

## Módulo de Função

Um módulo de função é uma unidade de modularização do ABAP/4. Ele é muito semelhante a uma sub-rotina externa sob esses aspectos:

- Ambos existem dentro de um programa externo;
- Ambos ativam os parâmetros a serem passados e retornados;
- Os parâmetros podem ser passados por valor, por valor e resultado, ou por referência.

As diferenças importantes entre módulos de função e sub-rotinas externas são as seguintes:

- Os módulos de função têm uma tela especial utilizada para definir parâmetros – os parâmetros não são definidos via instruções de ABAP/4;
- As áreas de trabalho tables não são compartilhadas entre o módulo de função e o programa de chamada;
- Para chamar um módulo de função, é utilizada uma sintaxe diferente da empregada para chamar uma sub-rotina.

Um nome de módulo de função tem um comprimento mínimo prático de três caracteres e um comprimento máximo de 30 caracteres. Os nomes dos módulos de função criados pelo usuário devem começar com Y ou Z. O nome de cada módulo de função é único dentro do sistema SAP ERP inteiro.

Um grupo de função é um programa que contém módulos de função. Com cada sistema SAP ERP, são fornecidos fornece mais de ;; grupos de função preexistentes. No total, eles contêm mais de ;; módulos de função. Se a funcionalidade que você exige já não é abordada por esses módulos de função fornecidos pela SAP, você também criar seus próprios grupos de função e módulos de função.

Alguns módulos de função tem uma característica especial, chamada “acesso remoto”. Módulos de função com esta característica podem ser chamados a partir de outros softwares. Os módulos de função de acesso remoto (também conhecidos como RFC – Remote Function Call) são muito usados para acessar dados do SAP ERP através da WEB.

### Definindo a interface de módulo de função

Para passar parâmetro para um módulo de função, você deve definir uma interface de módulo de função. Essa interface é a descrição dos parâmetros que são passados e recebidos do módulo de função. Ela é também simplesmente conhecida como interface.

Para definir parâmetros, você deve ir para uma de duas telas de definição de parâmetro:

- Import/Export Parameter Interface;
- Table Parameters/Exceptions Interface

Na tela de Import/Export Parameter Interface você pode definir o seguinte:

- Os parâmetros de importação;
- Os parâmetros de exportação;
- Os parâmetros de alteração;

Na tela de Table Parameters/Exceptions Interface você pode:

- Definir parâmetros internos de tabela;
- Documentar as exceções;

Você digita o nome do parâmetro na primeira coluna e os atributos do parâmetro nas colunas restantes. Digite um parâmetro por linha.

Parâmetros de importação são variáveis ou strings de campo que contêm valores passados no módulo de função do programa de chamada. Esses valores se originam fora do módulo de função e são importados para ele.

Parâmetros de exportação são variáveis ou strings de campo que contêm valores retornados do módulo de função. Esses valores se originam de dentro do módulo de função e são exportados para fora dele.

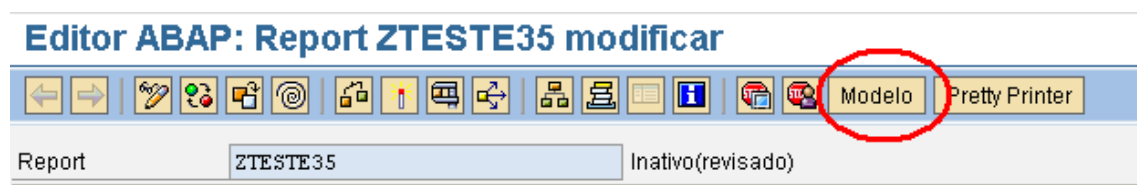
Parâmetros de alteração são variáveis ou strings de campo que contêm valores que são passados no módulo de função, alterados pelo código dentro do módulo de função e então retornados. Esses valores se originam de fora do módulo de função, que os recebe e os passa de volta.

Parâmetros de tabela são tabelas internas que são passadas para o módulo de função, alteradas dentro dele e retornadas. As tabelas internas devem ser definidas no programa que chama.

