**Academia ABAP**

**Aula 12 – Exits**



As “Exits” são pontos do SAP que podem ser alterados em programas Standards.

Não podemos alterar qualquer ponto de um programa standard, por isso a SAP reservou alguns locais específicos onde podemos inserir códigos para alterar ou acrescentar alguma lógica ao “Standard”. Temos diversos tipos de Exits no SAP e diversas formas de utilizar essas ferramentas e vamos conhecer abaixo os tipos de Exits e como utilá-las.

o User-Exit

As User-Exits são as exits mais utilizadas no ambiente SAP, envolvem o processo da ativação da um ponto do programa através de uma função onde podemos inserir a lógica necessária para a alteração e também podemos utilizar os pontos já ativos, ambos os cenários podem ser exibidos de acordo com cada ambiente SAP utilizado.

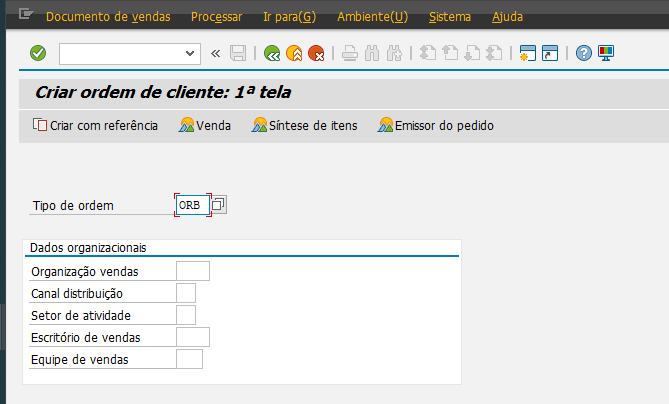
Podemos utilizar uma “User-Exit” para validar algum campo na tela do standard ou até mesmo modificar as informações baseadas na lógica de inserção.

⮚ Como encontrar exits

Antes de iniciarmos a edição de User Exits no SAP, precisamos primeiramente entender como encontrá-las e quais são as maneiras que temos para isso.

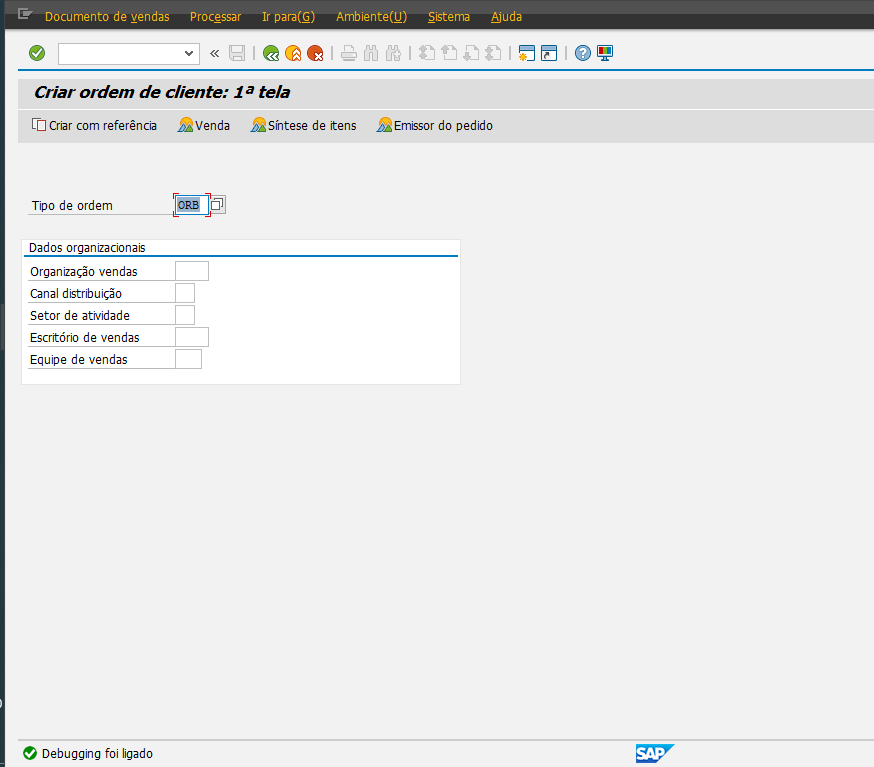
Uma das maneiras mais utilizadas para encontrar uma exit é através do Debug ABAP, utilizando um comando nas ferramentas do Debug para que sejam exibidas as exits entre um ponto e outro em um processo de uma transação Standard.

Transação VA01:



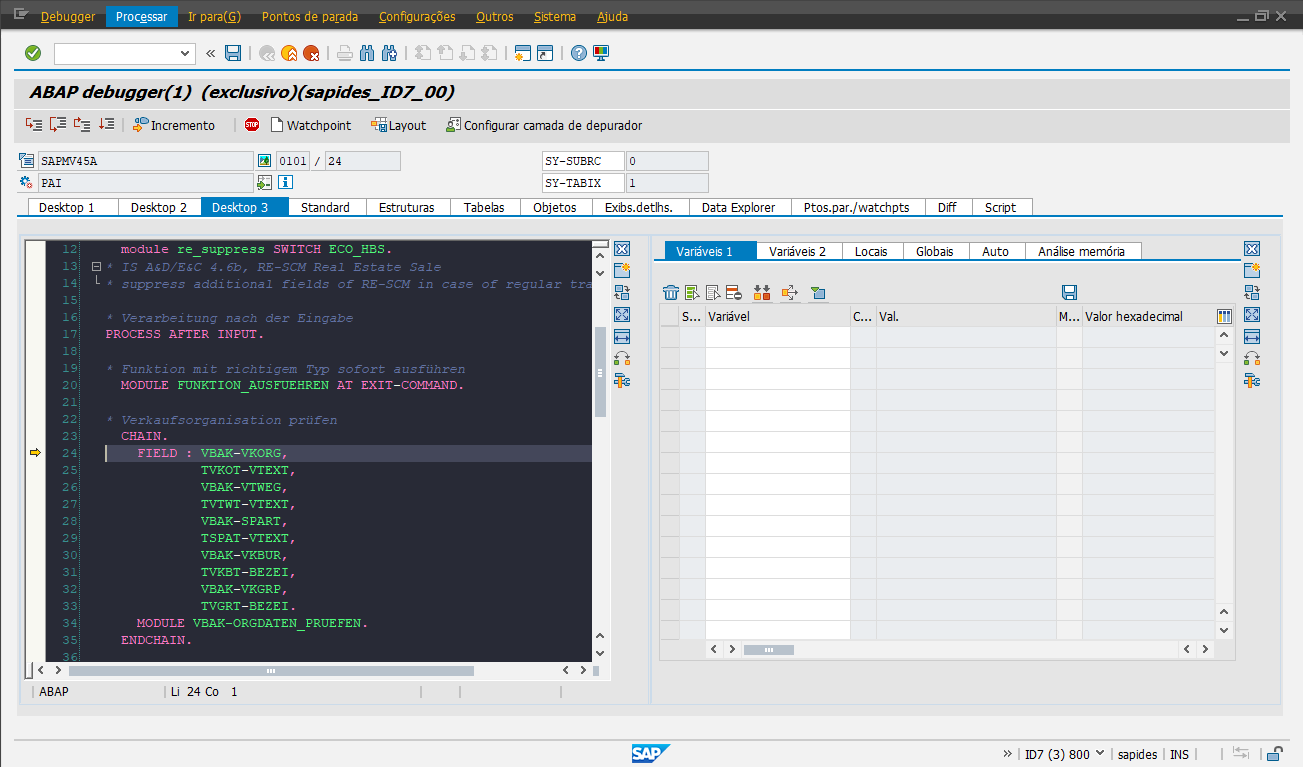
Neste momento, vamos verificar se existe alguma User-Exit entre a primeira tela da VA01 e a próxima tela onde serão preenchidas as informações da ordem de venda.

Depois de preencher o “Tipo de Ordem” como “ORB”, abra o Debug inserindo /H no campo de comando e siga conforme os passos abaixo:

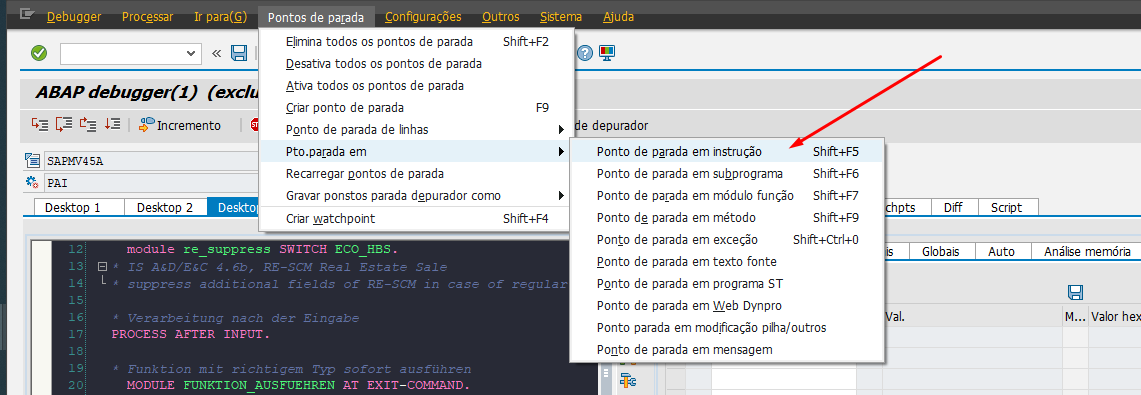


Pressione “Enter”

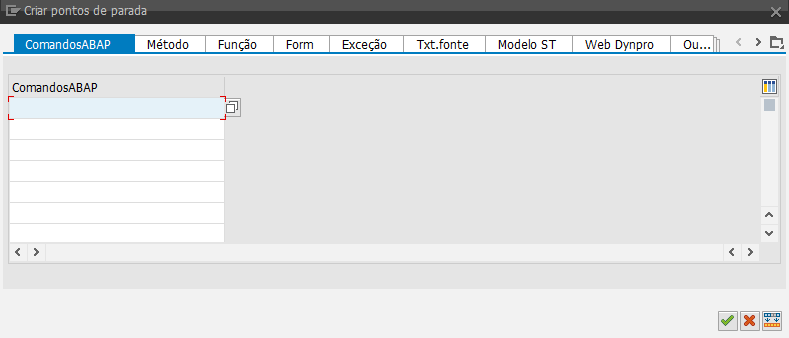
A tela abaixo será exibida, em modo de Debug:

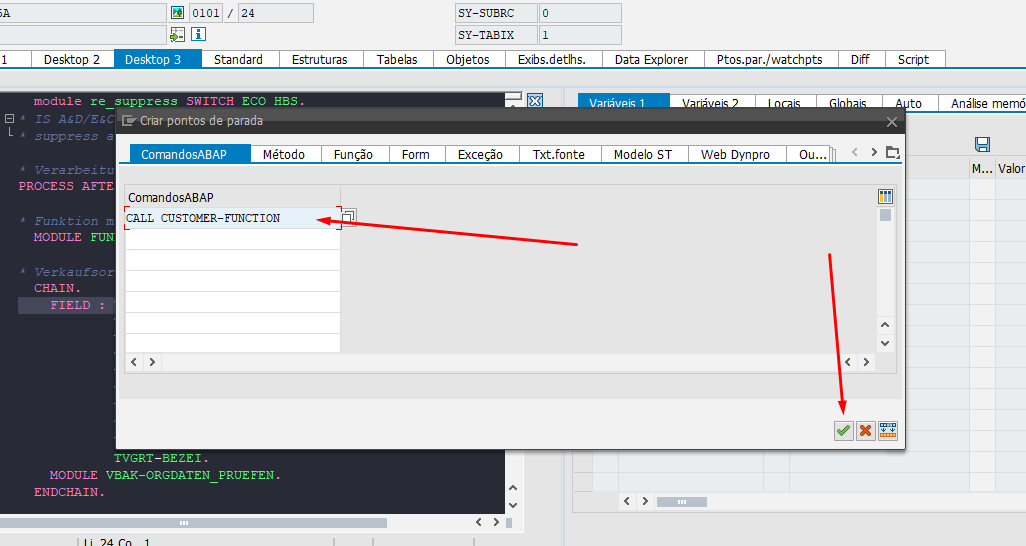


Navegue até a opção “Ponto de parada em Instrução” (SHIFT + F5)



