

Programas on-line (Pool de Módulos)

O objetivo deste documento é disponibilizar um roteiro mínimo para a criação de um programa on-line em abap. Seguindo o roteiro determinado, você será capaz de construir um programa on-line para alteração de e-mail de um cliente da tabela SCUSTOM. Ao final do texto são apresentadas sugestões para próximos estudos em programas on-line e algumas melhorias no programa do roteiro.

Observação importante: o conhecimento da criação e compilação de programas abap é pré-requisito para entendimento do roteiro deste documento. É desejável o conhecimento das ferramentas de depuração de programas abap (debug) e dicionário de dados, em especial as transações SE11 e SE16.

Usamos programas do tipo M (Pool de Módulos) quando precisamos fornecer ao usuário SAP interfaces de tela mais elaboradas em relação aos programas do tipo Report. Os programas do tipo report, de uma maneira geral, fornecem 3 passos de processamento:

1 – Tela de seleção (um tela por programa): opções de seleção de informações para um relatório e/ou processamentos diversos (gravação em tabelas, consolidação de informações). Conseguimos montar mais de uma tela em um programa report, mas não é uma operação usual;

2 – Processamento: após a digitação de informações na tela de seleção, usamos o evento START-OF-SELECTION para codificar o tratamento destas informações;

3 - Saída: após o processamento, exibimos as informações lidas/calculadas (em tela e/ou impressora) ou consolidamos alguma informação em tabelas (“totalização das vendas do mês”, por exemplo).

Esta apresentação do programa report atende a grande maioria das necessidades de desenvolvimentos de programas abap, mas não todas.

Quando precisamos utilizar mais de uma tela para exibição/entrada de informações e navegação entre telas, o programa do tipo report mostra-se limitado. Para este tipo de processamento a SAP disponibiliza o programa do tipo M (Pool de Módulos)

Programas do tipo M são dotados de uma grande capacidade de customização das telas (leia-se “editor gráfico”) e flexibilidade para criar uma interface mais amigável com o usuário.

Na maioria dos casos cada tela de um programa on-line contém os seguintes elementos:

- Título;
- Status GUI - conjunto de botões e menus. Lembre-se que as telas SAP tem local fixo para os botões: barra de botões padrão e barra de botões de aplicação (vide página seguinte);
- Layout: campos para exibição e digitação;
- Lógica de processamento (também conhecido como eventos de tela):
 - comandos para execução ANTES da tela ser exibida: Process Before Output – PBO;
 - comandos para execução DEPOIS de acionado um botão ou ação do usuário: Process After Input – PAI. Se no PAI não for acionado outra tela do programa, o processamento retorna para o PBO;
 - Ajuda de pesquisa (acionamento da tecla F4): Process On Value Request – POV
 - Help (acionamento da tecla F1): Process On Help Request - POH

Observação: os eventos POV e POH não serão tratados neste documento.

O Título e o Status GUI são feitos de forma separada do desenho da tela. De forma geral, os passos para criação de um tela em abap são:

1 – Criamos a tela “vazia” (não necessariamente é o primeiro passo). Cada tela tem uma identificação numérica de 4 posições e sempre está associada a um programa. Podemos ter várias telas com o número 9100, mas em programas diferentes;

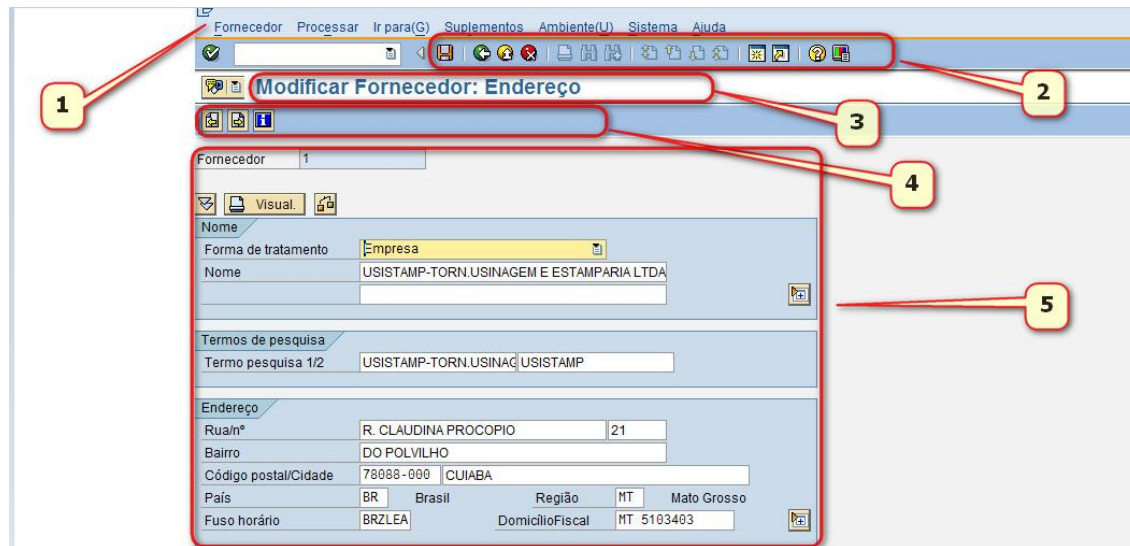
2 – Criamos o título e as ações (Status GUI) separados da tela. Estes elementos existem de forma independente da tela e no programa podem existir inúmeros textos e Status GUI. Podemos até mesmo copiar

estes elementos de outros programas existentes. Ratificando: o título e status GUI estão sempre associados a um programa, mas nunca a uma tela;

3 – Antes da tela ser exibida, informamos qual o título e o status GUI a ser usado, no PBO;

4 – Codificamos no PAI os resultados de cada ação do usuário, como por exemplo, um clique no botão salvar ou seleção de item no menu.

Veja na figura abaixo as áreas de tela que manipulamos em um programa on-line:



- 1 – Menu
- 2 – Barra de ferramentas padrão: em TODAS as telas SAP temos exatamente os botões mostrados acima, mas eles ficam habilitados ou não dependendo da funcionalidade a qual a tela se destina;
- 3 – Barra de título: a identificação da tela sempre é exibida nesta posição;
- 4 – Barra de ferramentas de aplicação: muda a cada tela. Nesta barra colocamos os botões que criamos. Dependendo da tela, esta área pode estar vazia, sem nenhum botão;
- 5 – Área de layout: exibição e digitação de dados do usuário. Podemos colocar botões na área de layout, mas precisamos tomar o cuidado para deixar a funcionalidade da tela coerente com a informação exibida. No exemplo acima temos dois botões na área de layout, mas a funcionalidade deles refere-se exclusivamente aos campos “Nome” e “Domicílio Fiscal”. Estes botões são criados no editor gráfico.

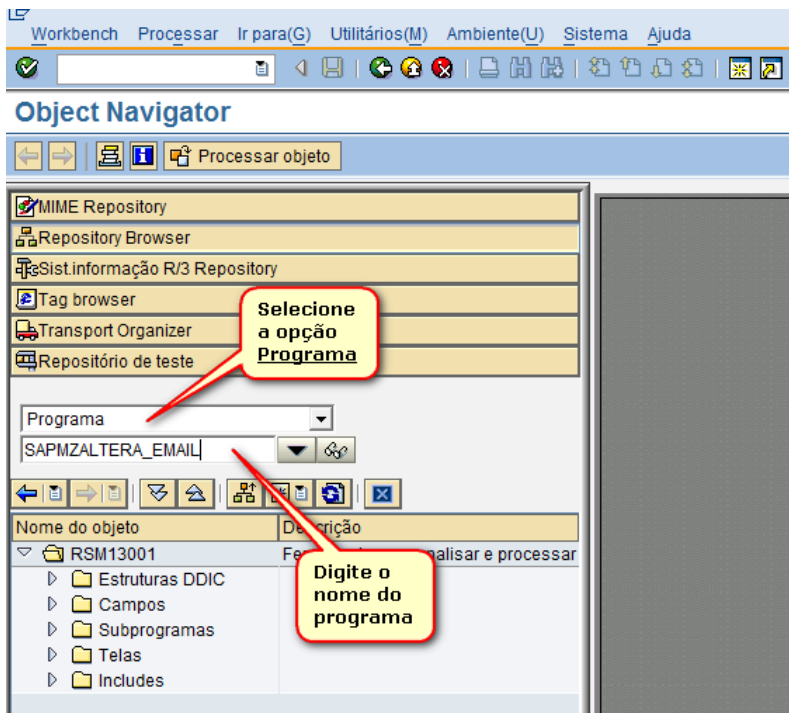
Concluindo:

- O menu e as barras de botões (áreas 1, 2 e 4) são criados no Status GUI;
- O título (conteúdo da área 3) é criado na opção “Títulos GUI”;
- O layout da tela (área 5) é criado no editor gráfico.

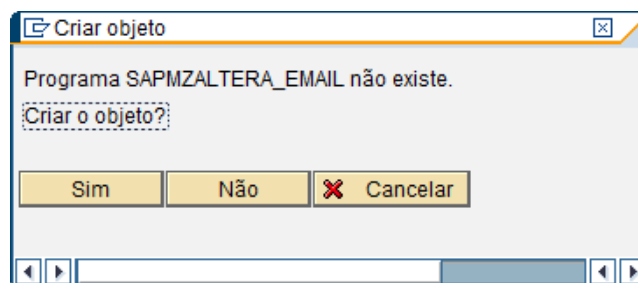
Criando um programa module-pool

Vejamos um programa on-line para alterar o e-mail de um cliente:

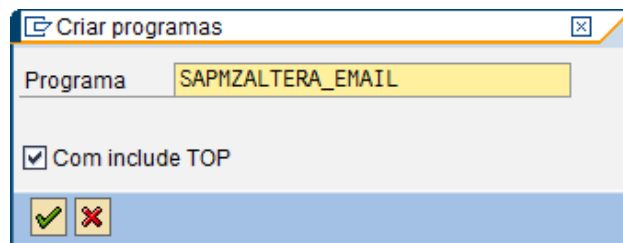
- 1 - Use a transação SE80 para criar o programa on-line (module pool):



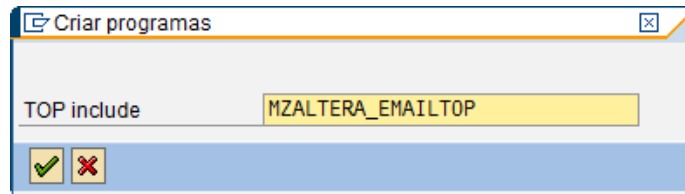
O nome de um programa on-line não-standard pode iniciar com a letra Z, mas a SAP recomenda usar o prefixo "SAPMZ" ou "SAPMY" (em oposição ao prefixo "SAPL", usado para programas on-line standard). Após informar o nome do programa tecele Enter. Surge a tela de confirmação:



Após confirmar aparece a tela:

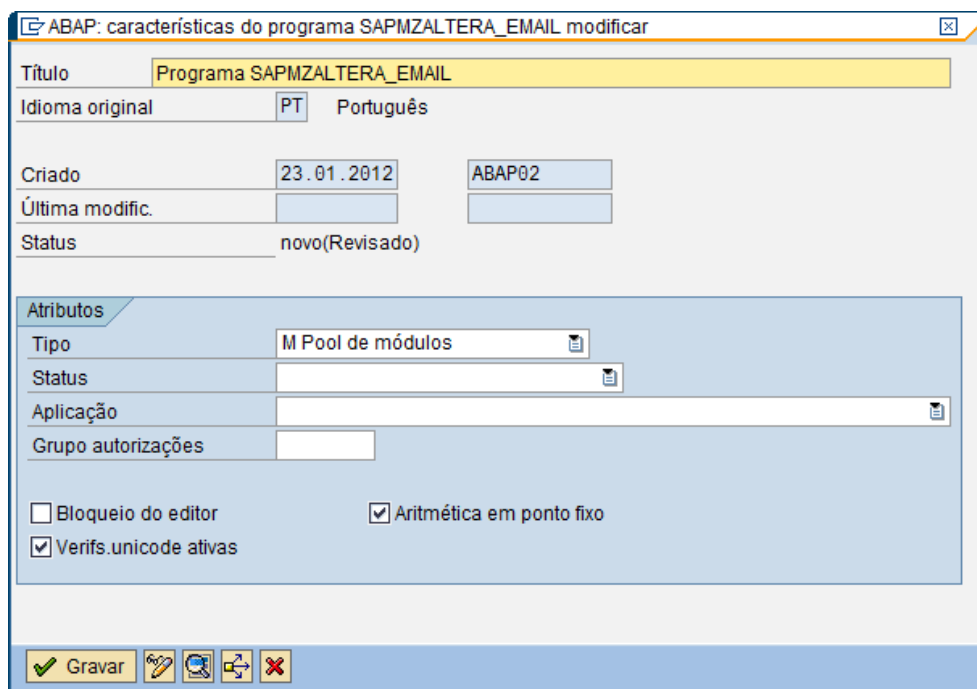


Deixe o checkbox "Com include TOP" marcado. Será criado automaticamente um novo include e ele será colocado no início do programa. Usamos este include para declaração de variáveis do programa module-pool. Após confirmar aparece a tela com a sugestão para o nome do include:



