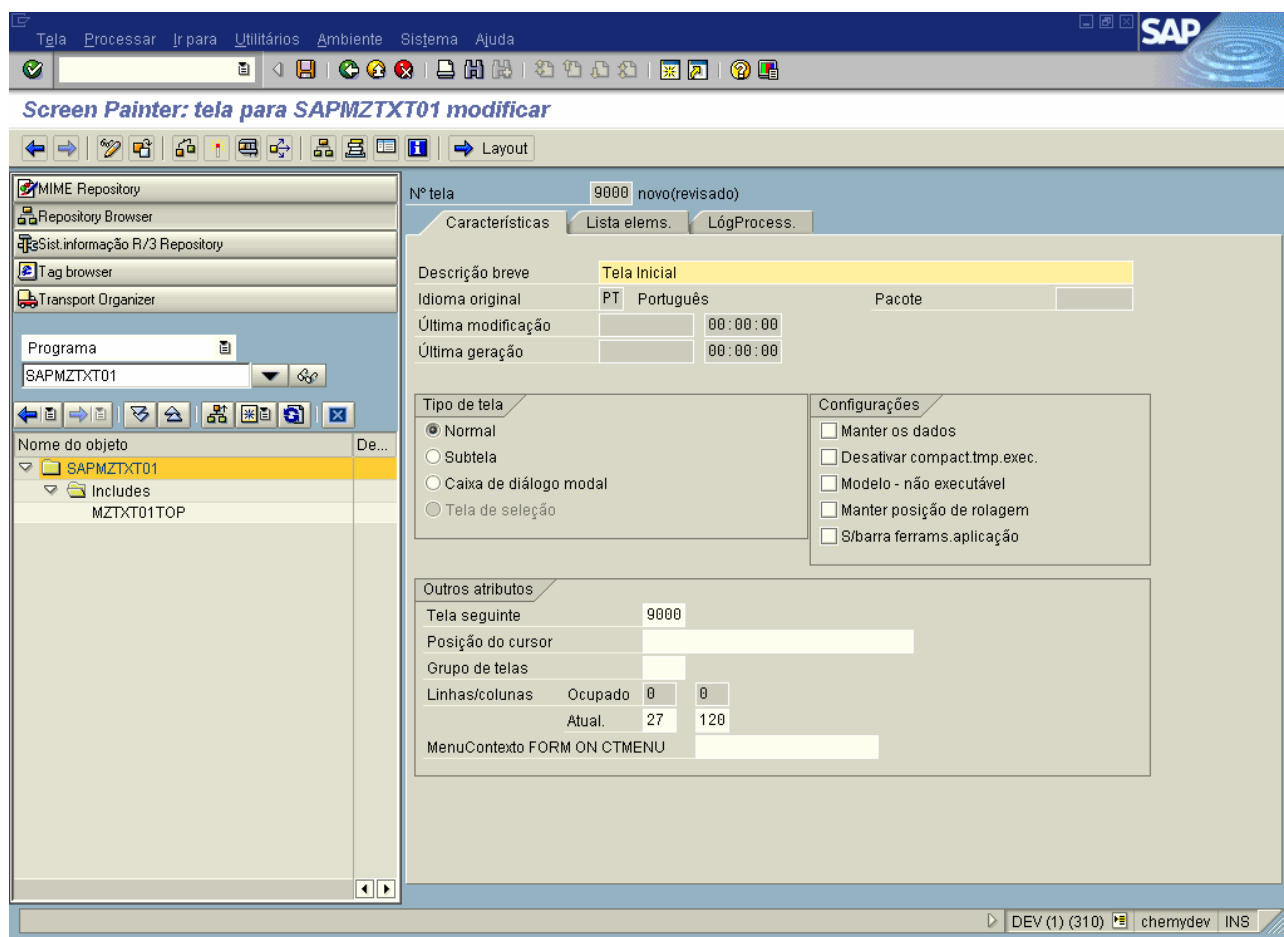
Padrões :
vg -> variável global
Tg -> tabela interna (global)



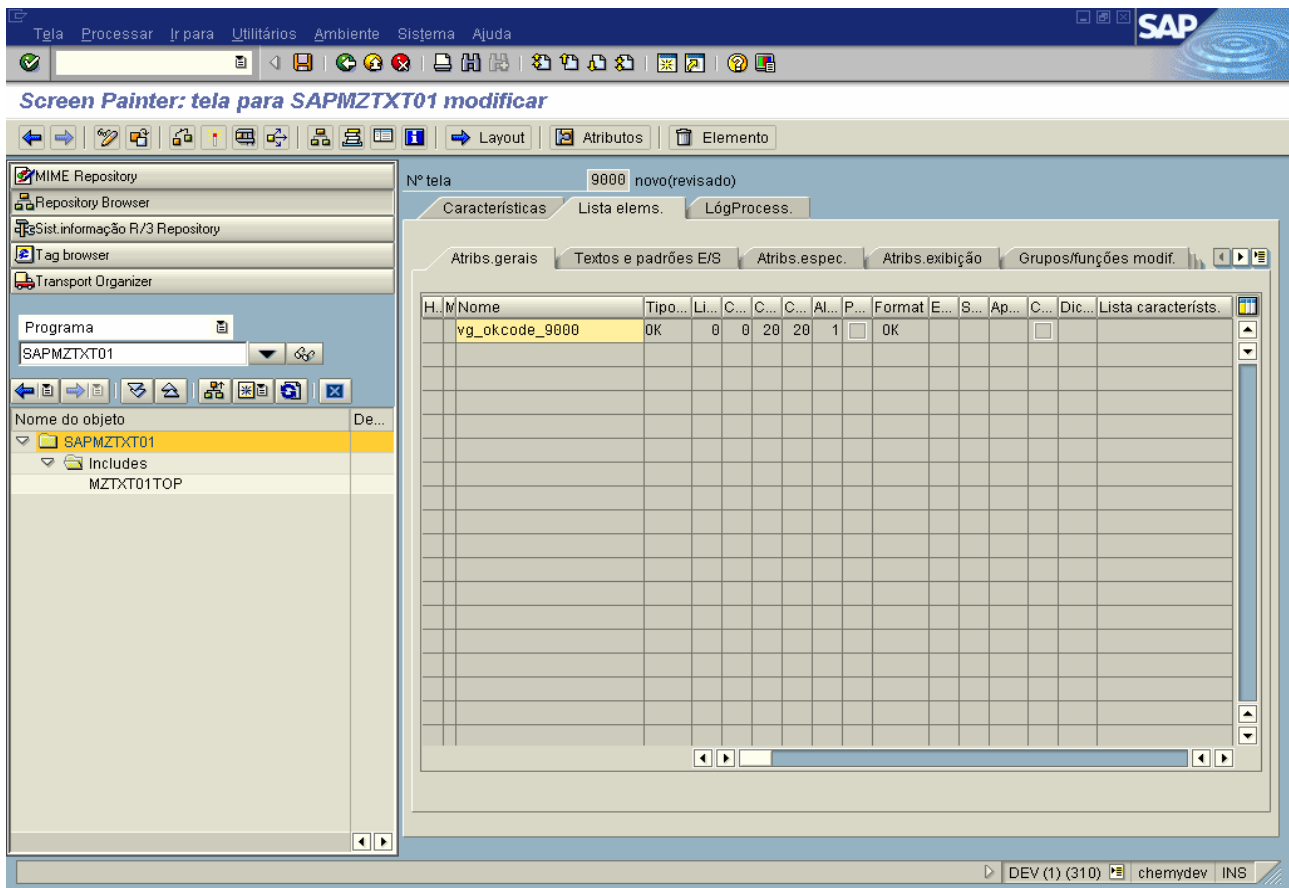
DEPOIS CRIAR A TELA (MOUSE SOBRE O NOME PRINCIPAL E BOTÃO DIREITO DO MOUSE)



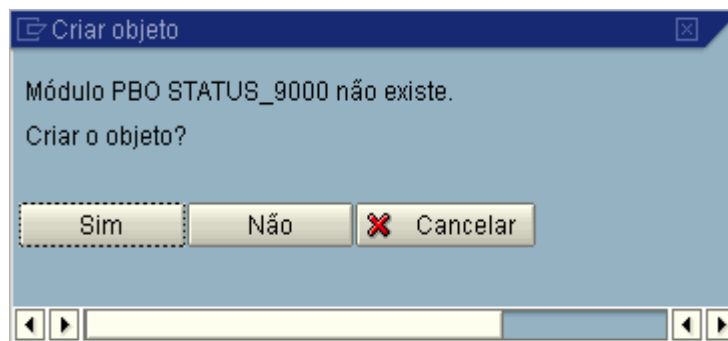
PREENCHER OS DADOS ABAIXO E IR PARA A PROXIMA TELA



Definir a variável que vai receber o código de retorno da pagina, depois declarar a variável no modulo TOP



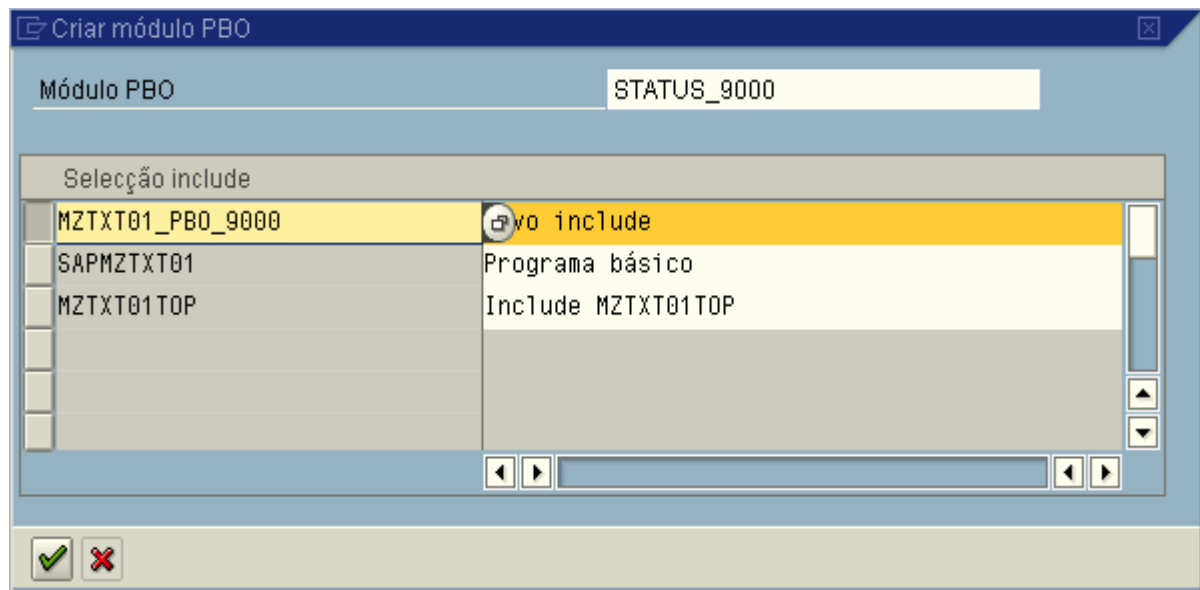
IR PARA A TELA, TIRAR O ASTERISCO DO MODULO PBO E DAR DUBLO CLICK



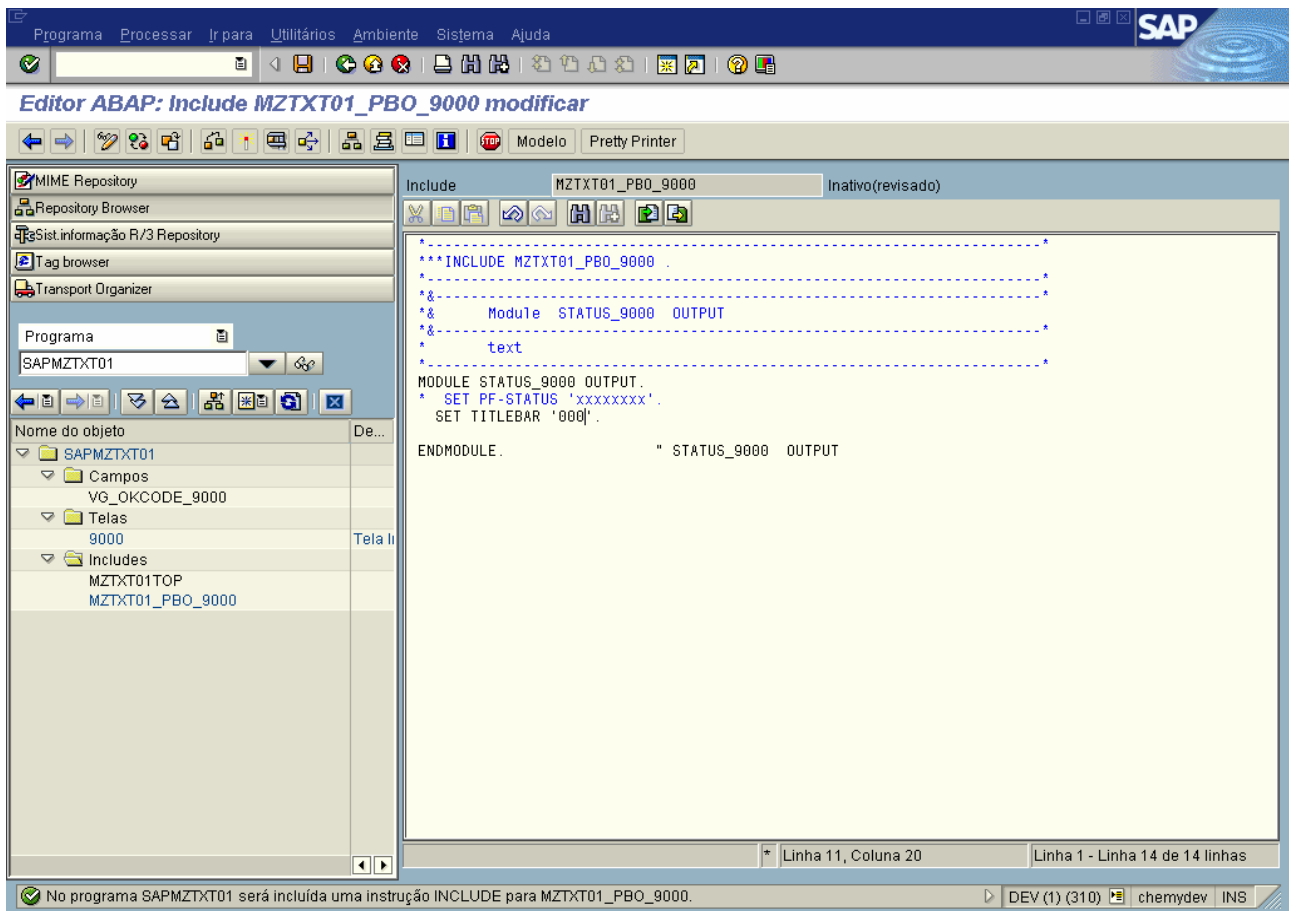
MUDAR O NOME SUGERIDO PARA O MODELO ABAIXO

Usar como padrão : nomesugerido_modulo_tela

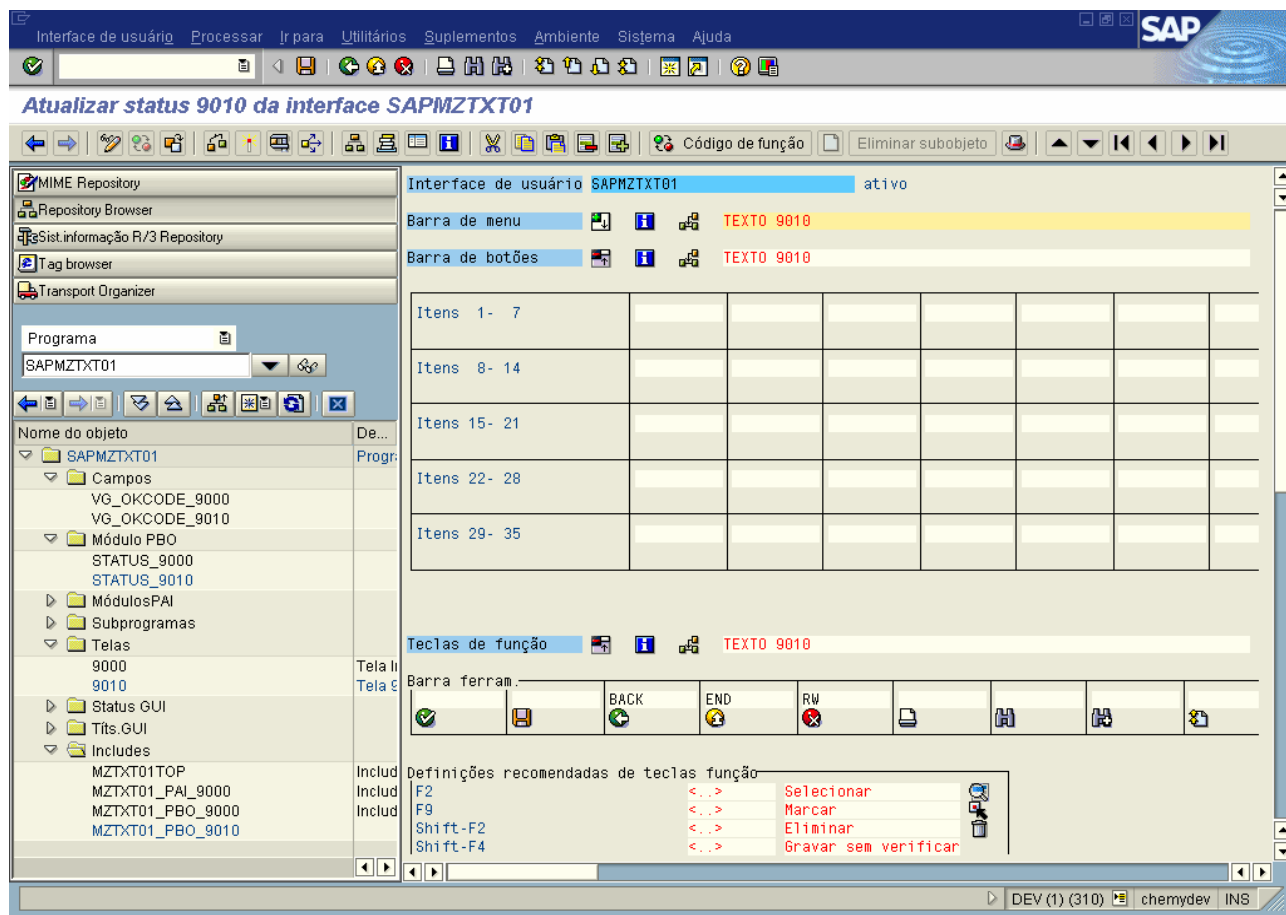
Exemplo : nomesugerido = MZTXT01 , módulo :PBO , Tela : 9000



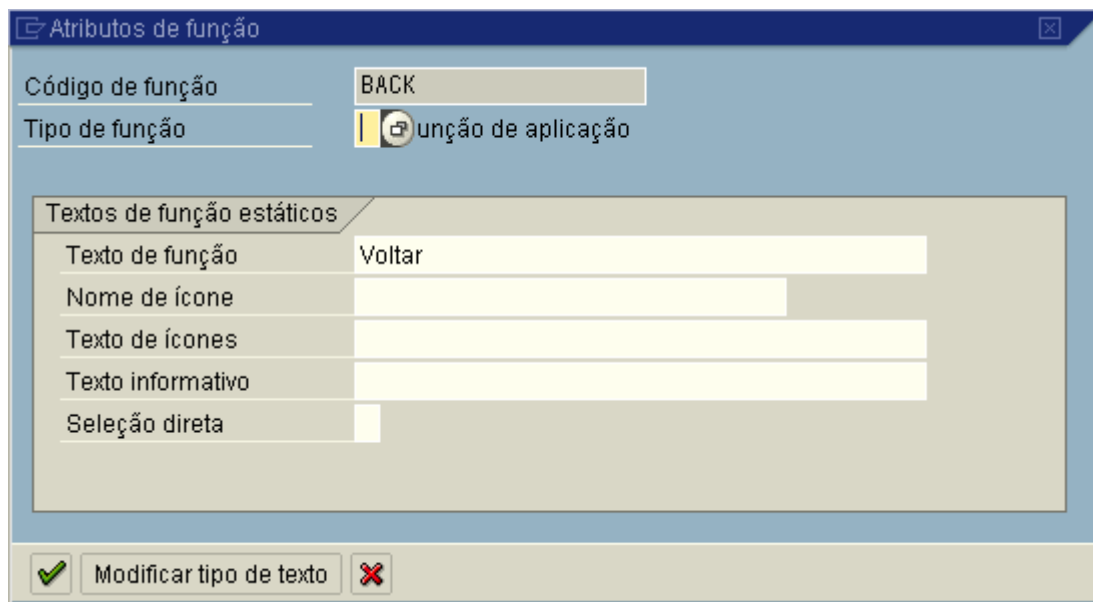
CRIAR SET TITLE BAR 'XXX' DAR DOIS CLICKS

