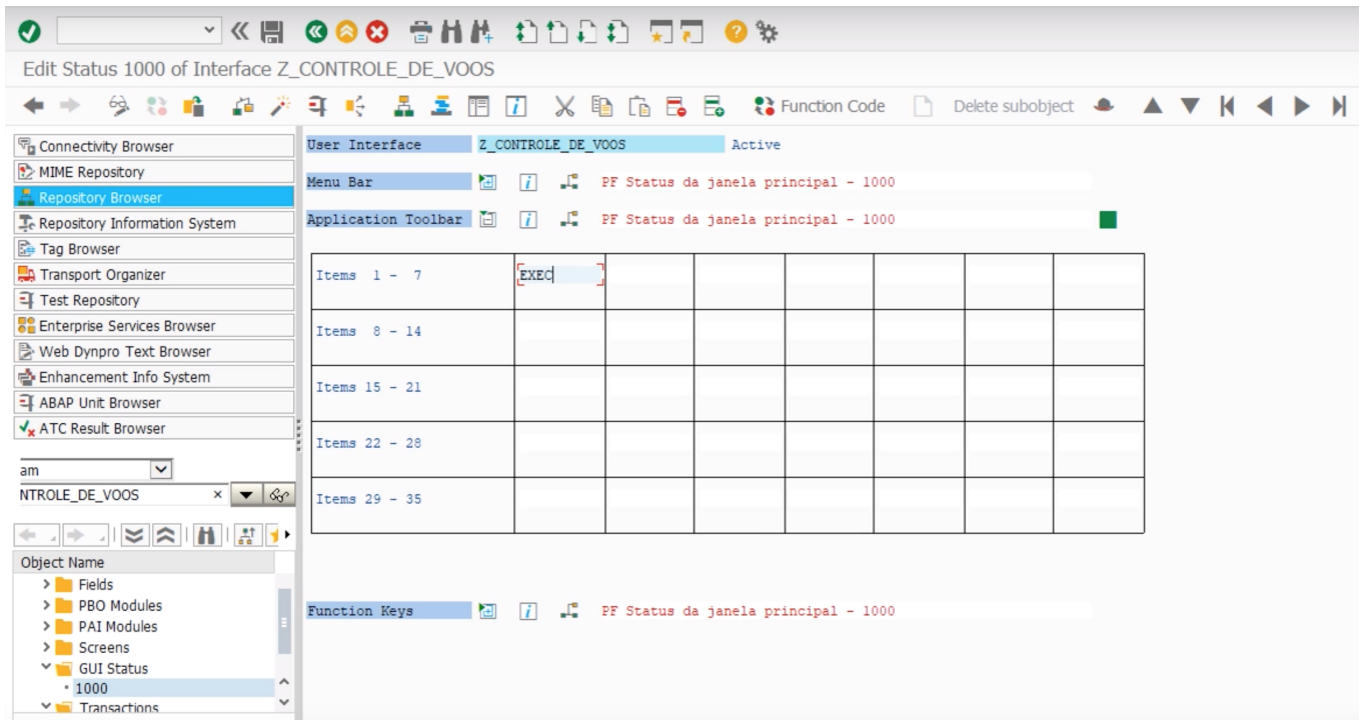


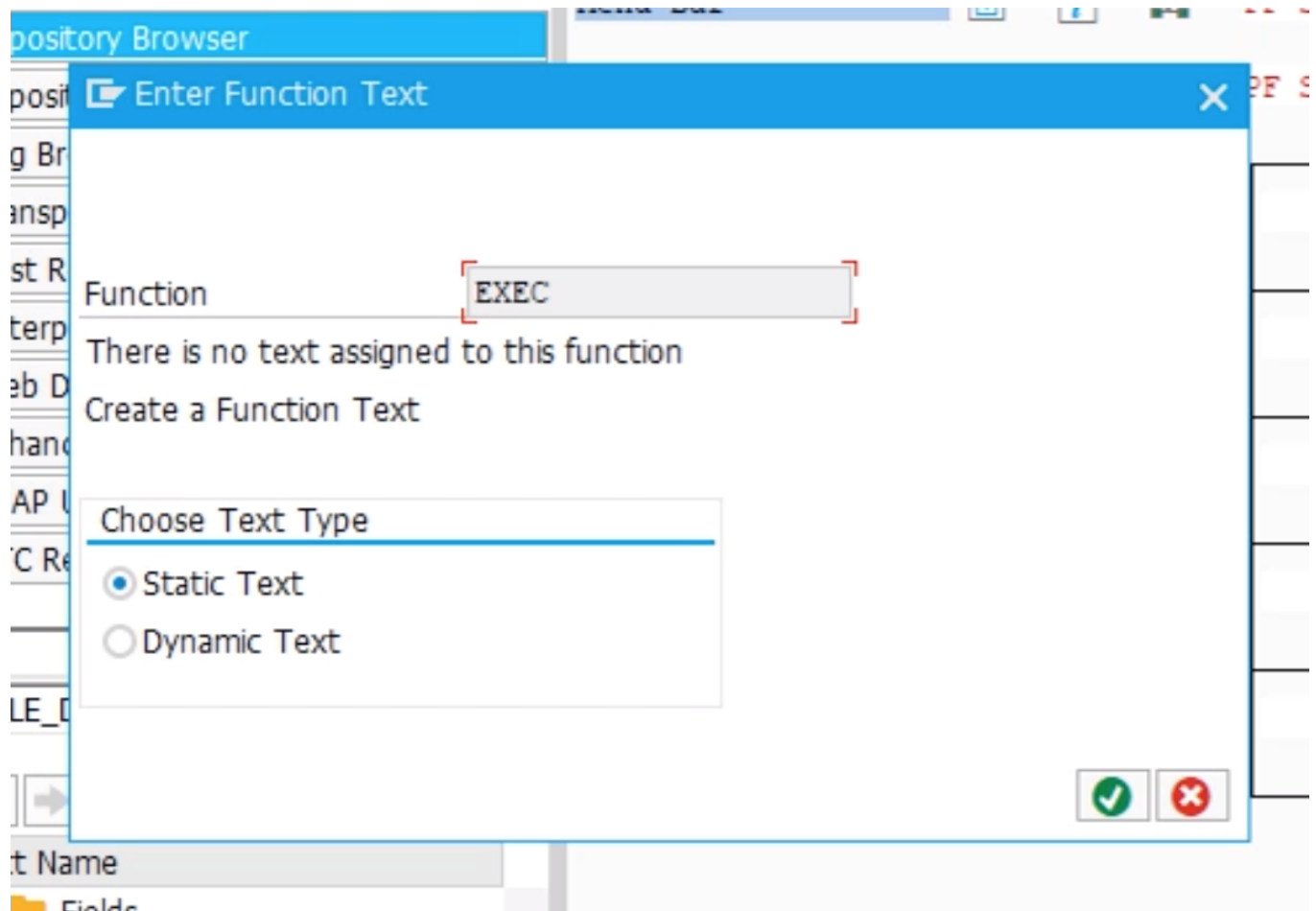
Faça o que eu fiz na aula

Nossa missão agora é preencher o nosso relatório com dados provenientes do banco de dados. Para isso, vamos inserir o botão responsável por executar a seleção no banco de dados.