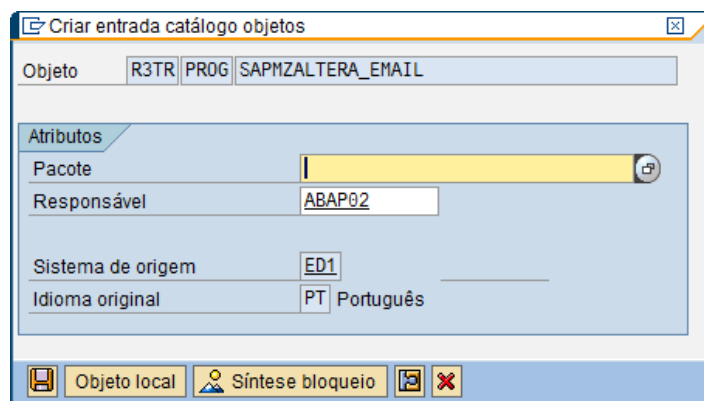
Observe a sugestão: MZALTERA_EMAILTOP. É um programa do tipo include e inicia pela letra M. Mais adiante criaremos outros includes, pois no programa on-line usamos 4 includes: declaração de variáveis (include TOP), rotinas para PBO, rotinas para PAI e rotinas gerais.

Confirme e veja a tela seguinte. Observe o tipo de programa sugerido pela SAP, Pool de módulos:

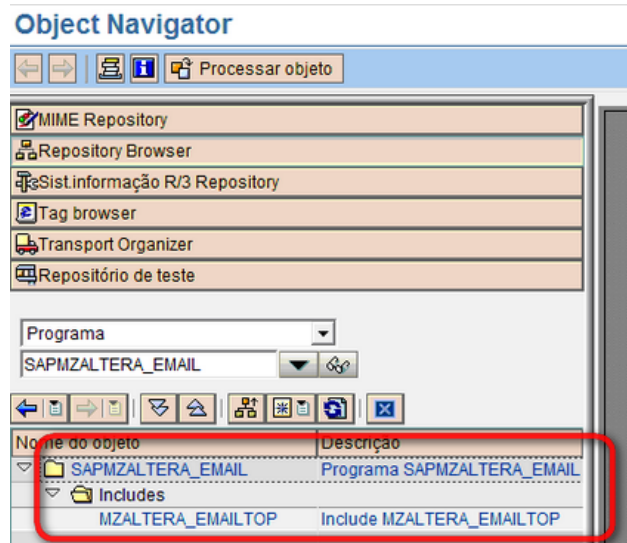


Observação: é possível criar telas em um programa tipo Report, mas não é usual. Uma exceção ocorre quando em um programa report precisamos de um elemento disponível apenas no editor gráfico (por exemplo, uma caixa de texto ou um relatório ALV orientado a objeto). Neste caso o processo é o mesmo, usamos a SE80 e criamos a tela e seus objetos.

Confirme a tela anterior e informe "Objeto local" na tela de requests (ou crie uma request):



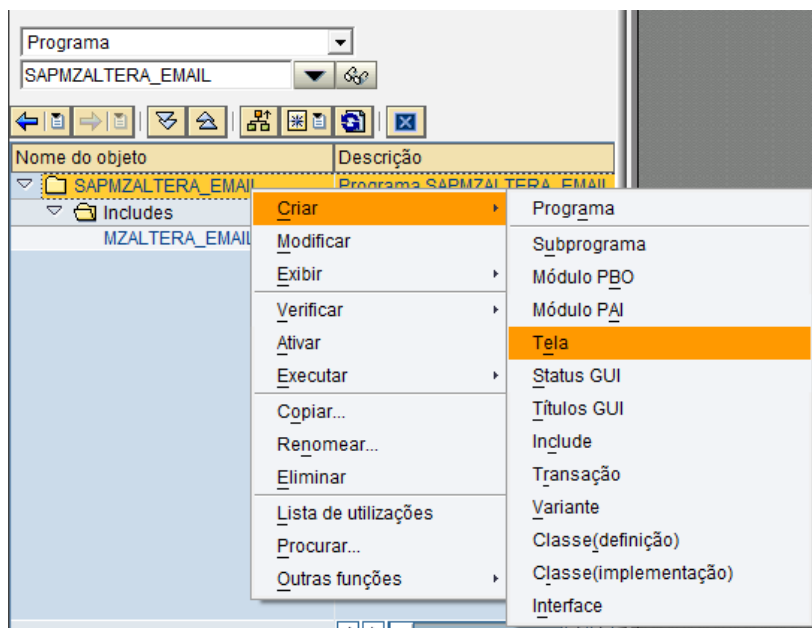
Após a criação, o lado esquerdo do painel mostra a árvore de objetos do programa on-line:



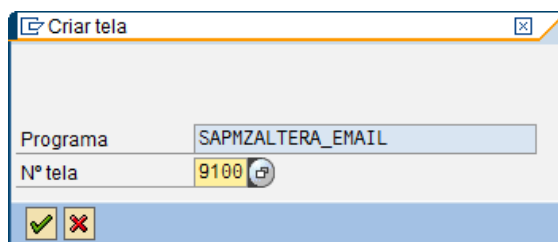
Observação: a cor azul no nome do programa e no nome do include indica que não foram ativados.

Criando uma tela

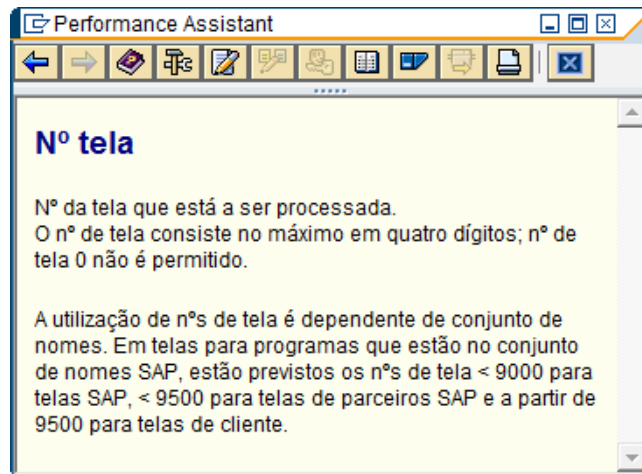
Posicione o cursor sobre o nome do programa e use o menu de contexto (botão direito do mouse) para criar uma tela:



Informe o número da tela:



A SAP recomenda que as telas de usuário iniciem com a numeração 9000, apesar de não haver restrições com outra numeração. Veja o que diz a ajuda do campo “N° tela” (aparece ao teclar F1):



Na aba “Características” informe a descrição breve para a tela (campo obrigatório). Esta descrição breve não aparece quando a tela é exibida, serve apenas como documentação.

N° tela: 9100 novo(Revisado)

Características | Lista elems. | LógProcmt.

Descrição breve: **Altera e-mail de cliente**

Idioma original: PT Português Pacote

Última modificação: 00:00:00

Última geração: 00:00:00

Tipo de tela

- ☒ Normal
- ☐ Subtela
- ☐ Caixa de diálogo modal
- ☐ Tela de seleção

Configurações

- ☐ Manter os dados
- ☐ Desativar compact.tmp.exec.
- ☐ Modelo - não executável
- ☐ Manter posição de rolagem
- ☐ S/barra ferrams.aplicação

Outros atributos

Tela seguinte: 9100

Posição do cursor:

Grupo de telas:

Linhas/colunas: Ocupado 0 0

Atual: 27 120

MenuContexto FORM ON CTMENU

Caracts.

Informe na aba “Lista elems.” o campo “OKCODE”. Esta variável armazenará o valor da ação do usuário na tela (clique de um botão, por exemplo). Veremos esta variável mais adiante neste documento:

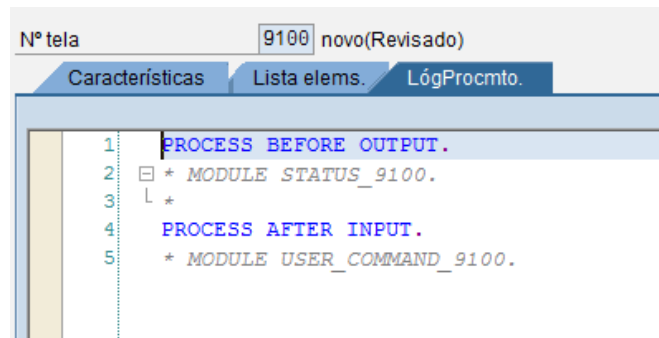
N° tela: 9100 novo(Revisado)

Características | **Lista elems.** | LógProcmt.

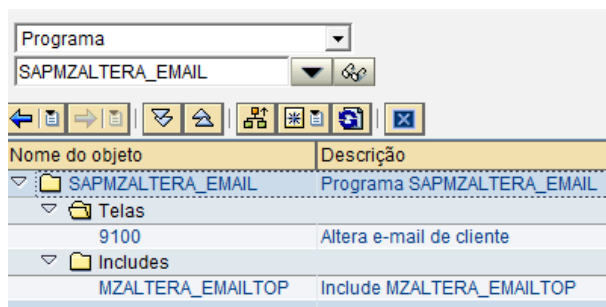
Atribs.gerais | Textos e padrões E/S | Atribs.espec. | Atribs.exibição

H	M	Nome	Tipo	Li	C	C	C	Alt	Po	Format	En
		OKCODE	OK	0	0	20	20	1		OK	

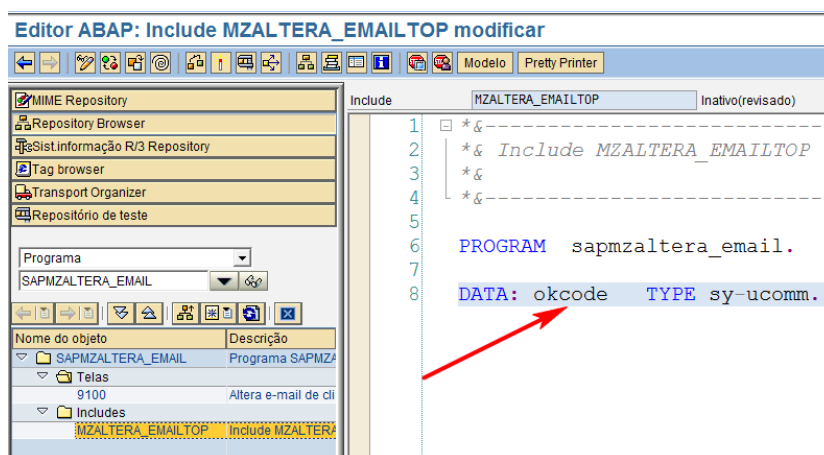
Clique agora na aba “LógProcmt”. Observe que o SAP cria o modelo da lógica de processamento (inicialmente com os MODULEs comentados). O comando MODULE para a lógica de processamento é como um PERFORM para um programa abap. Refere-se a chamada de uma “rotina” de tela. Os comandos que colocamos na lógica de processamento são específicos para telas: não é permitido colocar, por exemplo, PERFORM, CLEAR ou outros comandos de programas abap.