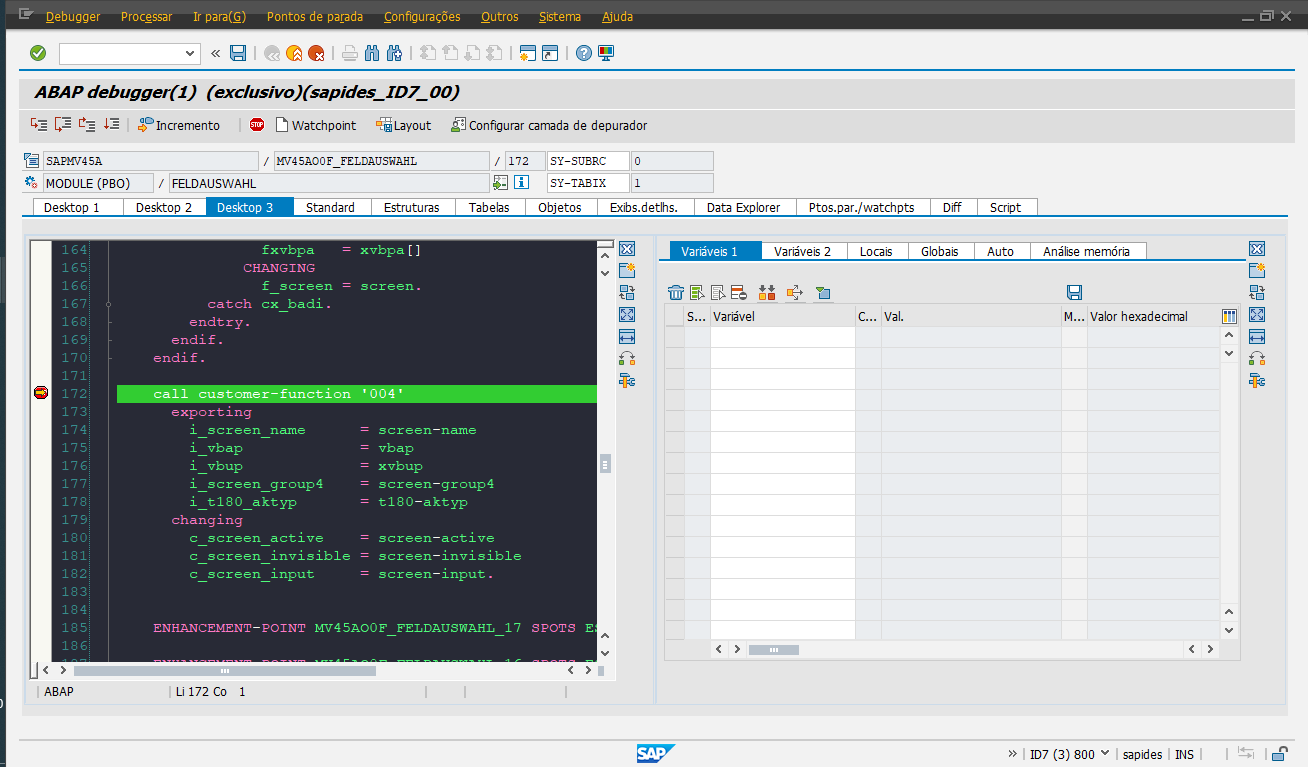
A tela abaixo será exibida e na aba “Comandos ABAP” podemos inserir qualquer sintaxe existente na base ABAP para que seja criado automaticamente um ponto de parada em cada linha em que a transação percorrer que contenha essa lógica, neste caso utilizaremos o comando “CALL CUSTOMER-FUNCTION” (Segunda Imagem)







Agora, ao pressionar F8 no teclado o Debug encontrará todos os “CALL CUSTOMER-FUNCTION” existentes na transação, essa sintaxe é utilizada para encontrar as “user-exits” existentes no processo da transação que está sendo executada.

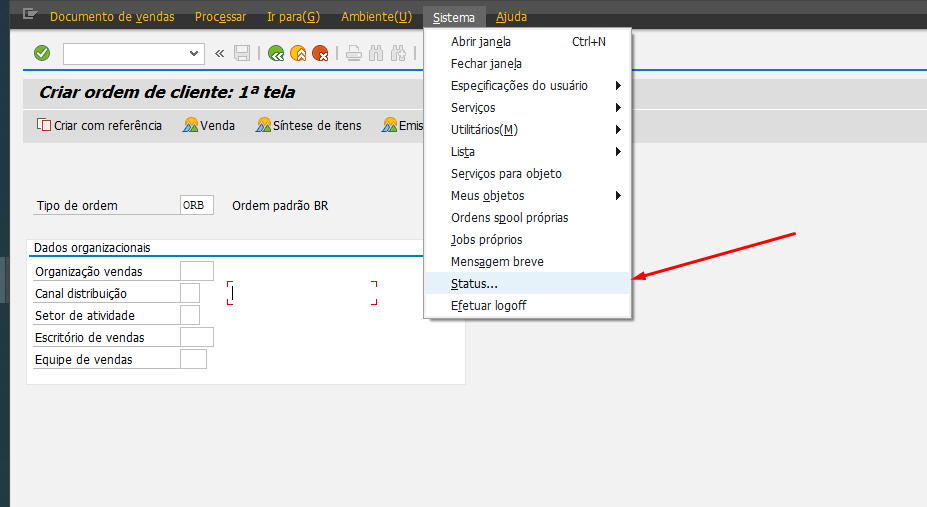


Na imagem acima, podemos ver o comando “CALL CUSTOMER-FUNCTION seguido do número da Exit “004”, como demonstrado neste exemplo, se precisarmos de uma exit entre a primeira e a segunda tela da transação VA01 poderíamos utilizar esse ponto para inserir a lógica necessária.

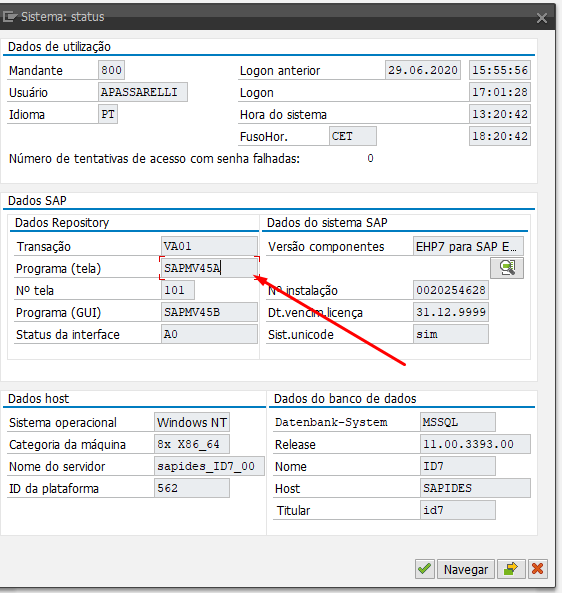
A questão da utilização da exit correta depende do processo que está sendo executado, pois existem diversas chamadas desse comando pela transação e pode ser que o ponto escolhido não seja o ideal para a inserção da lógica baseada no processo.

Uma outra forma de buscar uma exit é através do nome do programa principal da transação que está sendo executada, nesse passo, só podemos visualizar todas as exits criadas para a transação sem podermos verificar quais objetos estão preenchidos ou não em tempo de execução pois a execução é apenas demonstrativa como veremos abaixo:

Navegue até a transação VA01 e no menu “Sistema -> Status”.

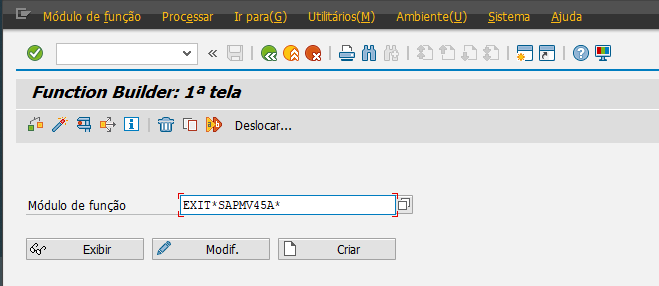


Copie o nome do programa principal, neste exemplo “SAPMV45A”.



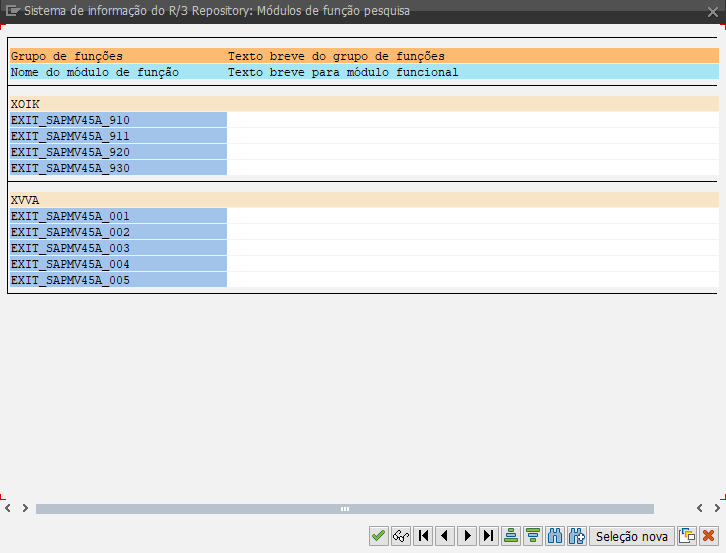
Após obtermos o nome do programa principal, vamos navegar até a transação SE37, onde vamos utilizar o nome do programa para encontrar as funções que fazem referência ao “CALL CUSTOMER-FUNCTION” da VA01.

Insira o texto “EXIT” seguido de “\*” no ínicio e no fim no nome do módulo de função:



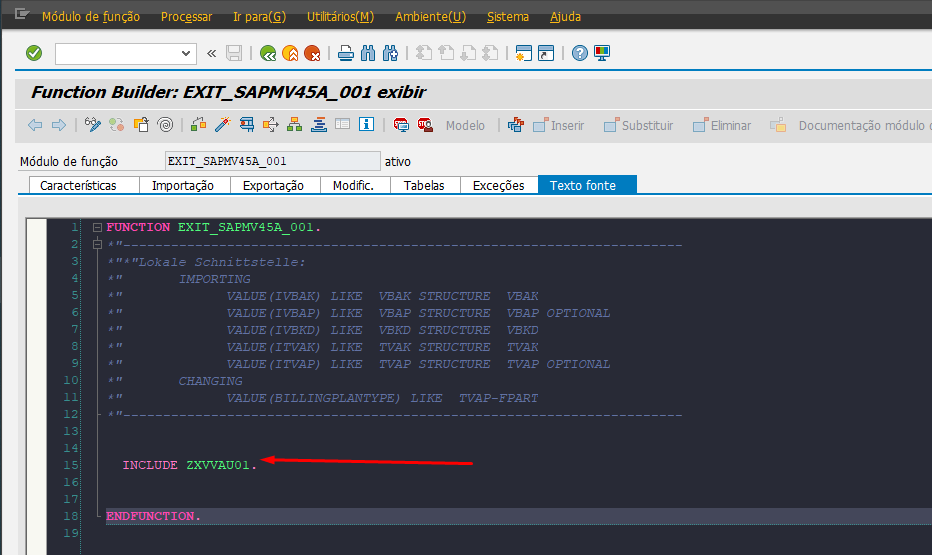
Conforme a imagem acima, o texto “EXIT” é usado no início de todas as User-Exits que pertencem a uma transação e sempre são seguidas do nome do programa principal, como temos sempre mais de uma ao final também inserimos um “\*” para que sejam encontradas todas as entradas correspondentes.

Depois de preencher as informações acima, pressione F4 no teclado ou o botão  para iniciar a pesquisa, o resultado podemos conferir abaixo:



Acima estão listadas todas as exits que pertencem ao conjunto do programa SAPMV45A (VA01), embora prático esse processo pode não listar todas as exits existentes em uma transação, pois internamente uma transação pode ter mais programas standards associados e neste caso só vamos exibir o conjunto que pertencem exatamente ao programa SAPMV45A, por isso o meio através do Debug é o mais recomendado para encontrar a Exit e o momento ideal para utilização.

Ao exibir uma Exit através da SE37, podemos visualizar um include Z previamente sugerido para utilização, é nele que faremos toda a lógica necessária para a alteração da User-Exit.



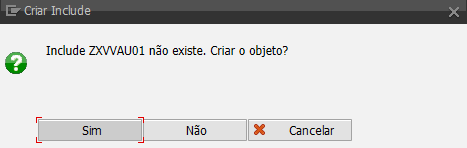
Na maioria dos clientes, as transações mais comuns como a VA01 já contém códigos ativos dentro desses includes, mas em alguns casos a Exit encontra-se apenas previamente criada sem estar ativa e nem utilizada por nenhuma lógica.

Mesmo que a Exit seja uma função como já vimos na aula anterior, não podemos modificar esse objeto sem estarmos dentro do include acima, pois esses são objetos standards que que não permitem alteração.