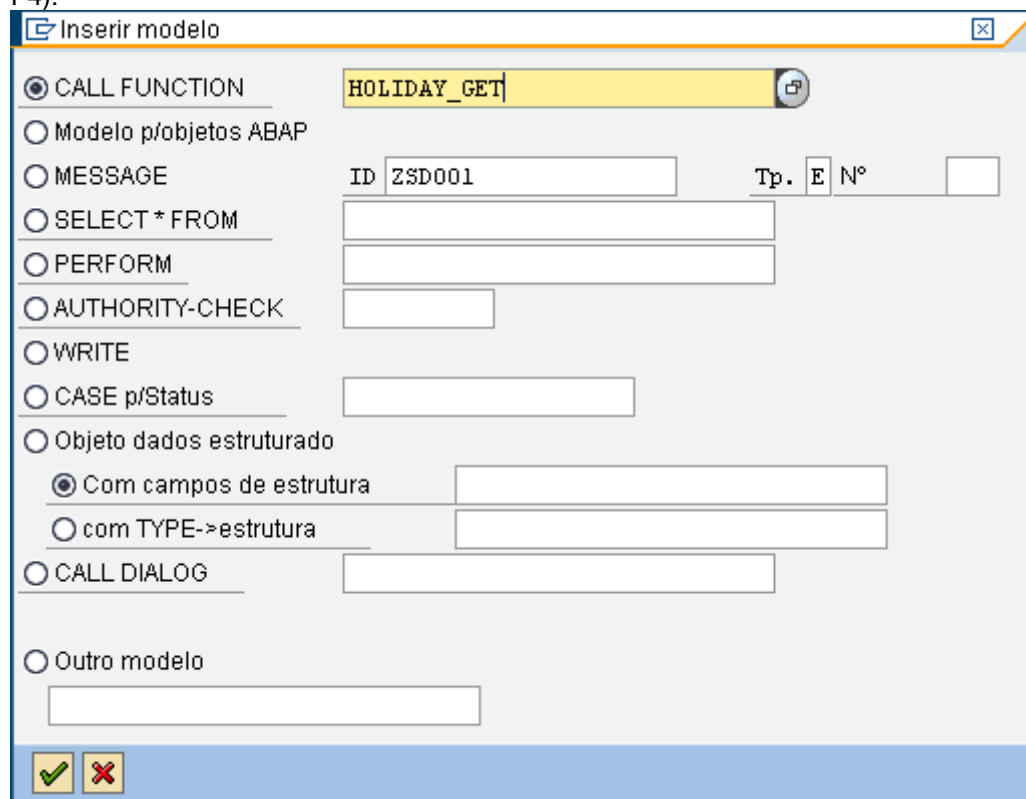
Uma exceção é um nome para um erro que ocorre dentro de um módulo de função. As exceções são descritas em detalhe na próxima seção.

### Usando uma função

A forma mais fácil de usar uma função é chamá-la dentro de um programa usando o botão “Modelo”, conforme a figura abaixo.



Informe em seguida o nome do módulo de função ou use o search-help no campo abaixo (tecla F4):



The 'Inserir modelo' dialog box is shown. It has a title bar with a search icon and a close button. The main area contains several radio buttons and input fields. The 'CALL FUNCTION' option is selected, and the text 'HOLIDAY\_GET' is entered in the adjacent field. Other options include 'Modelo p/objetos ABAP', 'MESSAGE', 'SELECT \* FROM', 'PERFORM', 'AUTHORITY-CHECK', 'WRITE', 'CASE p/Status', 'Objeto dados estruturado' (with sub-options 'Com campos de estrutura' and 'com TYPE->estrutura'), 'CALL DIALOG', and 'Outro modelo'. At the bottom, there are checkmark and cross icons.