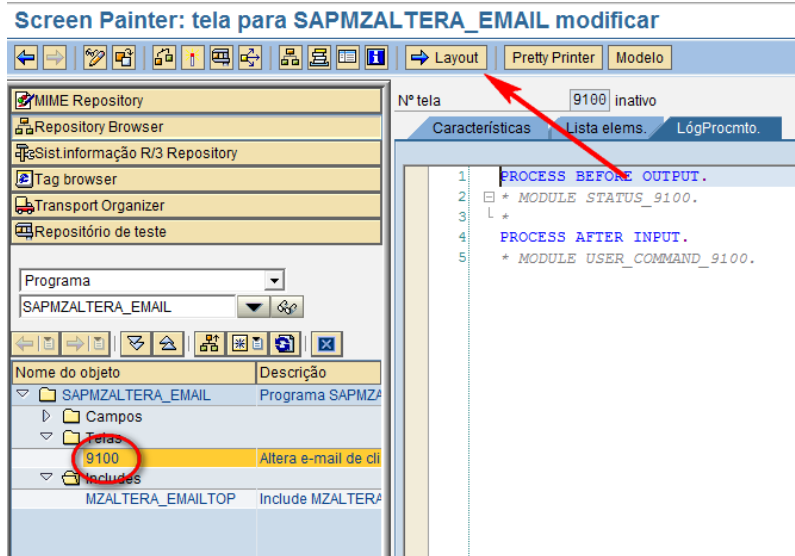
À medida que criamos elementos do programa, a árvore de objetos é atualizada. Salve o programa e observe a árvore de objetos até agora:



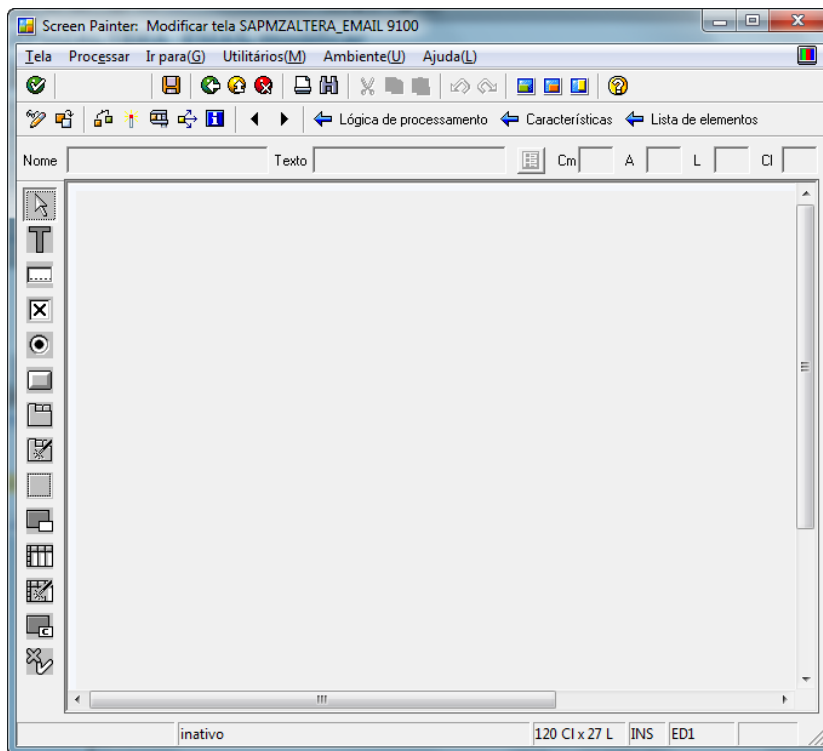
Crie agora a variável OKCODE. Selecione o include TOP criado (clique-duplo sobre o nome MZALTERA_EMAILTOP) e declare a variável OKCODE conforme abaixo:





Selecione a tela 9100 no painel esquerdo e clique no botão “Layout”:



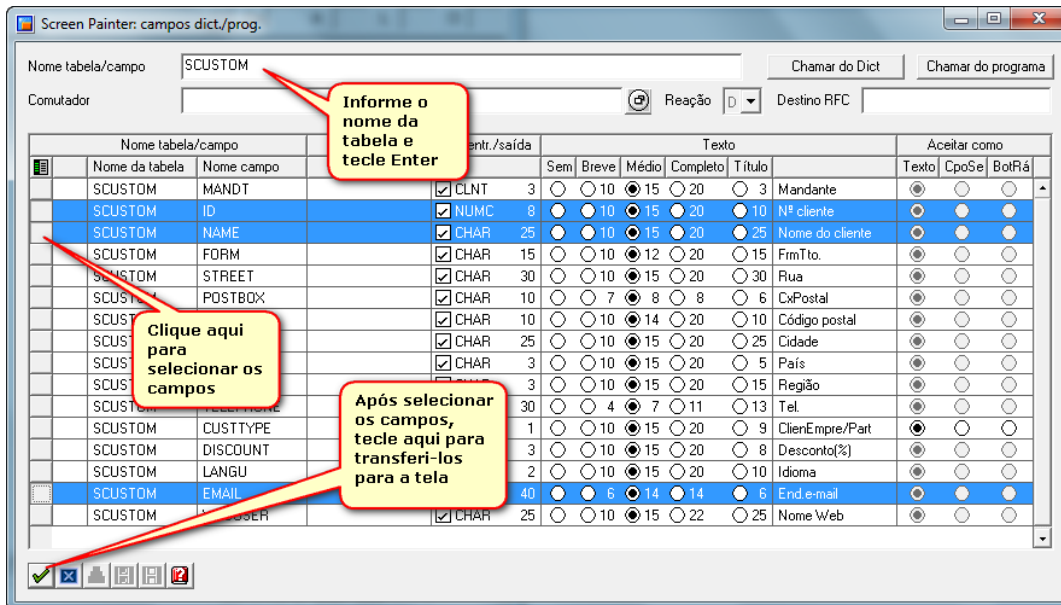
Aparece o editor gráfico:



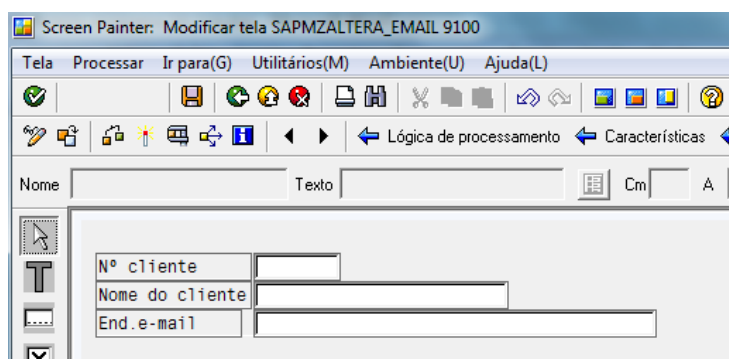
Há duas formas de colocar elementos nesta tela. A primeira consiste em selecionar o elemento desejado (por exemplo, um texto no símbolo  ou um campo de entrada de dados ) e arrastá-lo para a área de layout.

Na outra forma partimos de uma tabela ou estrutura do dicionário de dados, copiando os campos desejados e modificando os atributos. Esta forma tem a vantagem de trazer o tipo e tamanho corretos do campo para a tela.

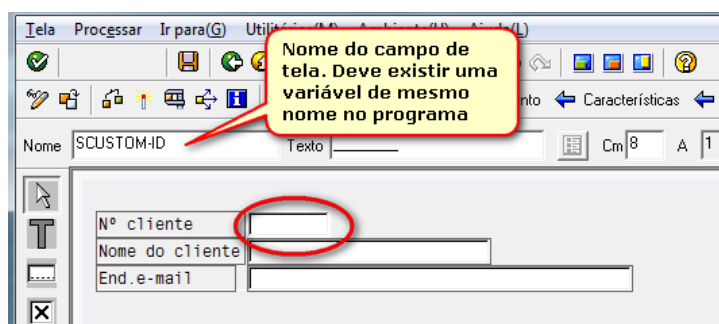
Acione a tecla F6 para exibir a tela de seleção de campos. Informe a tabela SCUSTOM e tecle Enter. Em seguida selecione os campos abaixo (clicando na coluna à esquerda de cada campo):



Coloque os campos selecionados à esquerda da tela:

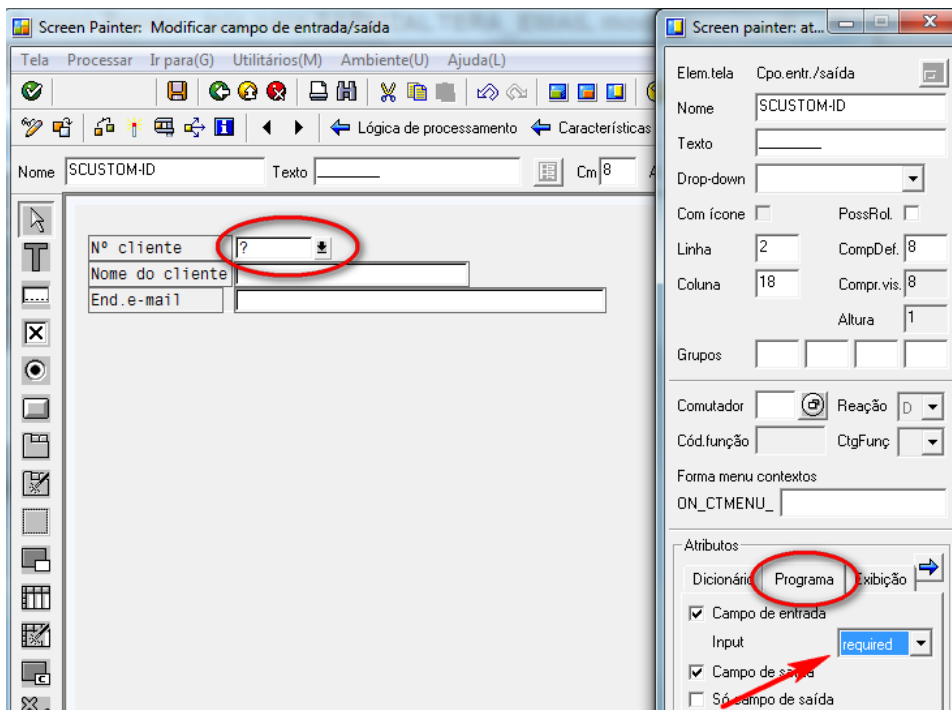


Clique no campo de digitação do número do cliente e veja o nome do campo. Como foi feita uma cópia a partir do dicionário de dados, o SAP sugere o nome SCUSTOM-ID para o campo. Lembre que para cada nome de campo alterável da tela deve ter uma variável de mesmo nome declarada no programa. Você pode usar o nome sugerido pela SAP ou alterá-lo. A falta da declaração da variável é um erro muito comum em iniciantes, pois a compilação/ativação não detecta erro neste caso (realmente não é um erro, as variáveis de tela e de programa existem de forma independente).

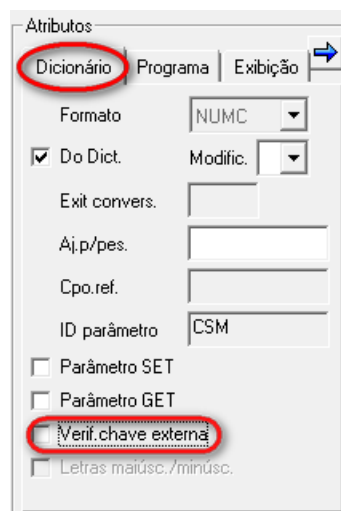


Este programa tem a intenção de alterar o e-mail de um cliente. Portanto, o campo de nome do cliente não pode estar habilitado para alteração, somente para exibição. Além disso, o número do cliente é digitação obrigatória.