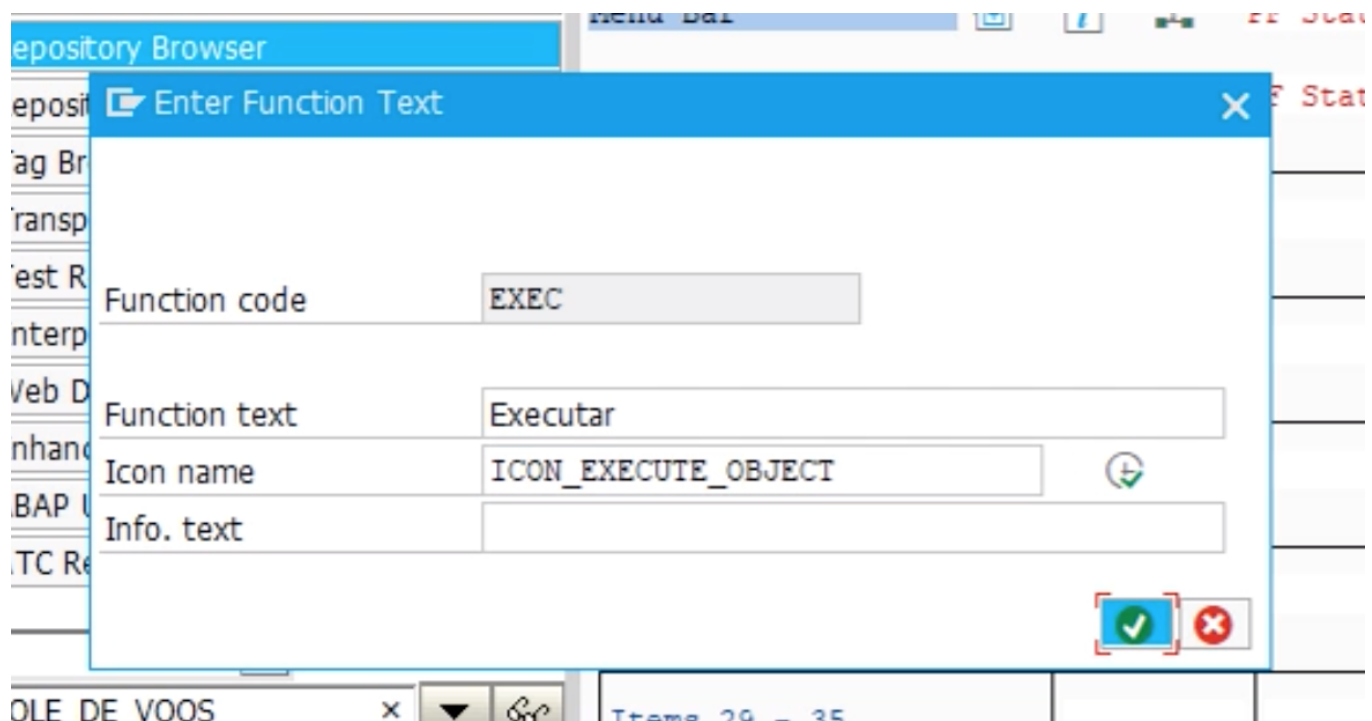
Primeiro, vamos voltar para o PF-STATUS 1000, navegando via GUI Status. Então, utilizando a opção **Application Toolbar**, vamos inserir o valor **EXEC**.