Após teclar Enter, o código de chamada é inserido no seu programa.

```

13
14     CALL FUNCTION 'HOLIDAY_GET'
15     * EXPORTING
16     *     HOLIDAY_CALENDAR           = ' '
17     *     FACTORY_CALENDAR           = ' '
18     *     DATE_FROM                   = SY-DATUM
19     *     DATE_TO                     = SY-DATUM
20     * IMPORTING
21     *     YEAR_OF_VALID_FROM          =
22     *     YEAR_OF_VALID_TO            =
23     *     RETURNCODE                  =
24     TABLES
25     *     holidays                    =
26     * EXCEPTIONS
27     *     FACTORY_CALENDAR_NOT_FOUND  = 1
28     *     HOLIDAY_CALENDAR_NOT_FOUND  = 2
29     *     DATE_HAS_INVALID_FORMAT      = 3
30     *     DATE_INCONSISTENCY           = 4
31     *     OTHERS                      = 5
32     .
33     IF sy-subrc <> 0.
34     * MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
35     *     WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
36     ENDIF.
37

```

Observe que nem todos os parâmetros são obrigatórios, mas os parâmetros que você utilizar DEVEM ter o mesmo tipo de dado e tamanho definidos na função. Use a transação SE37 para visualizar os parâmetros:

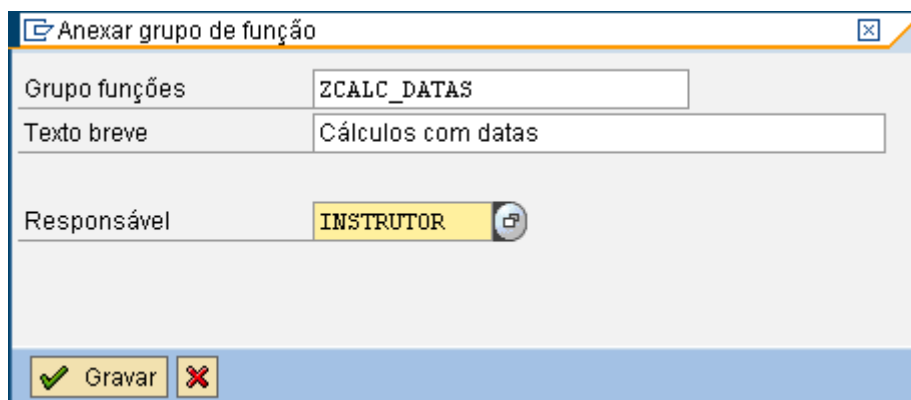
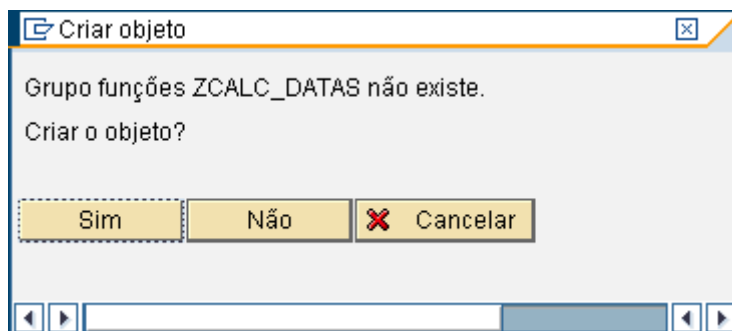
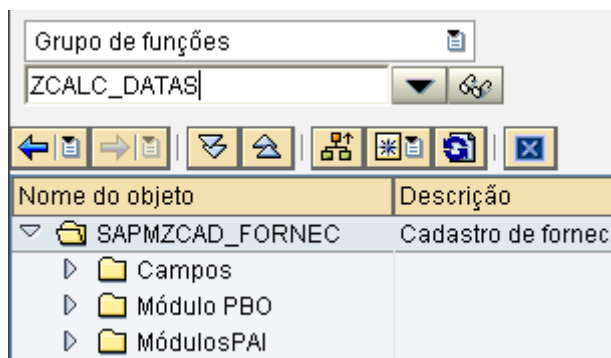
```

FUNCTION HOLIDAY_GET.
*-----
*""Lokale Schnittstelle:
*   IMPORTING
*       VALUE(HOLIDAY_CALENDAR) LIKE SCAL-HCALID DEFAULT SPACE
*       VALUE(FACTORY_CALENDAR) LIKE SCAL-FCALID DEFAULT SPACE
*       VALUE(DATE_FROM) LIKE SCAL-DATE DEFAULT SY-DATUM
*       VALUE(DATE_TO) LIKE SCAL-DATE DEFAULT SY-DATUM
*   EXPORTING
*       VALUE(YEAR_OF_VALID_FROM) LIKE SCAL-YEAR
*       VALUE(YEAR_OF_VALID_TO) LIKE SCAL-YEAR
*       VALUE(RETURNCODE) LIKE SY-SUBRC
*   TABLES
*       HOLIDAYS STRUCTURE ISCAL_DAY
*   EXCEPTIONS
*       FACTORY_CALENDAR_NOT_FOUND
*       HOLIDAY_CALENDAR_NOT_FOUND
*       DATE_HAS_INVALID_FORMAT
*       DATE_INCONSISTENCY
*-----

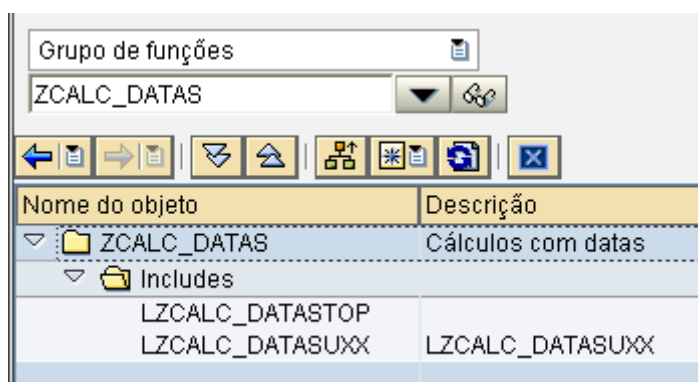
```