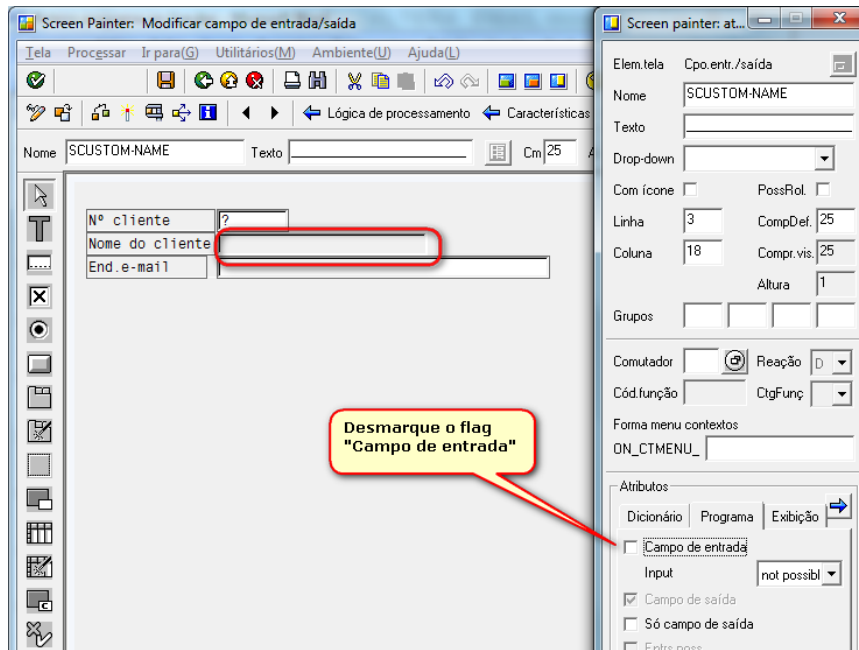
Primeiro marque o número do cliente como campo obrigatório. Selecione o campo de número do cliente e tecle F2 ou use o clique-duplo sobre ele. Selecione "required" conforme mostrado:



Na aba dicionário, desmarque o flag “Verificação de chave externa”:



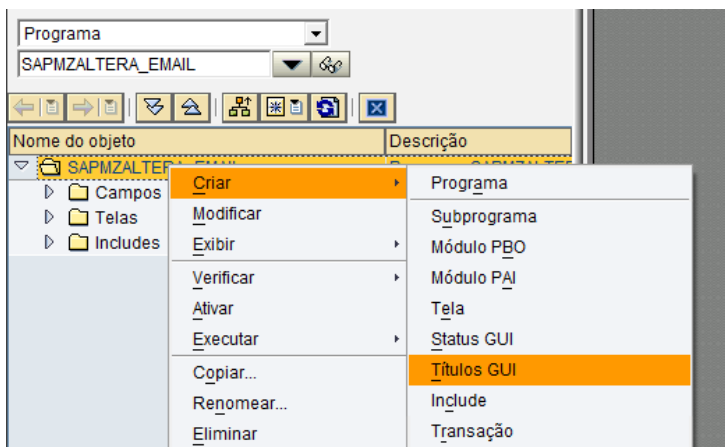
Deixe o campo nome do cliente apenas como exibição, desmarcando o flag “Campo de entrada” (figura abaixo). Toda informação variável que você precisar exibir deve usar o elemento de tela “campo de entrada e saída” (). O campo de texto () sempre representa um valor fixo e não pode ser alterado:



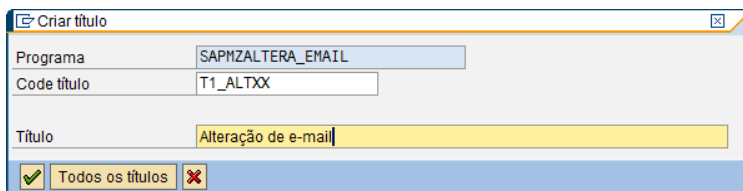
Volte para a tela anterior e clique no botão salvar.

Criando um título para a tela

Use o menu de contexto (botão direito do mouse) e selecione “Títulos GUI”:

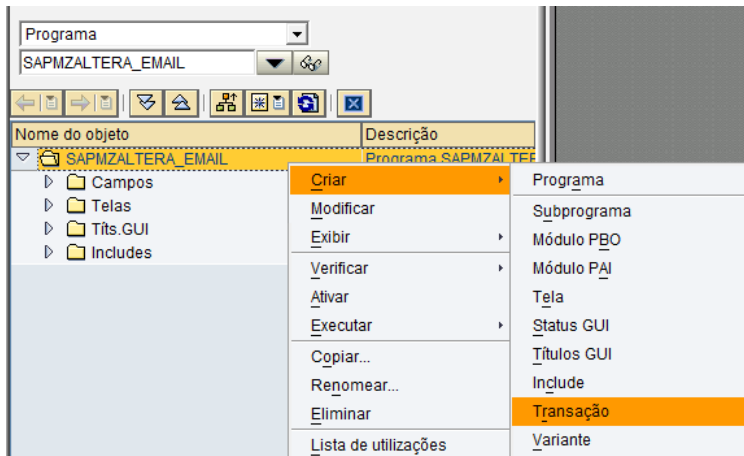


Informe o “Code título” e o “Título”:

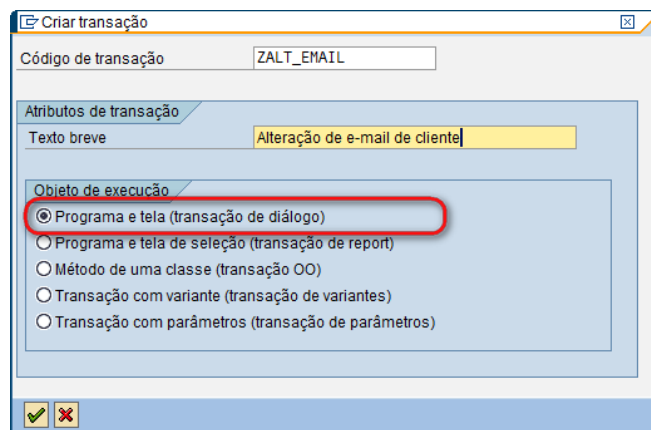


Criando uma transação para o programa

Um programa on-line só pode ser executado se tiver uma transação associada a ele. Novamente use o botão direito do mouse:

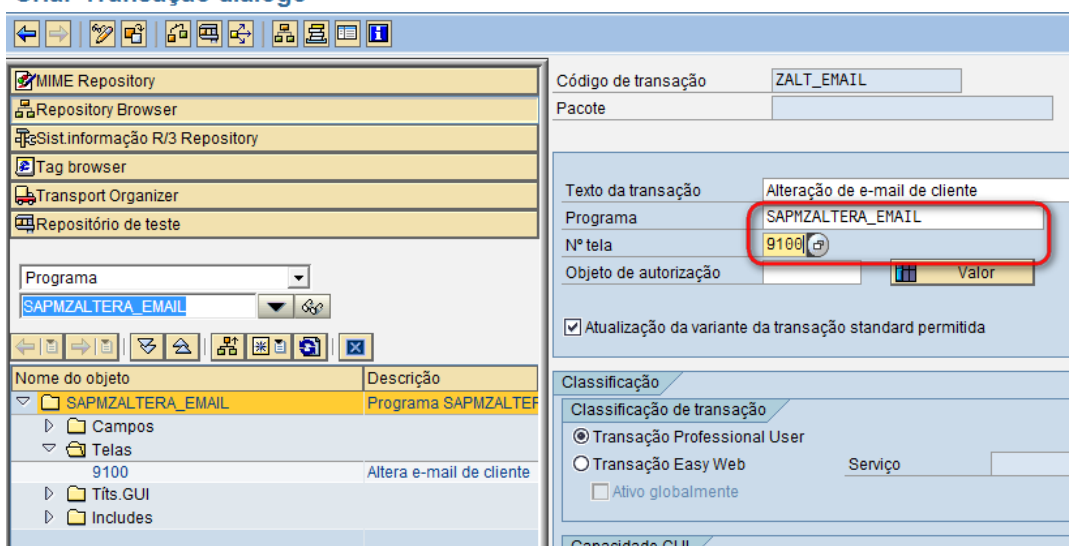


Informe o código da transação e o texto breve. O texto breve serve como documentação, ele não será exibido na tela:

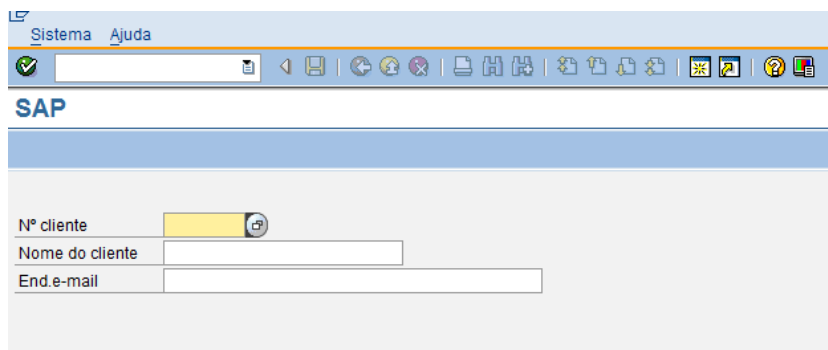


O código da transação DEVE iniciar pela letra Z ou Y. Informe o nome do programa e a tela inicial:

Criar Transação diálogo




Já criamos a tela, o título e a transação. Podemos até executar a transação ZALT_EMAIL para visualizar a tela, mas não teremos nenhum processamento, pois falta o Status GUI. Veja a chamada da transação ZALT_EMAIL até agora. Observe que não há botões disponíveis e também não há título:




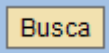
Criando o Status GUI

O Status GUI representa as ações possíveis para a tela. A cada ação associamos um “código de função”. É com o código de função que identificamos dentro do programa qual botão foi pressionado. Podemos ter vários Status GUI em um programa on-line, cada um com um conjunto de ações.

Neste exemplo colocaremos 3 botões:

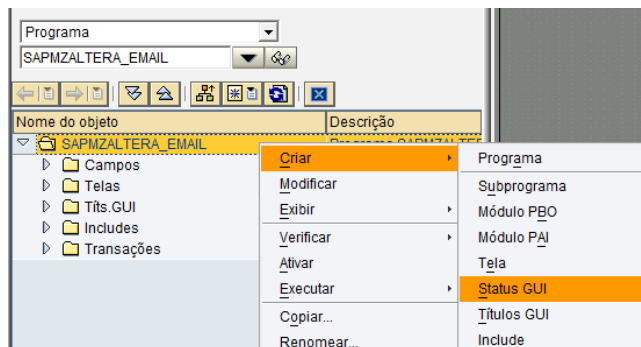
Voltar (): encerra o programa. Código de função: VOLTAR

Salvar (): grava o e-mail informado. Código de função: GRAVAR

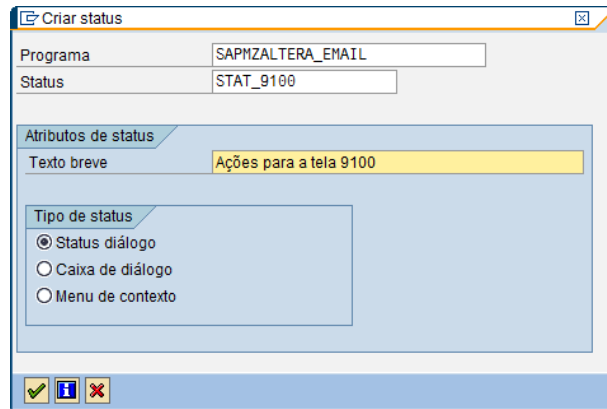
Buscar (): a partir do código do cliente informado na tela, busca na tabela de cliente (SCUSTOM) os dados do cliente. Código de função: PESQUISAR


Os dois primeiros botões estão na barra de botões padrão; o botão buscar será colocado na barra de botões de aplicação.