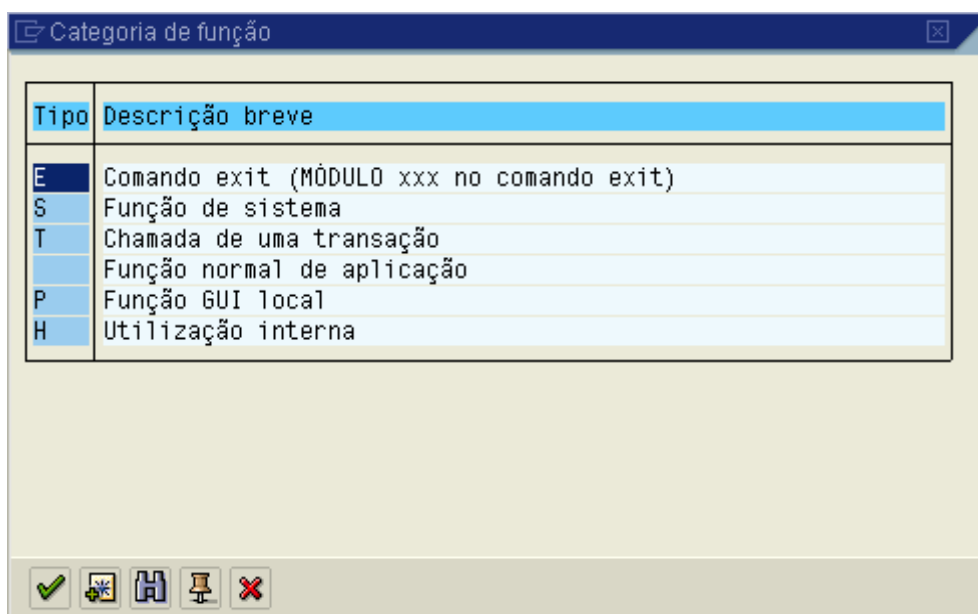


DEPOIS CRIAR OS BOTÕES

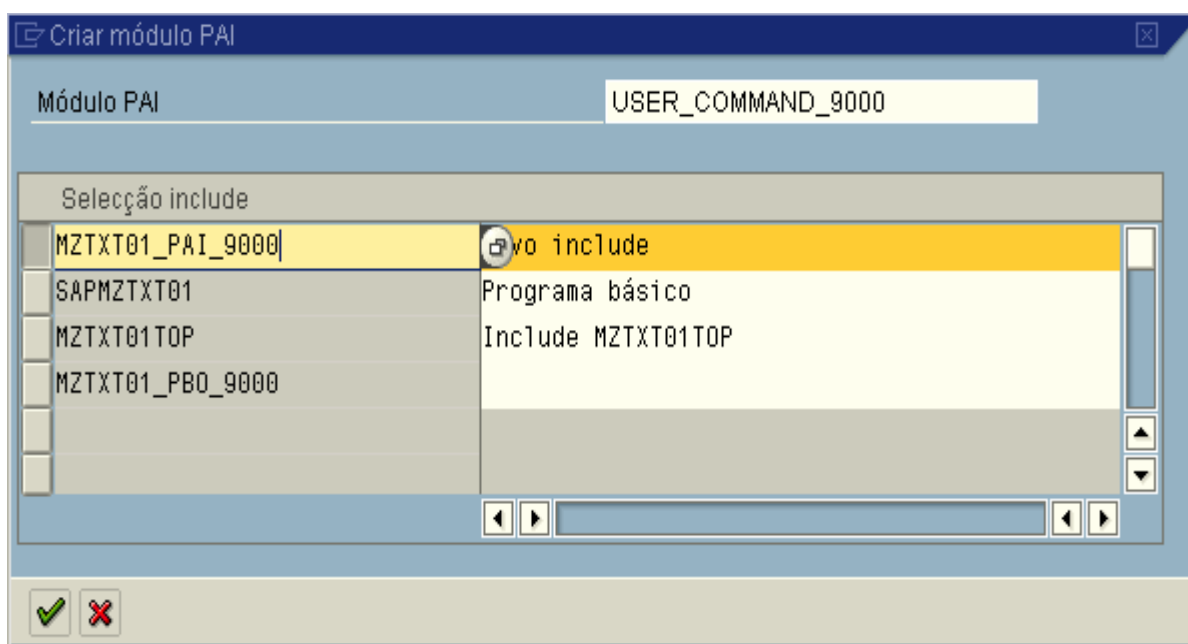


NOS BOTÕES PADRÕES TRABALHAR COM O TIPO DE FUNÇÃO E





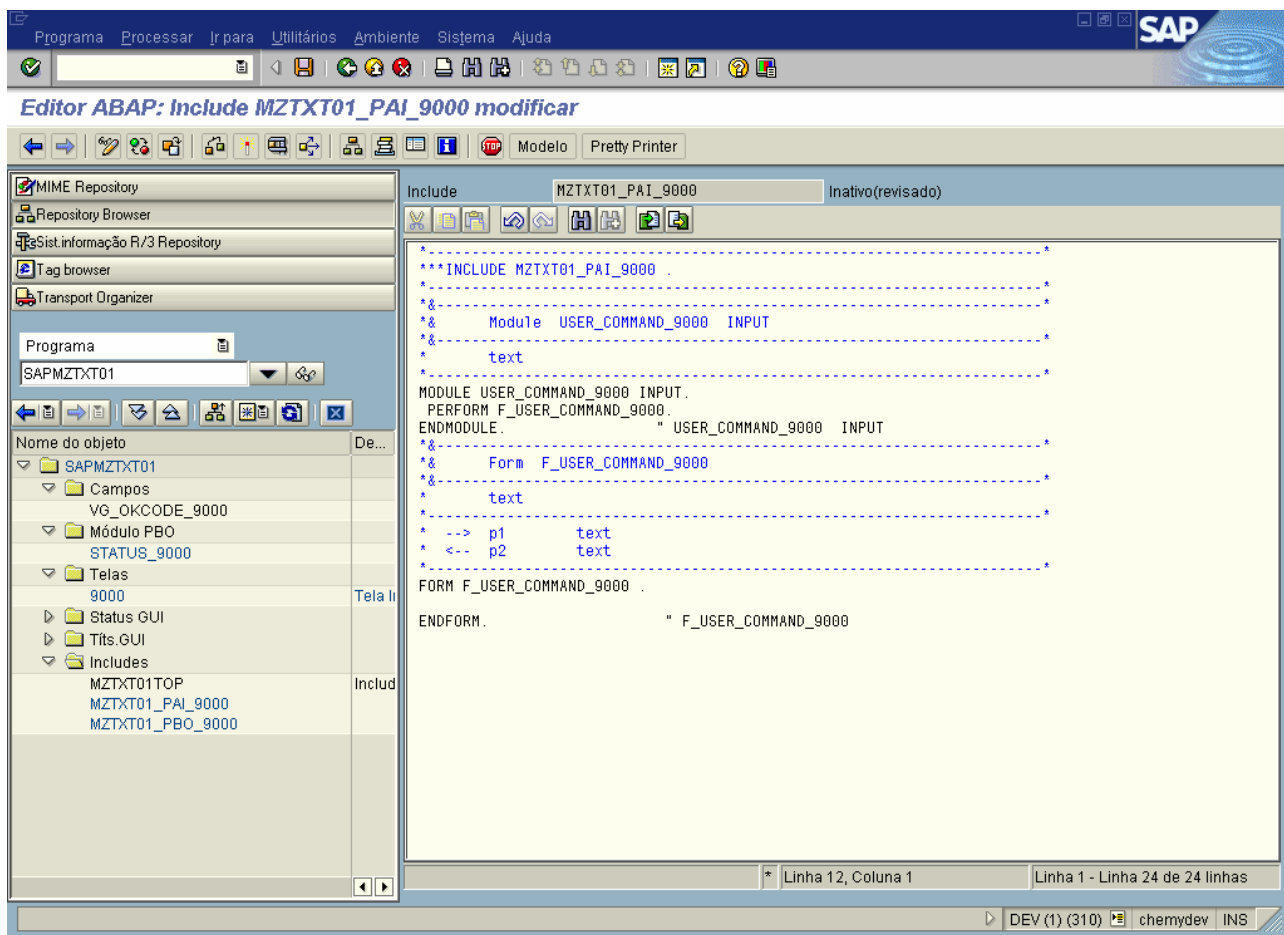
CRIAR OBJETO PAI – NA TELA TIRAR O ASTERISTICO (*) E DAR 2 CLICK
 SEGUIR O PADRÃO DE CRIAÇÃO DE NOMES : **nomesugerido_modulo_tela**



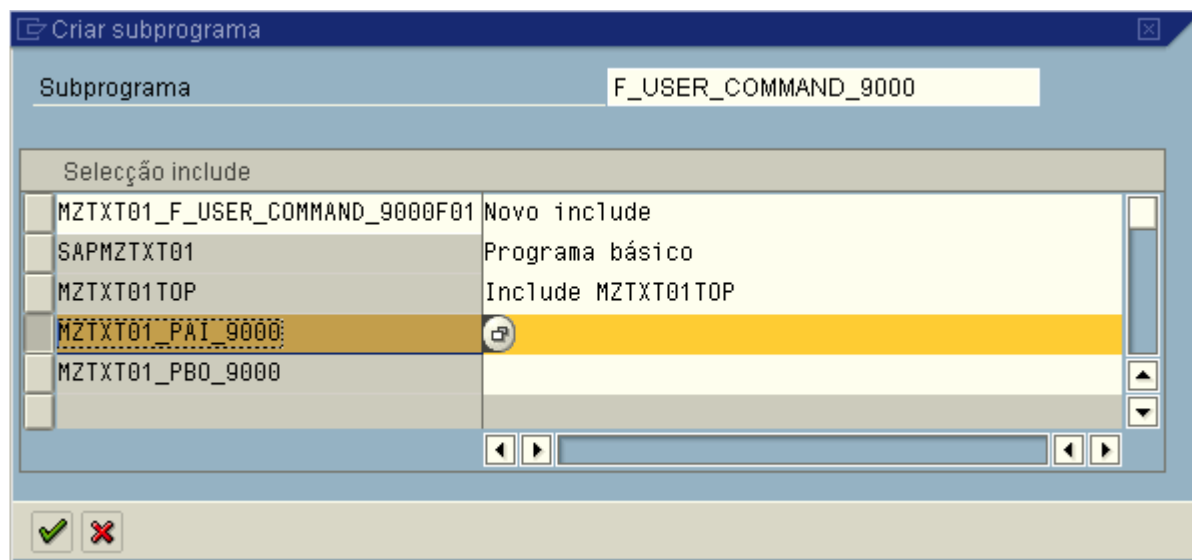
NOS MODULOS POB E PAI CRIAR SEMPRE CRIAR PERFORM PARA DECLARAÇÕES DE VARIÁVEIS, VER MODELO ABAIXO.

NOME DO PERFORM PARA FACILITAR ENDENTIMENTO DO PROGRAMA DEVE SER :


PERFORM **F_nome_do_modulo**. (F_USER_COMMAND_9000).

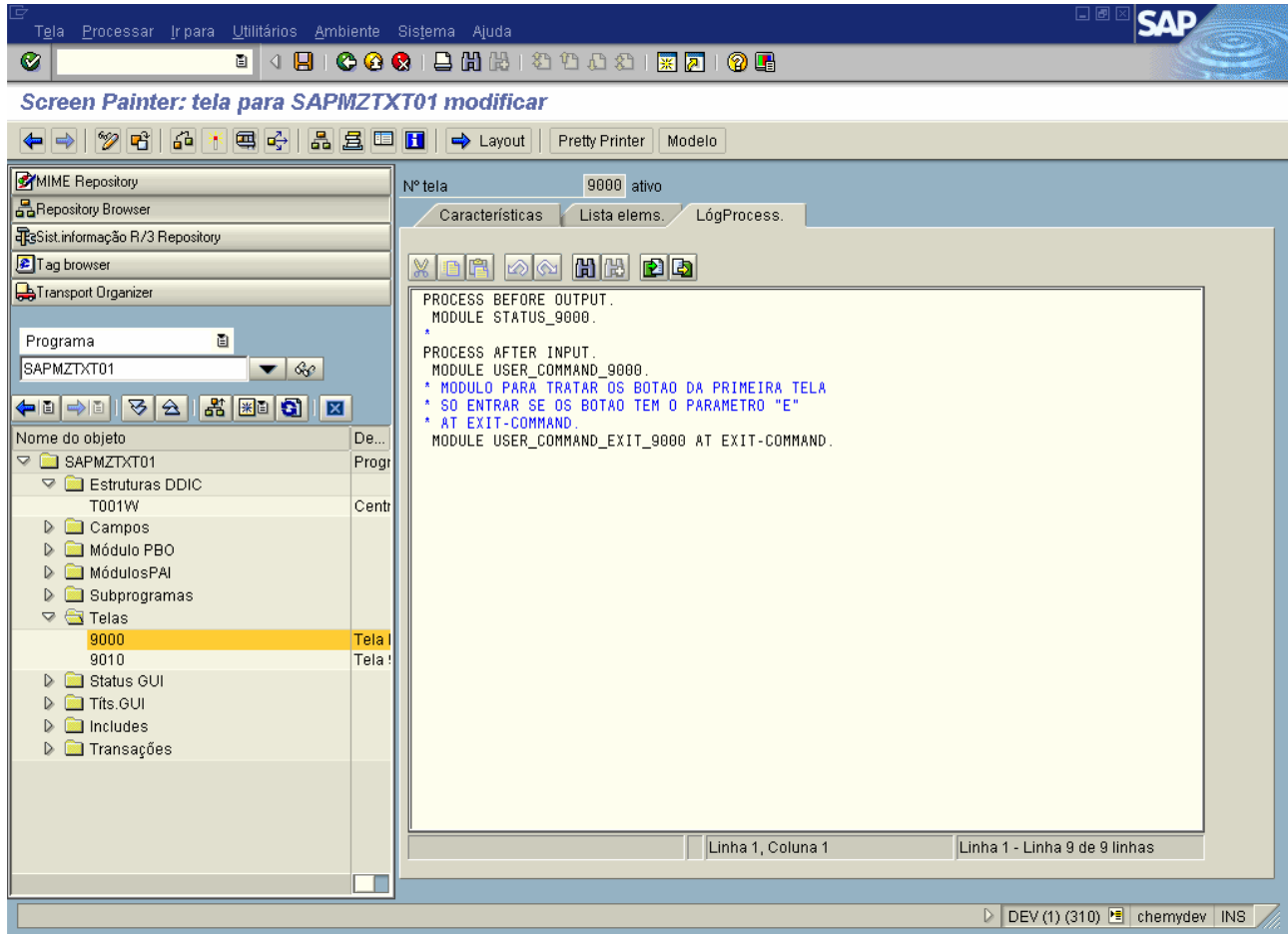


SEMPRE ATRIBUIR A CRIAÇÃO DOS PERFORMS AO OBJETO PAI.

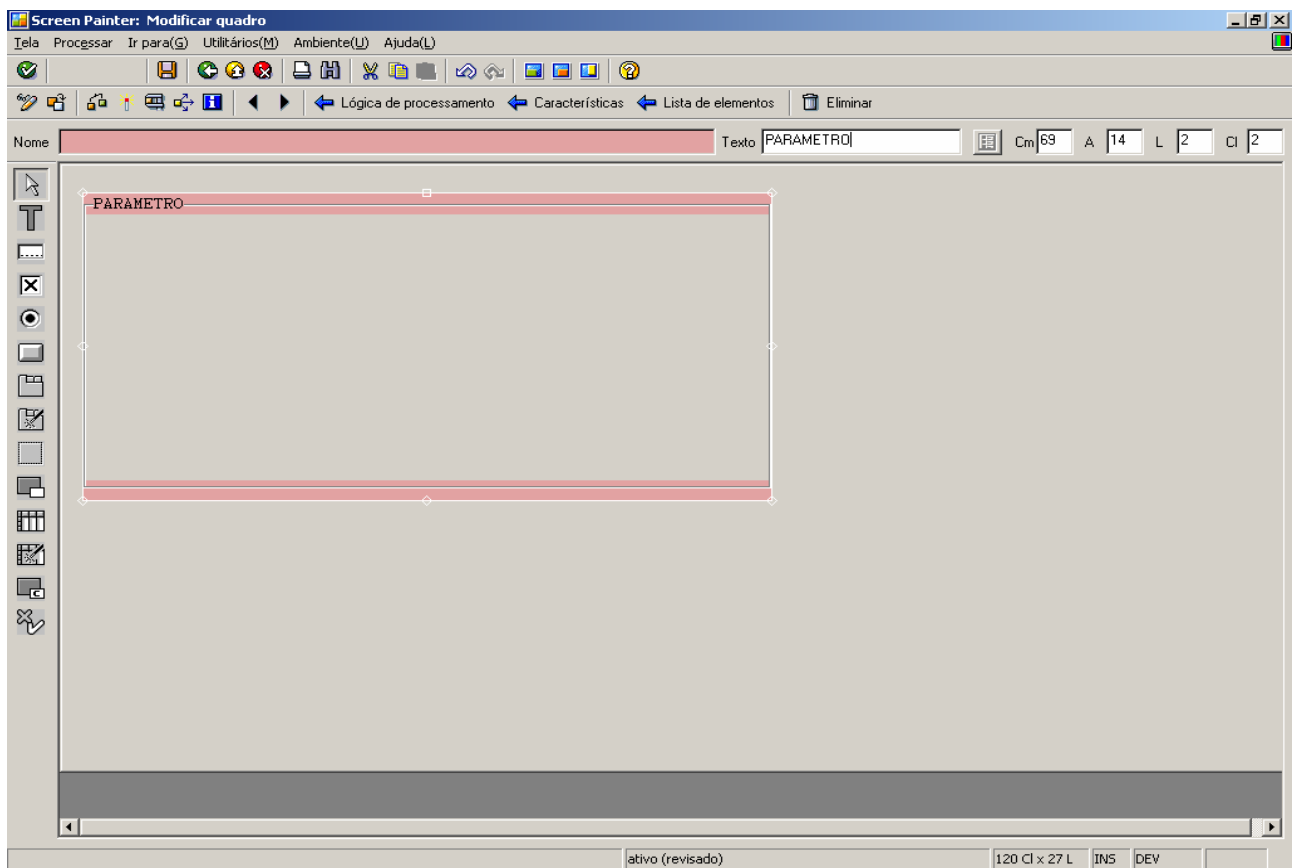


CRIAR JANELAS.

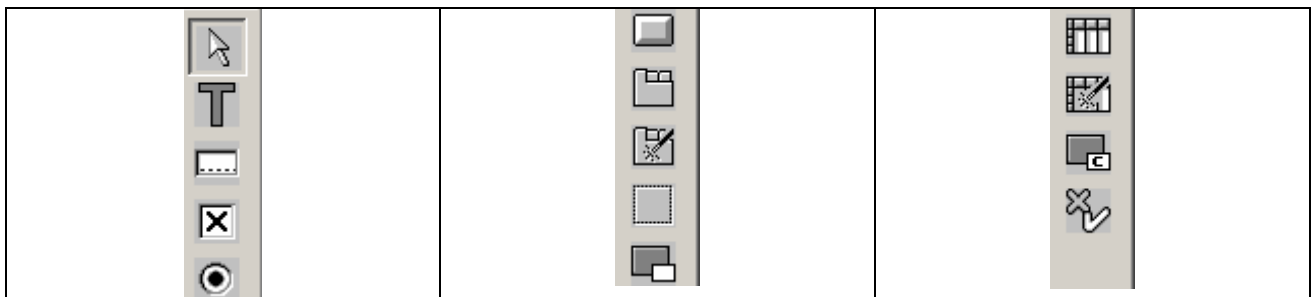
PARA CRIARMOS JANELAS (DESENHAR) COLOCAMOS O MOUSE SOBRE O NOME DA TELA, E CLICAMOS NO BOTÃO  Layout .



EM SEGUIDA SERA PROCESSADO O EDITOR GRAFICO E SERA MOSTRADO A TELA ABAIXO :




PARA DESENHAR A JANELA UTILIZAMOS OS OBEJTOS DE DESENHO

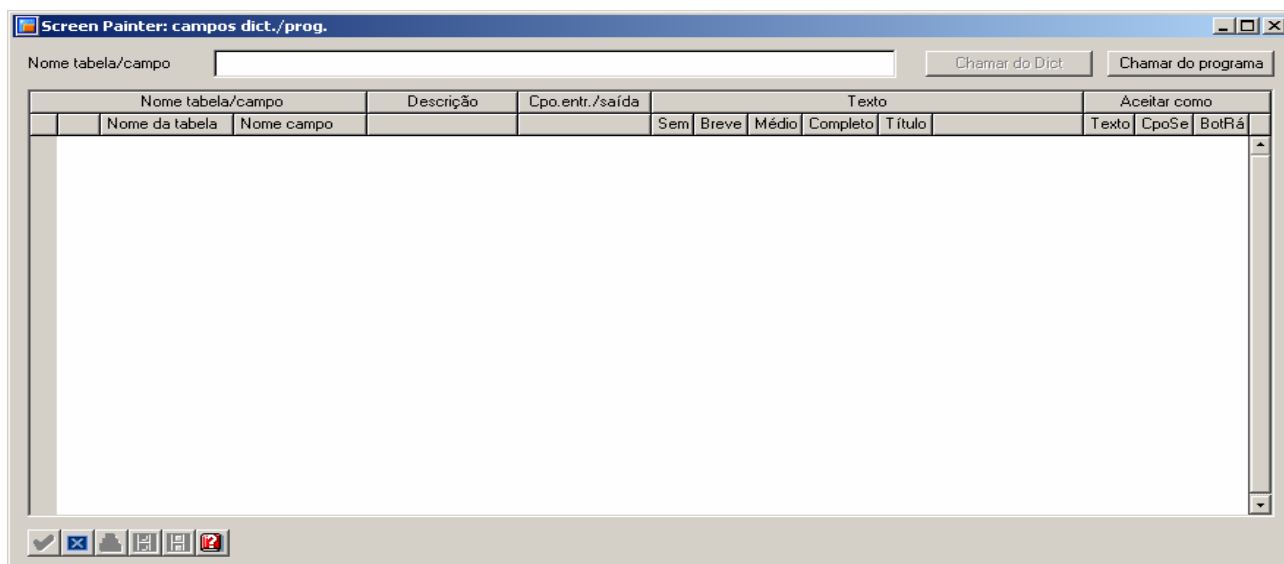


DEFINIR UMA JANELA – DEPOIS DE DESENHAR COLOCAR O NOME QUE SERA EXIBIDO EM Texto

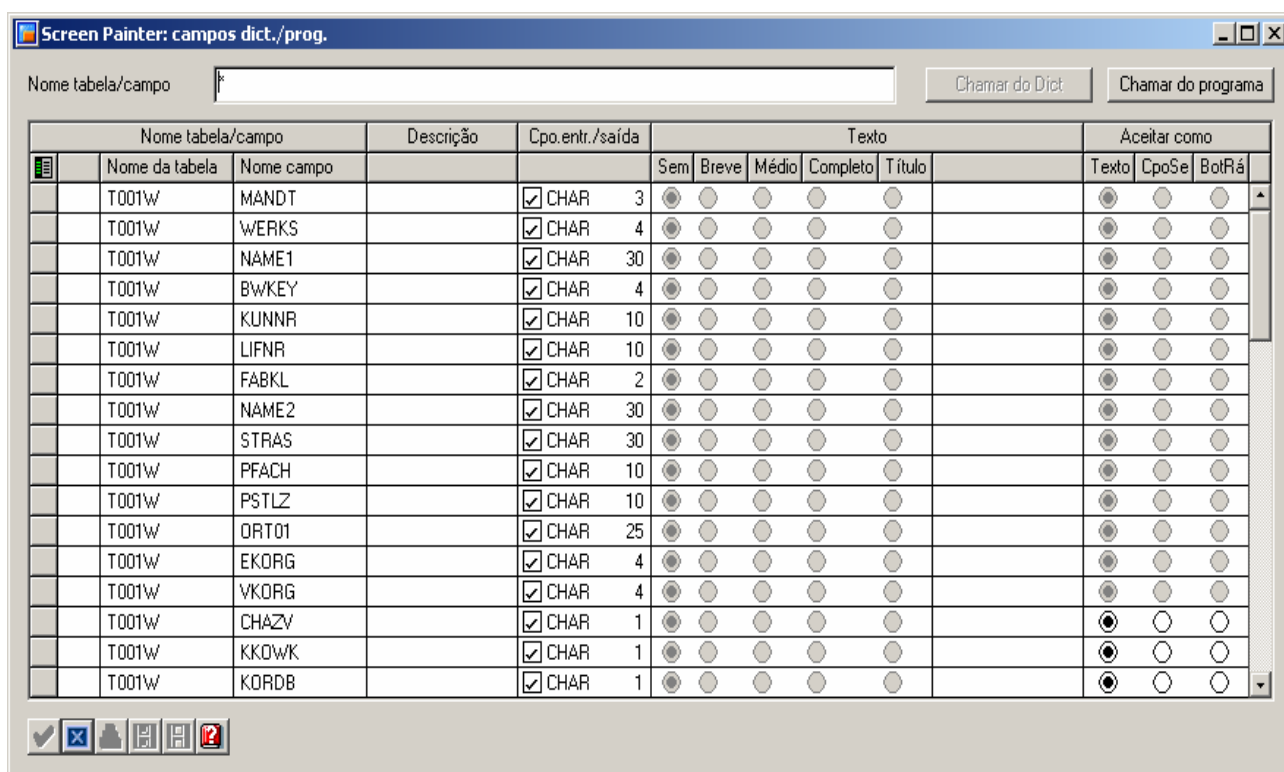


CAIXA DE ENTRADA (1)

LOGO APÓS INSERIRMOS A CAIXA DE ENTRADA (1) CLICAMOS NO ICONE  PARA ABRIR A TELA ABAIXO.