### Criando um módulo de função

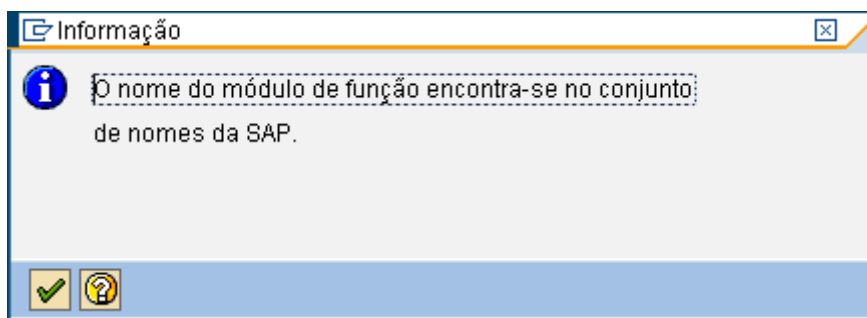
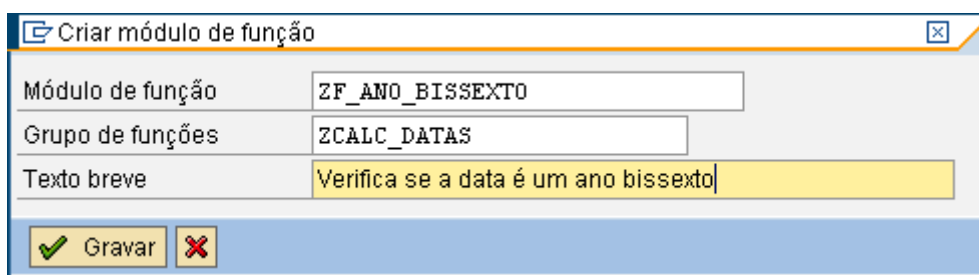
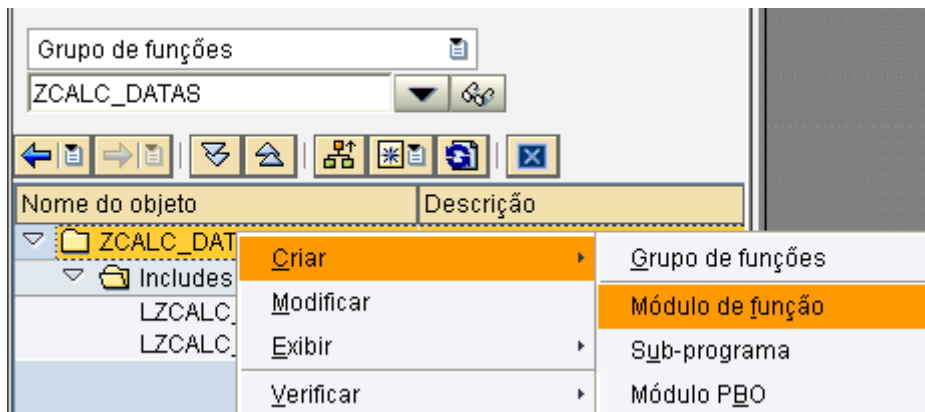
O primeiro passo para criar um módulo de função é a criação do grupo de função: um programa especialmente criado para agrupar módulos de função. Os módulos de função de um mesmo grupo podem compartilhar uma área de dados comum e um grupo de rotinas. Use a transação SE80 e selecione "Grupo de funções". Veja o exemplo:



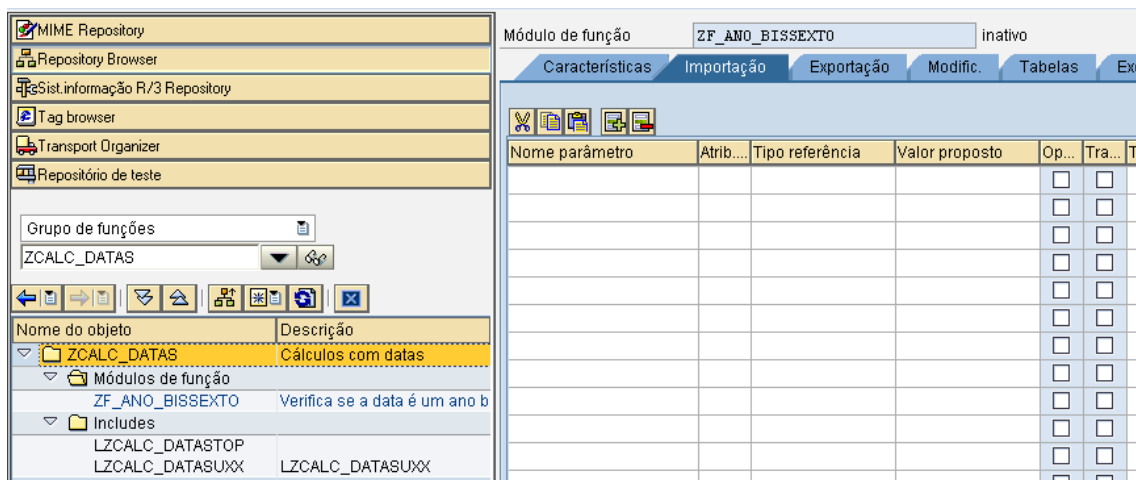
Após a criação do grupo, veja a árvore de includes criada. Na include LZCALC\_DATASTOP colocamos as variáveis comuns a todas as funções.