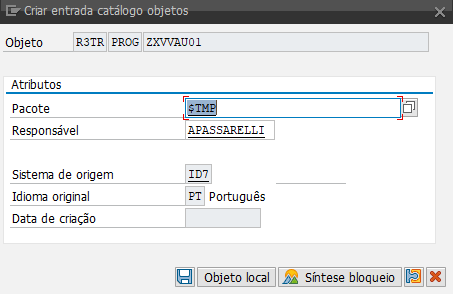
Ao clicarmos duas vezes sobre o Include da Exit, podemos visualizar que este objeto não existe e será exibida a mensagem abaixo, que antecede a criação do include:

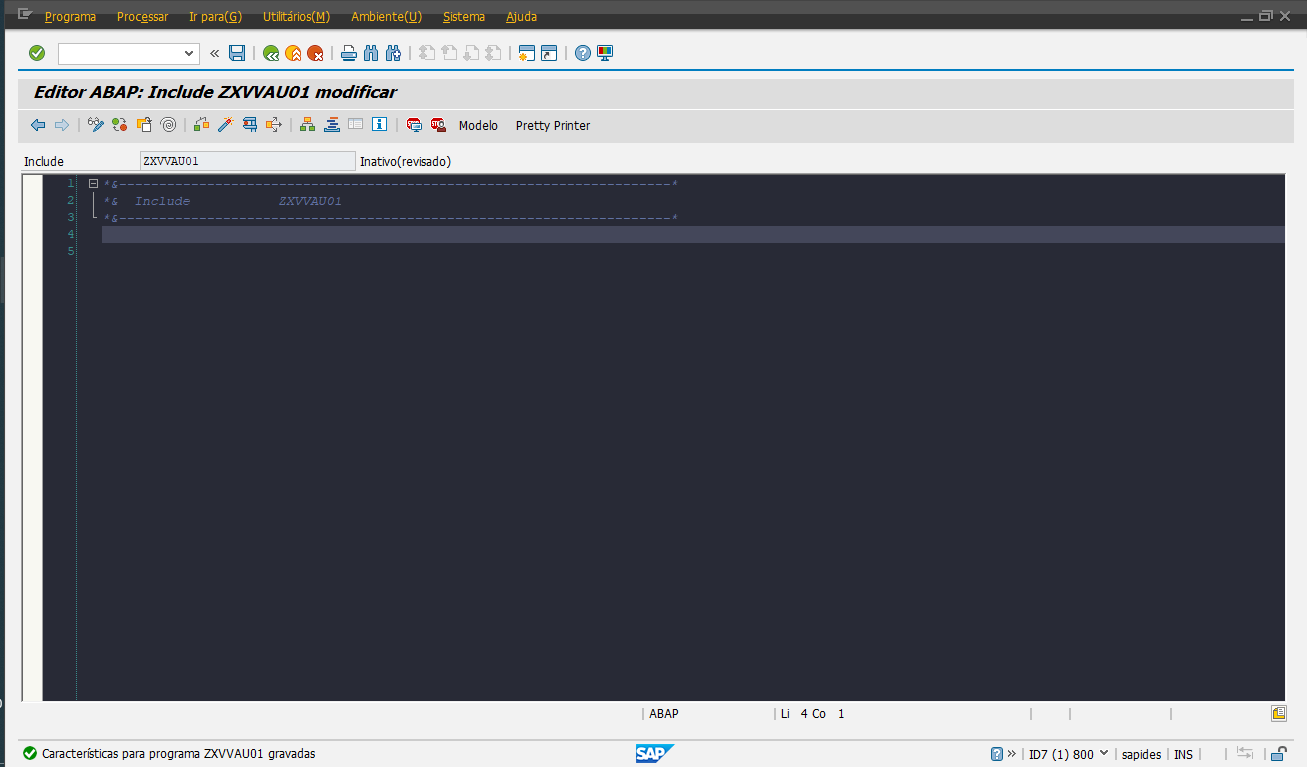


Ao pressionar “Enter”, será exibida a tela abaixo, solicitando a criação do Include:



Ao clicar em sim, o Include será criado e as telas abaixo serão exibidas:



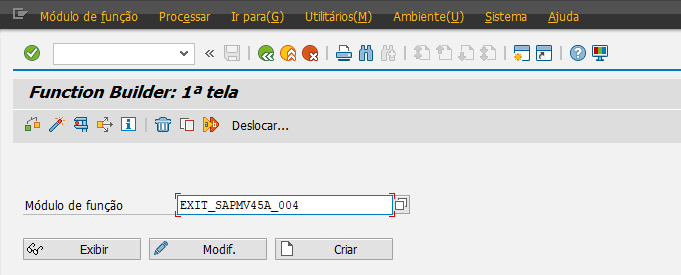


É possível excluir o Include criado através da transação SE38, porém, uma vez criado este será o objeto que será sempre utilizado pela exit nesse caso a exit “EXIT\_SAPMV45A\_001” e não teremos outro include associado a função principal para qualquer altaração, essas por sua vez devem ser realizadas dentro do include criado anteriormente.

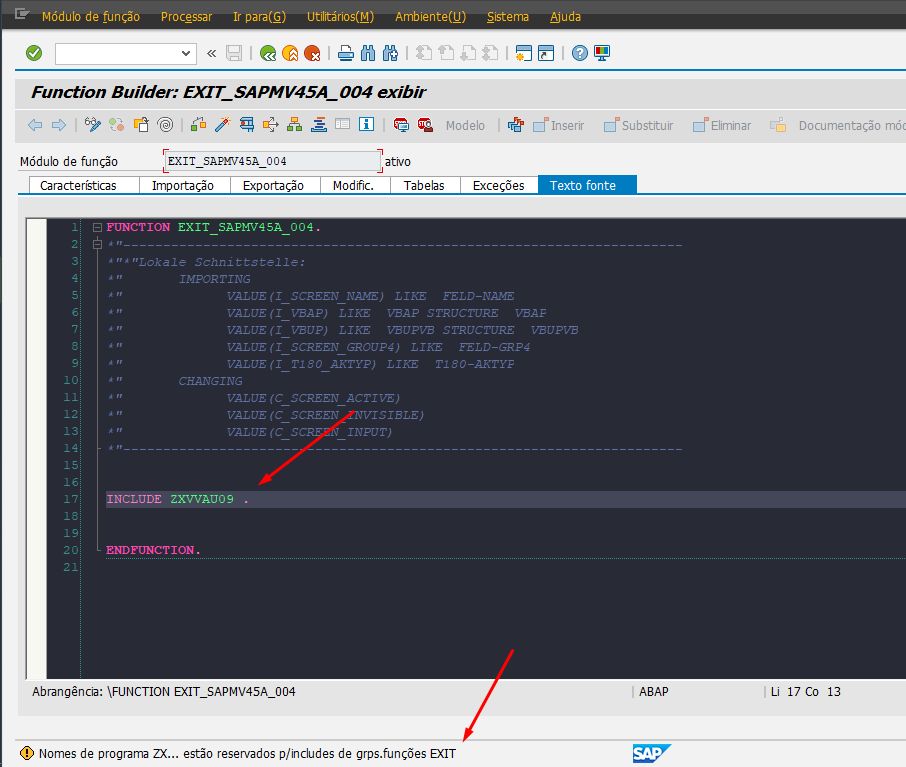
Vamos utilizar uma Exit da VA01 para inserir um bloqueio do tipo de ordem informado na tela, onde o usuário não poderá avançar da primeira tela se o Tipo de Ordem for igual ao definido na lógica da exit.

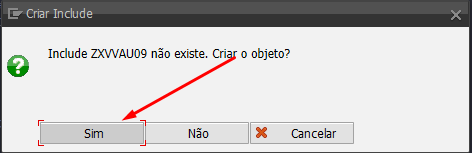
Neste exemplo vamos utilizar a Exit “EXIT\_SAPMV45A\_004”, pois já sabemos que ela encontra-se no conjunto entre a primeira e a segunda tela da VA01, conforme vimos acima via Debug, a “EXIT\_SAPMV45A\_001” não será utilizada nesse momento, pois não encontra-se neste momento necessário para inserir a lógica entre a primeira e a segunda tela.

Transação SE37:



Clique em “Exibir” e crie o include do Texto Fonte, dando um duplo clique sobre o nome do Include seguido de “Enter”.

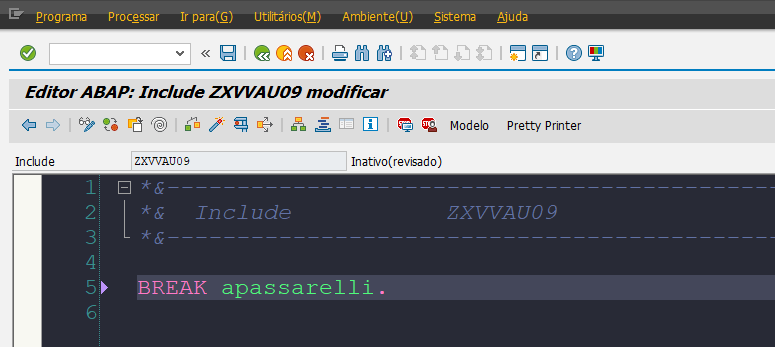




Para estes exemplos de criação de include, estamos salvando apenas como “Objeto Local” sem associar a nenhum pacote e Request.

Agora, dentro do include vamos primeiramente inserir um BREAK-POINT para verificar se realmente a exit está funcionando para o ponto que precisamos.

Como as Exits envolvem o processo geral de uma transação, devemos evitar utilizar diretamente o comando “BREAK-POINT”, pois este pode parar o processo para todos os usuários que estão utilizando essa transação, ao invés desse comando, vamos utilizar o Break Point por usuário, onde só será feita a chamada do Debug para o usuário informado no comando.

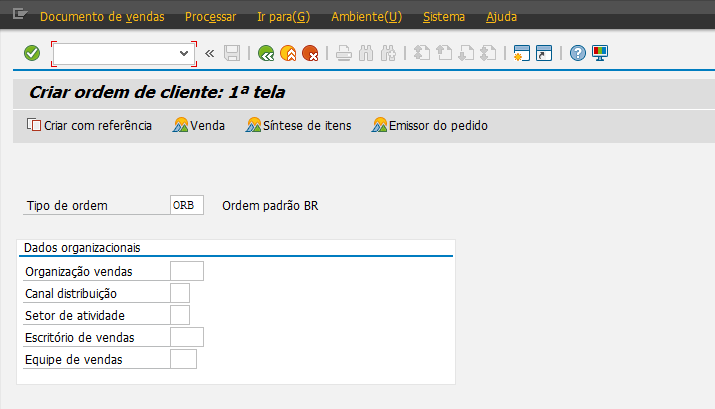


Ative o código:



Como essa exit nunca foi utilizada, devemos ativar a Exit para que ela seja processada quando acessarmos a transação VA01 entre a primeira e a segunda tela, para testar se a exit está ativa, apenas com o comando acima o Debug deverá ser aberto no caso de Exit Ativa, se não for aberto quer dizer que a exit está inativa, coforme as imagens abaixo:

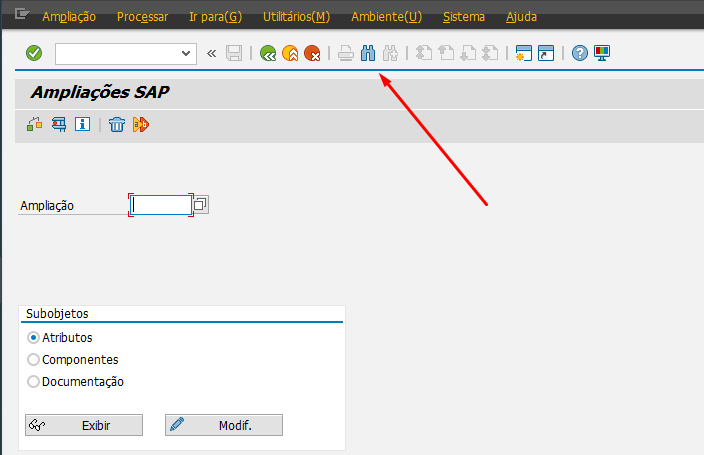
Transação VA01, informando o Tipo de Ordem “ORB”:



Ao pressionar “Enter” a segunda tela foi exibida sem parar dentro do include onde informamos o Break por usuário, isso indica que a Exit está inativa no momento de execução e por isso devemos ativar este objeto para que possamos utilizar essa Exit.