NESTA TELA PODEMOS BUSCAR AS TABELAS QUE ESTÃO DEFINIDAS EM NOSSO PROGRAMA E VARIÁVEIS, PARA ISSO BASTA CLICAR EM **Chamar do programa**.



DAÍ PODEMOS SELECIONAR O CAMPO DA TABELA OU VARIÁVEL QUE SERÁ ATRIBUÍDO À CAIXA DE ENTRADA (1).

Screen Painter: campos dict./prog.

Nome tabela/campo: T001W

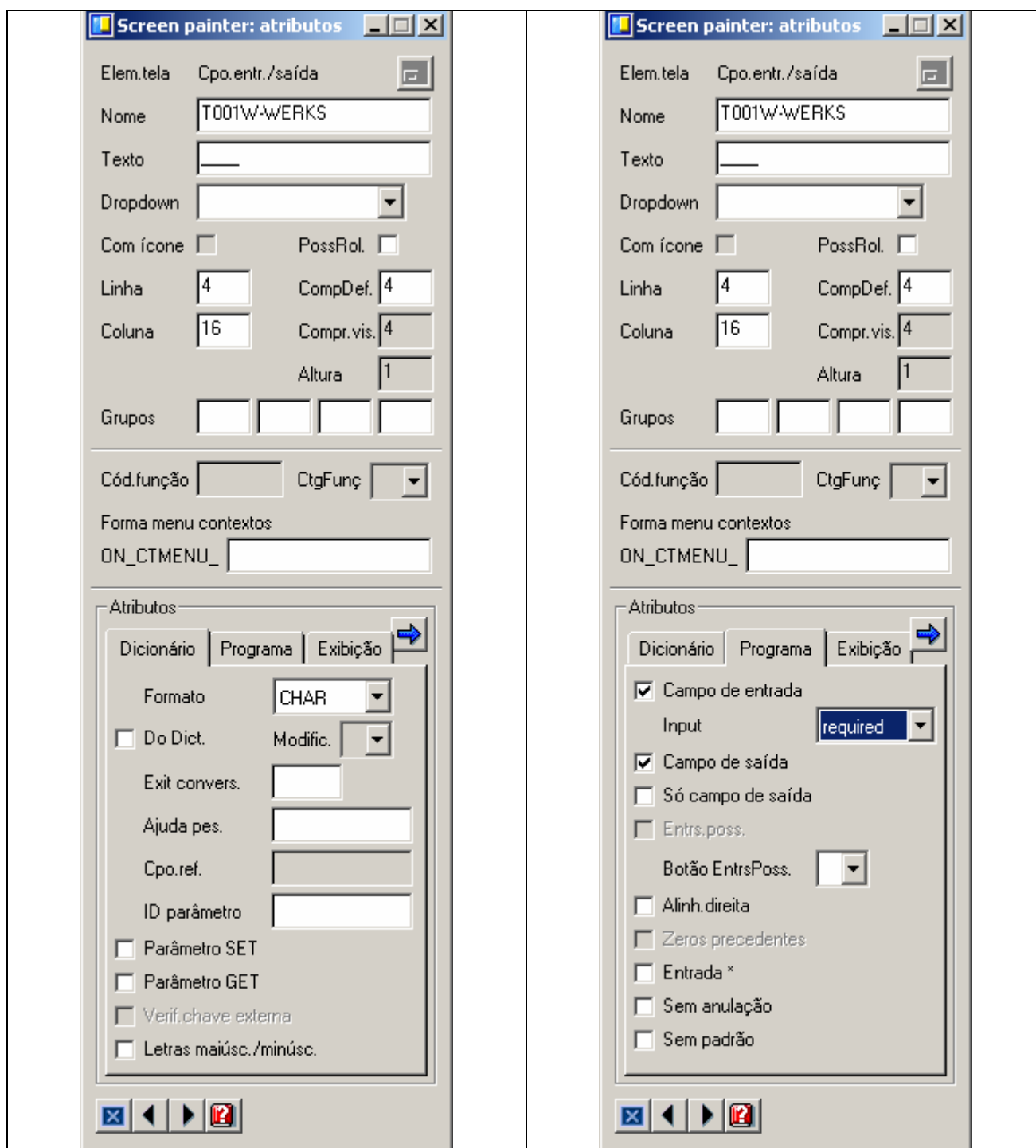
Chamar do Dict Chamar do programa

Nome tabela/campo		Descrição	Cpo.ent./saída	Texto					Aceitar como		
Nome da tabela	Nome campo			Sem	Breve	Médio	Completo	Título	Texto	CpoSe	BotRá
T001W	MANDT		<input checked="" type="checkbox"/> CHAR 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	WERKS		<input checked="" type="checkbox"/> CHAR 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	NAME1		<input checked="" type="checkbox"/> CHAR 30	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	BWKEY		<input checked="" type="checkbox"/> CHAR 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	KUNNR		<input checked="" type="checkbox"/> CHAR 10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	LIFNR		<input checked="" type="checkbox"/> CHAR 10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	FABKL		<input checked="" type="checkbox"/> CHAR 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	NAME2		<input checked="" type="checkbox"/> CHAR 30	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	STRAS		<input checked="" type="checkbox"/> CHAR 30	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	PFACH		<input checked="" type="checkbox"/> CHAR 10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	PSTLZ		<input checked="" type="checkbox"/> CHAR 10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	ORT01		<input checked="" type="checkbox"/> CHAR 25	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	EKORG		<input checked="" type="checkbox"/> CHAR 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	VKORG		<input checked="" type="checkbox"/> CHAR 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	CHAZV		<input checked="" type="checkbox"/> CHAR 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	KKDWK		<input checked="" type="checkbox"/> CHAR 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
T001W	KORDB		<input checked="" type="checkbox"/> CHAR 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

☒ ☐ ☐ ☐ ☐ ☐ ☐

SE DERMOS 2 CLICK NA CAIXA DE ENTRADA SERA ABERTA A TELA DE ATRIBUTOS DESTE OBJETO, QUE TAMBEM PODE SER ACESSADO PELO ICONE .

PODEMOS NAVEGAR PELOS ATRIBUTOS DESTE OBJETO USANDO AS ABAS : DICIONARIO, PROGRAMA, EXEBIÇÃO...



EM **DICIONÁRIO** DEVEMOS DEIXAR MARCADO O CAMPO **Do Dict.** DESTA FORMA O CAMPO VAI RECEBER AS CARACTERÍSTICAS DA TABELAxCAMPO.
E NA ABA **PROGRAMA** EM CAMPO DE ENTRADA PODEMOS DEIXAR A OPÇÃO REQUIRED MARCADA (pois isto torna o campo obrigatório)

Dicionário
Programa
Exibição

☒ Campo de entrada
Input required

Dicionário
Programa
Exibição

☐ Campo de entrada
Input not possible

Campo de entrada obrigatório

TAMBEM PODEMOS ACESSAR ESTE FLAG USANDO O BOTÃO  Lógica de processamento

ESTE BOTÃO LEVA DE VOLTA A TELA DE PROGRAMA, CLICAMOS NA ABA LISTA DE ELEMENTOS.

Screen Painter: tela para SAPMZTX01 modificar

Nº tela 9000 ativo(revisado)

Características Lista elems. LógProcess.

Atribs.gerais Textos e padrões E/S Atribs.espec. Atribs.exibição Grupos/funções modif.


H...	MNome	Tipo...	Li...	C...	C...	C...	Al...	P...	Format	E...	S...	Ap...	C...	Dic...	Lista característs
		Quadr	2	2	69	69	14								
		Texto	4	4	10	10	1								
	T001W-WERKS	I/O	4	16	4	4	1		CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Caracts.
	VG_OKCODE_9000	OK	0	0	20	20	1		OK						

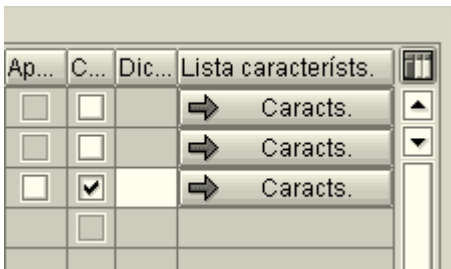
Nome do objeto

- Campos
 - VG_OKCODE_9000
 - VG_OKCODE_9010
- Módulo PBO
 - STATUS_9000
 - STATUS_9010
- MódulosPAI
- Subprogramas
- Telas
 - 9000 Te
 - 9010 Te
- Status GUI
- Títs.GUI
- Includes
 - MZTX01TOP Inc
 - MZTX01_PAI_9000 Inc
 - MZTX01_PAI_9010 Inc
 - MZTX01_PBO_9000 Inc
 - MZTX01_PBO_9010 Inc

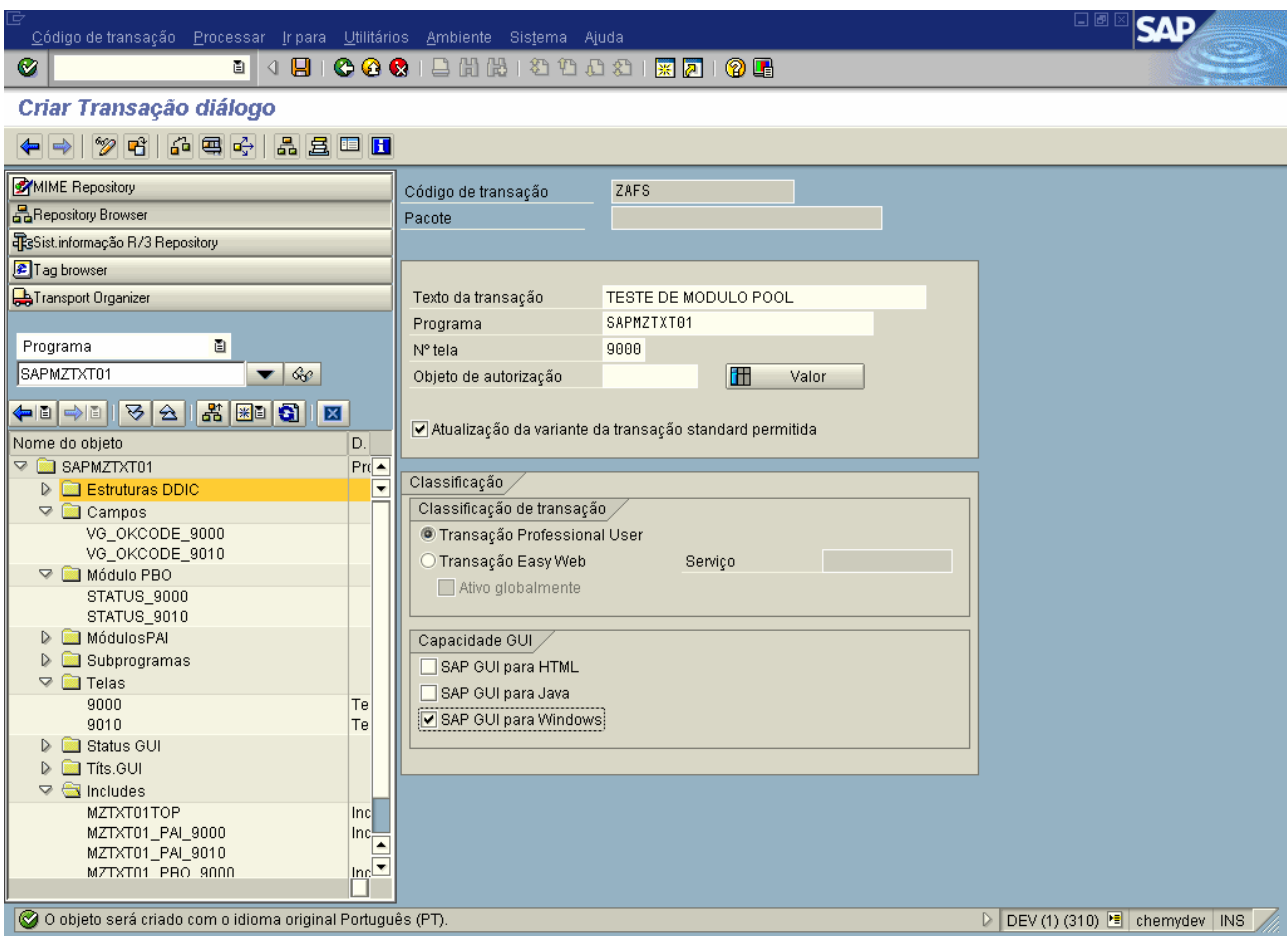
DEV (1) (310) chemydev INS

E LOCALIZAMOS A TabelaxCampo QUE ATRIBUIMOS A CAIXA DE ENTRADA E MARCAMOS A OPÇÃO C

T001W-WERKS	I/O	4	16	4	4	1	<input type="checkbox"/>	CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Caracts.
-------------	-----	---	----	---	---	---	--------------------------	------	-------------------------------------	-------------------------------------	--------------------------	-------------------------------------	---	----------



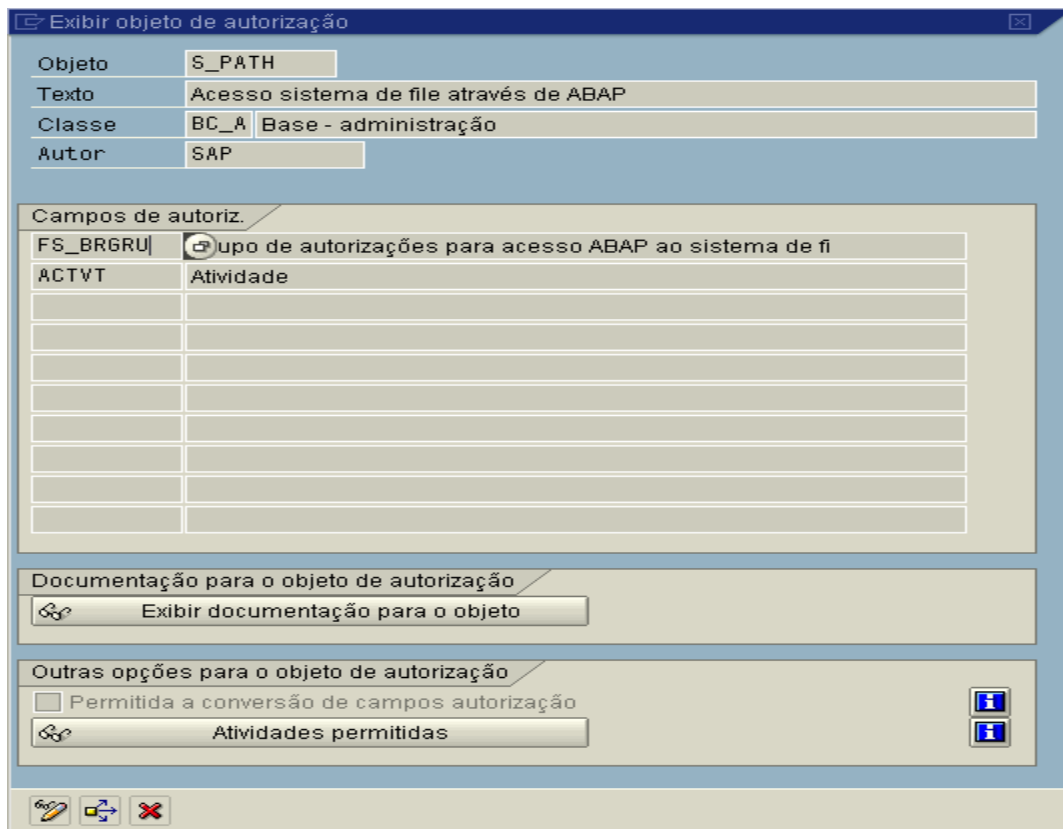
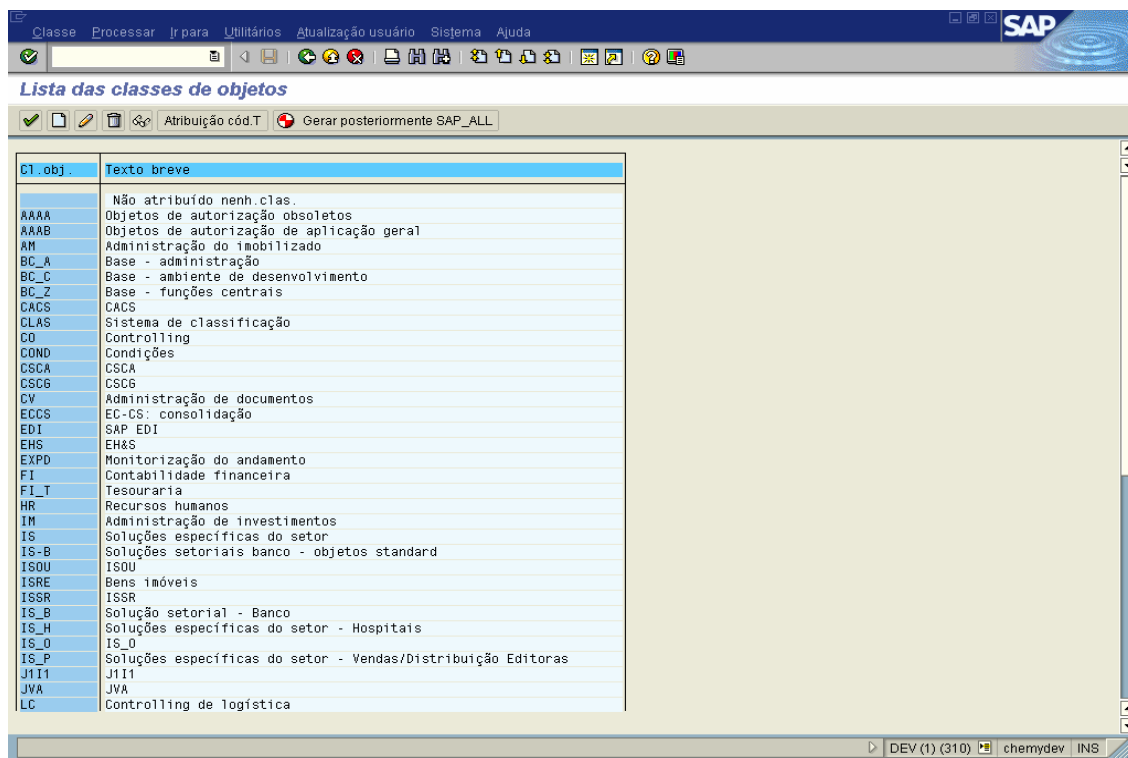
COLOCANDO O MOUSE SOBRE O NOME DO PROGRAMA E CLICANDO COM O BOTÃO DIREITO TEMOS UM MENU DE OPÇÕES. USAMOS A OPÇÃO ATIVAR, PARA GERAR TODOS OS PROGRAMAS E DEPOIS CRIAR->TRANSAÇÃO



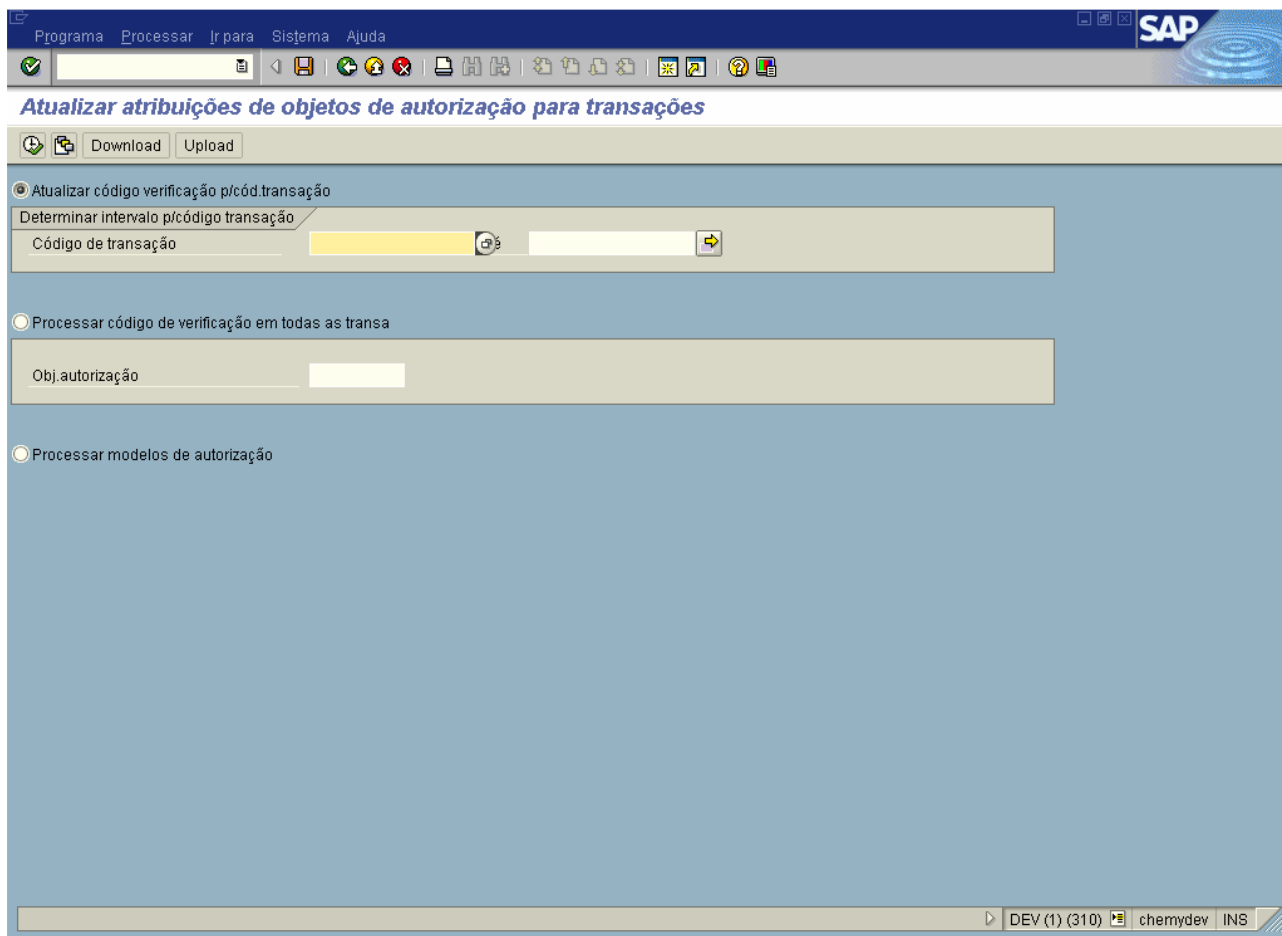
NA TELA DA TRANSAÇÃO INFORMAMOS O NOME DA TRANSAÇÃO, O PROGRAMA QUE SERÁ EXECUTADO E A TELA INICIAL, CLICAMOS EM SALVAR.

OBJETOS DE AUTORIZAÇÃO – TRANSAÇÃO SU21

LISTA E CRIAR OBJETOS COM AS SUAS AUTORIZAÇÕES



SU24 – PRENDE A TRANSAÇÃO COM O OBJETO DE AUTORIZAÇÃO



Como utilizar AUTHORITY-CHECKS

A amarração se dá com um Objeto de autorização para uma ou mais autorizações

O Perfil é composto por um ou mais objetos de autorizações....