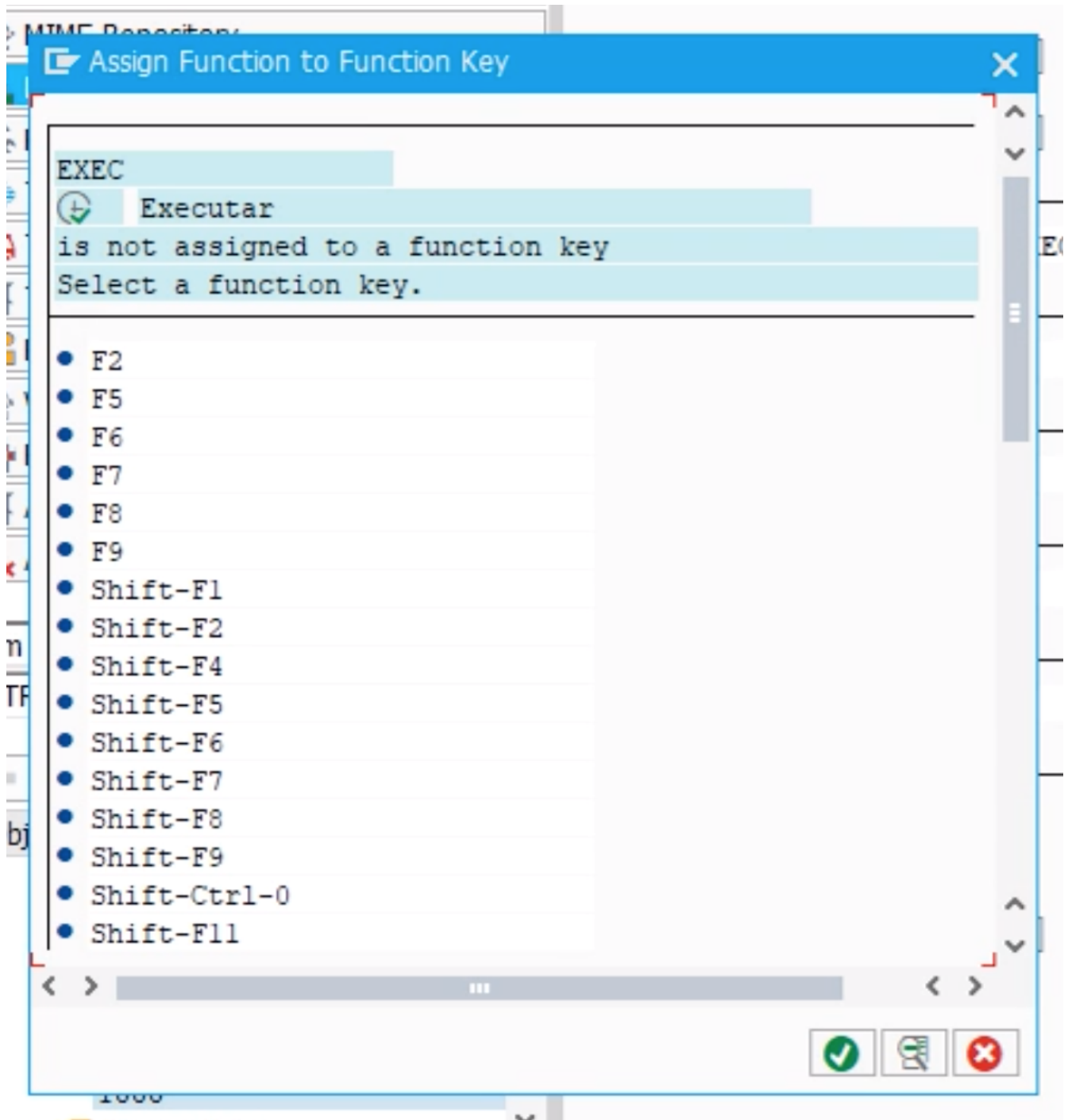
Ao pressionar **ENTER**, será exibida uma janela de confirmação para saber se vamos trabalhar com um valor estático ou dinâmico. Nesse caso vamos trabalhar com o valor estático.



Após confirmar uma nova janela será exibida, então informaremos o nome do botão e um ícone para sua representação.