Para criar o módulo de função, use o menu de contexto (botão direito do mouse) sobre o nome do grupo de função e selecione o módulo de função conforme mostrado:



Veja como ficou a árvore de objetos:



Informe o nome do parâmetro de entrada e o tipo:

Módulo de função		ZF_ANO_BISSEXTO		Inativo(revisado)	
Características		Importação		Exportação	
Nome parâmetro	Atrib...	Tipo referência	Valor proposto	Opci...	Tra...
I_DATA	TYPE	SY-DATUM		<input type="checkbox"/>	<input type="checkbox"/>

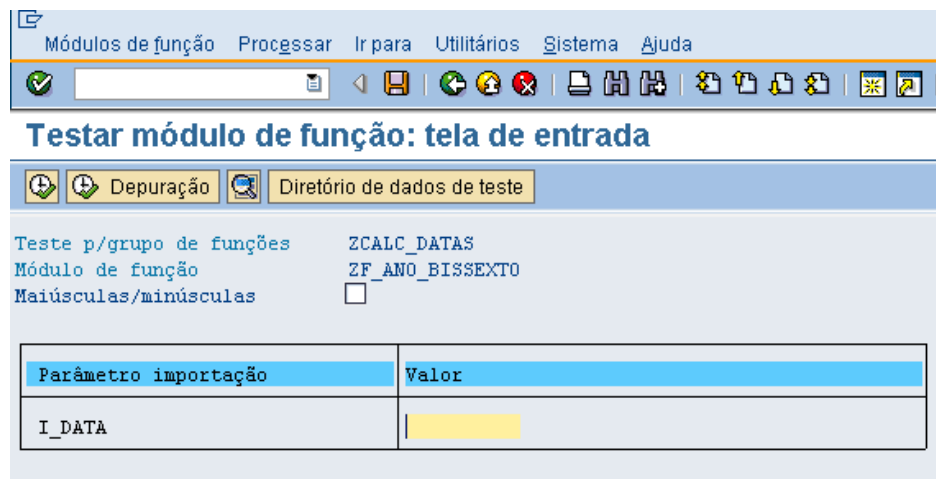
Informe o parâmetro de saída:

Módulo de função		ZF_ANO_BISSEXTO		Inativo(revisado)	
Características		Importação		Exportação	
Nome parâmetro	Atrib.tipo	Tipo referência	Transfer...	Texto bre	
ANO_BISSEXTO	TYPE	CHAR1	<input type="checkbox"/>	Código di	

Use o botão direito sobre o nome do módulo de função e ative-o. Observe que o cabeçalho do módulo de função é criado pela SAP, não altere-o.

Módulo de função		ZF_ANO_BISSEXTO		ativo	
Características		Importação		Exportação	
<pre> 1  FUNCTION ZF_ANO_BISSEXTO. 2  *----- 3  *""Interface local: 4  *  IMPORTING 5  *    REFERENCE(I_DATA) TYPE SY-DATUM 6  *  EXPORTING 7  *    REFERENCE(ANO_BISSEXTO) TYPE CHAR1 8  *----- 9 10 11 12 13 14  ENDFUNCTION. </pre>					

Escreva agora o código da função:



Tecla F8 para executar a função:

## Testar módulo de função: tela de resultado

Parâmetro importação	Valor
I_DATA	10.12.2008

Parâms.export.	Valor
ANO_BISSEXTO	X

### Definindo dados dentro de um módulo de função

As definições de dados dentro de módulos de função são semelhantes àquelas de sub-rotinas. Dentro de um módulo de função, utilize a instrução `data` para definir variáveis locais que são reinicializadas toda vez que o módulo de função é chamado. Utilize a instrução ***statics*** para definir variáveis locais que são alocadas na primeira vez que o módulo de função é chamado. O valor de uma variável estática é lembrado entre chamadas. Defina parâmetros dentro da interface de módulos de função para criar definições locais de variáveis que são passadas no módulo de função e retornadas dele. Você não pode utilizar a instrução local dentro de um módulo de função. Em vez disso, parâmetros de interface globalizados servem para o mesmo propósito.

### Passando parâmetros

Os métodos para passar parâmetros para módulos de função são muito semelhantes àqueles para passar parâmetros para sub-rotinas externas.

Por padrão:

- Os parâmetros de exportação e importação são passados por valor;
- Os parâmetros de alteração são passados por valor e resultado;
- As tabelas internas são passadas por referência;

Você pode fazer com que a importação, exportação e parâmetros de alteração sejam passados por referência, colocando uma marca de verificação.