O Usuário pode estar presente em N perfis...

Os objetos de autorização, podem ser compostos de vários campos. Conforme o interesse do objeto. Exemplo, pode-se testar a atividade do usuário de acordo com o valor preenchido de um determinado campo...como: Autorização de usuário por empresa campos:

BUKRS - Empresa

ACTVT - Atividade (01-Criação, 02-Alteração e 03-Exibição)

As autorizações podem conter preenchimentos diversos para os objetos de autorização exemplo anterior:

Autorização = S_EXIBIR, campos/valores:

BUKRS → EMP1

ACTVT → 03;

OBSERVAÇÃO:

Estes valores podem ser RANGES ou PARÂMETROS; FIXOS ou VARIÁVEIS.

Em qualquer campo de autorização, o (*) eqüivale a todas as entradas possíveis;

- **Pesquisando os Objetos de Autorização:**
- Caminho: R/3→Ferramentas→ ABAP Workbench →Desenvolvimento→Outras Ferramentas→Objetos de Autorização→Objetos;
- **Transação SU21;**
- Selecionar a classe de Objetos do seu interesse;
- Consultar o Objeto em questão;
- Poderá ser criado novos objetos de autorização....

- **Criando Autorizações;**
- Caminho: R/3→Ferramentas→Administração→Atualização Usuário→Autorização
- **Transação SU03;**
- Selecciona, novamente, a Classe de Objetos desejado;
- Consultar os Objetos existentes, conforme passo anterior;
- Optar pelo objeto de interesse;
- Consultar as Autorizações já existentes para este objeto;
- Poderá ser criado novas autorizações...
- **Atualizar Perfil:**
- Caminho: R/3→ Ferramentas→Administração→Atualização Usuário→Perfil;
- **Transação SU02;**
- **Criação;**
 - Digite o nome do perfil a ser criado;
 - Clicar em ENTER ou GERAR ÁREA TRAB.;
 - Exibirá uma lista vazia;
 - Clicar em Criar, colocar descrição;
 - Perfil individual ou coletivo;
 - Na coluna "Objeto", inserir o nome do Objeto de Autorização;
 - Na coluna "Autorização", inserir o nome da(s) autorizações desejadas;
 - Gravar;
 - Observar STATUS, deve estar ATIVO e GRAVADO
- **Exibição/Modificação;**
 - Digitar o nome do Perfil;
 - Clicar ENTER ou GERAR ÁREA TRAB.;
 - Exibirá a lista contendo a autorização selecionada;
 - Clicar em MODIF, EXCLUIR,...

- **Atualizar Perfil do USUÁRIO:**
- Caminho: R/3→ Ferramentas→Administração→Atualização Usuário→Usuário;
- **Transação SU01;**
- Digitar o usuário e nterar como modificação;
- Navegar até a pasta "Perfis";
- Adicionar o novo perfil à lista do usuário;
- Gravar

Código Fonte Modelo.

Report com duas possibilidades de resposta (AUTORIZADO ou Não Autorizado). Contém uma tela de seleção com um parâmetro para CIA ÁREA.

Quando executado, acionará o evento da tela de seleção e efetuará a verificação de autorização do usuário, conforme digitação em tela...

REPORT ZZZTESTE.

DATA: V_RESULT(20) TYPE C VALUE 'AUTORIZADO'.

PARAMETERS P_CARR LIKE SPFLI-CARRID.

AT SELECTION-SCREEN ON P_CARR.

 AUTHORITY-CHECK OBJECT 'S_CARRID'

 ID 'CARRID' FIELD P_CARR

 ID 'ACTVT' FIELD '02'.

IF SY-SUBRC NE 0.

 MOVE 'Não Autorizado' TO V_RESULT.

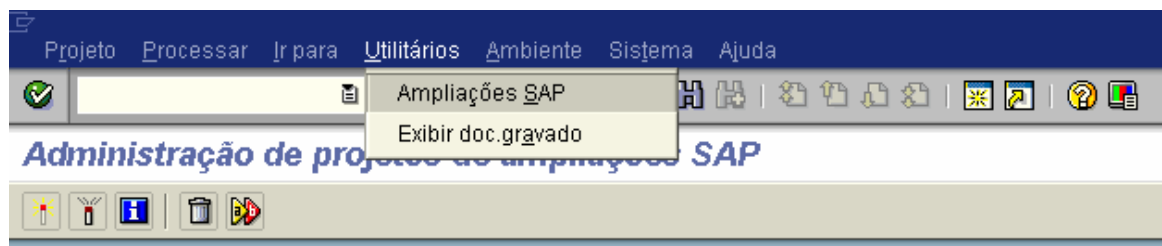
ENDIF.

START-OF-SELECTION.

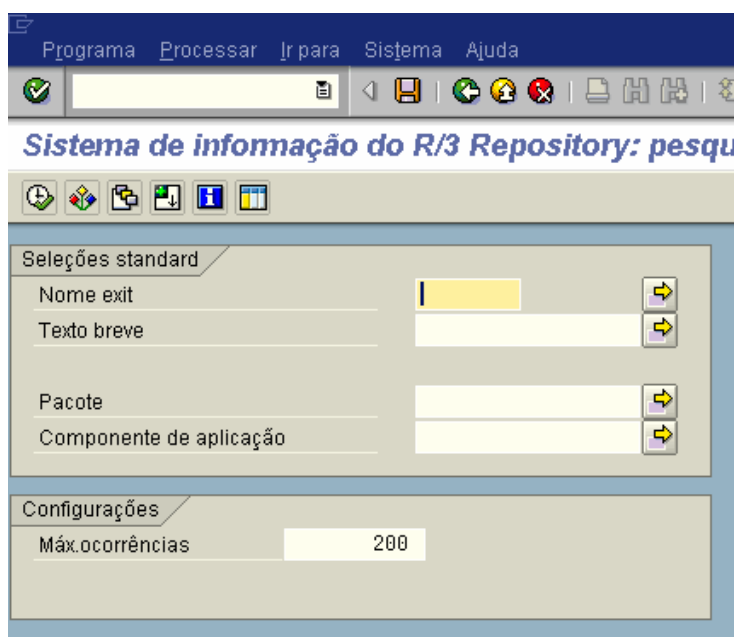
WRITE V_RESULT.



USER-EXIT – TRANSAÇÃO CMOD

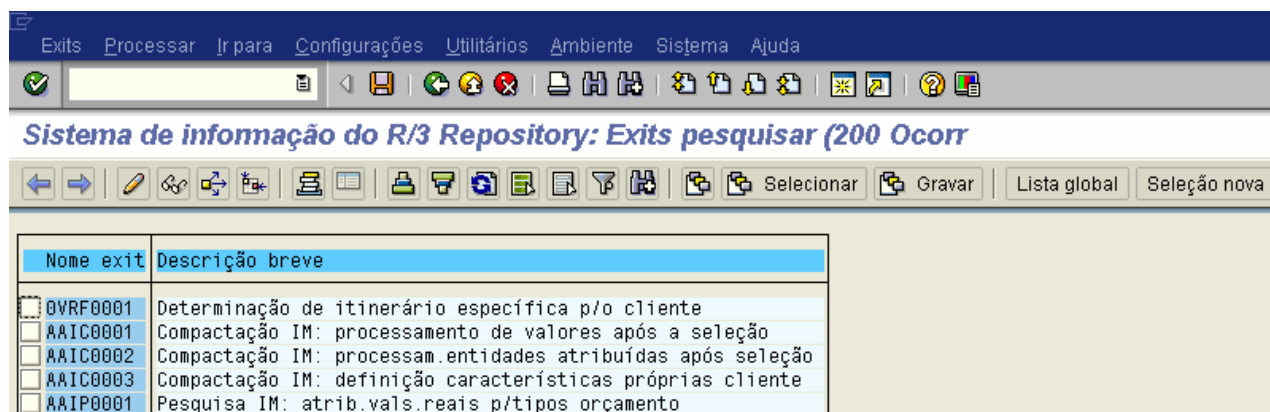
- Ir para : Utilitários > Ampliações SAP



- Executar (F8)

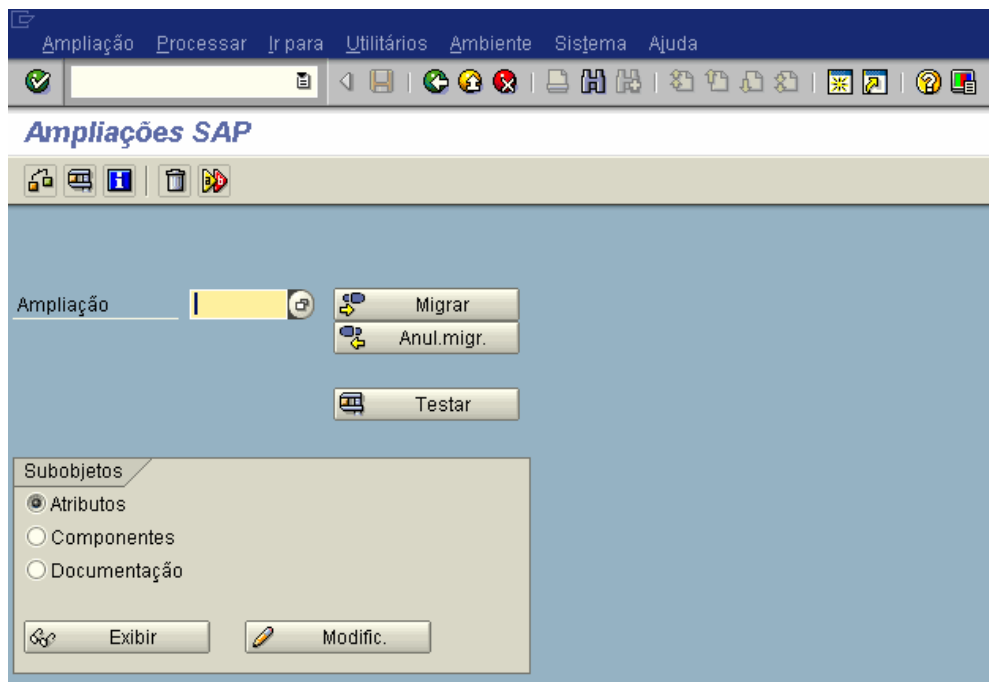


- Selecionar a User exit e clicar em   para exibir ou modificar.

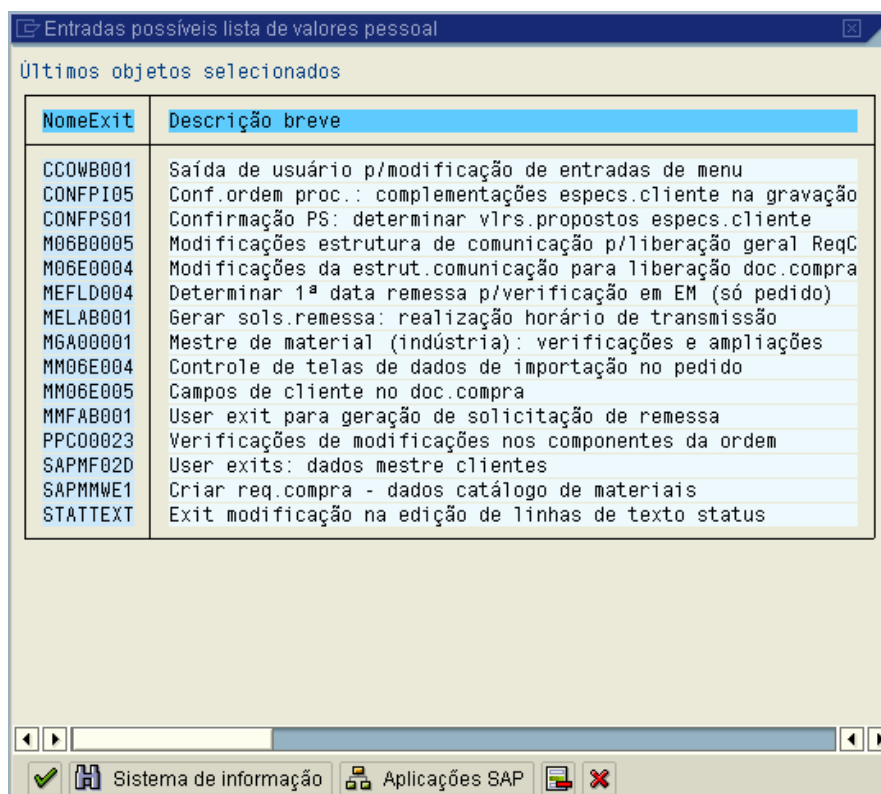


Ativar uma User-EXIT.

- Procure o nome da função chamada por esta customer-function;
- Vá na SMOD, clique sobre o search help (ou F4) na ampliação;



- Ao aparecer a próxima tela, clique no botão de "Sistema de Informação"

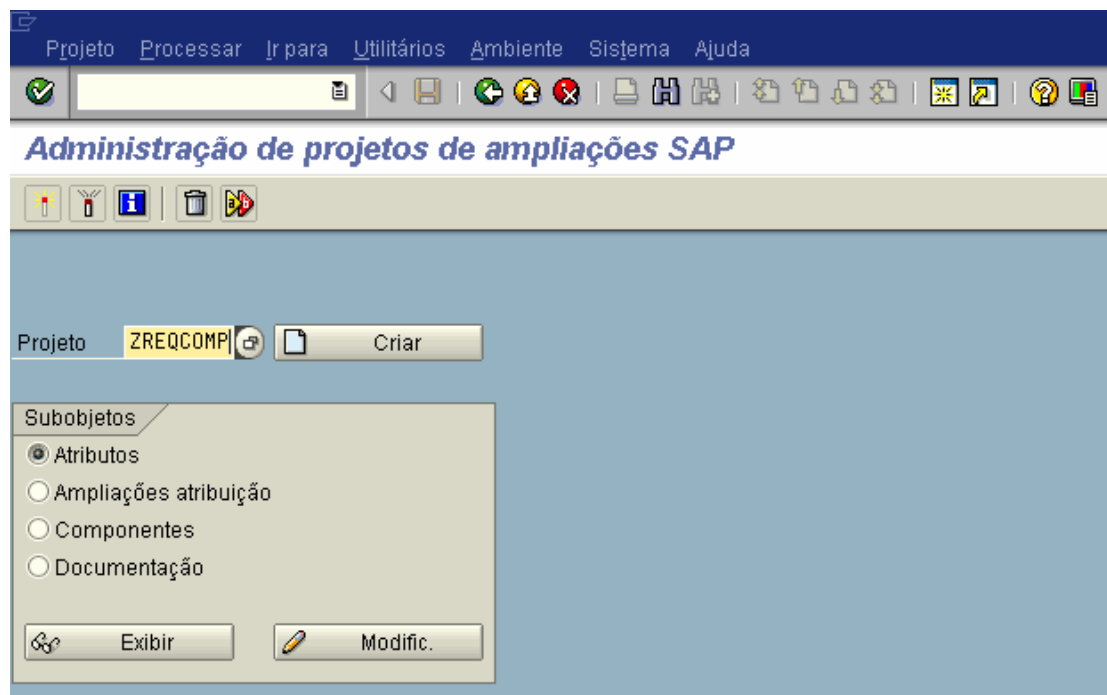


- Na tela seguinte tem um botão que ativa todas as opções de pesquisa (um que tem uma seta para baixo); clique que irá aparecer todas as formas de pesquisas possíveis;

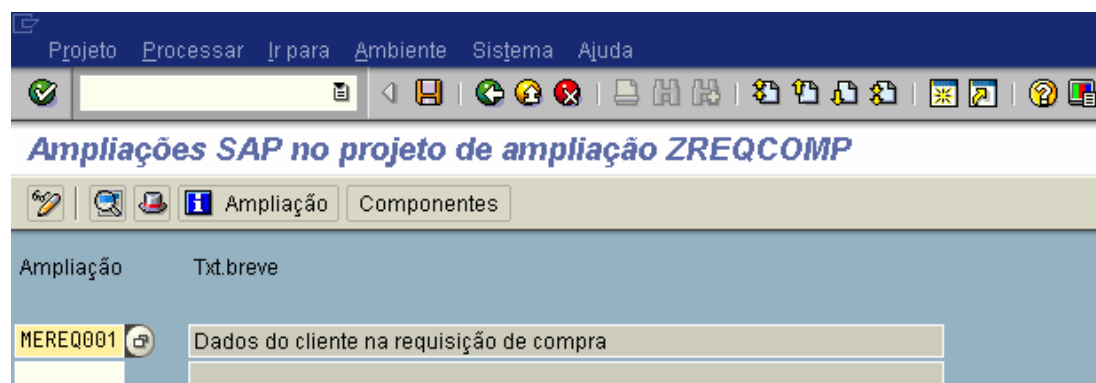
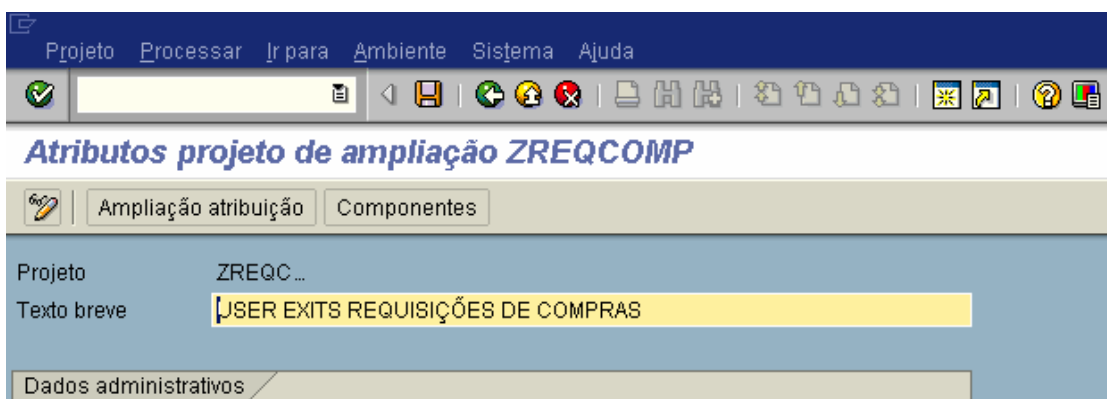
- Em nome do componente, digite o nome da função que você pegou na primeira opção e mande procurar;
- Irá aparecer o nome de uma ampliação. Guarde esse nome.

Nome exit	Descrição breve
M06B0001	Determinação de função para liberação req.compra
M06B0002	Modificações de estrut.comunicação para liberação req.c
M06B0003	number range and document number
M06B0004	number range and document number
M06B0005	Modificações estrutura de comunicação p/liberação geral
M06E0004	Modificações da estrut.comunicação para liberação doc.c
M06E0005	Determinação de função para liberação docs.compra
ME590001	Agrupamento de reqs.compra para partição de pedido na M
MEAG0001	Distribuição contrato/programa de remessas do CRM
MEETA001	Determinar tipo de divisão (atraso, nec.imediata, previ
MEFLD004	Determinar 1ª data remessa p/verificação em EM (só pedi
MELAB001	Gerar sols.remessa: realização horário de transmissão
MEQUERY1	Extensão para síntese de docs. ME21N/ME51N
MEREQ001	Dados do cliente na requisição de compra
MEVME001	Cálculo qtd.proposta EM e tolerância fornec.incomp./exc
MM06E001	User exits para EDI entrada e saída de documentos de co
MM06E003	number range and document number
MM06E004	Controle de telas de dados de importação no pedido
MM06E005	Campos de cliente no doc.compra
MM06E007	Doc.modificação para ReqCs ao converter em pedido
MM06E008	Monitorização do valor fix.contrato para solic.sobre co
MM06E009	Textos relevantes para código "Existem textos"
MM06E010	Seleção de campos para o endereço de fornecedor
MM06E011	Ativação bloqueio de requisição de compra
MMAL0001	ALE distribuição LOF: proc.de saída
MMAL0002	ALE distribuição LOF: proc.de entrada

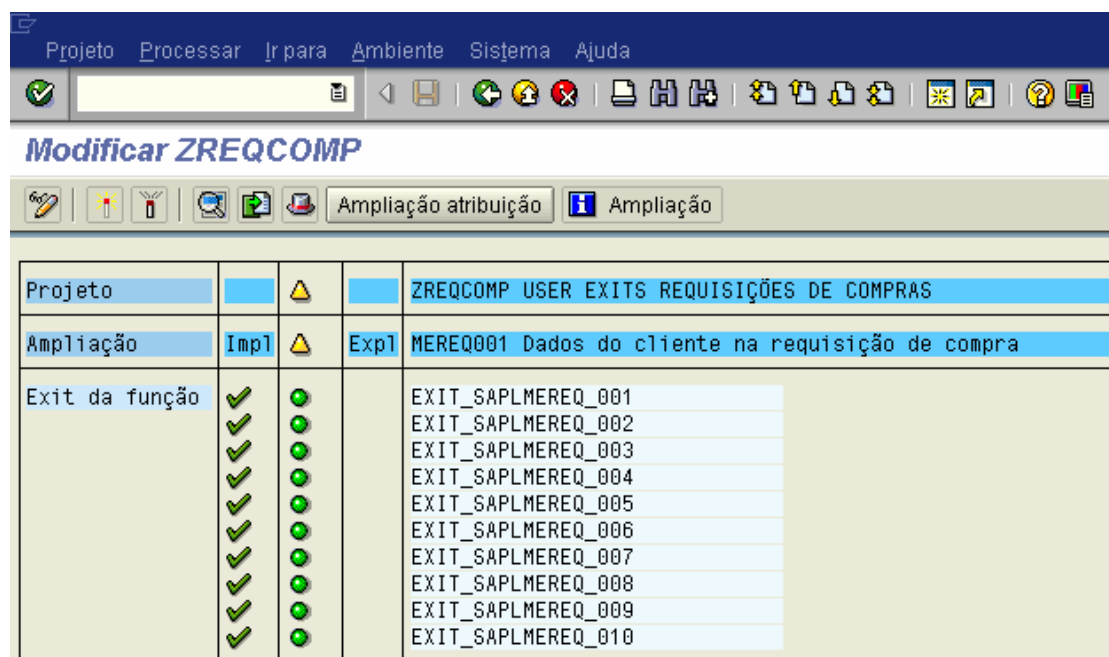
- Vá na transação CMOD e crie um projeto.



- Tem um botão para associar ampliação ao projeto. Digite o nome da ampliação que vc pegou no item 6.