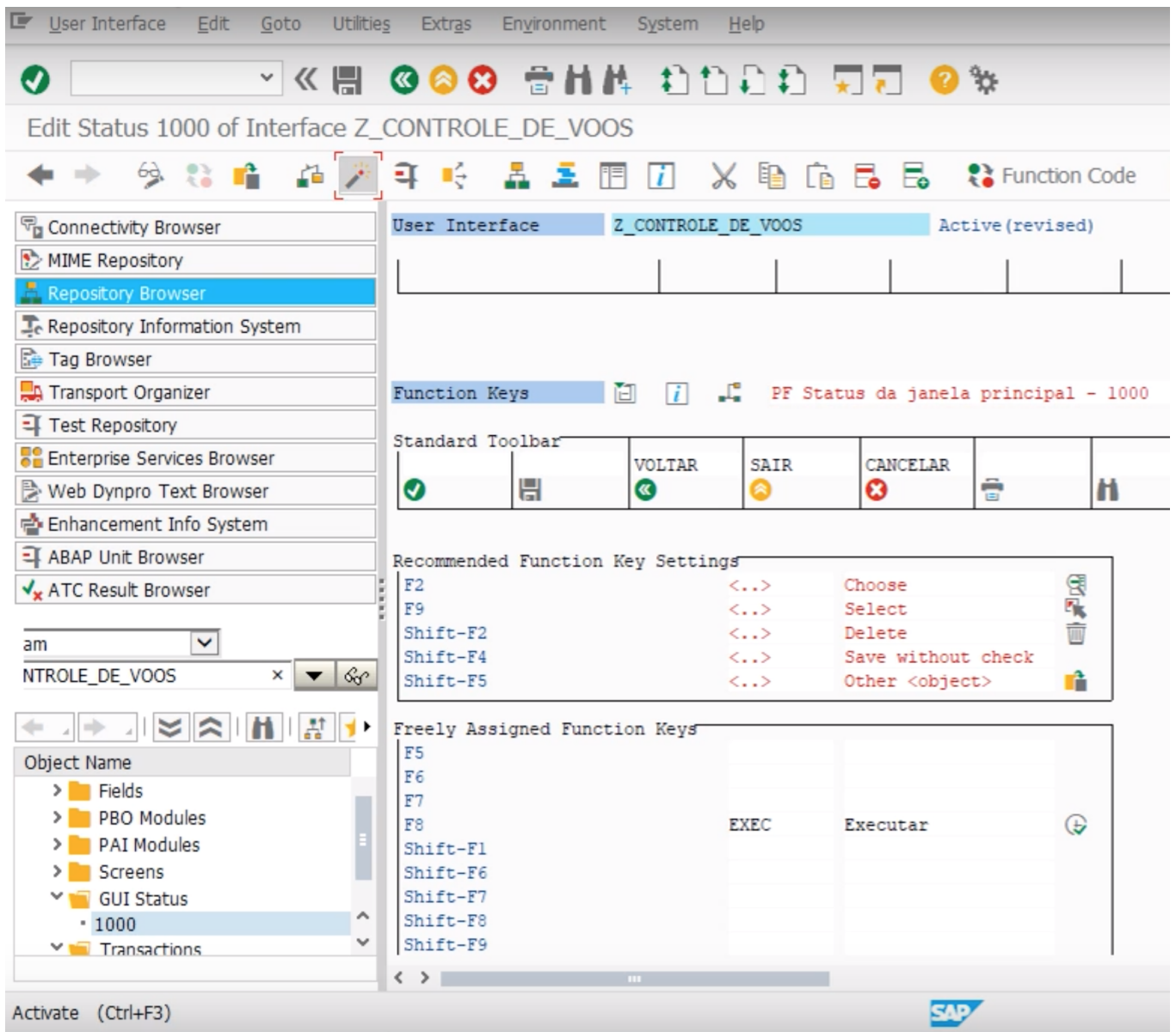


Após confirmação da uma última configuração, o SAP nos solicita que indiquemos uma tecla de atalho para este botão. Vamos utilizar a opção **F8**.