### Subrotinas - PERFORM

Para criar uma subrotina em um módulo de função, proceda da mesma forma como se estivesse em um programa report: insira a chamada `PERFORM MINHA_ROTINA` e dê um clique duplo no nome da rotina. Diferente do programa report, crie sempre um include para as suas rotinas. O include será automaticamente inserido no programa do grupo de função. Veja o exemplo: primeiro escreva a chamada e dê um clique duplo no nome da rotina:

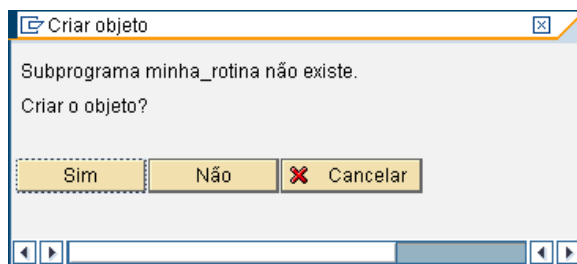


```

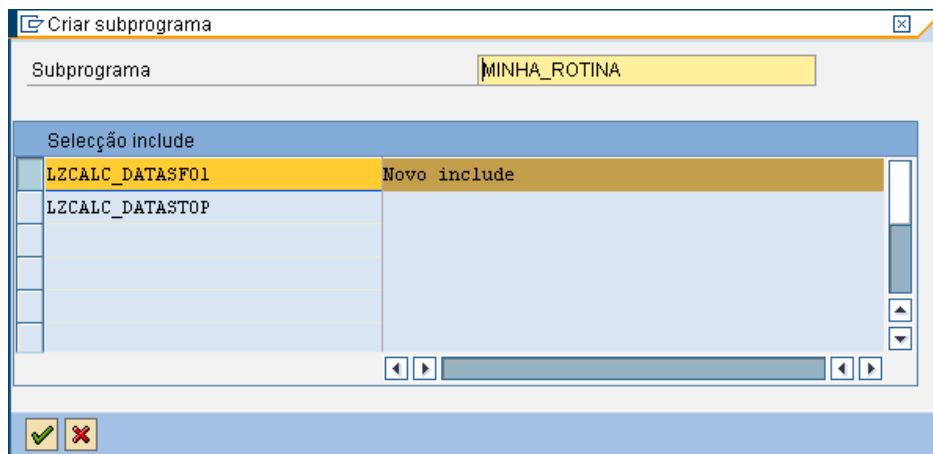
14      INTO l_data.
15      l_data = l_data - 1. " Último dia
16      IF l_data+6(2) = '29'.
17          ano_bissexto = 'X'.
18      ELSE.
19          CLEAR ano_bissexto.
20      ENDIF.
21
22      PERFORM minha_rotina.
23
24  ENDFUNCTION.

```

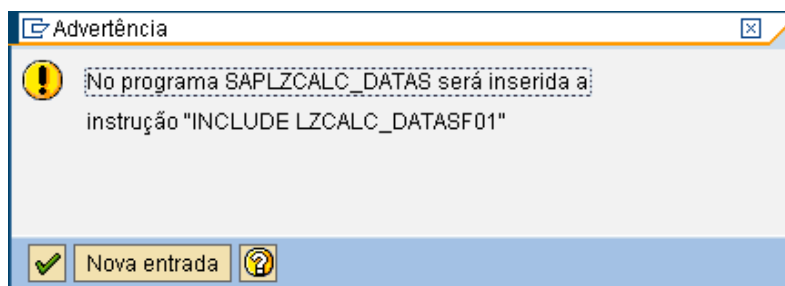
Confirme a criação, acionando o botão “Sim”:



Confirme o nome da rotina:



O SAP avisa que colocará o include no seu programa do grupo de função:



Se a função ainda não foi gravada o SAP pede uma confirmação:

**Editor ABAP: Include LZCALC\_DATASF01 modificar**