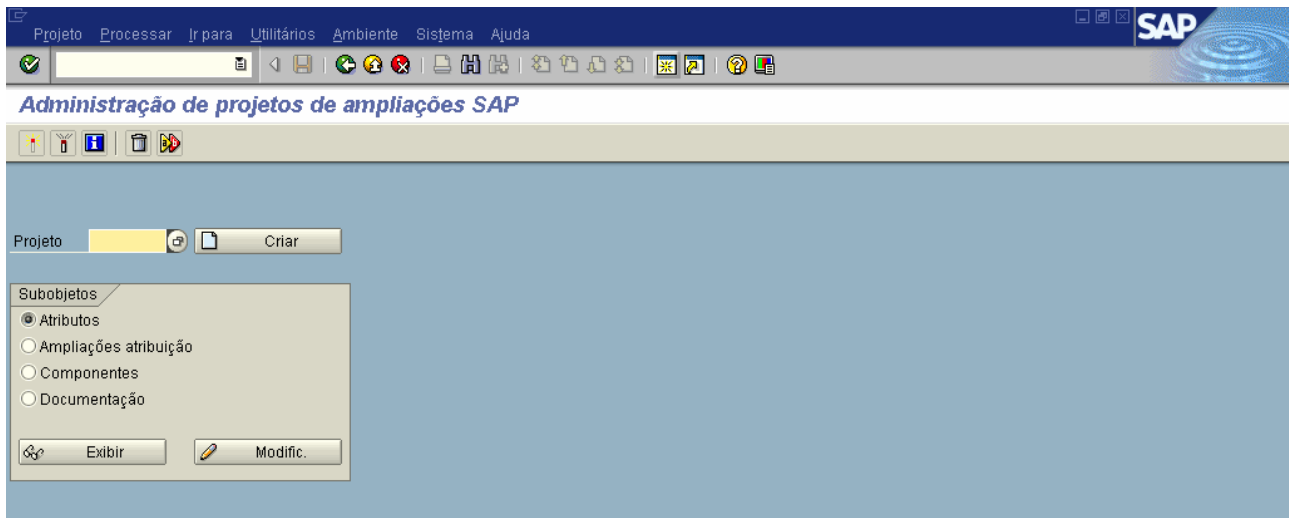


- O próximo passo é criar o código desejado para o funcionamento da user-exit. Neste caso, entre na função (ou via SE37 ou dentro da própria CMOD) e faça o código (clique sobre o include que aparece dentro da função para criar um programa separado).



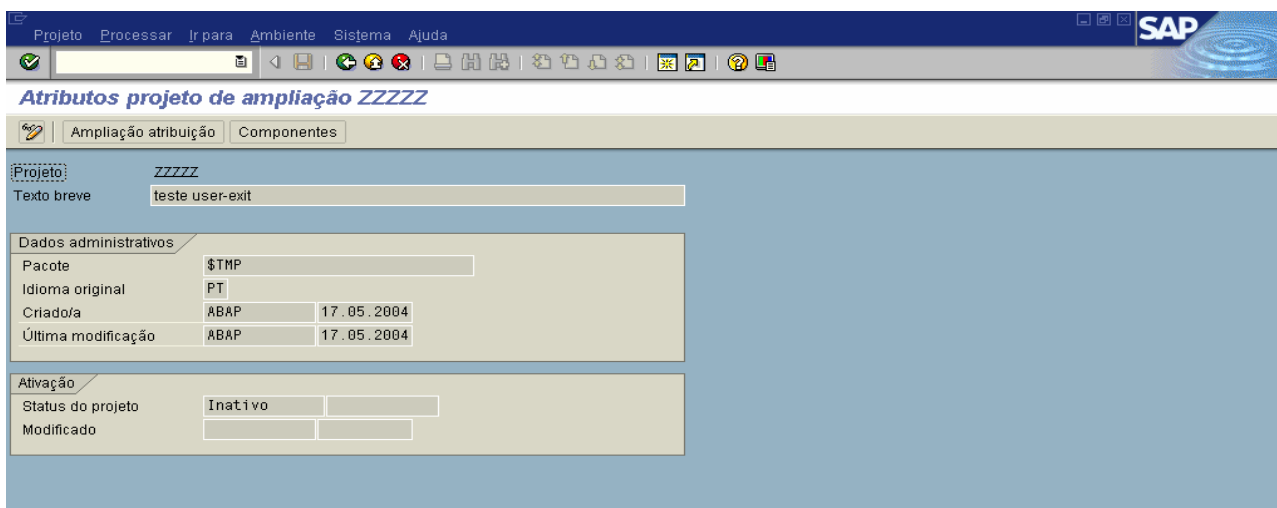
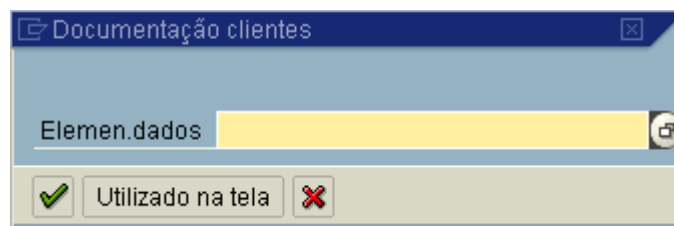
Para ativar, na tela inicial da CMOD tem um botão específico.

FIELD-EXIT -> (ESTA RELACIONADA A LOGICA DO CAMPO, AO PASSAR O CURSOR)
TRANSAÇÃO CMOD – para localizar uma FIELD-EXIT

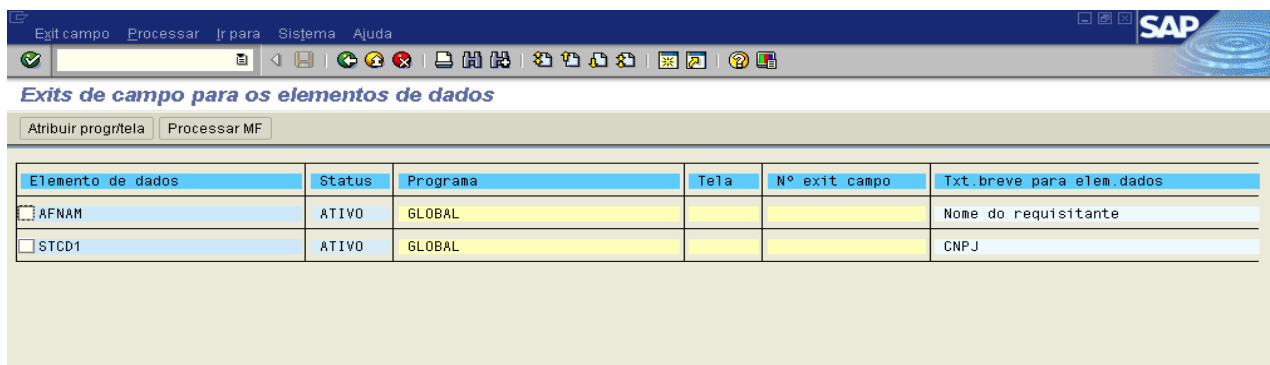


MENU : IR PARA-> AMPLIAÇÕES GLOBAIS->ELEMENTO DE DADOS->NOVO DOC.CLIENT.ED

FIELD AMARRAR EM UM PROJETO

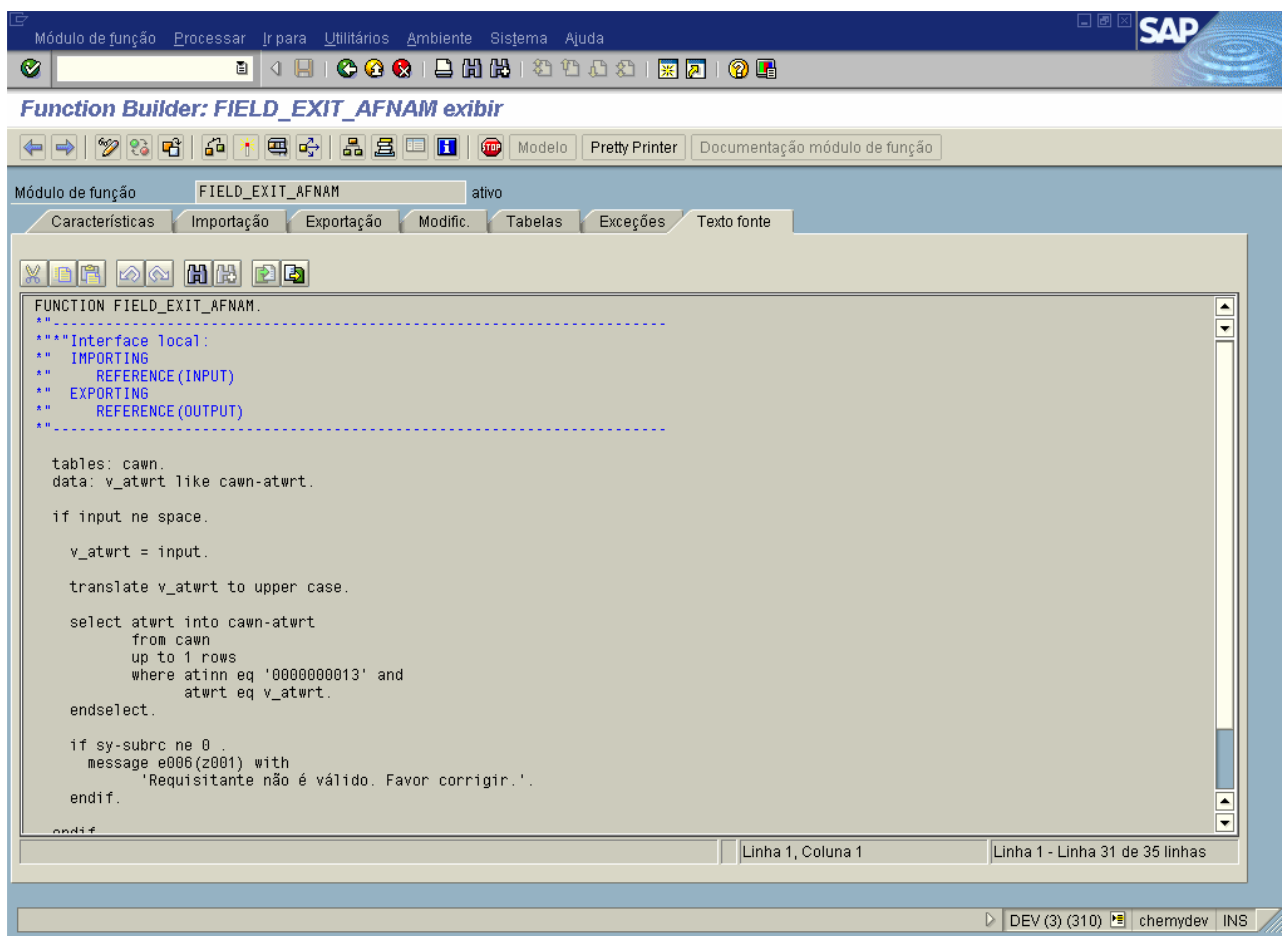
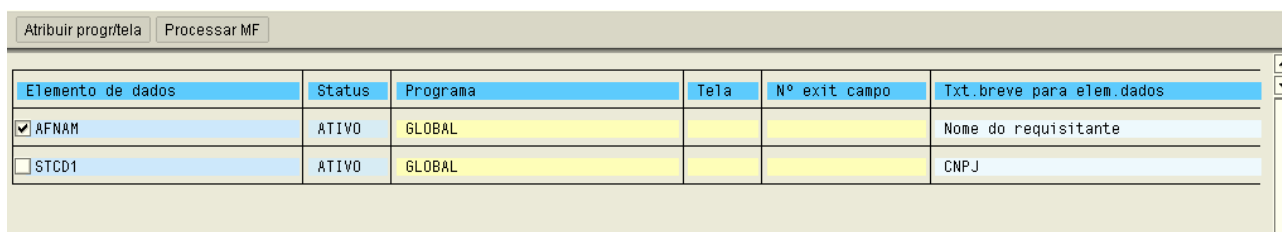


FIELD-EXIT - NA TELA CMOD DIGITAR =PRFB e DAR ENTER



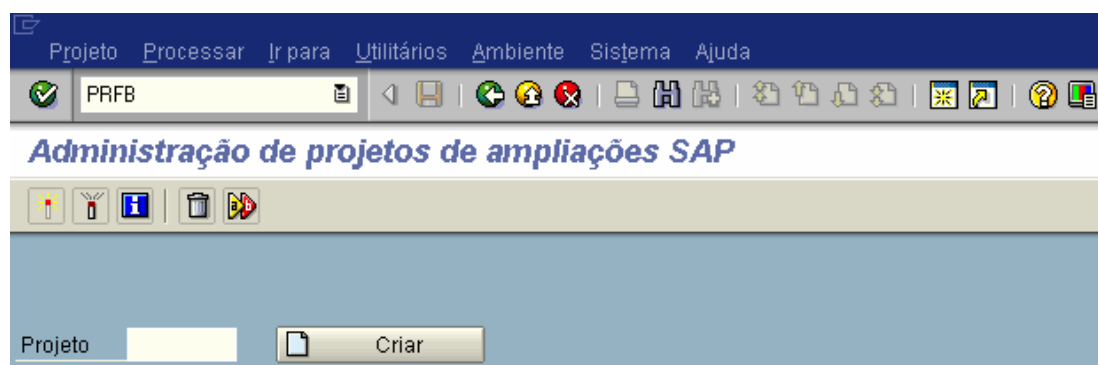
MARCAR QUAL ELEMENTOS DE DADOS E CLICAR EM

Processar MF

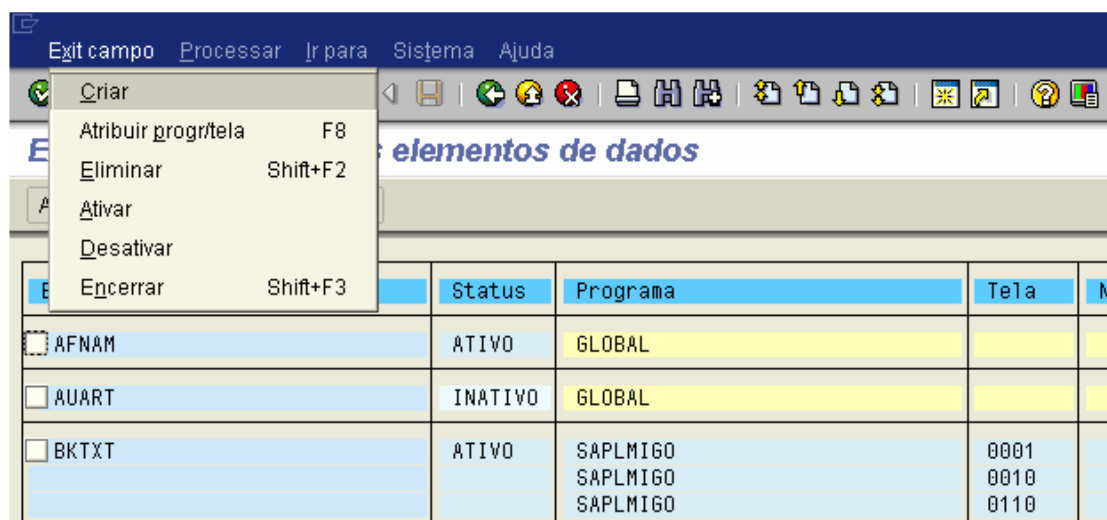


PARA CRIAR UM FIELD-EXIT

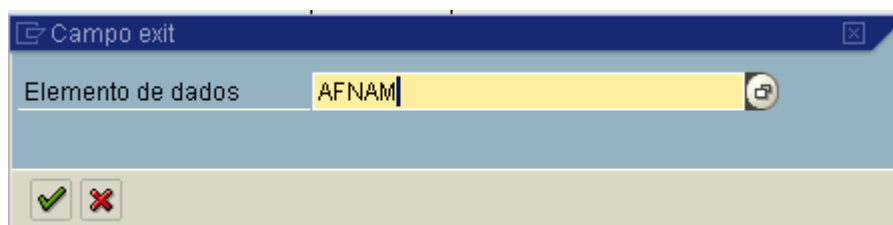
- Como primeiro passo, devemos executar a transação a onde iremos colocar o Filed-Exit, e coletar os dados :
 1. nome do campo
 2. elemento de dados
 3. nome do programa
 4. numero da tela
- Em seguida executar a transação CMOD : Dentro da transação digitar na caixa de dialogo PRFB e dar enter.



- No menu superior escolher : Exit campo > Criar.



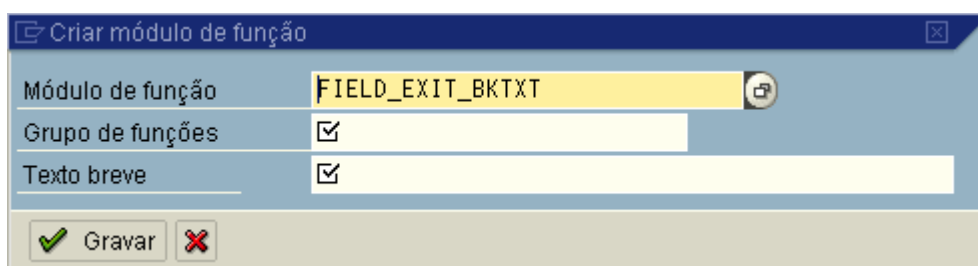
- Informar o nome do elemento de dados



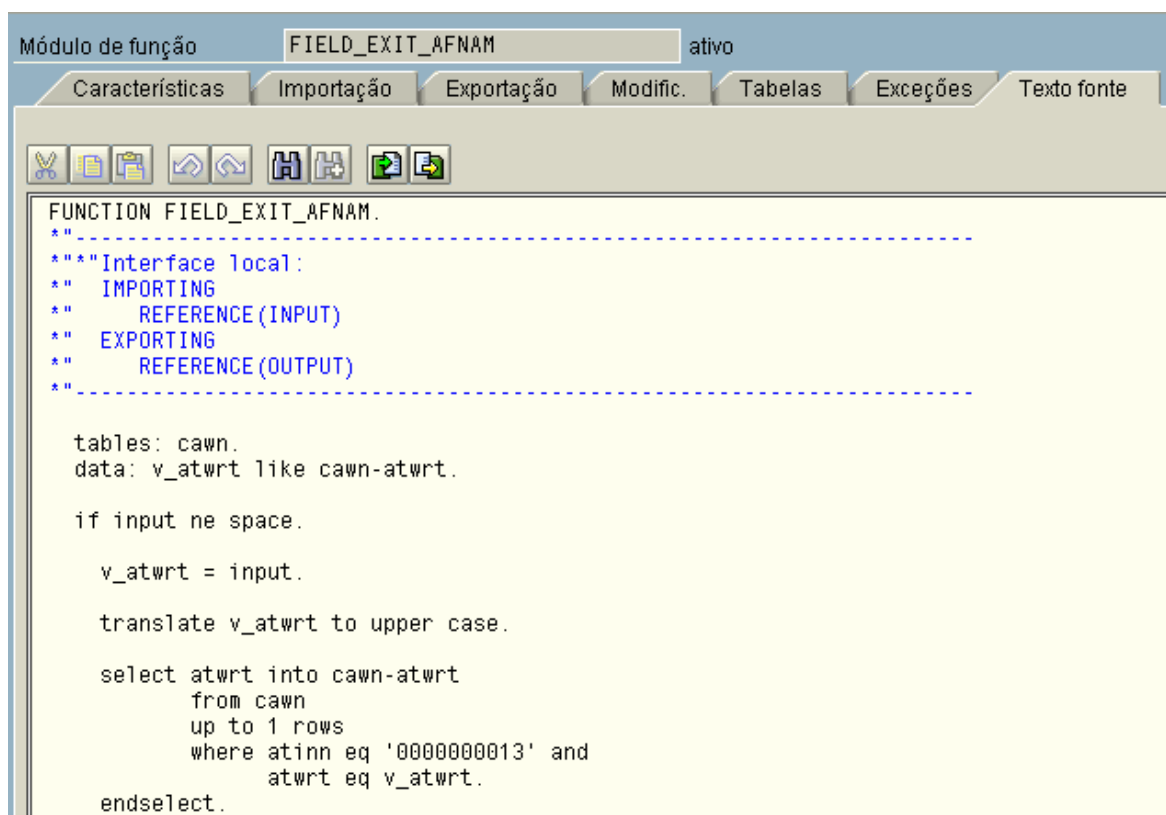
- Criar a função para a Field-Exit

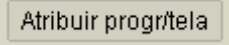


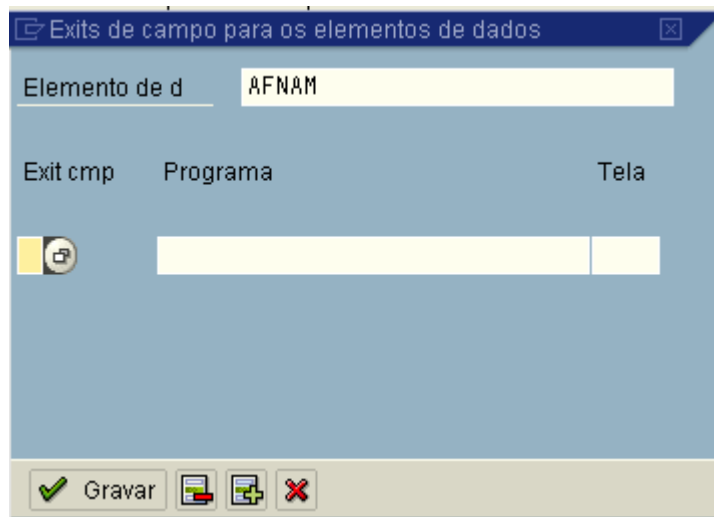
- Informar grupo de funções e texto breve



- Digitar o código fonte da função e salvar.



- Marcar o item criado e clicar em  e preencher a tela que será mostrada com o nome do programa e tela. Desta forma a Field-Exit será validada para este programa. Para ativar para todos os programas, não preencher a tela, apenas confirmar.



A Field exit, permite que seja feita alguma seleção ou checagem de um determinado campo no programa e tela desejados.

Para isso, se faz necessário, buscar o elemento de dados do campo que se deseja fazer a field exit.

Ir até a transação CMOD, clicar AMPLIAÇÕES TEXTO(menu), depois escolher Exits campo, aparecerão todas as fields existentes.

Para se criar uma nova:

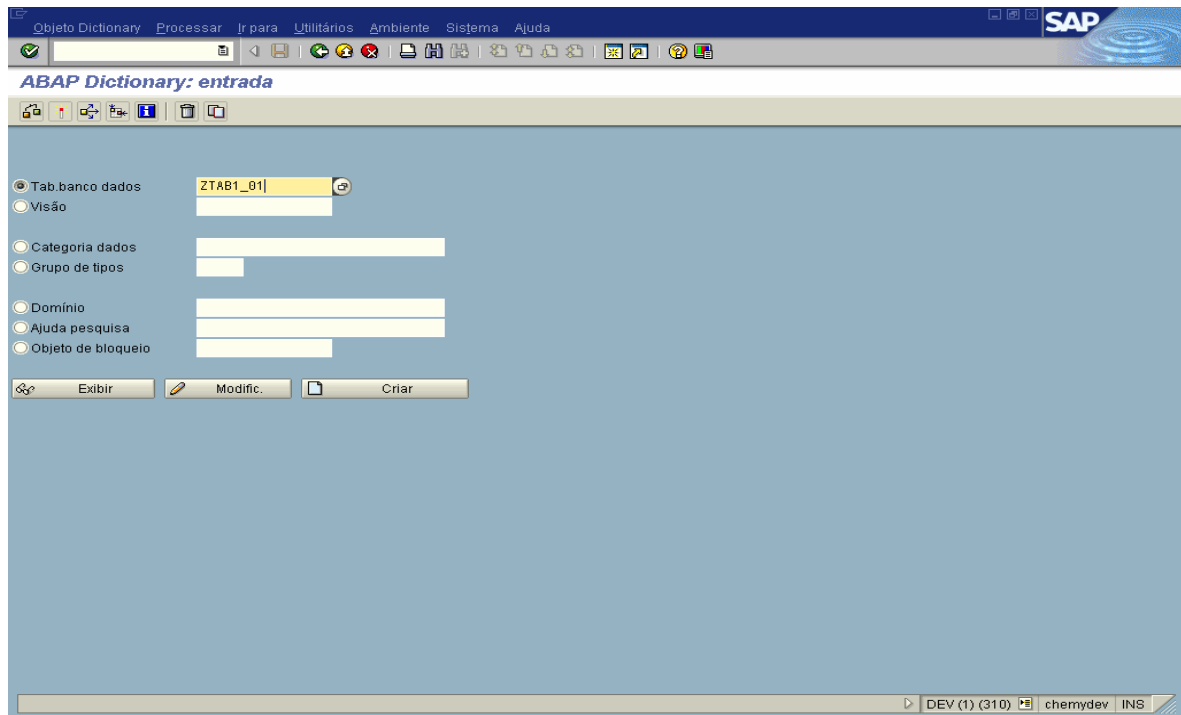
- 1) Exit campo (menu)
- 2) Criar
- 3) Digitar o elemento de dados - Avançar
- 4) Digitar o código, como uma função
- 5) Depois clicar no botão Atribuir progr/tela, colocando o nome do programa e o número da tela, p/ pegar estas informações, clicar F1 e F9, no campo desejado
- 6) Visualiza ou modifica o conteúdo da field, no botão Processar MF, deve-se selecionar o elemento de dado desejado
- 7) Ativar a field exit

IMPORTANTE: Na field exit, vc precisa pegar o valor digitado no campo desejado, p/ isso existe a importação e a exportação, ou seja, as variáveis INPUT e OUTPUT, vc precisa sempre colocar OUTPUT = INPUT, p/ que o valor possa voltar p/ tela origem.