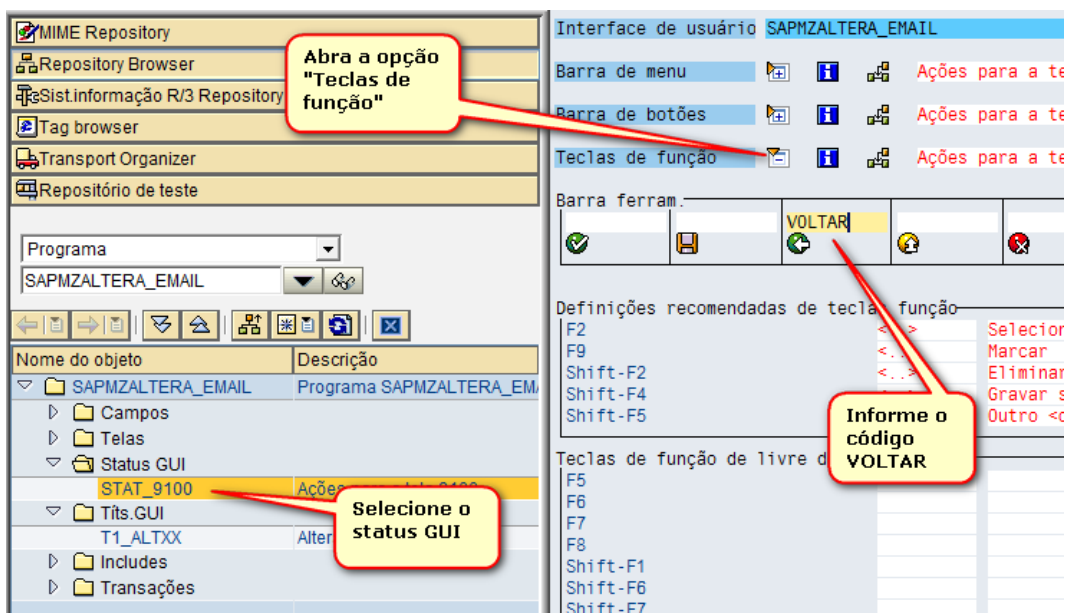
Use o menu de contexto (clique com o botão direito do mouse) para criar o status GUI:



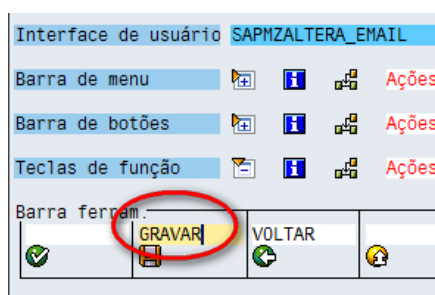
Apesar de nomear o status GUI como STAT_9100, ele poderia ser usado em qualquer tela.



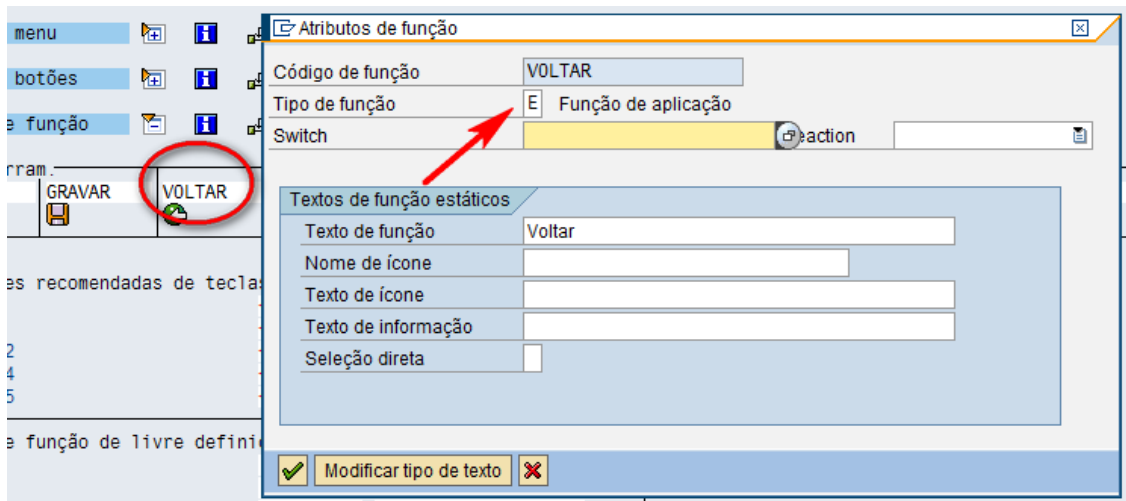
A barra de ferramentas padrão contém os botões mais usados, mas para fazê-los disponíveis para o usuário precisamos definir o “Código de função”. Cada “Código de função” deve ter um conjunto de teclas de atalho associado (CTRL+F5, por exemplo). O nome do código de função é usado na lógica de processamento da tela para direcionarmos o processamento desejado (gravar, imprimir, excluir e assim por diante). Selecione (clique-duplo no nome do status GUI) e abra a opção “Teclas de função”, conforme mostrado. Informe o código de função VOLTAR para o botão : Lembre de gravar a sua alteração:




Coloque agora o código de função GRAVAR, como mostrado abaixo:

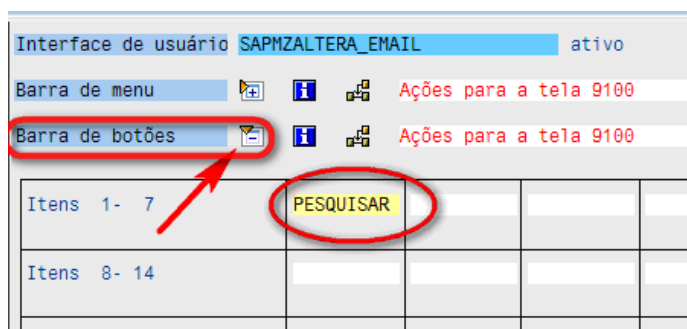


Dê um clique duplo no código de função VOLTAR e altere o tipo de função para E. Esta operação indica que podemos ter um MODULE exclusivo para este botão (será explicado mais adiante):

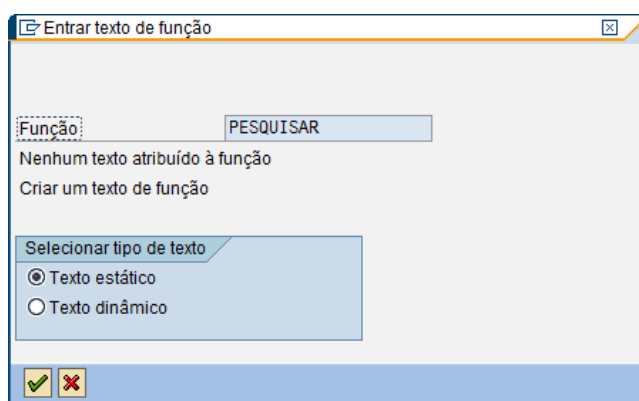


Observação importante: não crie botões desnecessários na sua tela. Se a sua tela tem a funcionalidade de “gravação”, use o botão  da barra de ferramentas padrão.

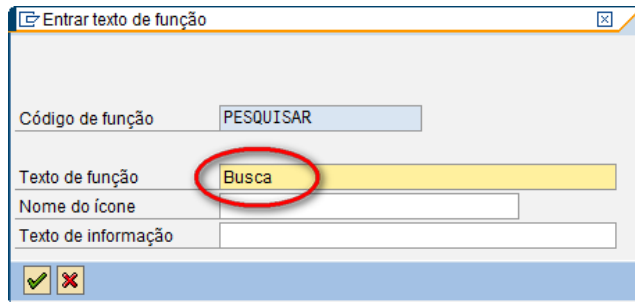
Coloque agora o botão “Pesquisar” na barra de botões de aplicação. Informe o código de função e tecle Enter:



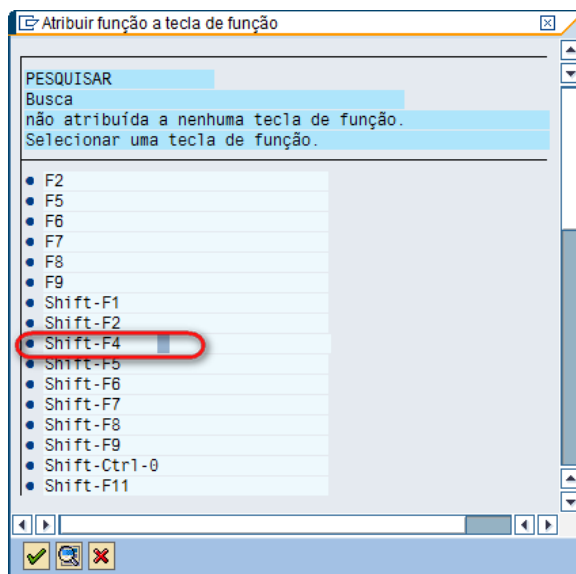
Aparece a tela solicitando o texto. Tecle Enter:



Informe o texto para aparecer no botão e tecle Enter. Nesta tela podemos informar também um ícone para o botão. Atenção! O código de função deste botão é “PESQUISAR”, mas o nome que aparecerá na tela é “Busca”:



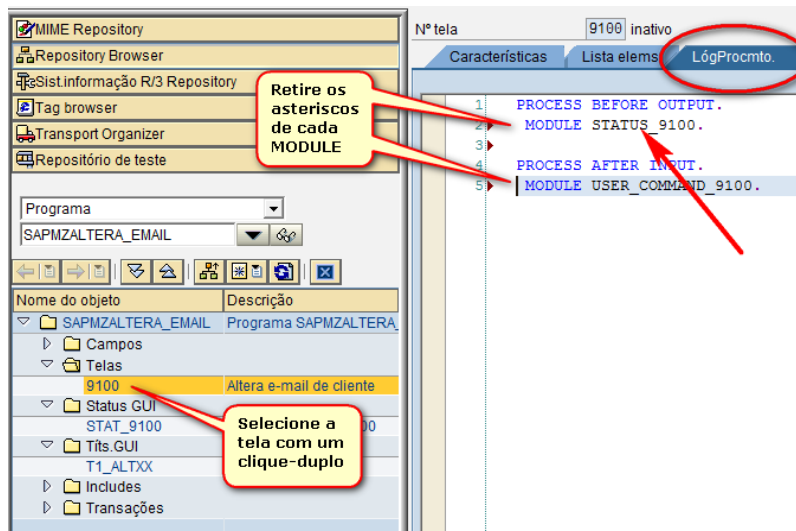
Toda ação de tela deve ter um conjunto de teclas de atalho associado. A tela seguinte pede as teclas de atalho. Selecione alguma (por exemplo, Shift-F4) e tecle Enter:



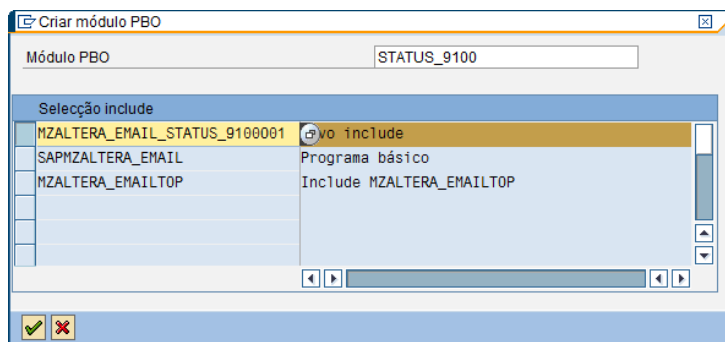
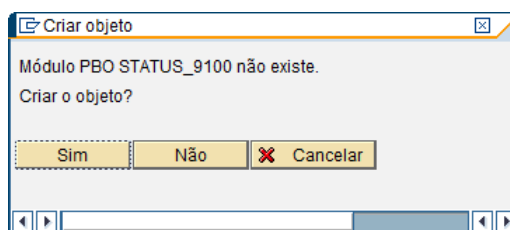
Observe que as ações da barra de ferramentas padrão tem teclas de atalho pré-definidas: F3 para voltar (🏠), Ctrl+S para salvar (💾) e outros.