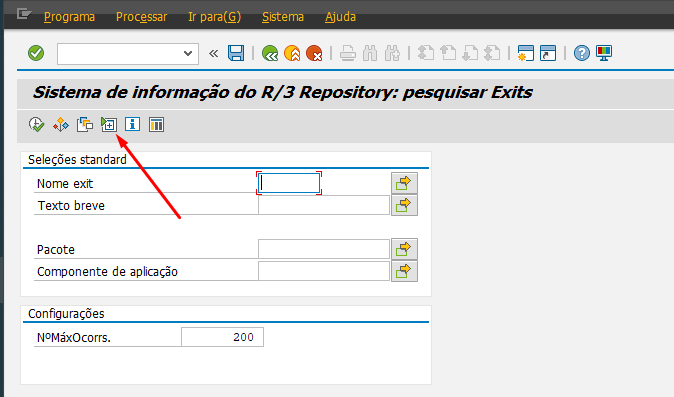
⮚ Ativar e desativar exits

Utilizamos a transação SMOD (Ampliações SAP) para encontrar a qual “Projeto” a exit que estamos utilizando pertence, para encontrar o projeto, entre na transação SMOD, e siga os passos conforme demonstrados abaixo:

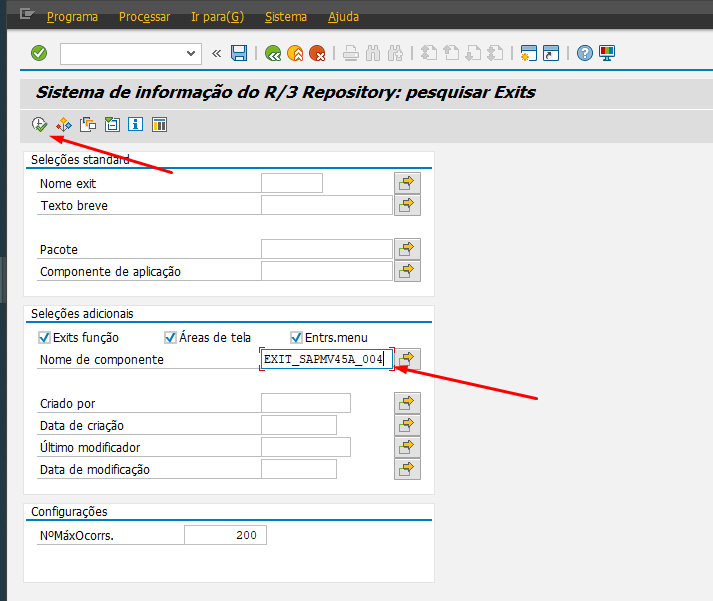


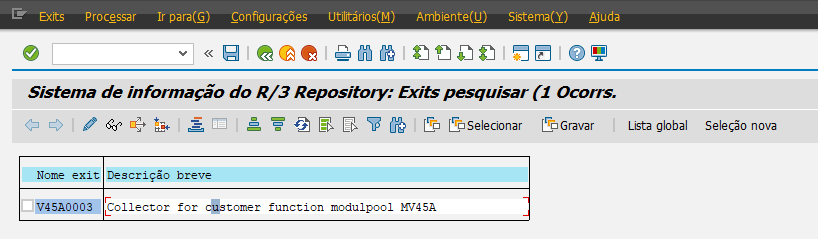
Clique em pesquisar conforme a imagem acima ou pressione o conjunto de teclas “Control + F”:

Clique no botão “Todas as Seleções” ou pressione o conjunto de teclas “SHIFT + F9”.



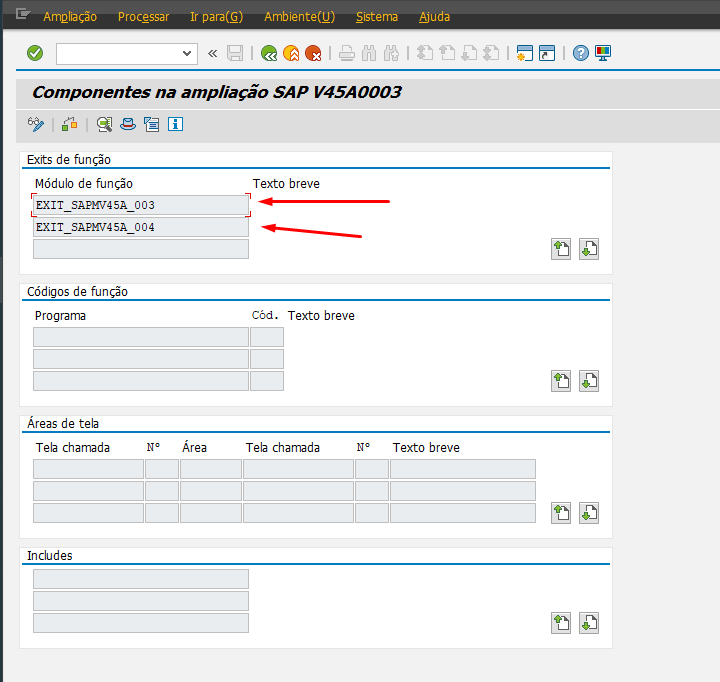
Em “Nome do Componente” informe o nome da Exit que está sendo utilizada “EXIT\_SAPMV45A\_004” e clique em Executar ou Pressione F8:



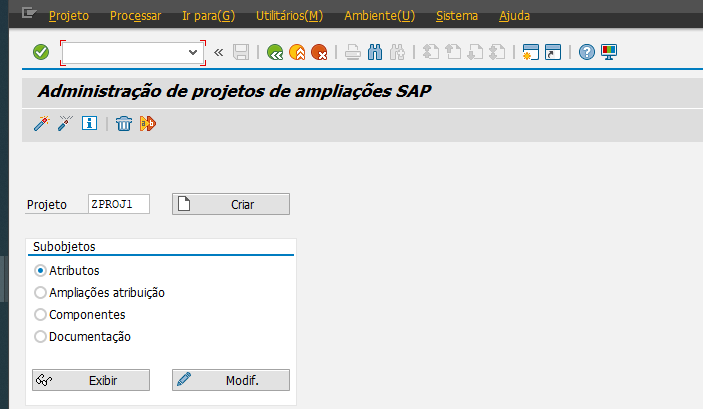


Será exibido no campo “Nome Exit” qual “Ampliação” aquela exit pertence, precisamos encontrar a “Ampliação” sempre que for necessário ativar uma exit.

Uma “Ampliação” pode conter mais de um tipo de exit ou mais de uma exit do mesmo tipo, geralmente elas são agrupadas dentro das etapas de processo de cada transação, se clicarmos duas vezes sobre o “Nome da Exit”, podemos visualizar que existem duas exits nesta ampliação, a 003 e a 004 que estamos utilizando nesse exemplo:



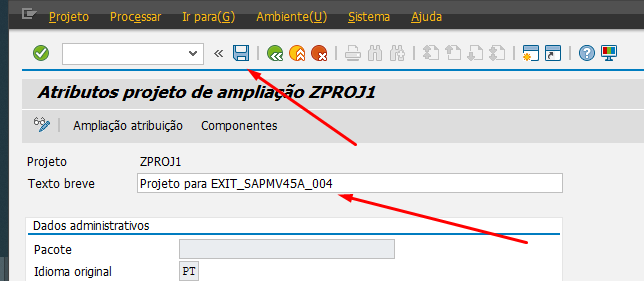
Guarde o nome da “Ampliação” (V45A0003) informado na pesquisa da SMOD para que possamos criar um projeto para ativar a Exit, a criação de projetos para exits inativas também é obrigatória, pois não podemos modificar o objeto de Ampliação standard da transação SMOD, para criarmos um projeto Z de uma Ampliação, precisamos navegar até a transação CMOD, informando o nome do projeto na primeia tela, conforme a imagem abaixo:



Conforme a imagem acima, foi informado o nome do projeto “ZPROJ1” para o projeto atual, este nome também é definido de acordo com as regras de cada cliente através do Workboook ABAP.

Após informar o nome do projeto, clique no botão .

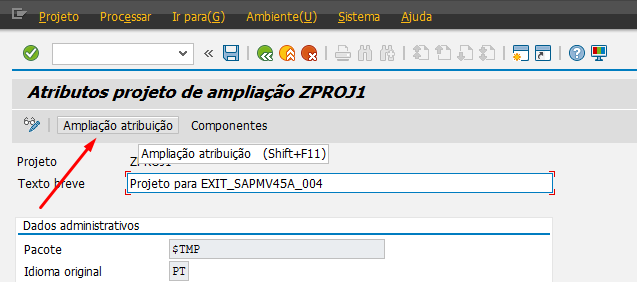
Informe a descrição breve do projeto e clique em “Salvar”



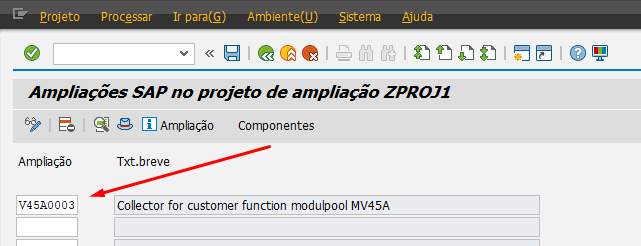
Escolha objeto local ou associe a um pacote de desenvolvimento que esteja utilizando, em seguida a mensagem será exibida:



Após os passos acima, clique no botão “Ampliação Atribuição”, para associar o nome da Ampliação a este Projeto:

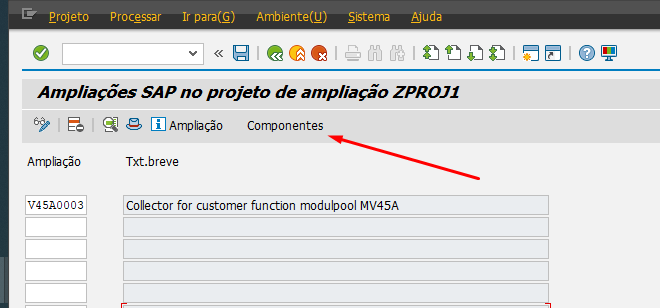


Informe o nome da Ampliação que encontramos na transação SMOD, conforme a imagem abaixo e pressione “Enter”

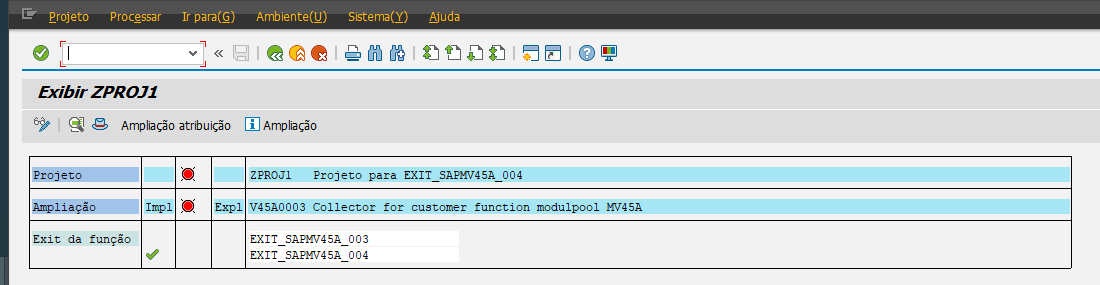


Nesta parte, você pode informar várias ampliações para o mesmo projeto, ou apenas um como neste exemplo, quanto mais ampliações presentes no projeto mais Exits serão disponibiliadas para ele.

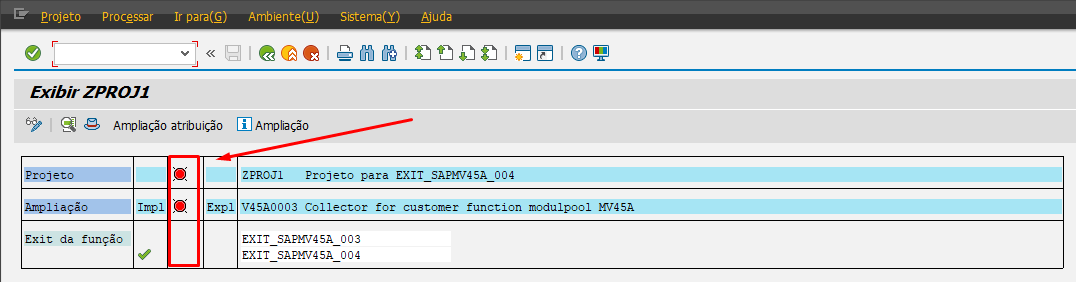
Após informar a Ampliação, navegue até o botão “Componentes”



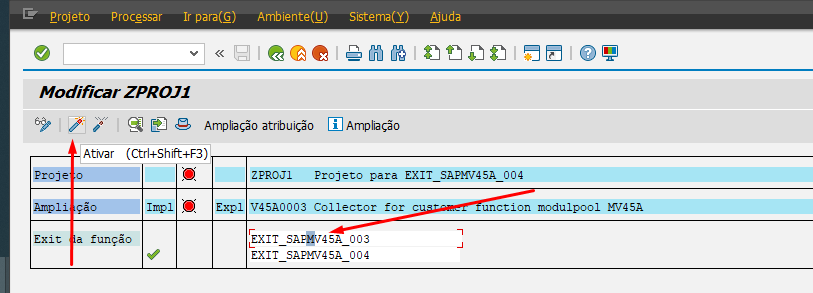
Grave as informações e a tela abaixo será exibida:



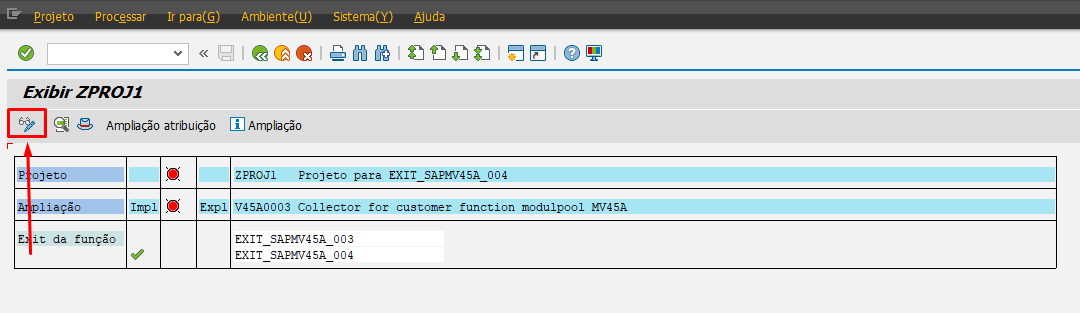
Conforme demonstrado na imagem acima, as duas exits 003 e 004 da VA01 estão sendo exibidas, porém estão inativas conforme podemos visualizar pelo status:



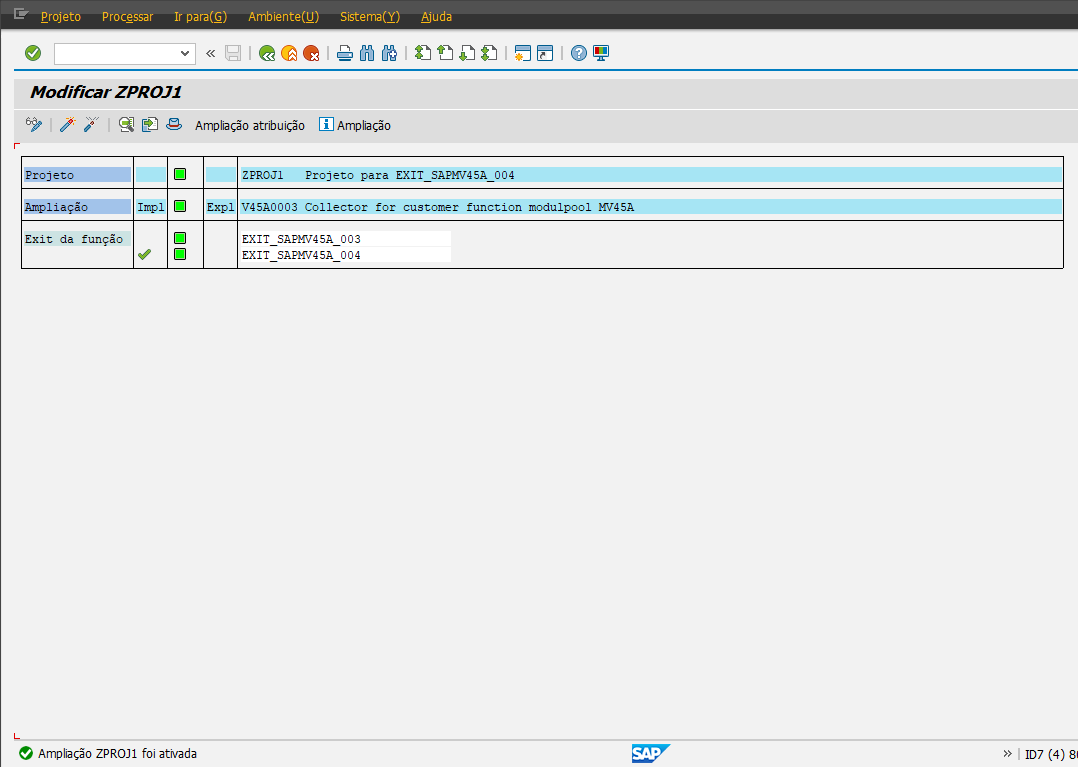
Para ativar os objetos, clique com sobre o nome de uma Exit e em seguida clique no botão “Ativar”, conforme a imagem abaixo.



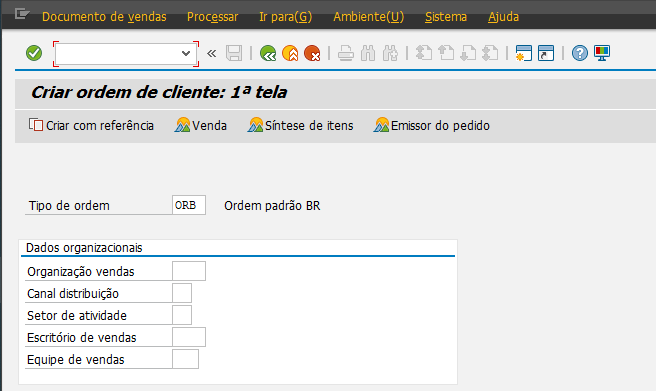
Se as opções para a ativação não estão sendo exibidas, você precisa entrar em modo de modificação através do botão abaixo.



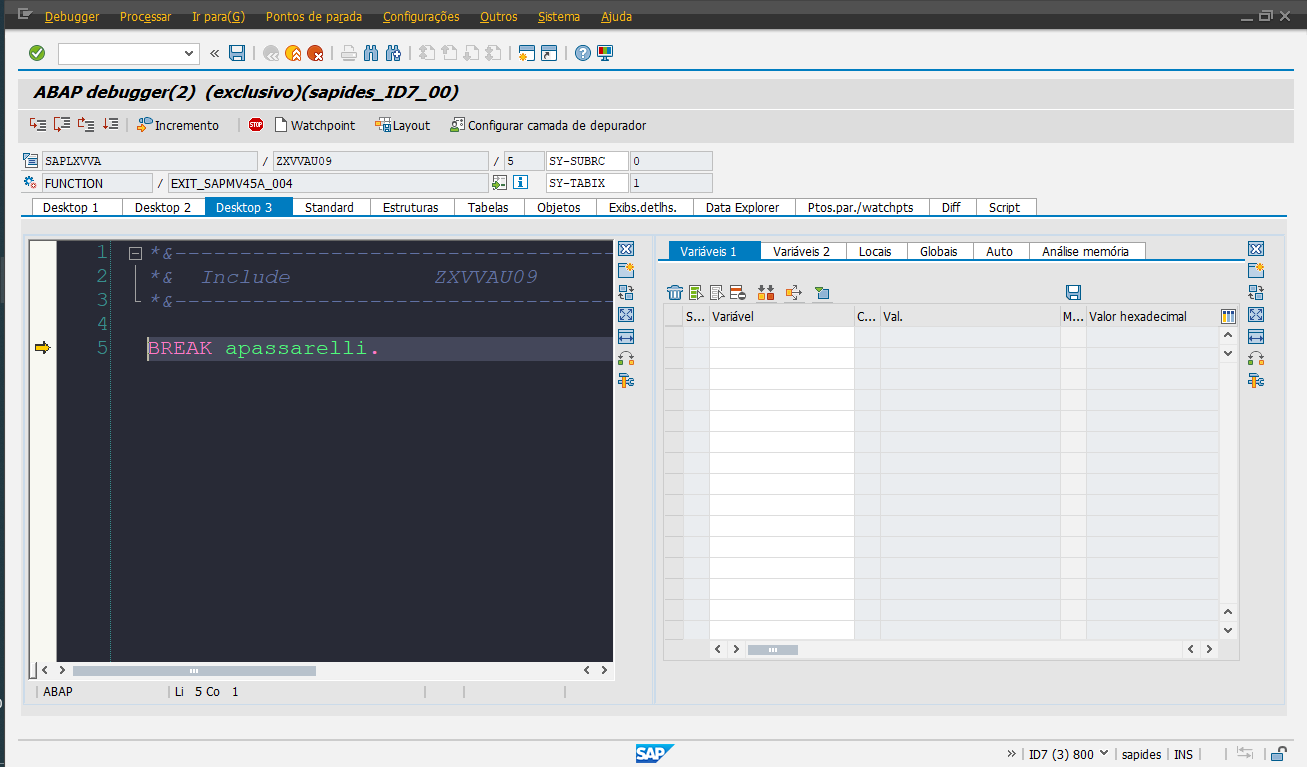
Após realizar a ativação das Exits, o Status mudará da cor vermelha para a cor verde e a tela abaixo será exibida:



Ao finalizar essa etapa, nossa Exit 004 já está ativa e pronta para ser utilizada, vamos simular novamente a entrada pela VA01, informando o Tipo de Ordem “ORB” e pressionando “Enter”.



Como nossa Exit está ativa, o Debug será aberto no ponto onde inserimos o código de Break para o usuário, conforme a imagem abaixo:

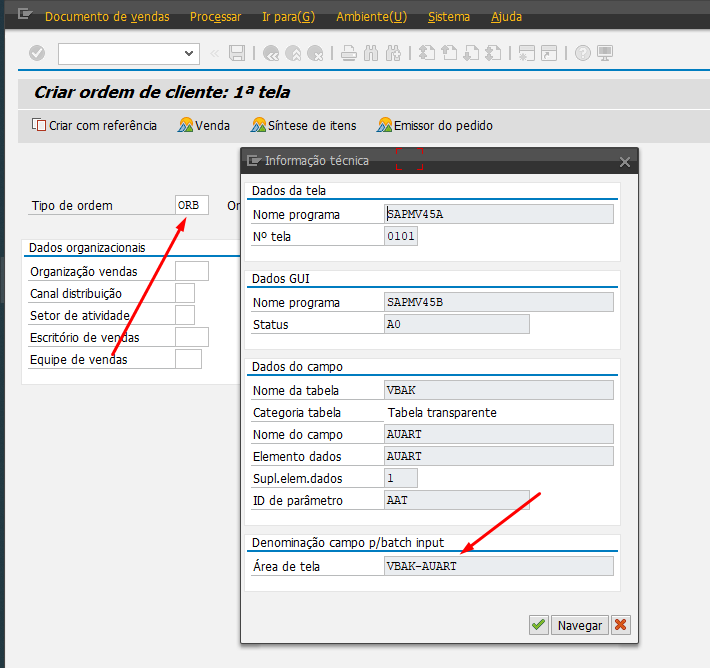


Caso todos os passos sejam realizados corretamente e a Exit seja chamada através do Debug e o comando Break por usuário, vamos poder inserir uma lógica para impedir que algum “Tipo de Ordem” não seja permitida.

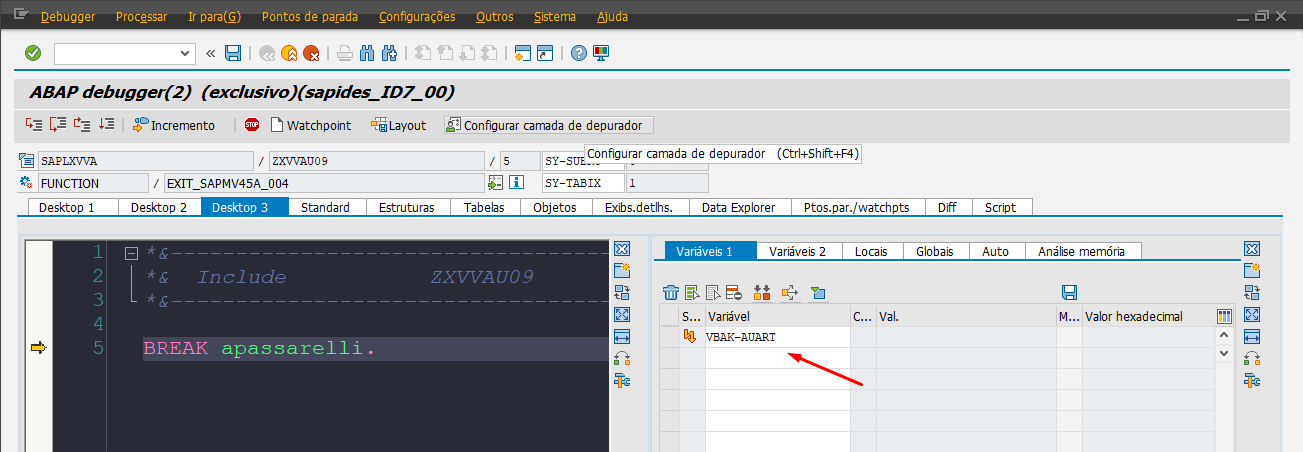
Antes de fazermos a alteração, precisamos entender o conceito dos “Field-Symbols” dentro de uma Exit.

Os Field-Symbols são objetos que chamamos de ponteiros no ABAP, eles podem se transformar em algum campo para serem utilizados em qualquer processo do SAP de qualquer tela que já tenha sido executada no processamento da transação.

No exemplo em questão, vamos utilizar o campo “Tipo de Ordem” para fazermos a validação, podemos visualuzar através da primeira tela da VA01, utilizando o F1 sobre o campo que seu nome é VBAK-AUART



Porém, ao chamarmos esse campo no Debug dentro da nossa Exit, podemos ver que o campo não existe naquele momento, mesmo que o processo esteja em andamento:

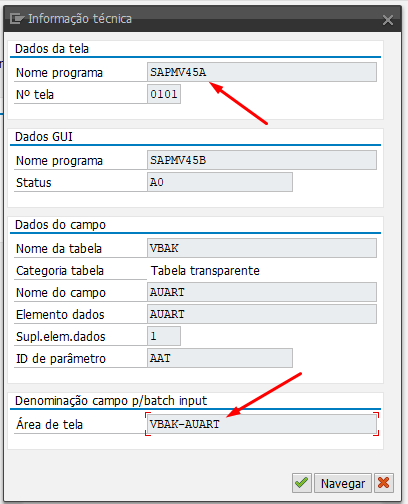


Utilizar um campo diretamente nos obriga a ter aquele campo exatamente no ponto em que estamos alterando, nesse caso, como estamos dentro de um Include de uma Função, o objeto não está referenciado diretamente ao ponto que estamos precisando, por isso, precisamos transformar esse campo em um “Ponteiro” para que ele possa ser utilizado nesse ponto sem perder a referência ao objeto original.