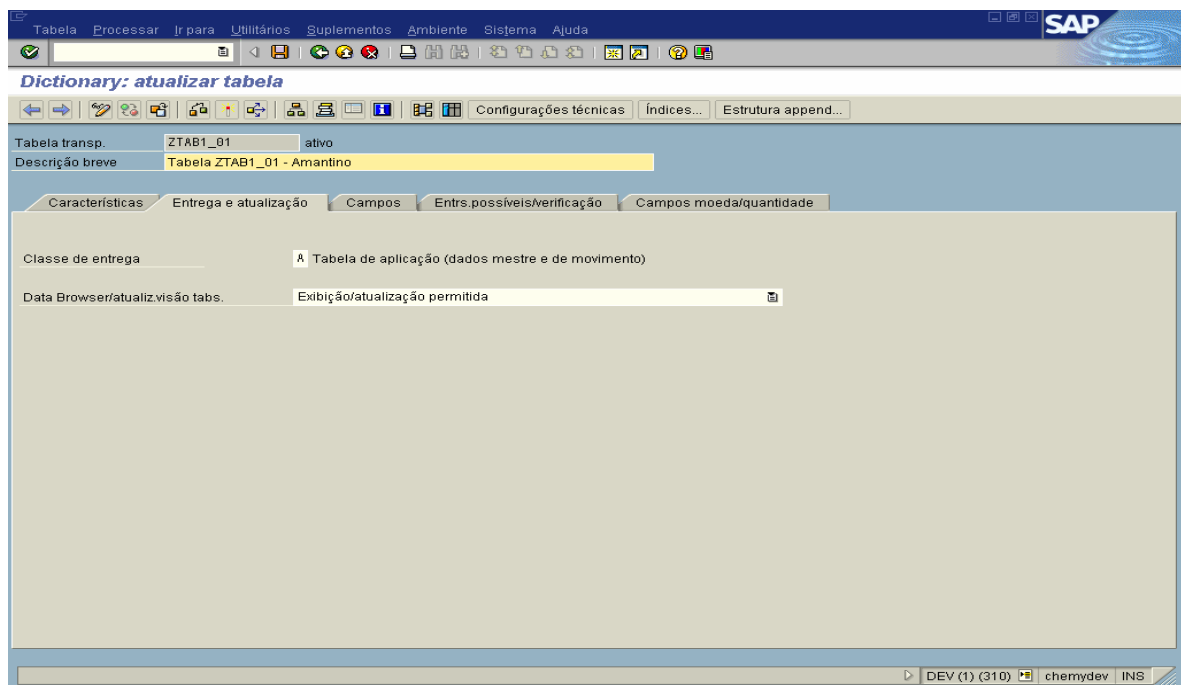
Observação: A Field exit só funcionará, se a mesma estiver ativa.

CRIAR TABELAS

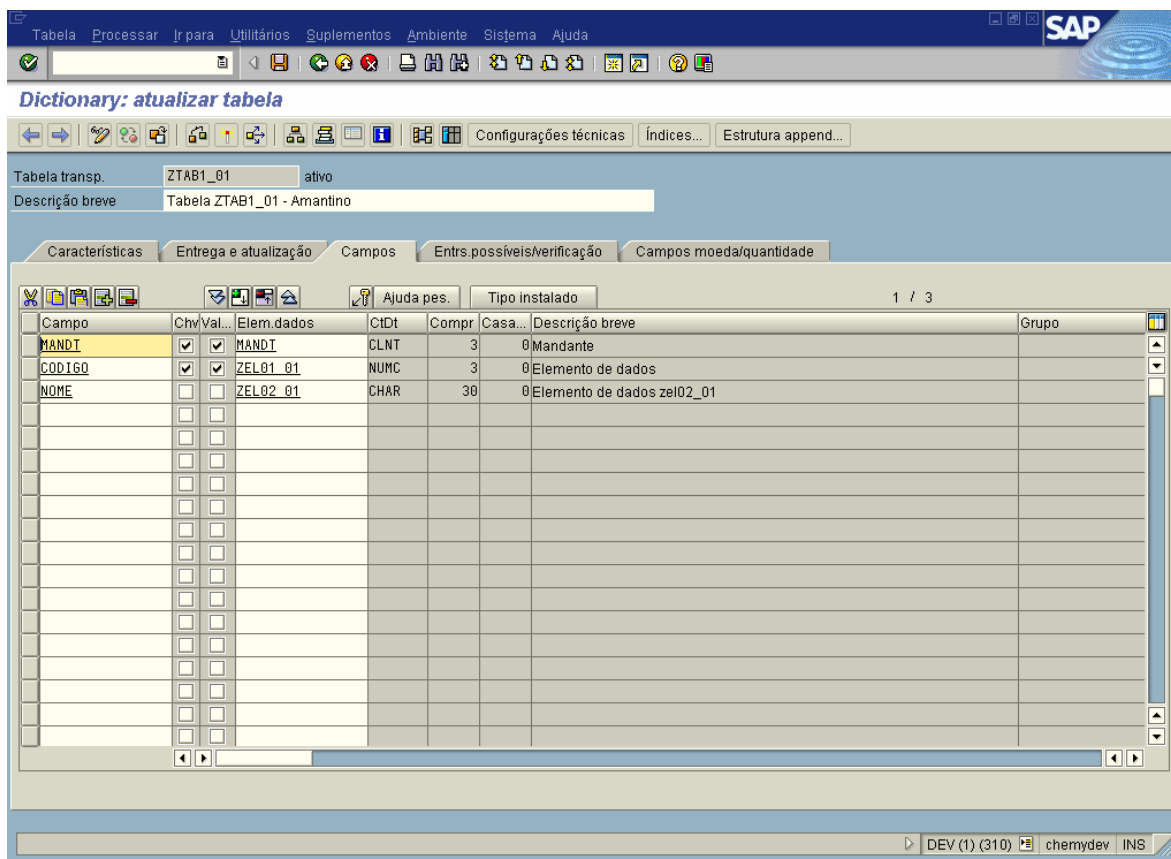
TRANSAÇÃO SE11



COLOCAR O NOME DA TABELA E BOTÃO **CRIAR**



COLOCAR EM CLASSE DE ENTREGA SEMPRE **A**
EM **DATA BROWSER/ATUALIZAÇÃO** COLOCAR CONFORME CARACTERISTICA DA TABELA



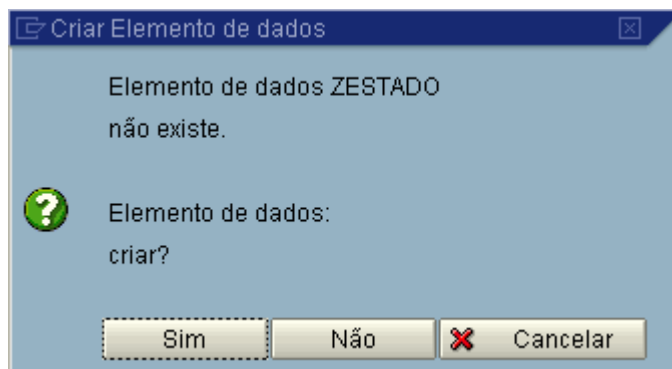
COLOCAR OS CAMPOS E FLAG CONFORME A NECESSIDADE DA TABELA.

CHAVE -> **INDICA SE O CAMPO É CHAVE**
Val. -> **SE O CAMPO PODE CONTER NULOS**

OBS : SEMPRE COLOCAR O CAMPO MANDT (MANDANTE)

ENTRS.POSSIVEIS DE VERIFICAÇÃO E CAMPOS MOEDAS, SERVEM PARA INDICAR SE HAVERA TABELAS ASSOCIADAS PARA PESQUISA E VALORES.

PARA CRIAR UM ELEMENTO DE DADOS, DEVEMOS ESCREVER O NOME DO ELEMENTO DE DADOS E CLICAR DUAS VEZES SOBRE ELE.



CONFIRMAR A CRIAÇÃO DO ELEMENTO DE DADOS.

Elemento de dados: ZESTADO novo(revisado)

Descrição breve: ESTADO

Caracts. Ctg.dds. Características adicion. Denomin.campo

☒ Categoria elementar

☒ Domínio

Ctg.dds. Compr 0 Casas decimais 0

☐ Tipo instalado

Ctg.dds. Compr 0 Decimais 0

☐ Tipo de referência

☐ Nome tipo refer.

☐ Referência a tipo instalado

Ctg.dds. Compr 0 Casas decimais 0

DEV (1) (310) chernydev INS

ESCREVER O NOME DO DOMINIO E CLICAR DUAS VEZES SE ELE NÃO EXISTIR.

Elemento de dados: ZESTADO novo(revisado)

Descrição breve: ESTADO

Caracts. Ctg.dds. Características adicion. Denomin.campo

Breve Compr Denominador de campo

0 0 0

0 0 0

0 0 0

0 0 0

DEV (1) (310) chernydev INS

EM **DENOMINAÇÃO CAMPOS**, COLOCAR OS NOMES QUE DEVERÃO APARECER NAS TELAS E RELATÓRIOS.

The dialog box 'Criar entrada catálogo objetos' has a title bar with a close button. It contains a tabbed interface with the 'Objeto' tab selected. The 'Objeto' field contains 'R3TR DTEL ZESTADO'. Below this is the 'Atributos' (Attributes) section with the following fields: 'Pacote' (empty), 'Responsável' (ABAP), 'Sistema de origem' (DEV), and 'Idioma original' (PT Português). At the bottom, there are four buttons: 'Objeto local' (with a floppy disk icon), 'Síntese bloqueio' (with a person icon), and two standard window control buttons (a magnifying glass and a red X).

DEFINIR O OBJETO LOCAL OU REQUEST

The dialog box 'Criar Domínio' has a title bar with a close button. It displays the message 'Domínio ZDMESTDAO não existe.' (Domain ZDMESTDAO does not exist.) followed by a question mark icon and the text 'Domínio: criar?' (Domain: create?). At the bottom, there are three buttons: 'Sim' (Yes), 'Não' (No), and 'Cancelar' (Cancel).

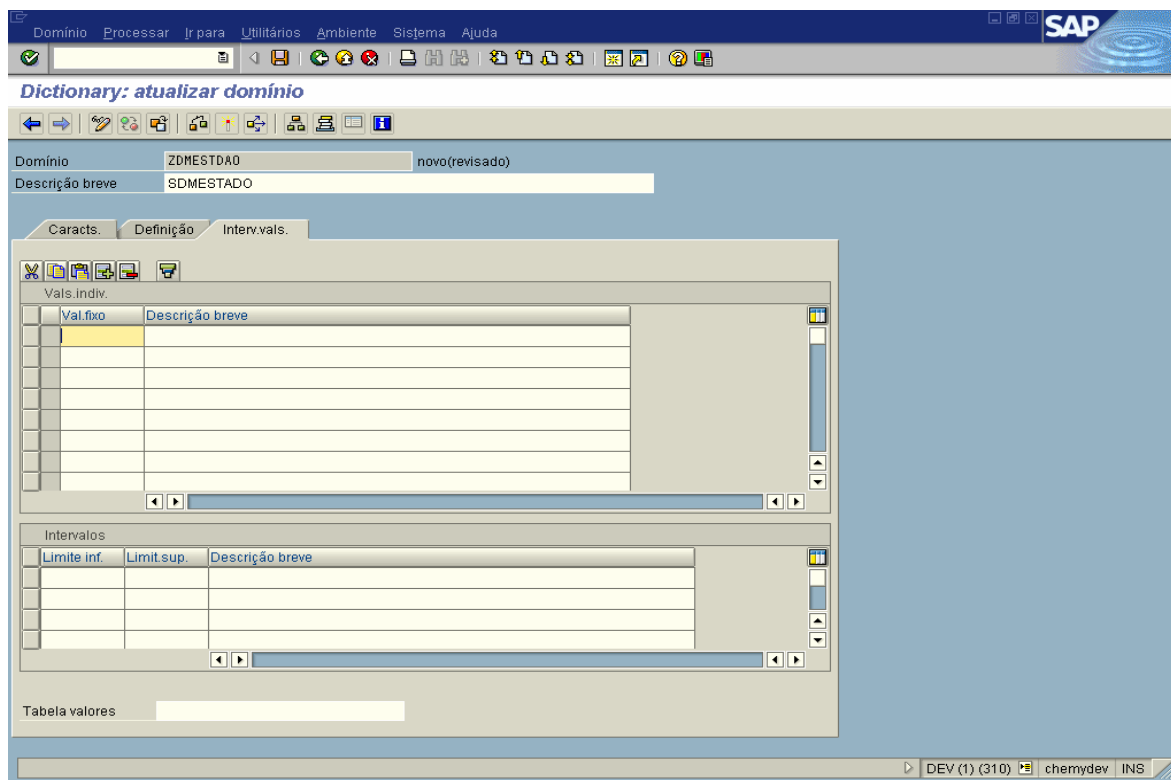
CRIAR O DOMINIO

The screenshot shows the SAP Dictionary: atualizar domínio (Update Domain) screen. The title bar includes the SAP logo and menu items: Domínio, Processar, Ir para, Utilitários, Ambiente, Sistema, Ajuda. The main area has a toolbar with various icons. Below the toolbar, the 'Domínio' field contains 'ZDMESTDAO' and 'novo(revisado)'. The 'Descrição breve' field contains 'SDMESTADO'. There are three tabs: 'Caracts.', 'Definição', and 'Interv.vals.'. The 'Definição' tab is active, showing the 'Formato' (Format) section with fields for 'Categoria dados' (empty), 'Núm.posições' (empty), and 'Decimais' (empty). Below this is the 'Características da saída' (Output Characteristics) section with fields for 'Comprim.saída' (empty), 'Rotina conversão' (empty), and checkboxes for 'Sinal +/-' and 'Letr.minúsculas'. At the bottom, there is a status bar with a red X icon and the text 'Preencher todos os campos obrigatórios' (Fill in all mandatory fields). The right side of the status bar shows 'DEV (1) (310)', 'chemydev', and 'INS'.

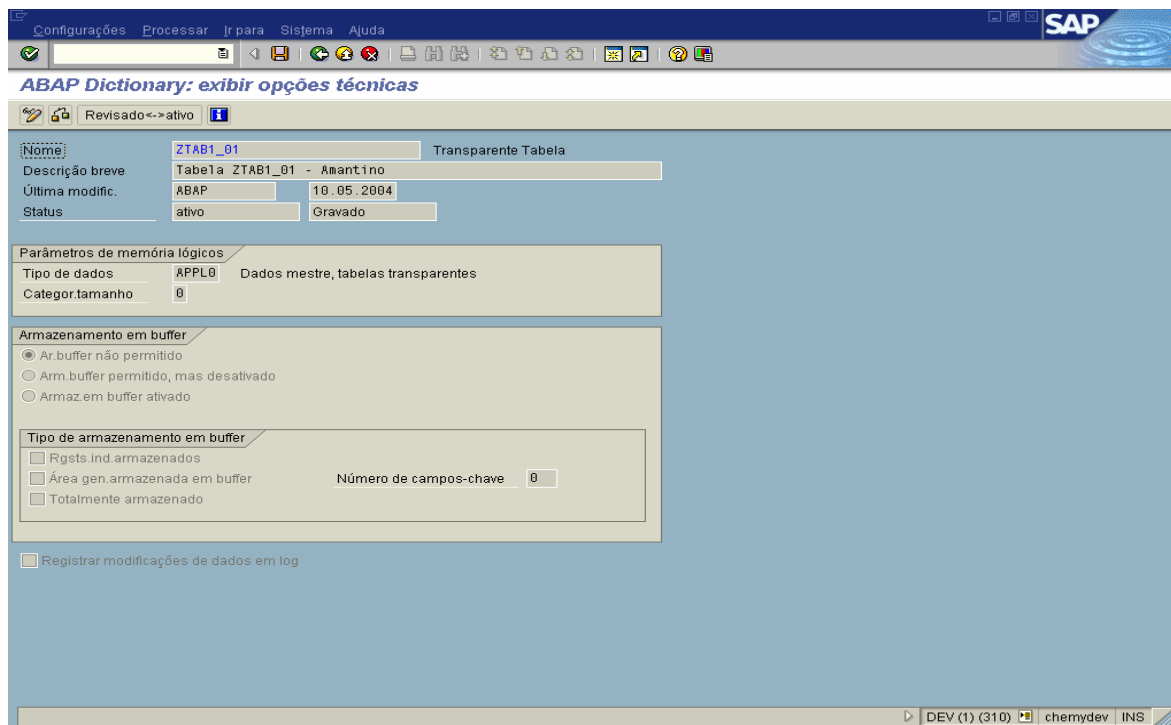
EM **DEFINIÇÃO** COLOCAR A CATEGORIA DE DADOS, NUMERO DE POSIÇÕES, EM **SAIDA** DEFINIR O TAMANHO DO CAMPO PARA RELATORIOS, SE ELE PODERA SER NEGATIVO E ACEITAR LETRAS MINUSCULAS

OS TIPO DE DADOS:

Categoria de dados no ABAP Dictionary	
CtDt	Descrição breve
ACCP	Período contábil AAAAMM
CHAR	Cadeia de caracteres
CLNT	Mandante
CUKY	Código da moeda, é referido por campos CURR
CURR	Campo de moeda (divisa), arquivado como DEC
DATS	Campo de data (AAAMDD), arquivado como carac(8)
DEC	Campo de cálculo ou montante com vírgula e sinal pos./neg.
FLTP	Número em ponto flutuante com precisão de 8 bytes
INT1	Nº inteiro de 1 byte, nº inteiro <= 255
INT2	2 byte nº inteiro, só p/campo de compr.antes de LCHR ou LRAW
INT4	Nº inteiro de 4 byte, nº inteiro com sinal positivo/negativo
LANG	Código de idioma
LCHR	Seq.longa de caracteres, necessita campo INT2 anterior
LRAW	Seq.longa de bytes, necessita campo INT2 anterior
NUMC	Cadeia de caracteres só com algarismos
PREC	Precisão de um campo QUAN
QUAN	Campo quantidade, aponta p/campo de unidades de formato UNIT
RAW	Sequência de bytes não interpretada
RAWSTRING	Seqüência de bytes de comprimento variável
SSTRING	Cadeia de caracteres breve de comprimento variável
STRING	Cadeia de caracteres de comprimento variável
TIMS	Campo da hora (hhmmss), arquivado como campo de texto (6)
VARC	Cad.longa caracteres, já não suport.a partir do release 3.0
UNIT	Código de unidades para campos QUAN



EM INTERVALOS PODEMOS DEFINIR UMA FAIXA OU VALORES QUE O CAMPO PODEM ACEITAR. VOLTAR ATIVANDO AS TELAS. QUANDO CHEGAR NA TELA PRINCIPAL, IR EM **CONFIGURAÇÕES TÉCNICAS**



COLOCAR O TIPO DE DADOS E A CATEGORIA DE TAMANHO E SALVAR E DEPOIS ATIVAR A TELA.

PRINCIPAIS TABELAS GENÉRICAS

- JEST - Status Individual por Objeto
- JSTO - Informações sobre Objeto de Status
- KNA1 - Mestre de Clientes (Parte Geral)
- KNB1 - Mestre de Clientes (Empresa)
- KNB4 - Histórico de Pagamentos do Cliente
- KNB5 - Mestre de Clientes (Dados de Reclamação)
- KNC1 - Mestre de Clientes (Movimentação no Período)
- KNC3 - Mestre de Clientes (Movimentação no Período - Razão Especial)
- KNVV - Mestre de Clientes (Vendas e Distribuição)
- LFA1 - Mestre de Fornecedores (Parte Geral)
- LFAS - Mestre de Fornecedores (Parte Geral Ident. Fiscal IVA)
- LFAT - Mestre de Fornecedores (Agrupamento de Impostos)
- LFB1 - Mestre de Fornecedores (Empresa)
- LFB5 - Mestre de Fornecedores (Dados de Reclamação)
- LFBK - Mestre de Fornecedores (Banco)
- LFBW - Mestre de Fornecedores (Categoria de Imposto Retido na Fonte)
- LFC1 - Mestre de Fornecedores (Movimento no Período)
- LFC3 - Mestre de Fornecedores (Movimento no Período - Razão Especial)
- LFM1 - Mestre de Fornecedores (Organização de Compras)
- T000 - Mandantes
- T001 - Empresas
- T001Z - Dados Adicionais para Empresa
- T005T - Denominação dos Países
- T012K - Bancos
- T074T - Denominação dos Códigos de Razão Especial
- T074U - Características dos Códigos de Razão Especial
- TBTCO - Síntese de estado de job
- TCURR - Taxas para conversão monetária
- TJ30T - Textos Relativos a Status de Objetos
- V_T015Z - Visão dos valores em extenso para conversão de numéricos na função standard SPELL_AMOUNT.