Criando a lógica de processamento – PBO

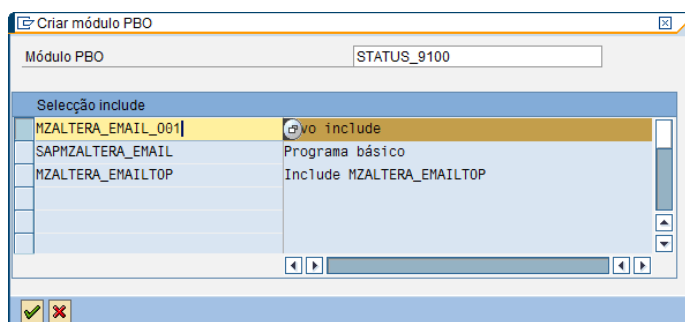
Após criarmos o título e status GUI (botões) do nosso programa, precisamos associá-los à tela que foi criada. Selecione a lógica de processamento da tela e retire o comentário dos MODULE. Lembre sempre de salvar o seu trabalho:



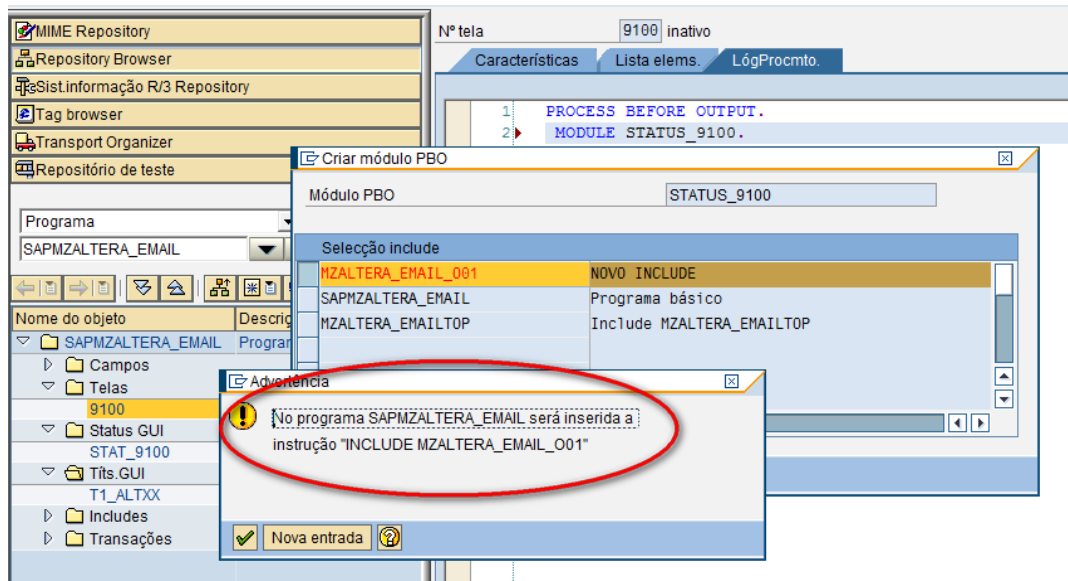
Com um clique duplo (ou F2) sobre o nome do module STATUS_9100 criamos o module e o include (se não tiver sido criado). Observe que este é um nome sugerido para o module, mas podemos trocá-lo. Se você não salvou sua alteração anterior, o SAP pedirá para salvar:



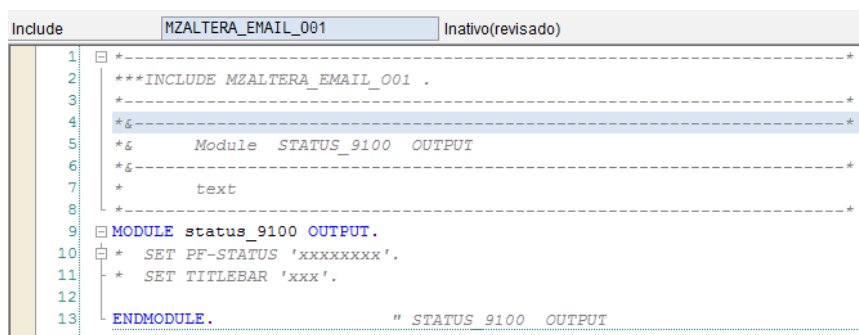
O SAP sugere o nome MZALTERA_EMAIL_STATUS_9100001 para o include, mas não precisamos seguir este nome. Observe que o nome sugerido contém o nome do module (STATUS_9100) e um sufixo 001 ("Ó zero um"). A letra O aqui refere-se ao evento PBO. Vamos resumir este nome para MZALTERA_EMAIL_O01, para não parecer que teríamos somente o module STATUS_9100 neste include. Ficamos assim:



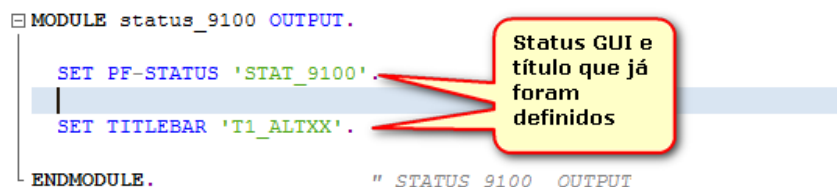
Após a confirmação vemos o lembrete da colocação do include no programa principal. Tecle Enter nesta tela:



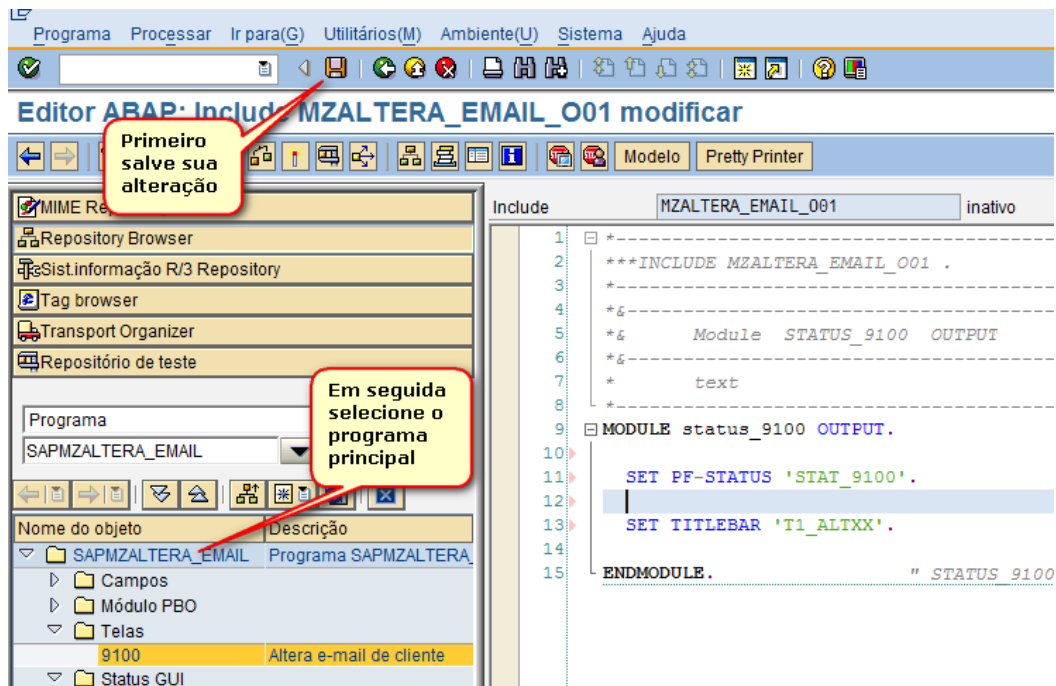
Lembre-se que o processamento do module STATUS_9100 será feito ANTES da tela ser exibida (Process Before Output). O próximo passo é a codificação do module STATUS_9100. Observe que foi sugerido dois comandos para este module: um para indicar qual status GUI será usado e outro para indicar o título:



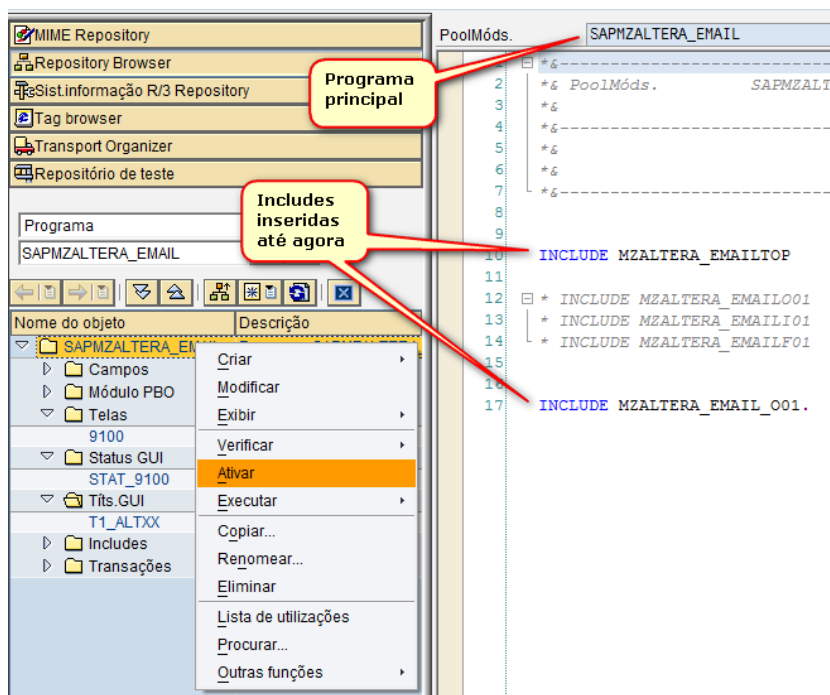
Retire os comentários do module, informando o status GUI e título nos comandos, sempre em letras maiúsculas. O código final deste module deve ficar assim:



Está em um bom momento para salvar o seu trabalho e ativá-lo. Salve esta alteração e selecione o programa principal em seguida:



Ative agora todo o programa, com o botão direito do mouse:



A tela de ativação mostra os elementos não ativados. Pode ser que sua tela esteja diferente, caso tenha ativado algum item separadamente. Selecionando o programa principal primeiro garante a ativação de todos os elementos inativos:

Objetos inativos de ABAP02

Objetos transportáveis **Objetos locais**

Nome do objeto		
O	Objeto	Nome obj.
	REPS	MZALTERA_EMAILTOP
	REPS	MZALTERA_EMAIL_O01
	REPS	SAPMZALTERA_EMAIL
	CUAD	SAPMZALTERA_EMAIL
	DYNP	SAPMZALTERA_EMAIL
		9100

Objetos:
 REPS: fontes abap
 CUAD: status GUI
 DYNP: tela

Veja como ficou a árvore de objetos do programa até agora:

Programa

SAPMZALTERA_EMAIL

Nome do objeto	Descrição
▼ SAPMZALTERA_EMAIL	Programa SAPMZALTERA_EMAIL
▼ Campos	
OKCODE	
▼ Módulo PBO	
STATUS_9100	
▼ Telas	
9100	Altera e-mail de cliente
▼ Status GUI	
STAT_9100	Ações para a tela 9100
▼ Títs.GUI	
T1_ALTX	Alteração de e-mail
▼ Includes	
MZALTERA_EMAILTOP	Include MZALTERA_EMAILTOP
MZALTERA_EMAIL_O01	Include MZALTERA_EMAIL_O01
▼ Transações	
ZALT_EMAIL	Alteração de e-mail de cliente

Veja a tela após acionar a transação ZALT_EMAIL:

Sistema Ajuda

Alteração de e-mail

Busca

Nº cliente

Nome do cliente

End.e-mail

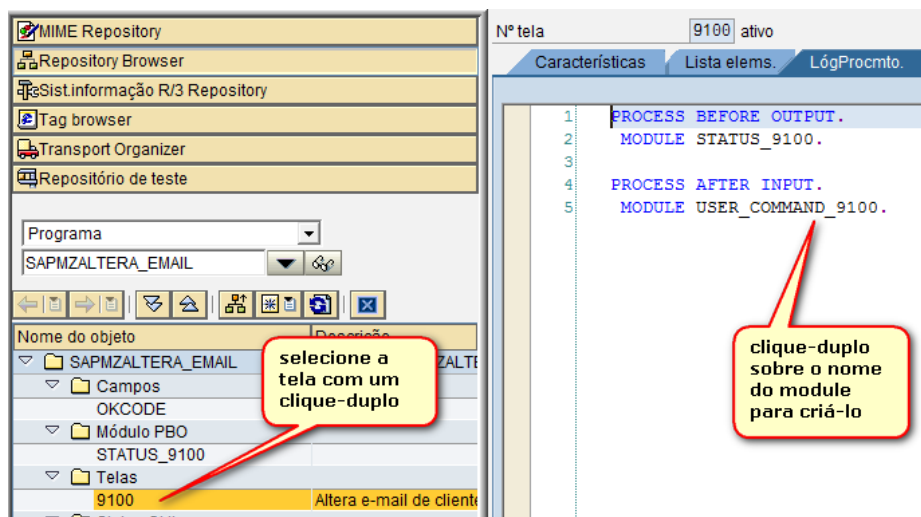
Observe:

- Os botões VOLTAR (↶), GRAVAR (💾) e PESQUISAR (🔍) estão ativos devido ao comando SET PF-STATUS;
- O título “Alteração de e-mail” apareceu devido ao comando SET TITLEBAR;
- O N° cliente é campo obrigatório (☒);
- O Nome do cliente é campo somente para exibição.