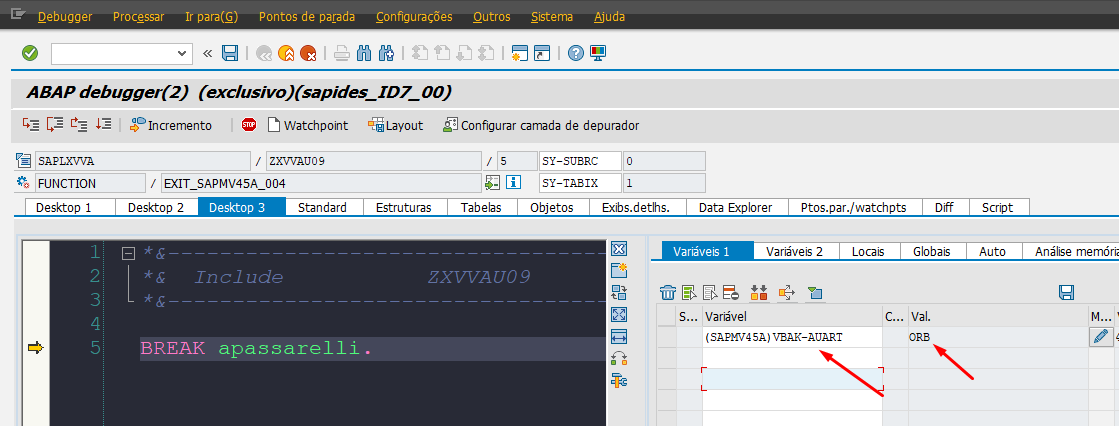
Utilizando o nome do programa principal informado no F1 do campo “Tipo de Ordem” e inserindo o nome do campo, podemos recuperar via Debug aquele campo para verificar se a informação que vamos validar existe naquele momento, para isso, devemos sempre informar no campo de variáveis do Debug a seguinte condição:

(PROGRAMA)ESTRUTURA-CAMPO

(SAPMV45A)VBAK-AUART

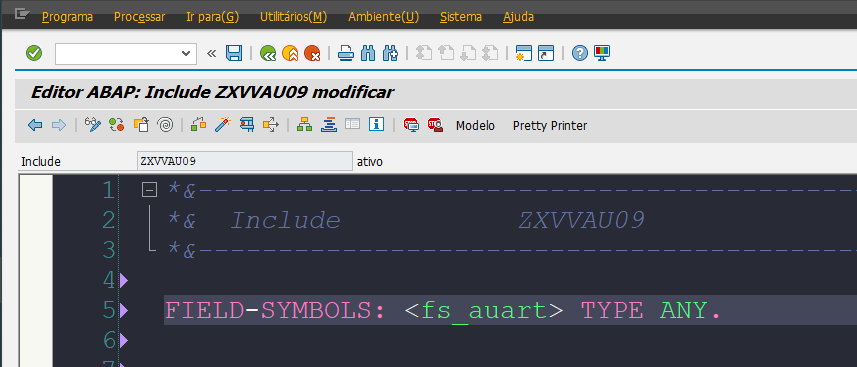


Ao inserir a informação no Debug, podemos visualizar o campo preenchido na primeira tela da transação VA01:



Enquanto estivermos processando a transação VA01, poderemos utilizar o modo de Ponteiro do campo para visualizar a informação do campo e também editá-la se necessário, isso funciona em qualquer transação, desde que o processo tenha sido executado anteriormente para que o Ponteiro esteja “Assinado” na memória ABAP.

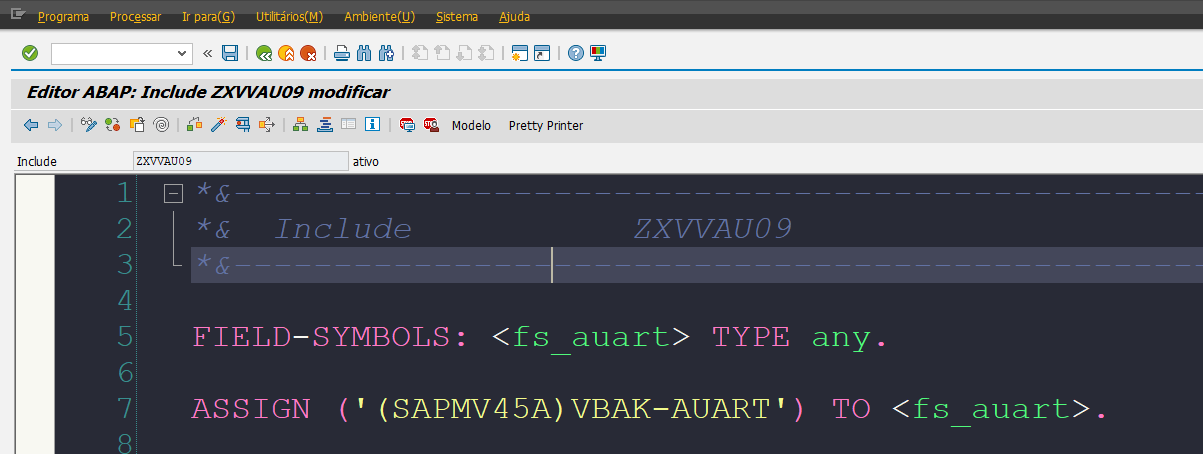
Para modificarmos um ponteiro via Código ABAP, precisamos criar um FIELD-SYMBOL com essas mesmas caraterísticas, assim esse objeto será criado internamente no código e pode ser utilizado ou modificado assim como no Debug através da associação do ponteiro, vejamos abaixo como criar um Ponteiro (Field-Symbol) no código ABAP:



Conforme a imagem acima, um Field-Symbol deve ser criado com o nome entre <>, utilizando o padrão SAP informamos um FS para indicar que é um Field-Symbol em seguida colocamos o nome do objeto ao qual fará referência que é o campo AUART da estrutura VBAK.

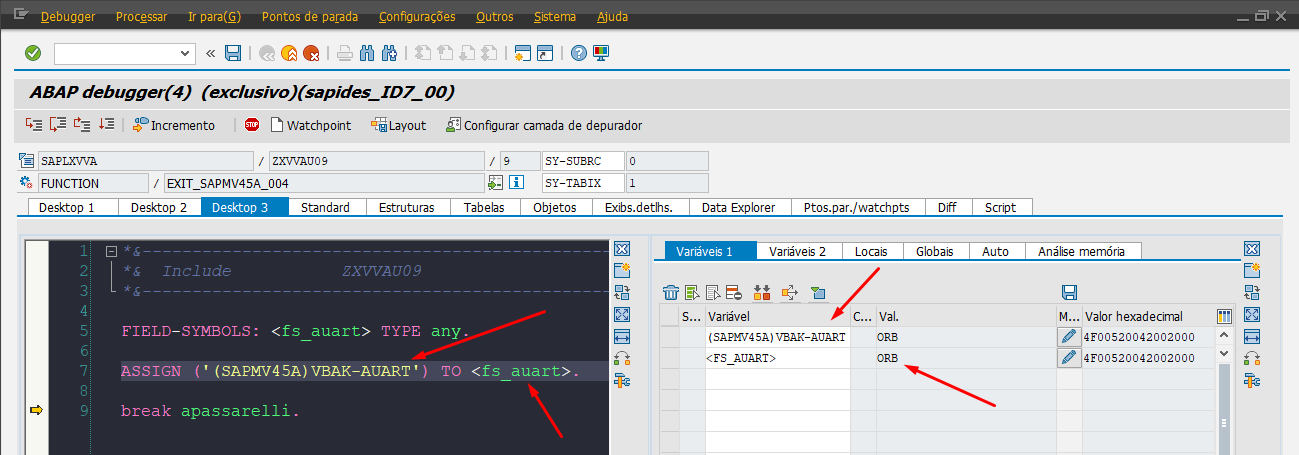
O tipo deve ser declarado como ANY, para indicar que ele poderá ser transformado em qualquer objeto, sem associar a um tipo fixo como fazemos com outros objetos como variáveis.

Em seguida, devemos utilizar o comando abaixo para Assinar os valores do Ponteiro ao Filed-Symbol:

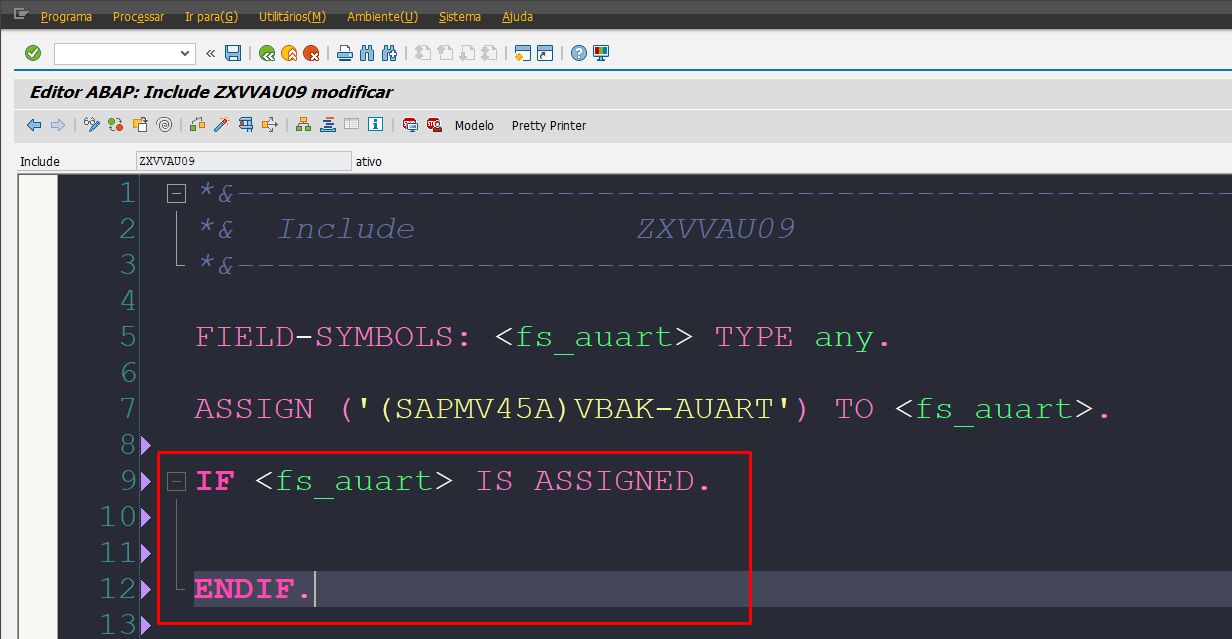


Dessa forma, o objeto <fs\_auart> agora é o mesmo que o objeto da estrutura da primeira tela VBAK-AUART e qualquer modificação associada no Field-Symbol será refletida em todo o processamento da transação para este campo.

Demonstração via Debug do Objeto:

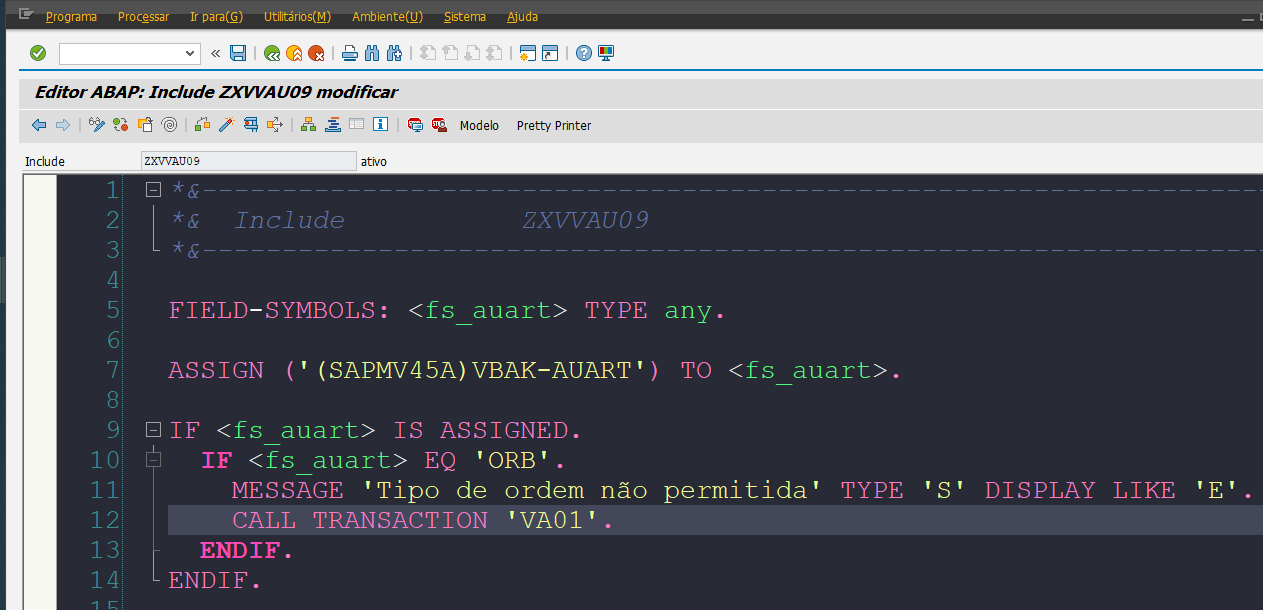


Ao utilizar um Field-Symbol, sempre é necessário verificar se sua assinatura está correta, para isso utilize sempre a instrução abaixo:

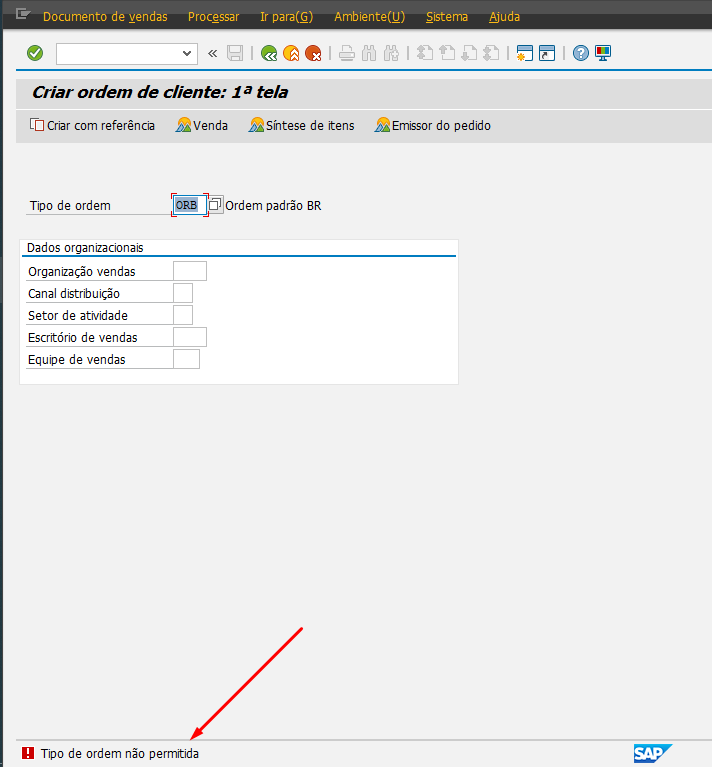


Caso o objeto não esteja assinado e seja utilizado o SAP retornará um Dump no sistema, por isso nunca devemos esquecer de verificar se o processo foi realizado com sucesso.

Agora, vamos inserir uma condição para bloquear o tipo de ordem “ORB” entre a primeira a e segunda tela da VA01, para isso, utilizaremos a lógica abaixo:



Será exibida uma mensagem na tela da VA01 e em seguida será chamada novamente a transação VA01 para que seja entrado um novo Tipo de Ordem.



É importante ressaltar que nunca devemos utilizar mensagens do tipo “E” diretamente em Exits, pois as Exits são lógicas que estão entre processos no SAP, ao utilizar uma mensagem do Tipo “E” sem uma inteligência que não encerre a transação, toda a tela será fechada independente de qual parte esteja, no exemplo do código acima estamos utilizando a chamada da transação VA01 novamente para não fecharmos a tela ao usuário pressionar enter, em alguns casos o usuário pode ter informado diversos campos em uma tela Standard e a execução de uma mensagem de erro encerrará o processo fechando toda a transação e prejudicando assim o funcionamento da transação.

Field-Exit

Quando há necessidade de consistências adicionais para um determinado campo na tela, o SAP disponibiliza o recurso de FIELD EXIT. Trata-se de uma pequena rotina em ABAP que será executada e que pode ter lógica, por exemplo, de validação do conteúdo.

Diferente da Exit que podem ser usada para qualquer objeto, sendo Campos, Tabelas, Estruturas e qualquer definição desejada, as FIELD EXITs são usadas apenas para um campo específico em uma tela.

Devemos tomar alguns cuidados ao utilizar essa ferramenta, pois a mesma apresenta os seguintes detalhes:

- Não é possível Debugar

- É considerada obsoleta pela SAP

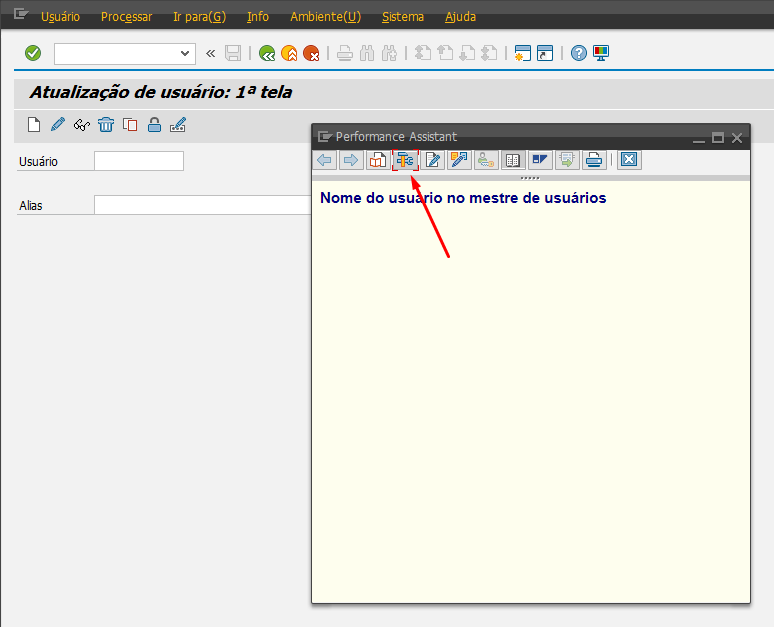
- A execuçã é feita apenas em telas, se for utilizada em conjunto com BAPIs podem não funcionar corretamente.

Em alguns ambientes SAP a utilização de Field-Exits encontra-se inativa, sendo necessário que um Basis faça a ativação do perfil para essa ferramenta pode ser utilizada, caso não seja ativada, mesmo que seguirmos todos os passos descritos nesse material e na aula ainda sim não será possível utilizar a Field-Exit.

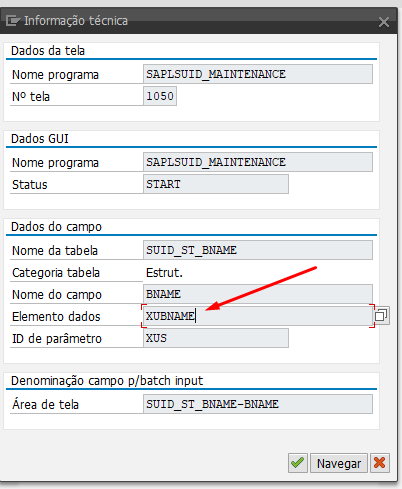
Podemos criar uma Exit de Campo (Field-Exit) para qualquer campo de qualquer transação do SAP, para essa ferramenta não precisamos encontrar previamente a Field-Exit antes de utilizar, como fizemos para o caso da User Exit, pois podemos criar a Field-Exit baseada no elemento de dados do campo, juntamente com as informações da tela onde esse campo se encontra.

No exemplo abaixo, vamos utilizar uma Field-Exit para bloquear a alteração de um usuário específico na transação SU01, o que pode ser utilizado para usuários de sistema de forma geral, impedindo que qualquer pessoa com acesso a transação faça a modificação só por tê-la liberada.

Primeiramente, vamos acessar a transação SU01, clicando no campo nome de usuário e pressionando F1 no teclado:



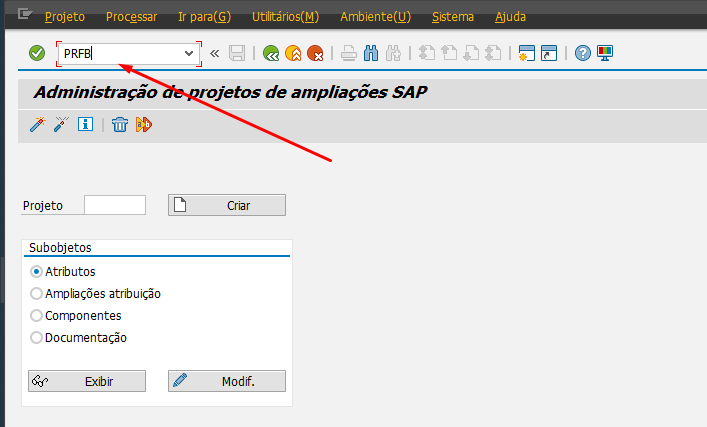
Clique no botão demonstrado na imagem acima:



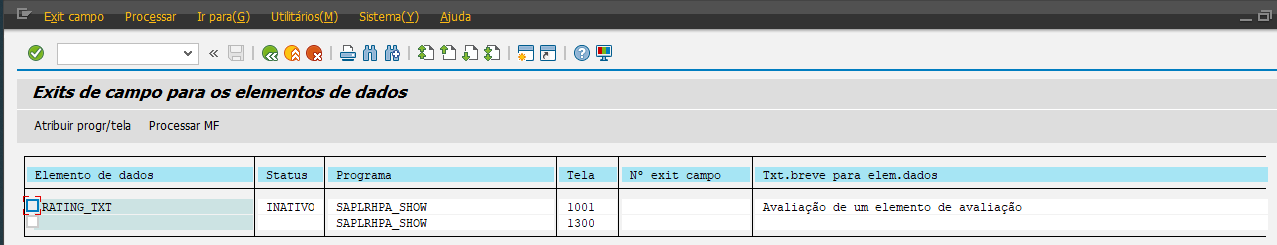
No primeiro passo, sempre guardamos o nome do Elemento de Dados, que é a primeira informação necessária para a criação da Field-Exit.

Agora, com a informação em mãos, vamos até a transação CMOD:

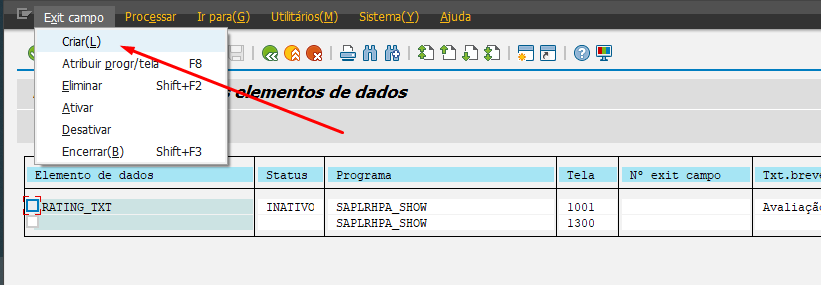
Com a tela da CMOD aberta, informe a sequência “PRFB” no campo de comando, conforme a imagem abaixo e pressione “Enter”:



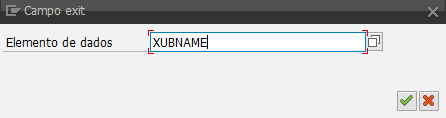
A tela abaixo será exibida:



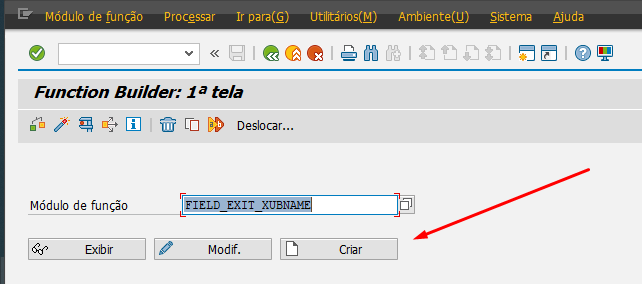
Neste ponto demonstrado na imagem acima ficam listadas todas as Field-Exits criadas no SAP, podemos modificar as Field-Exits já existentes ou criar novas entradas, para isso clique no menu “Exit Campo” e selecione “Criar”, conforme a imagem abaixo:



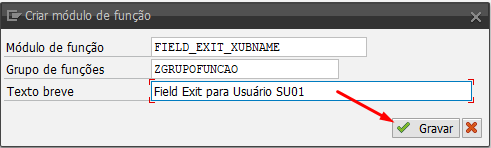
Informe os dados do Elemento de Dados que guardamos previamente e clique em confirmar 



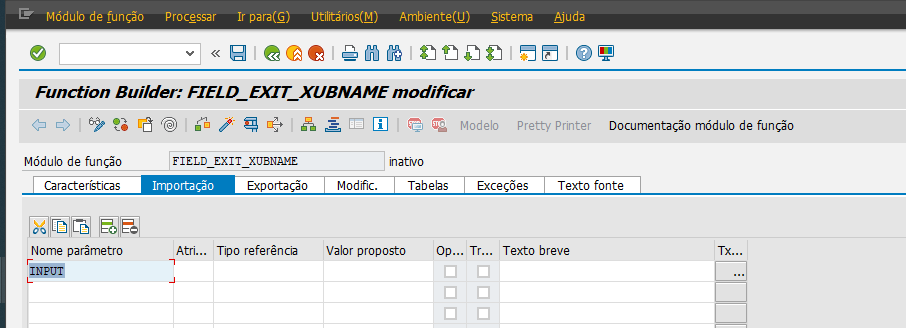
Será exibida a tela abaixo, clique em Criar:



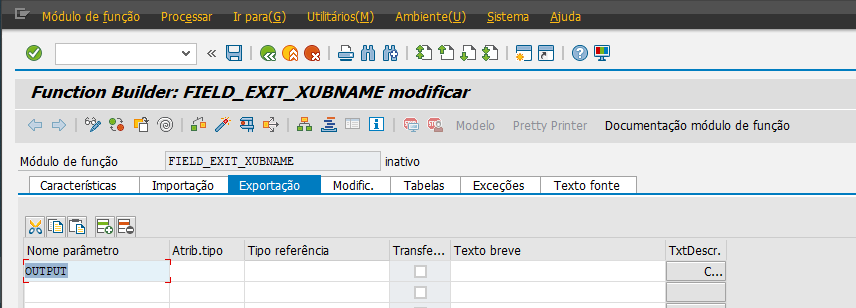
Da mesma forma que fazemos na criação de uma função normal, preencha os dados abaixo incluindo um grupo de função pré existente e clique em Gravar:



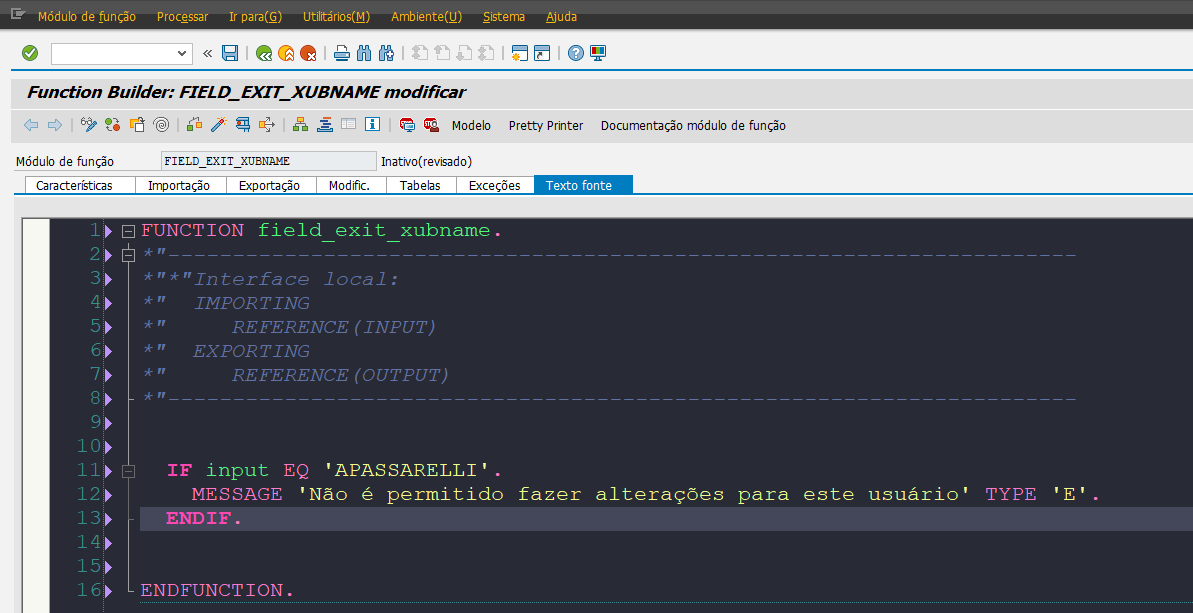
O Parâmetro de Importação terá o campo INPUT, que é o campo que receberá a informação diretamente do campo usuário da SU01:



Assim como o campo Exportação terá o campo OUTPUT que retornará para a tela da SU01, caso a informação seja substituída.



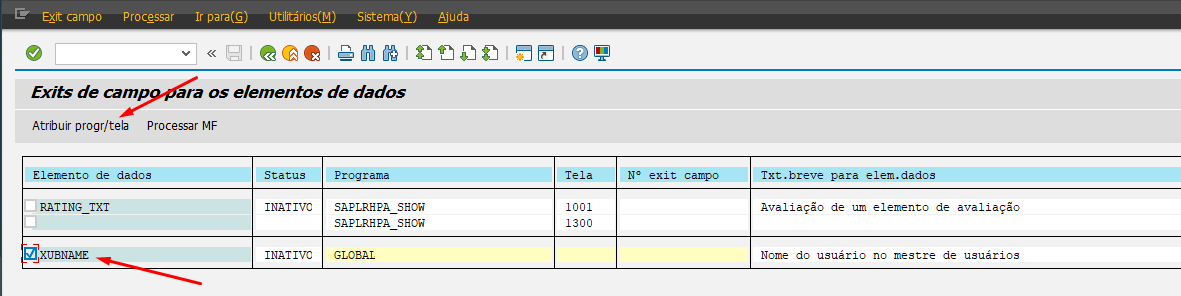
Na aba “Texto Fonte”, vamos inserir nossa lógica, que impedirá que um usuário específico seja alterado na transação SU01.



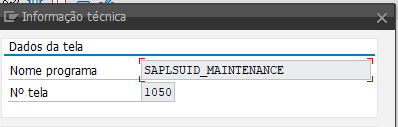
Como estamos utilizando uma Field-Exit, para esse caso podemos usar sem problemas uma mensagem de erro para travar o usuário que tentar fazer a alteração para o usuário exemplo “apassarelli”.

Ative a função normalmente e volte para a tela da CMOD utilizando a PRFB para abrir a atribuição de Field-Exits.

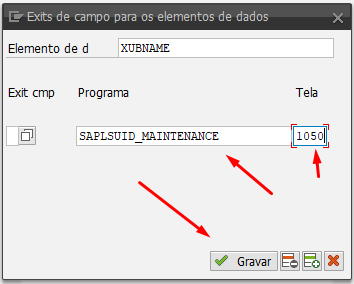
Selecione a linha do Elemento de Dados que foi utilizado para esse processo e clique em “Atribuir progr/tela.



Vamos utilizar ainda as informações técnicas do elemento de dados abertos anteriormente.

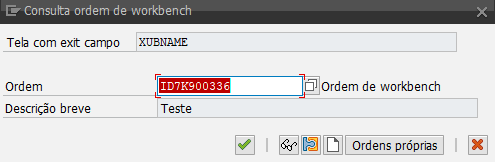


Usamos as informações acima para preencher as atribuições de programa e tela da Field-Exit:

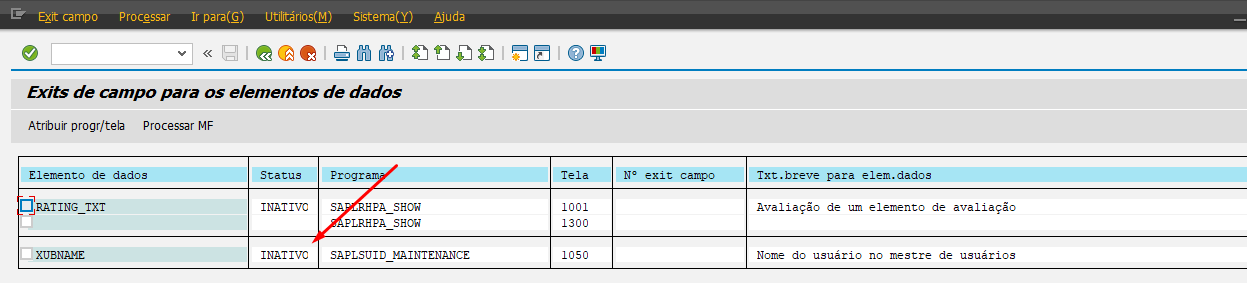


Após informar os dados, clique em “Gravar”.

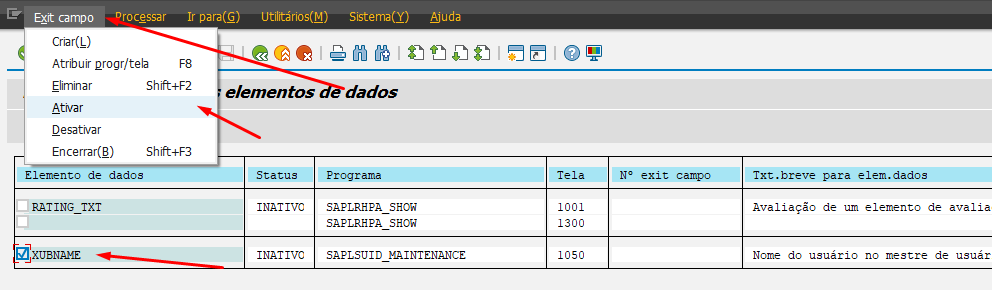
Será solicitada uma request para essa alteração, caso não possua utilize a request abaixo como teste:



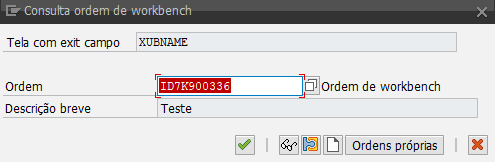
Note que o Status encontra-se ainda Inativo para a Field-Exit que estamos criando:



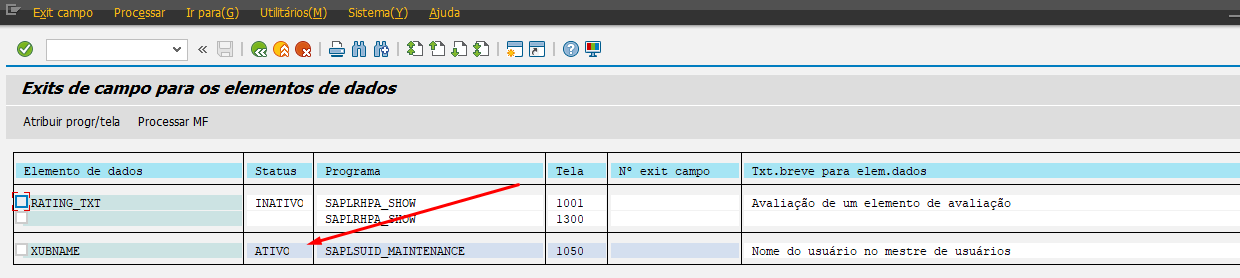
Para ativá-lo, selecione a linha com o Elemento de Dados utilizado para esse exemplo e navegue até o menu, conforme a imagem abaixo:



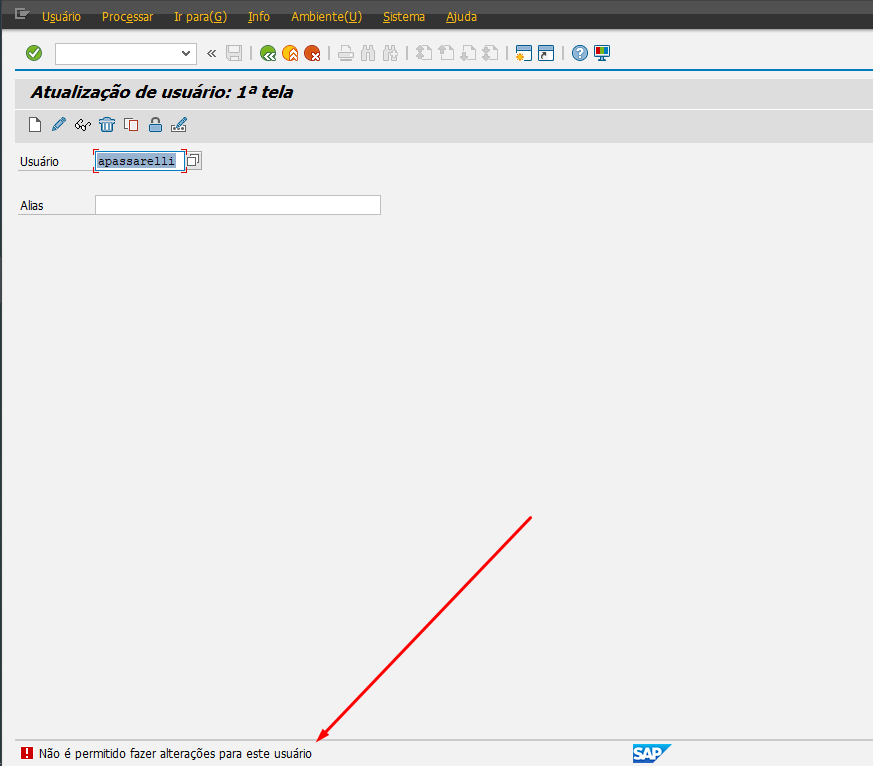
Informe novamente a request para a ativação:



Quando o status mudar para ativo, conforme a imagem abaixo, sua Field-Exit está pronta para ser testada, visto que não temos como depurar o sistema para ver seu funcionamento



Transação SU01, informando o usuário bloqueado por código:



Screen-Exit

BTE