Principais Tabelas de CO e FI

- BKPF - Cabeçalho de Documentos Gerais de FI
- BSAD - Itens de Partidas Compensadas de Clientes
- BSAK - Itens de Partidas Compensadas de Fornecedores
- BSAS - Itens de Partidas Compensadas de Contas do Razão
- BSEG - Itens de Documentos Gerais de FI
- BSID - Itens de Partidas em Aberto de Clientes
- BSIK - Itens de Partidas em Aberto de Fornecedores
- BSIS - Itens de Partidas em Aberto de Contas do Razão
- COBK - Cabeçalho do Documento Nº CO Referente ao Período
- COEP - Partidas no CO Referente ao Período
- COSP - Valores das Ordens de Investimento e Outras Ordens
- CSKS - Mestre de Centro de Custo
- CSKT - Texto de Centro de Custo
- GLT0 - Mestre da Conta do Razão (Movimentação no Período)
- GLT1 - Totais para General Ledger Local
- J_1AT059Z - Códigos de IRF (Nova Funcionalidade)
- J_1AWITH - Dados de Operações de IRF
- J_1AWTOFF - Código Oficial de Imposto de Renda na Fonte
- SKA1 - Mestre das Contas do Razão
- T001S - Encarregado da Contabilidade

- T011 - Estrutura de Balanço L/P
- T030A - Operações (Ex. "WIT" Operações de IRF)
- T894 - Versões de Ledger
- TBSL - Mestre da Chave de Lançamento
- TBSLT - Denominação das Chaves de Lançamento
- TGSB - Divisões
- TGSBT - Denominação das Divisões
- TKA01 - Área de Contabilidade de Custos
- TKA09 - Opções Básicas Versões de Ordens
- TKA50 - Perfis de Planejador - Tabela de Entidades
- TKO08 - Área de Apropriação de Custo para Ordem
- TKO09 - Texto para Área de Apropriação de Custo para Ordem

Principais Tabelas HR

- HRS1200 - Infotipo standard 1200 (SAP) atribuição de função a tarefa
- HRS1203 - Infotipo standard 1203 (SAP) funções executáveis
- PA2001 - Registro horas pessoal: infotipo 2001 (ausências)
- PA2002 - Registro horas pessoal: infotipo 2002 (presenças)
- PA2003 - Registro horas pessoal: infotipo 2003 (substituição)
- PA2004 - Registro horas pessoal: infotipo 2004 (disponibilidade)
- PA2005 - Registro horas pessoal: infotipo 2005 (horas extras)
- PA2006 - Registro horas pessoal: infotipo 2006 (contingente ausências)
- PA2007 - Registro horas pessoal infotipo 2007 (autorização presença)
- PA2010 - Registro horas pessoal: infotipo P2010 (info remun.empr.)
- PA2012 - Registro horas pessoal: infotipo 2012 (revisão do saldo)
- PA2013 - Infotipo reg.tempo pess.2013 (correções contingente)

Principais Tabelas de MM

- EKBE - Histórico de Documento de Compras
- EKKN - Classificação Contábil de Documento de Compras
- EKKO - Cabeçalho de Documento de Compras
- EKPO - Item de Documento de Compras
- MAKI - Textos breves de Materiais
- MARA - Mestre de Materiais
- MARC - Segmento C do Mestre de Materiais
- MARM - Unidades de Medida
- MBEW - Avaliação do Material
- MKPF - Cabeçalho de Documento de Materiais (Movimentações)
- MSEG - Item de Documento de Materiais (Movimentações)
- T134T - Denominação dos Tipos de Material

Principais Tabelas de PP

- AFKO - Cabeçalho da Ordem PCP
- AFPO - Item da Ordem PCP
- AUFK - Mestre de Ordens
- MAST - Ligação entre Materiais (Lista Técnica)
- PLAF - Ordem Planejada
- STKO - Cabeçalho da Lista Técnica
- STPO - Item da Lista Técnica
- T003P - Textos de Tipos de Ordens

Principais Tabelas de SD

- J_1BBRANCH - Filial do CNPJ
- J_1BNFDOC - Nota Fiscal - Cabeçalho
- J_1BNFLIN - Nota Fiscal - Itens
- LIKP - Remessa/Fornecimento - Cabeçalho
- LIPS - Remessa/Fornecimento - Itens
- T171T - Clientes Zona de Distribuição - Textos
- TVFK - Documento de Faturamento - Tipo de Documento
- TVGRT - Unidade de Organização - Grupo de Vendedores
- VBAK - Ordens de Venda - Cabeçalho
- VBAP - Ordens de Venda - Itens
- VBFA - Fluxo de Documento de Venda
- VBRK - Faturamento - Cabeçalho
- VBRP - Faturamento - Itens
- VTTK - Transporte - Cabeçalho
- VTPP - Transporte - Itens

Obter as tabelas relacionadas a uma determinada tabela, via Lógica Database

Uma das maneiras de se saber quais outras tabelas estão associadas a uma tabela, é verificando se existe um logical database para tabela em questão. Através da seguinte seleção de opções de menu:

```
FERRAMENTAS >  
  ABAP/4 WORKBENCH > DESENVOLVIMENTO >  
    AMBIENTE DE DESENVOLVIMENTO > BANCO DE DADOS LÓGICO
```

(Chega-se à transação SE36)

SUPLEMENTOS > UTILIZAÇÃO DE TABELA

Informar a tabela que se deseja verificar

Verificar, através dos Bancos de Dados Lógicos listados, as ligações entre a tabela pretendida e as demais.

SAPSCRIPT – Etapas de Criação

The screenshot shows the SAP Form Painter interface. The title bar includes menu items: Formulário, Processar, Ir para, Utilitários, Configurações, Ambiente, Sistema, Ajuda. The main window title is "Form Painter: 1ª tela". On the left, there is a "Subobjetos" (Subobjects) list with radio buttons for: Cabeçalho (selected), Páginas, Janela, Janela de página, Formatos parágrafo, Formatos caracteres, and Documentação. Below this list are buttons for "Exibir" and "Modific.". At the top left, there are fields for "Formulário" and "Idioma" (set to PT), and a "Criar" button. The status bar at the bottom right shows "DEV (1) (310)", "chemydev", and "INS".

The screenshot shows the "Exibir cabeçalho do formulário: Z_NOTA_FISCAL" screen. The title bar includes menu items: Formulário, Processar, Ir para, Atributos, Utilitários, Configurações, Sistema, Ajuda. The main window title is "Exibir cabeçalho do formulário: Z_NOTA_FISCAL". At the top, there are tabs: "Páginas", "Janela", "Janelas de página", "Formatos de parágrafo", "Formatos de caracteres", and a help icon. Below these are two main sections: "Dados administrativos" and "Configurações globais". The "Dados administrativos" section contains the following information:

Informação de administração					
Formulário	Z_NOTA_FISCAL				
Significado	Nota fiscal				
Status	ativo - gravado				
Classificação					
Pacote	ZPRD	Package (Classe de Desenvolvimento) PRD			
Nº mandante	310				
Criado em	17.01.1996	13:00:09	de	SAP	Release 30B
Modificado em	15.03.2004	08:29:23	de	DDIC	Release 620

Below this is the "Atributos idioma" (Language Attributes) section with the following fields:

Código de idioma	PT
Idioma original	PT
Traduzir	
<input type="radio"/> Em todos os idiomas <input checked="" type="radio"/> em alguns idiomas <input type="radio"/> Não traduzir	

The status bar at the bottom right shows "DEV (1) (310)", "chemydev", and "INS".

CLICAR EM

Configurações globais

Formulário Processar Ir para Atributos Utilitários Configurações Sistema Ajuda

Exibir cabeçalho do formulário: Z_NOTA_FISCAL

Páginas Janela Janelas de página Formatos de parágrafo Formatos de caracteres

Dados administrativos Configurações globais

Configurações globais

Instalar página

Formato página ZNFISC

Orientação

☐ Formato vertical

☒ Formato horizontal

Linhas/polegada 6,00

Caracs/polegada 10,00

Página inicial NF

Valores propostos p/formatação texto

Parágrf.default HD

Stop tabulador 1,00 CM

Família fontes TIMES

Tamanho fonte 10,0 Ponto

☐ Negrito

☐ Itálico

☐ Sublinhado

Espaçamento 0,00

Espessura 0,00

Intensidade 0 %

DEV (1) (310) chemydev INS

Parágrafo

Parág.	Significado
FA	Fatura
HD	Cabeçalho
IS	Emissor
LI	Linha de detalhes
MG	Message
NF	Tipo da nota fiscal
PA	Parceiro
T1	Transporte (volume)
TT	Nota fiscal - totais
TX	Textos

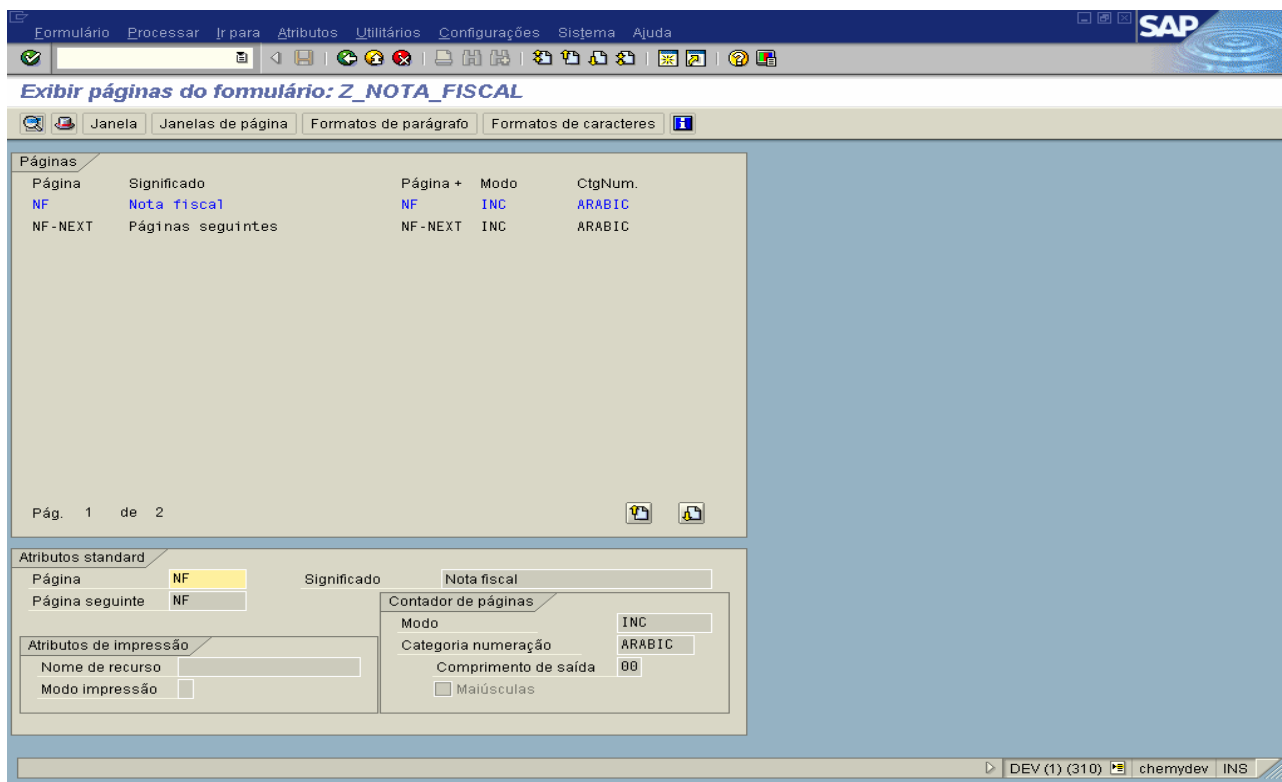
☒
☐
☐
☒

Atributo de fonte nome da família de fontes

Família	Significado	PROP
A_SIMP	Arabic Simplified	<input checked="" type="checkbox"/>
A_SIMP_B	Arabic Simplified Bold	<input checked="" type="checkbox"/>
A_SIMP_F	Arabic Simplified Fixed	<input type="checkbox"/>
A_TRAD	Arabic Traditional	<input checked="" type="checkbox"/>
A_TRAD_B	Arabic Traditional Bold	<input checked="" type="checkbox"/>
A_TRAN	Arabic Transparent	<input checked="" type="checkbox"/>
A_TRAN_B	Arabic Transparent Bold	<input checked="" type="checkbox"/>
ANDALE_J	Andale Mono WT J - UTF8	<input type="checkbox"/>
ANDALE_K	Andale Mono WT K - UTF8	<input type="checkbox"/>
ANDALE_S	Andale Mono WT S - UTF8	<input type="checkbox"/>
ANDALE_T	Andale Mono WT T - UTF8	<input type="checkbox"/>
CNHEI	DB Font China	<input type="checkbox"/>
CNKAI	DB Font China	<input type="checkbox"/>
CNSONG	Standard DB font China	<input type="checkbox"/>
COUR_I7	ISO-7: Greek Courier	<input type="checkbox"/>
COURCYR	ISO-5: Cyrillic Courier	<input type="checkbox"/>
COURIER	Courier	<input type="checkbox"/>
DBGOTHIC	Standard DB Font Japan	<input type="checkbox"/>
DBMINCHO	DB Font Japan	<input type="checkbox"/>
HELV_I7	ISO-7: Greek Helvetica	<input checked="" type="checkbox"/>
HELVCYR	ISO-5: Cyrillic Helvetica	<input checked="" type="checkbox"/>
HELVE	Helvetica	<input checked="" type="checkbox"/>
JPMINCHO	Standard DB font Japan	<input type="checkbox"/>
KPBATANG	Standard DB Font Korea	<input type="checkbox"/>
KPDODUM	Standard DB Font Korea	<input type="checkbox"/>
KPGULIM	DB Font Korean	<input type="checkbox"/>
KPGUNGSE	Standard DB Font Korea	<input type="checkbox"/>
KPSAMMUL	Standard DB font Korea	<input type="checkbox"/>
LETGOTH	Letter Gothic	<input type="checkbox"/>
LNPRINT	Line Printer	<input type="checkbox"/>
MICR_C	MICR CMC-7 magnetic ink font	<input type="checkbox"/>
MICR_E	MICR E-13B magnetic ink font	<input type="checkbox"/>
OCRA	OCR-A	<input type="checkbox"/>

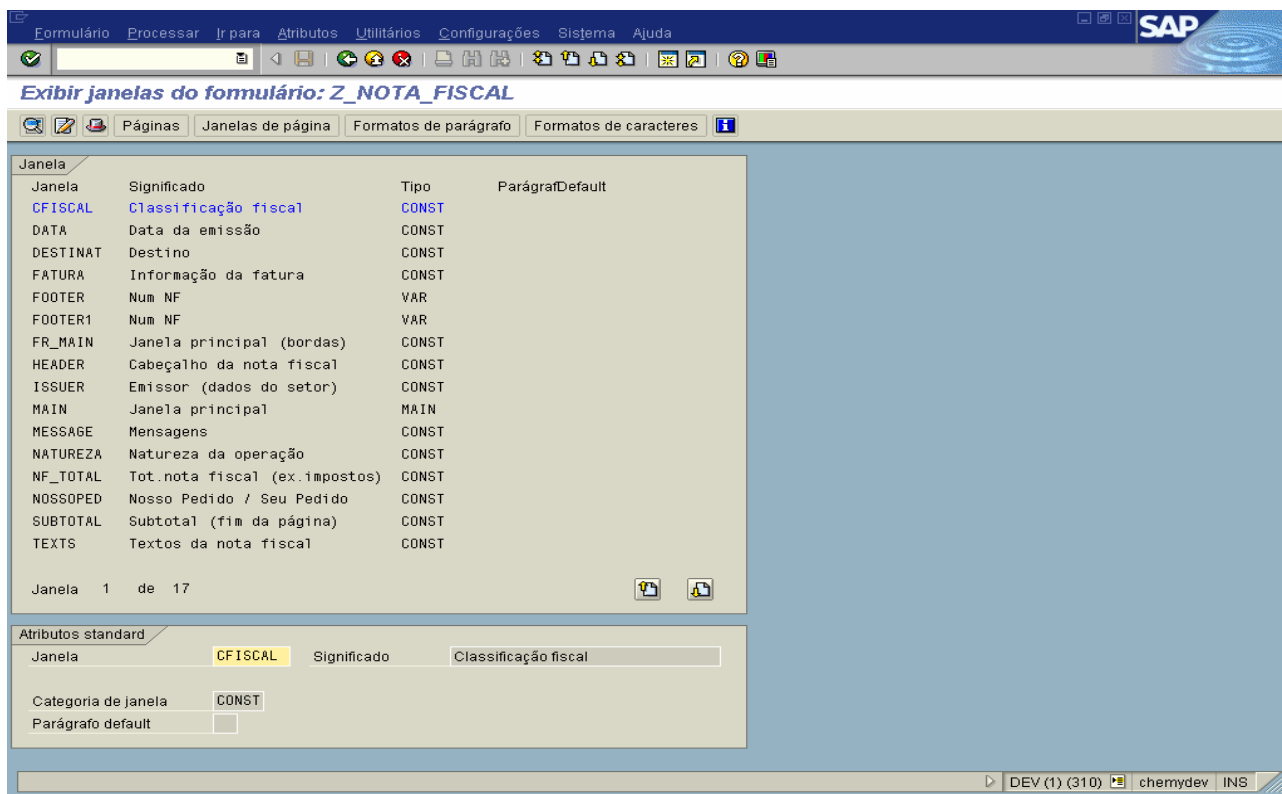
☒
☐
☐
☐
☒


Clicar em Páginas



NF-NEXT -> RESET E ENVIA PARA NF . SEMPRE TER 2 PAGINAS.

CLICAR EM **Janela**.



CLICAR 2 VEZES SOBRE A JANELA E CLICAR EM  PARA DETALHAR O CONTEUDO DA JANELA.

Pré-visualização para LOCL página 00001 de 00001

Pág. NF

DEV (1) (310) chemydev INS

Janelas de página

Formulário Processar Ir para Atributos Utilitários Configurações Sistema Ajuda

Exibir janelas de página do formulário: Z_NOTA_FISCAL

Páginas Janela Formatos de parágrafo Formatos de caracteres

Janela de página

Página NF

Janela	Significado	Esq.	Cima	Larg.	Alt.
MAIN 00	Janela principal	0,00 CM	9,00 CM	29,50 CM	7,50 CM
DATA	Data da emissão	22,00 CM	2,95 CM	3,00 CM	2,50 CM
DESTINAT	Destino	0,50 CM	2,97 CM	26,30 CM	2,50 CM
FATURA	Informação da fatura	0,00 CM	5,90 CM	29,50 CM	2,20 CM
FOOTER	Num NF	11,00 CM	25,50 CM	2,00 CM	1,00 CM
FOOTER1	Num NF	24,00 CM	25,50 CM	2,00 CM	1,00 CM
HEADER	Cabeçalho da nota fiscal	0,50 CM	0,00 CM	28,00 CM	2,00 CM
NF_TOTAL	Tot.nota fiscal (ex.impostos	0,50 CM	16,40 CM	29,50 CM	1,70 CM
SUBTOTAL	Subtotal (fim da página)	18,00 CM	25,50 CM	1,00 CM	1,00 CM
TEXTS	Textos da nota fiscal	0,50 CM	22,00 CM	15,00 CM	7,00 LN
TRANSPOR	Transporte	0,50 CM	19,00 CM	29,50 CM	2,50 CM

Janelas pág. 1 de 11

Atributos standard

Janela MAIN Significado Janela principal

CategoriaJanela MAIN 0

Margem esquerda 0,00 CM Largura janela 29,50 CM

Margem superior 9,00 CM Altura janela 7,50 CM

DEV (1) (310) chemistrydev INS

CLICAR EM Formatos de caracteres

Formulário Processar Ir para Atributos Utilitários Configurações Sistema Ajuda

Exibir cadeias de caracteres do formulário: Z_1BNOTA_FISCAL

Páginas Janela Janelas de página Formatos de parágrafo Formatos de caracteres

Formatos de caracteres

FrmCarac.	Significado	Marc.	Proteg	Ocul.	Sob.	Sub.
B	Negrito	Off				
HD	Bloco do cabeçalho	Off				
I	Itálico	Off				
LF	Fonte grande	Off				
SM	Texto pequeno	Off				
U	Sublinhado	Off				

FormatoCaract 1 de 6

Atributos standard

Formato caract. B Signific. Negrito

Standard

Fonte

☐ Marcação
 ☐ Protegido
 ☐ Oculto
 ☐ Sobrescrito
 ☐ Subscrito

Cód.barras

DEV (1) (310) chemistrydev INS

CLICAR EM

Fonte

Atributos de fonte

Formato caract. HD Signific. Bloco do cabeçalho

Standard

Fonte

Família TIMES

Tam. 8,0 Ponto

Negrito ON Off Manter

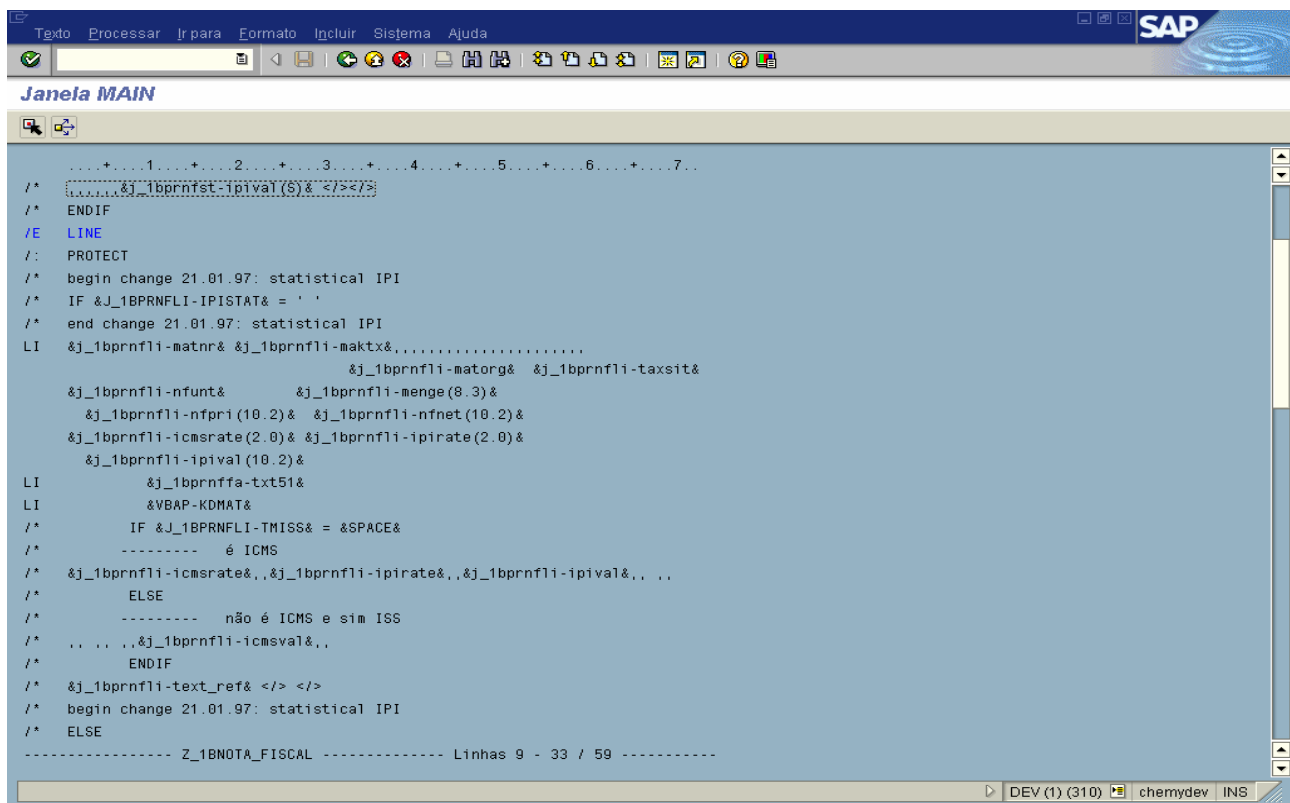
Itálico

Sublinhado

➔

Coluna de formato	
Formato	Significado
*	Parágr.default
FA	Fatura
HD	Cabeçalho
IS	Emissor
LI	Linha de detalhes
MG	Message
NF	Tipo da nota fiscal
PA	Parceiro
T1	Transporte (volume)
TT	Nota fiscal - totais
TX	Textos
	Txt.cont.
=	Lnh.longa
(Lnh.Bruta
/	Avanço linha
/=	Avanço linha e linha longa
/ (Avanço linha e linha bruta
/:	Linha de comando
/*	Linha de comentário
/E	Elemento de texto