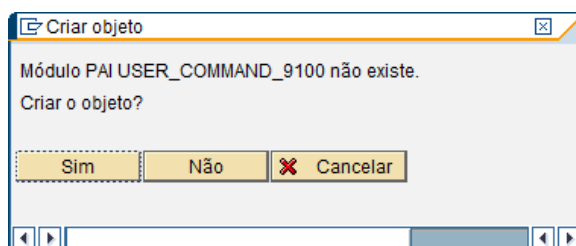
Criando a lógica de processamento – PAI

Até agora finalizamos os passos necessários para a exibição da tela. Desenhamos o layout, títulos e botões da tela, mas não codificamos nenhuma ação do usuário. Vamos agora colocar algum código para responder às ações do usuário (Process After Input).

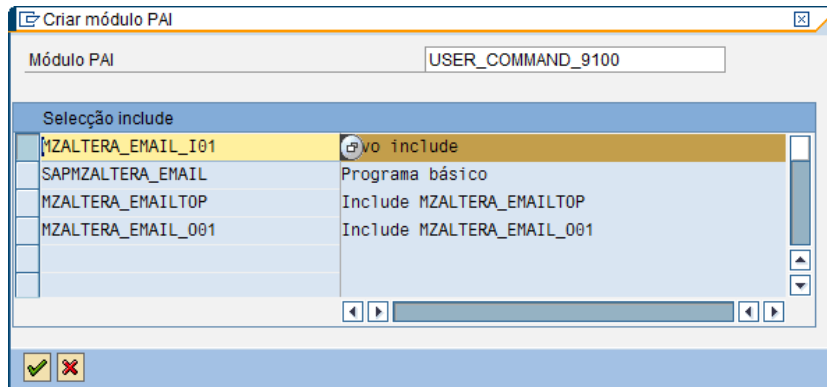
De modo semelhante ao PBO, selecione a tela e dê um clique duplo sobre o nome do module USER_COMMAND_9100 (nome sugerido pelo SAP: “comandos de usuário para a tela 9100”):



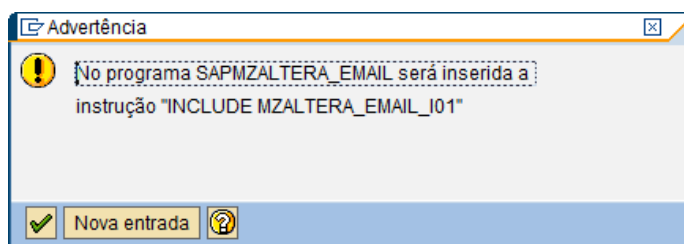
Confirme a criação:



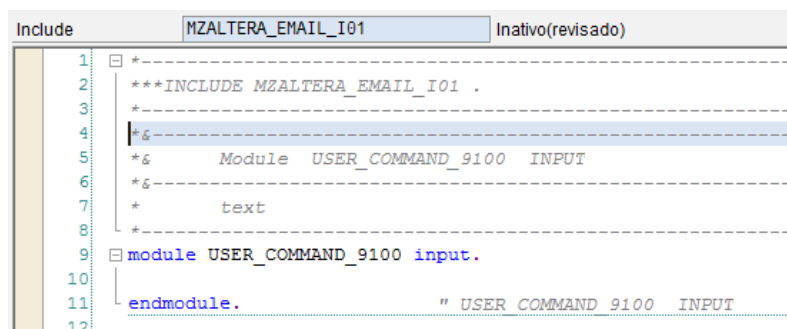
Altere o nome do include para MZALTERA_EMAIL_I01. No sufixo “I zero um”, a letra I refere-se ao PAI:



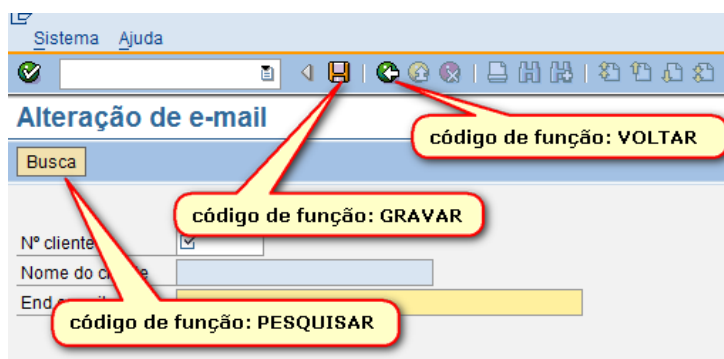
Confirme a mensagem seguinte com um Enter:



A tela de edição do module aparece:



Toda vez que o usuário teclar Enter, clicar um botão ou acionar um item do menu, os comandos dentro do module USER_COMMAND_9100 serão executados. Estas ações possíveis do usuário são os códigos de função definidos no status GUI. Lembrando os códigos de função já definidos:



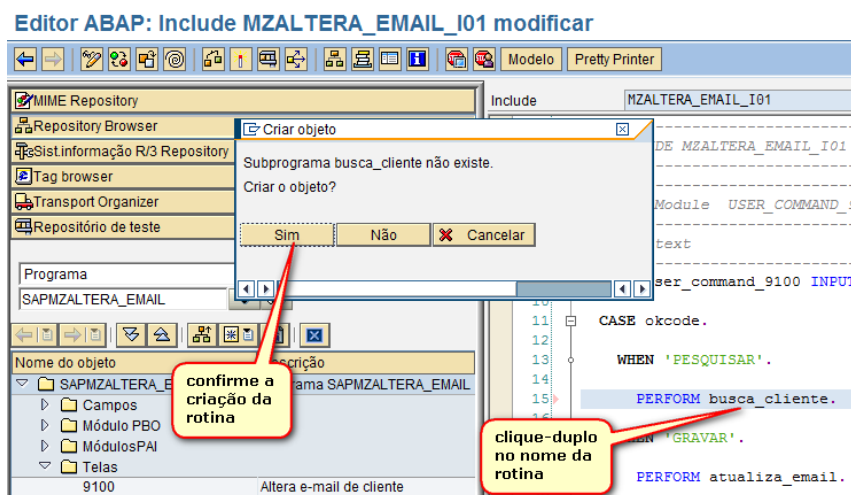
Ao acionar um dos botões acima, a variável OKCODE (vide página 7 para lembrar sobre esta variável) será preenchida com algum dos valores acima (PESQUISAR, GRAVAR ou VOLTAR). Portanto, no module USER_COMMAND_9100 questionamos o valor desta variável e executamos a ação desejada pelo usuário. Codifique o module conforme mostrado abaixo e salve seu trabalho. Cuidado ao escrever os literais, coloque-os em letras maiúsculas:

```

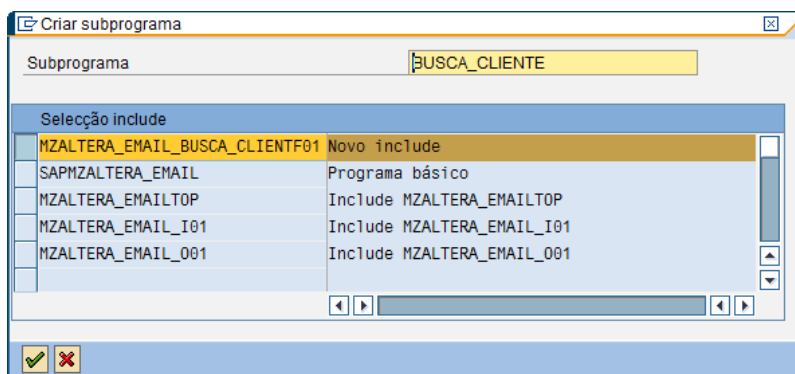
4  *-----
5  *      Module  USER_COMMAND_9100  INPUT
6  *-----
7  *      text
8  *-----
9  MODULE user_command_9100 INPUT.
10
11  CASE okcode.
12
13      WHEN 'PESQUISAR'.
14
15          PERFORM busca_cliente.
16
17      WHEN 'GRAVAR'.
18
19          PERFORM atualiza_email.
20
21      WHEN 'VOLTAR'.
22
23          LEAVE PROGRAM.
24
25  ENDCASE.
26
27  ENDMODULE.                  " USER COMMAND 9100 INPUT

```

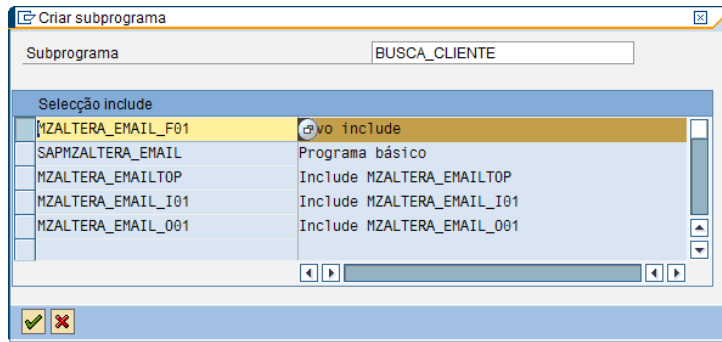
Agora criamos as rotinas BUSCA_CLIENTE e ATUALIZA_EMAIL. Estas rotinas serão colocadas em um novo include (releia a página 4). Dê um clique duplo sobre o nome da rotina BUSCA_CLIENTE:



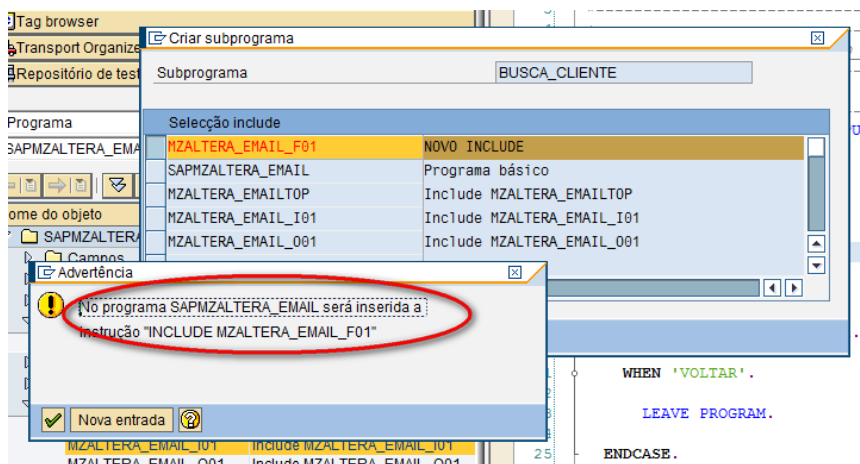
Novamente o SAP sugere o nome para o novo include:



Altere o nome do include para MZALTERA_EMAIL_F01 e tecla Enter:



Após a confirmação aparece o aviso. Tecle Enter:



Codifique a rotina conforme abaixo (abaixo da figura está o código-fonte para você copiar/colar):

```

1  *-----
2  ***INCLUDE MZALTERA_EMAIL_F01 .
3  *-----
4  *-----
5  *      Form  BUSCA_CLIENTE
6  *-----
7  *      text
8  *-----
9  *  -->  p1      text
10 *  <--  p2      text
11 *-----
12 FORM busca_cliente .
13
14     SELECT SINGLE *
15     FROM scustom
16     WHERE id = scustom-id.
17 IF sy-subrc <> 0.
18     CLEAR scustom.
19 ENDIF.
20
21 ENDFORM.                                " BUSCA_CLIENTE

```

FORM busca_cliente .

```

SELECT SINGLE *
FROM scustom
WHERE id = scustom-id.
IF sy-subrc <> 0.
    CLEAR scustom.
ENDIF.