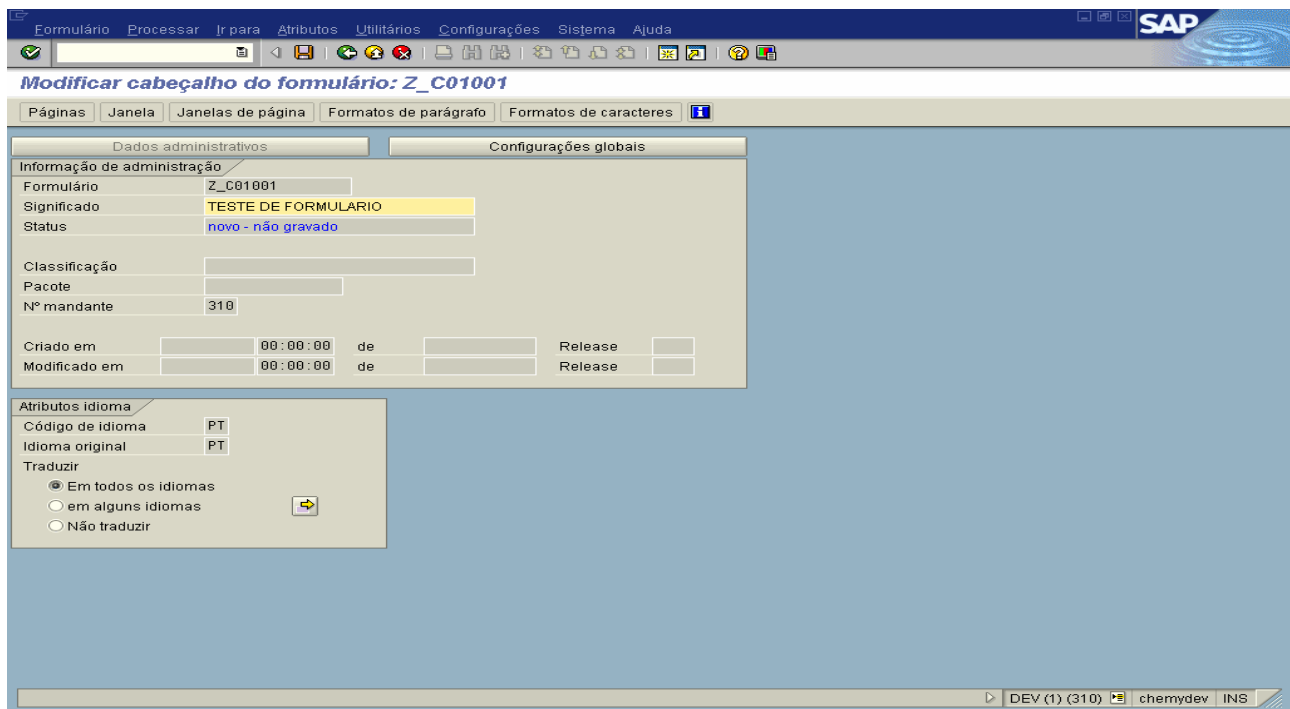
✓ ⚙️ ➕ 🏠 ✖

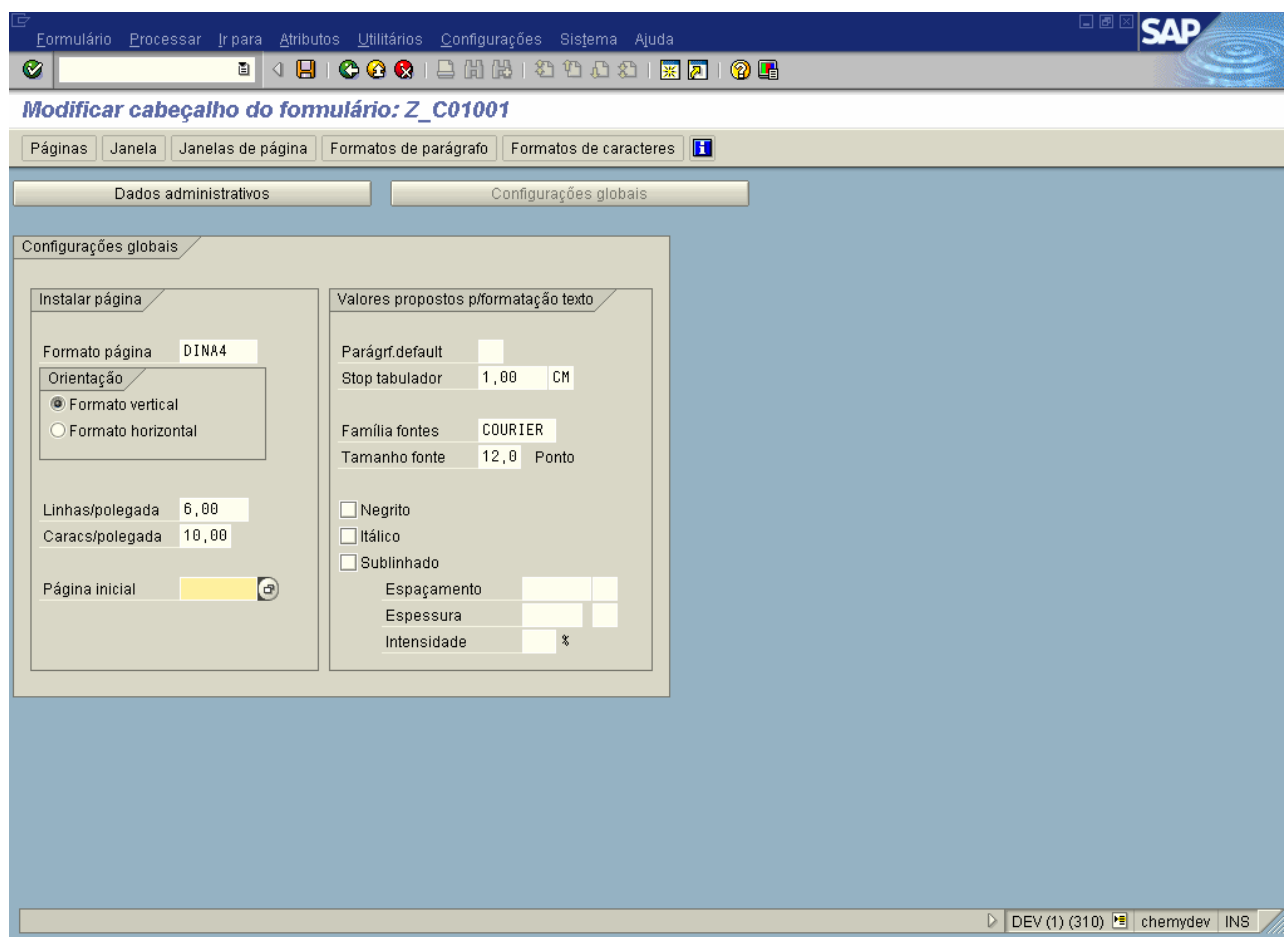


`&j_1bprnfli-menge(8.3)&` tamanho do campo

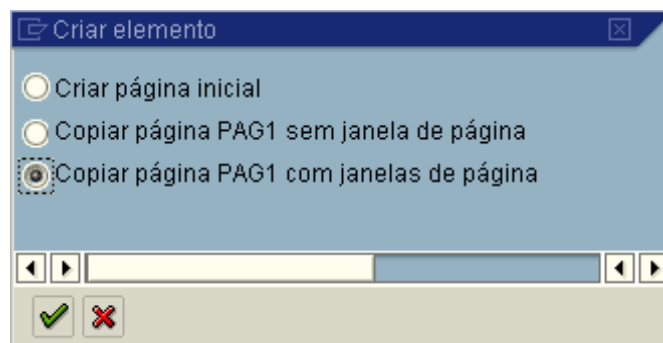
CRIAR UM FORMULÁRIO SAPSCRIPT

COLOCAR O SIGNIFICADO, DEPOIS CLICAR EM **CONF.GLOBAIS**

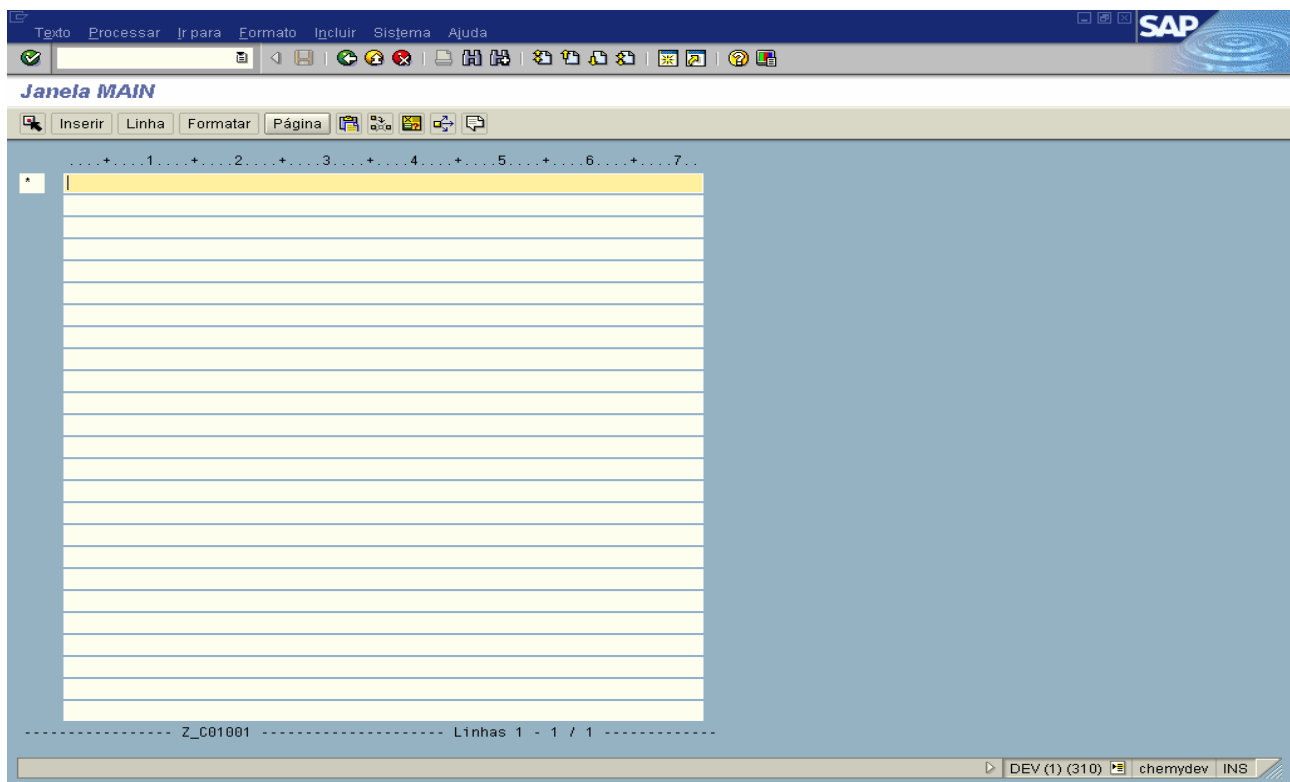




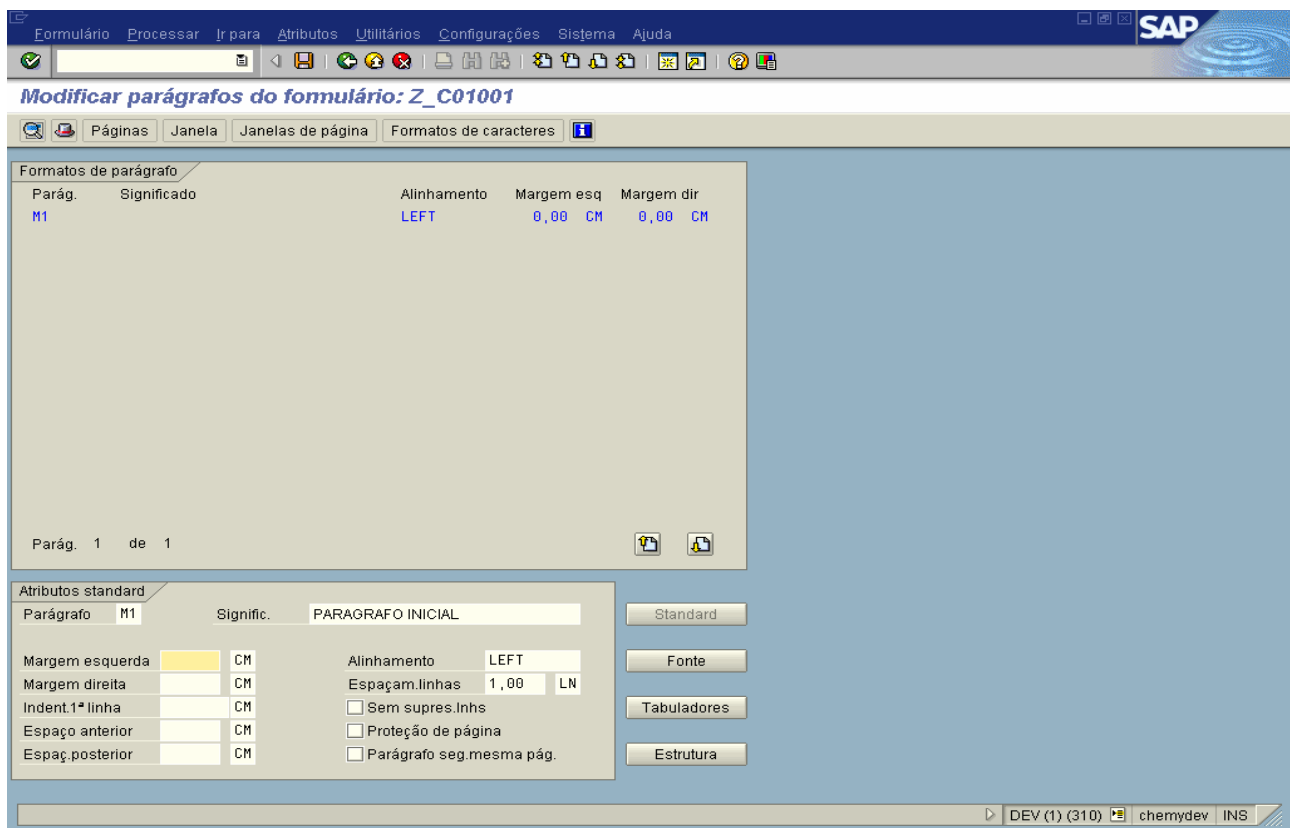
CRIAR EM PAGINAS



PRIMEIRO CRIAR A NEXT, DEPOIS A PRIMEIRA PAGINA



CRIAR FORMATOS DE PARAGRAFO E FORMATOS DE TEXTO



Formulário Processar Ir para Atributos Utilitários Configurações Sistema Ajuda

Modificar parágrafos do formulário: Z_C01001

Páginas Janela Janelas de página Formatos de caracteres

Formatos de parágrafo

Parág.	Significado	Família	Tam.	Negr	Itál.	Subl
M1	PARAGRAFO INICIAL	COURIER	10,0			

Parág. 1 de 1

Atributos de fonte

Parágrafo M1 Signific. PARAGRAFO INICIAL

Standard

Fonte

Tabuladores

Estrutura

DEV (1) (310) chemistrydev INS

Formulário Processar Ir para Atributos Utilitários Configurações Sistema Ajuda

Modificar parágrafos do formulário: Z_C01001

Páginas Janela Janelas de página Formatos de caracteres

Formatos de parágrafo

Parág.	Significado	Número tabuladores
M1	PARAGRAFO INICIAL	

Parág. 1 de 1

Tabuladores

Parágrafo M1 Signific. PARAGRAFO INICIAL

Standard

Fonte

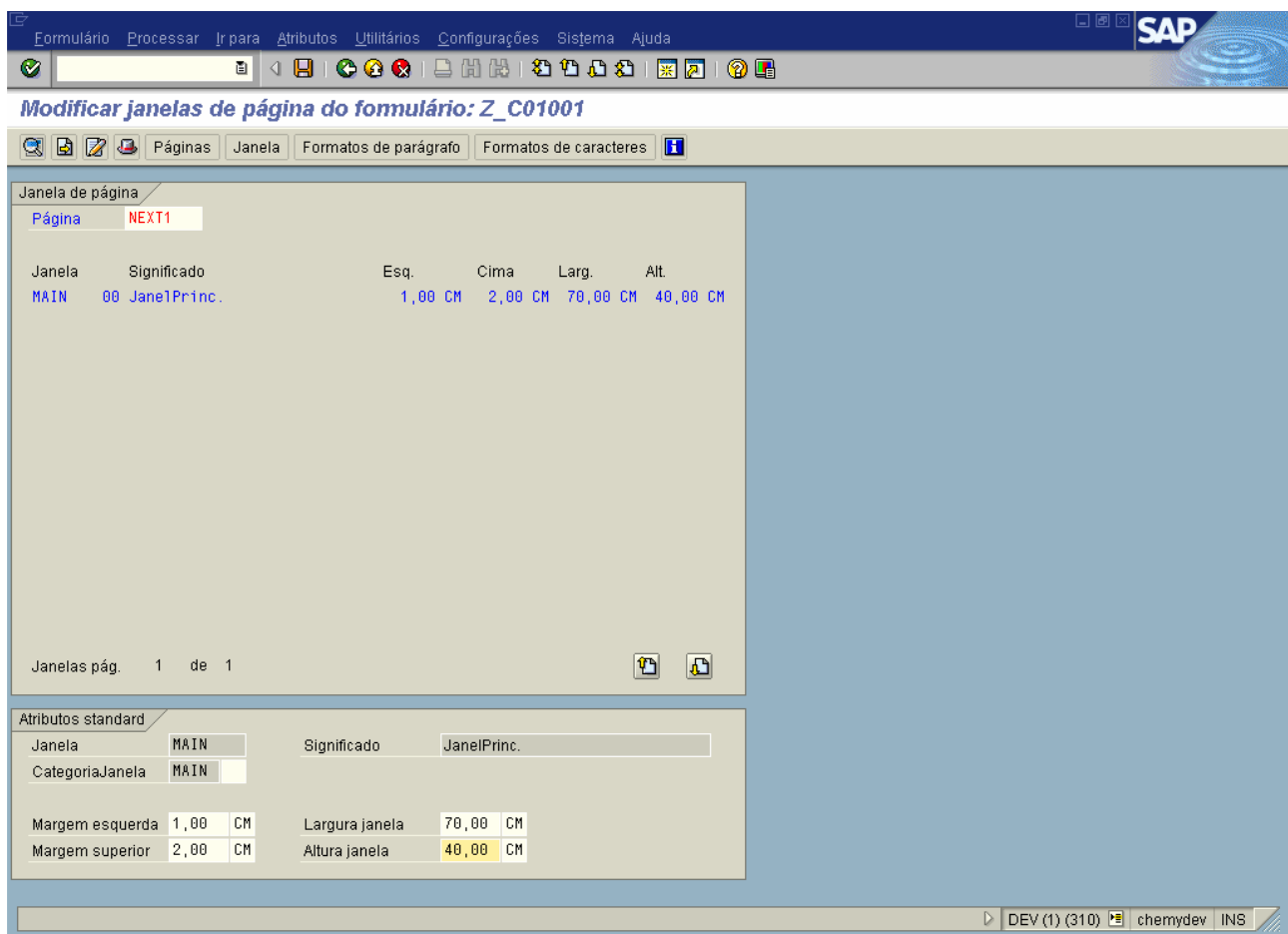
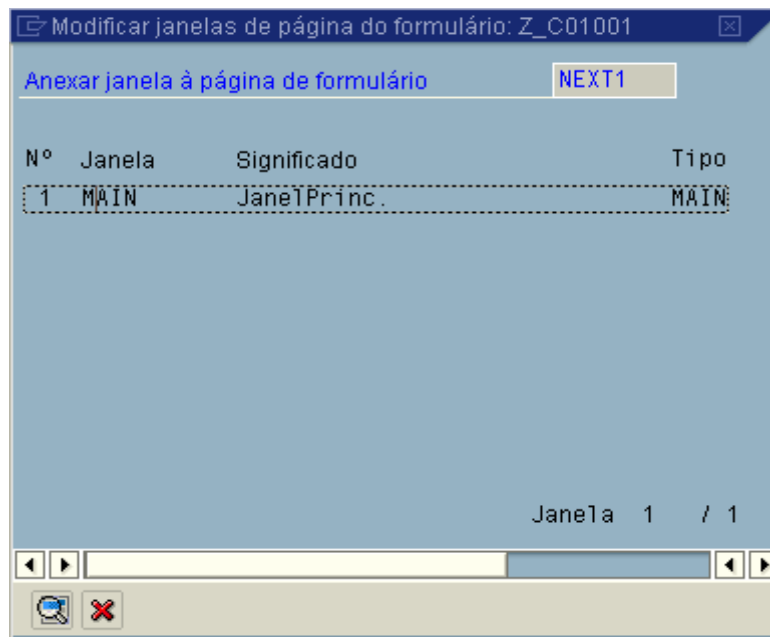
Tabuladores

Estrutura

Nº	Posição tabulador	Alinhamento
1	2 CM	LEFT
2	12 CM	LEFT
3		

DEV (1) (310) chemistrydev INS

CLICAR EM **JANELA DE PAGINA**



PARA AMARRAR AS JANELAS (FAZER LOOP)

EM **JANELAS**

PRIMEIRA JANELA (PAGINA)

Atributos standard		Significado	
Página	PAG1	PAGINA 1	
Página seguinte	NEXT1		
Atributos de impressão		Contador de páginas	
Nome de recurso		Modo	START
Modo impressão		Categoria numeração	ARABIC
		Comprimento de saída	
		<input type="checkbox"/> Maiúsculas	

SEGUNDA JANELA (OU PROXIMA PAGINA)

Atributos standard		Significado	
Página	NEXT1	PROXIMA PAGINA	
Página seguinte	PAG1		
Atributos de impressão		Contador de páginas	
Nome de recurso		Modo	INC
Modo impressão		Categoria numeração	ARABIC
		Comprimento de saída	
		<input type="checkbox"/> Maiúsculas	

PARA DESENHAR LINHAS (BOX)

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..	
/E	TITULO
/:	POSITION XORIGIN -2 MM YORIGIN -2 MM
/:	SIZE WIDTH +0 MM
/:	BOX FRAME 1 TW
T1	CODIGO,,DESCRIÇÃO DO CLIENTE,,ENDEREÇO
T1	

PARA DEIXAR EM DESTAQUE UM COLUNA

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..	
/E	CABE
M1	&KNA1-KUNNR&,,<B1>&KNA1-NAME1&</>,,&KNA1-STRAS&
*	

ONDE **B1** -> FORMATO DE CHARACTER (OU TEXTO)

Formatos de parágrafo