```

ENDFORM. " BUSCA_CLIENTE

Explicando o código: a variável SCUSTOM-ID recebe da tela o valor digitado para o código de cliente. Para declarar esta variável coloque no include TOP: TABLES SCUSTOM, conforme mostrado abaixo.


```

1  * &-----
2  * & Include MZALTERA_EMAILTOP
3  * &
4  * &-----
5
6  PROGRAM sapmzaltera_email.
7
8  TABLES scustom.
9
10 DATA: okcode TYPE sy-ucomm.

```

Observação: a declaração TABLES não significa necessariamente que vamos usar a tabela SCUSTOM no programa. Podemos ler e gravar na tabela SCUSTOM sem usar esta declaração. Basicamente a declaração TABLES indica duas coisas:

- Cria uma variável estruturada de mesmo nome e tipo da tabela transparente SCUSTOM;
- Se a work-área em comandos do open SQL for omitida, esta variável será usada.

Portanto, os códigos abaixo produzem o mesmo efeito:

Código 1: SELECT SINGLE * FROM SCUSTOM WHERE ID = '00003404'.

Código 2: SELECT SINGLE * FROM SCUSTOM **INTO SCUSTOM** WHERE ID = '00003404'.

O código 1 só pode ser usado se houver a declaração TABLES. Se você não usar a declaração TABLES, é necessário criar uma work-área como mostrado abaixo:

DATA WA_CLIENTE TYPE SCUSTOM.

SELECT SINGLE * FROM SCUSTOM **INTO WA_CLIENTE** WHERE ID = '00003404'.

Salve seu trabalho e retorne ao module USER_COMMAND_9100. Crie agora a rotina ATUALIZA_EMAIL:

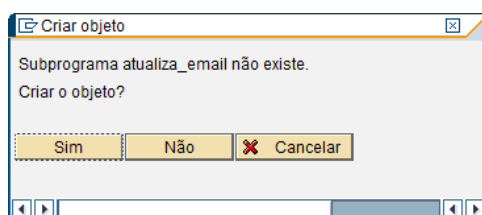
```

9  MODULE user_command_9100 INPUT.
10
11  CASE okcode.
12
13    WHEN 'PESQUISAR'.
14
15      PERFORM busca_cliente.
16
17    WHEN 'GRAVAR'.
18
19      PERFORM atualiza_email.
20
21    WHEN 'VOLTAR'.
22
23      LEAVE PROGRAM.
24
25  ENDCASE.
26
27  ENDMODULE.

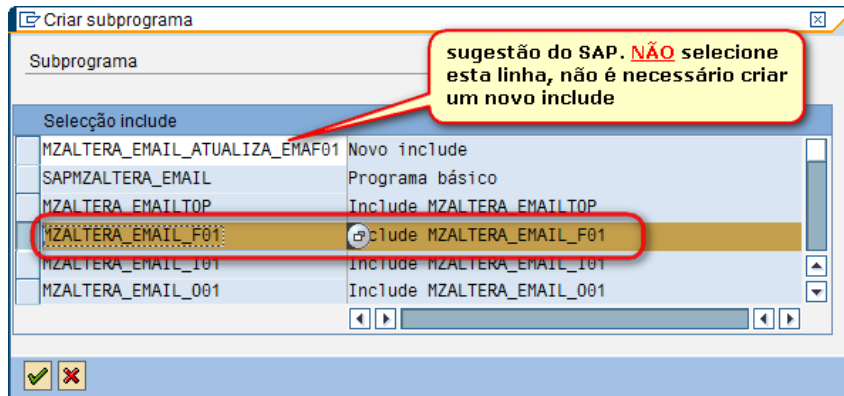
```

clique duplo no nome da rotina para criá-la

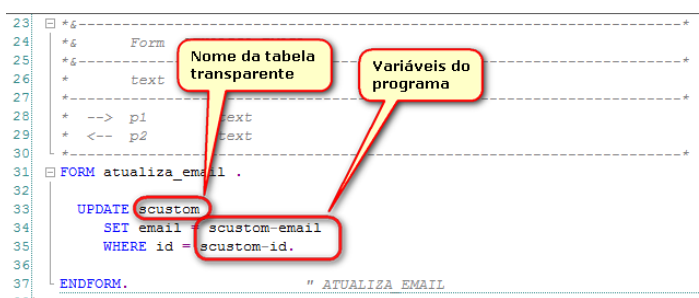
Confirme a criação com um Enter:



O SAP sugere o nome de um novo include, mas já criamos um para as rotinas. Selecione o include já criado:

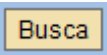




Codifique a rotina conforme figura abaixo e salve seu trabalho:



```
FORM atualiza_email .
  UPDATE scustom
    SET email = scustom-email
    WHERE id = scustom-id.
ENDFORM.          " ATUALIZA_EMAIL
```

Ative todos os seus objetos (veja nas páginas 20 e 21). Use a transação ZALT_EMAIL para testar o seu programa:

- Informe o código do cliente (ou use o search-help do campo) e tecla o botão PESQUISAR ();
- Se o cliente existir, o nome e e-mail devem aparecer na tela. Caso contrário todos os campos da tela serão apagados, inclusive o código digitado;
- Altere o e-mail e clique no botão Salvar (). Saia da transação e entre novamente, selecionando o mesmo cliente. Clique no botão Buscar: o novo e-mail deve aparecer na tela;
- Deixe o campo código em branco e tecla o botão Voltar (). O que aconteceu?

Module de saída – AT EXIT-COMMAND

Na montagem do layout da tela, marcamos o campo código como obrigatório (vide página 11). Se ele não for informado, não conseguimos abandonar a tela usando o botão VOLTAR, pois o SAP emitirá a mensagem “Preencher todos os campos obrigatórios”.

A opção AT EXIT-COMMAND em um module do PAI indica que, mesmo com campos obrigatórios, o module será executado ANTES de qualquer verificação de tela. Desta forma conseguimos colocar código de saída da tela. Atenção: um module com a opção AT EXIT-COMMAND será executado somente se o botão tiver com o tipo E (vide página 16).

Inclua um novo module no PAI conforme mostrado:

Nº tela 9100 inativo

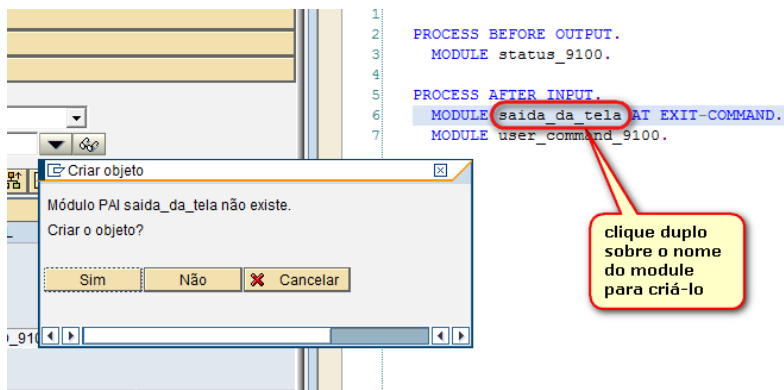
Características Lista elems. LógProcmt.

```

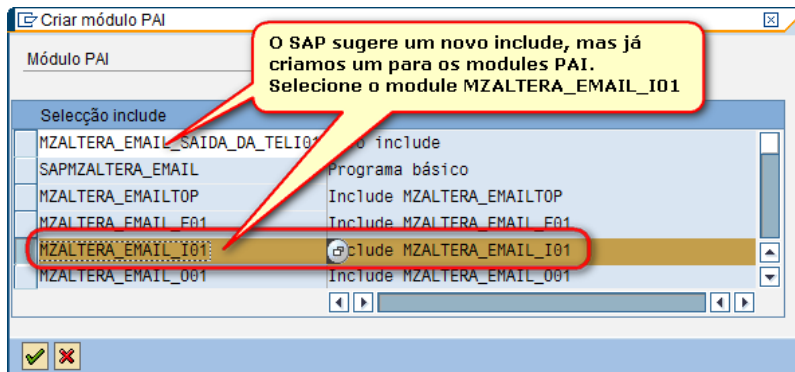
1
2 PROCESS BEFORE OUTPUT.
3   MODULE status_9100.
4
5 PROCESS AFTER INPUT.
6   MODULE saida_da_tela AT EXIT-COMMAND.
7   MODULE user_command_9100.

```

Dê um clique duplo sobre o nome do module para criá-lo:



Selecione o include de module já criado:




Coloque o comando de saída no module. Aproveite para ativar o programa:

Include MZALTERA_EMAIL_I01 ativo

```

27 ENDMODULE. " USER COMMAND 9100 INPUT
28
29 *-----*
30 *&      Module  SAIDA_DA_TELA  INPUT
31 *&-----*
32 *      text
33 *-----*
34 MODULE saida_da_tela INPUT.
35   LEAVE PROGRAM.
36
37 ENDMODULE. " SAIDA DA TELA INPUT
38

```

A partir de agora o clique no botão Voltar () acionará o module SAIDA_DA_TELA antes de qualquer outro module do PAI. O comando LEAVE PROGRAM no module USER_COMMAND_9100 não é mais necessário. Edite o module USER_COMMAND_9100 e retire este código, conforme mostrado abaixo:

```
Include MZALTERA_EMAIL_I01 ativo
9 | MODULE user_command_9100 INPUT.
10 |
11 | CASE okcode.
12 |
13 | WHEN 'PESQUISAR'.
14 |
15 |     PERFORM busca_cliente.
16 |
17 | WHEN 'GRAVAR'.
18 |
19 |     PERFORM atualiza_email.
20 |
21 | WHEN 'VOLTAR'.
22 |     LEAVE PROGRAM.
23 |
24 | ENDCASE.
25 |
26 | ENDMODULE. " USER COMMAND 9100 INPUT
27 |
28 |
```

Após ativar todo o programa faça o teste: entre na transação, deixe o campo código em branco e tecle o botão Voltar. Esta ação deverá mostrar a tela inicial do SAP.

CONSIDERAÇÕES FINAIS

- O exemplo citado neste roteiro altera diretamente uma tabela standard, a SCUSTOM. Lembre que alterações diretas em tabelas standard são procedimentos raros em programação abap;
- Os erros mais comuns para iniciantes no desenvolvimento de programas on-line são:
 - Esquecer de ativar algum elemento do programa (include, tela, status gui);
 - Omitir o campo OKCODE na lista de elementos da tela;
 - Não declarar no programa as variáveis de tela. Se você fizer isto, a compilação não apresenta erro, pois as variáveis de tela e de programa existem de forma independente;
 - Declarar variável no programa com nome diferente da variável de tela;
 - Não codificar os modules ou nomeá-los de forma diferente daquela que consta na lógica de processamento. A compilação não apresenta erro se você escrever MODULE na lógica de processamento e não existir código para ele;
- Uma boa ferramenta de aprendizado para programas on-line (e outros também...) é o debug. Use-o antes de acionar um comando de tela (com o /H) e inspecione os valores das variáveis no PBO e PAI, principalmente o OKCODE. Logo você descobrirá a necessidade de limpá-la no PBO, pois se não definimos um código de função para a tecla Enter, o OKCODE mantém o último código de função selecionado;
- Coloque agora no seu programa uma validação no campo de e-mail. Por exemplo, emita uma mensagem de erro caso o endereço de e-mail digitado não contenha um "@". Para isto você precisará do comando FIELD no PAI;
- Crie um novo botão Detalhe (código de função DETALHE) para exibir em uma outra tela (9200, por exemplo) os outros dados do cliente: endereço, cidade, telefone, forma de tratamento. Use o comando LEAVE TO SCREEN 9200 (ou SET SCREEN 9200. LEAVE SCREEN.) para chamar a tela 9200.

Espero tê-lo ajudado nos primeiros passos em programas on-line.

Sugestões, críticas e correções são bem vindas. Caso tenha alguma, por favor envie para um dos endereços de e-mail: camilo@geraessolutio.com.br / camilolelisveloso@gmail.com