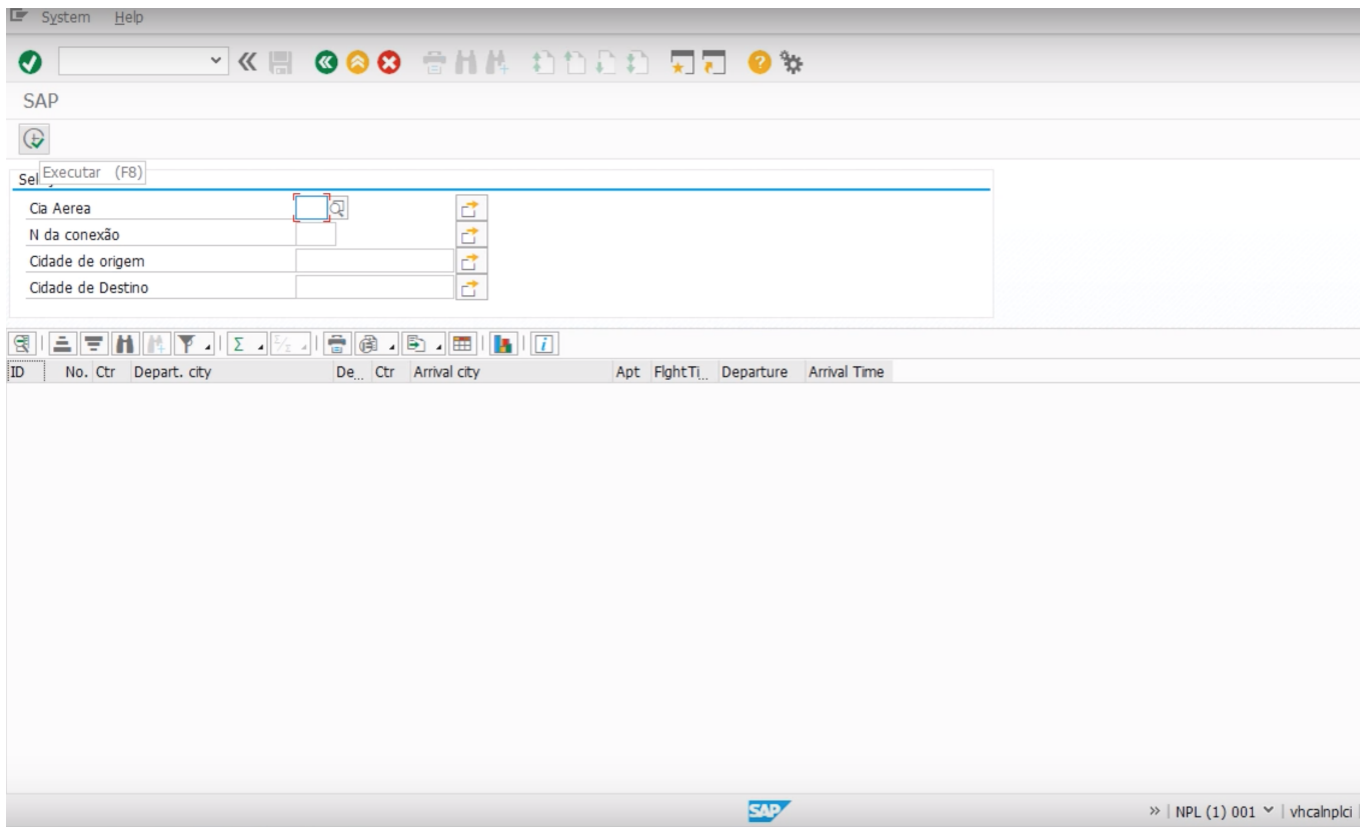


Vamos ativar o nosso código e então confirmar se o botão está aparecendo na nossa aplicação.

Para isso vamos clicar no botão com o ícone de ativação do PF-STATUS , e após a verificação de sintaxe que acontece de forma automática, teremos a ativação do PF-STATUS .



Executando a nossa aplicação, veremos a nossa tela com um novo botão na nossa janela.



Vamos implementar uma nova ação para este botão. Vamos refatorar o nosso código quebrando-o em seções, e então podemos implementar essa nova ação.

Primeiro vamos alterar o fluxo lógico da nossa aplicação com o seguinte trecho de código:

```
PROCESS BEFORE OUTPUT.
```

```
MODULE pbo .
```

```
CALL SUBSCREEN subscreen_selecao INCLUDING sy-repid '9999'.
```

```
*
```

```
PROCESS AFTER INPUT.
```

```
MODULE pai.
```

```
CALL SUBSCREEN subscreen_selecao .
```

E após a alteração do nosso fluxo lógico, vamos quebrar o nosso código em seções. Para isso, será necessário criar quatro INCLUDES :

```
INCLUDE z_controle_de_voos_declaracoes.
```

```
INCLUDE z_controle_de_voos_selecao.
```

```
INCLUDE z_controle_de_voos_pbo.
```

```
INCLUDE z_controle_de_voos_pai.
```

Dentro da seção `z_controle_de_voos_declaracoes` vamos implementar o seguinte trecho de código:

```
*&-----*
*& Include          Z_CONTROLE_DE_VOOS_DECLARACOES
*&-----*
```

```
DATA spfli TYPE spfli .

DATA: custom_container TYPE REF TO cl_gui_custom_container,
      alv               TYPE REF TO cl_gui_alv_grid,
      fieldcat          TYPE lvc_t_fcat.

DATA it_spfli TYPE STANDARD TABLE OF spfli .
```

Dentro da seção `z_controle_de_voos_selecao` vamos implementar o seguinte trecho de código:

```
*&-----*
*& Include      Z_CONTROLE_DE_VOOS_SELECAO
*&-----*

SELECTION-SCREEN: BEGIN OF SCREEN 9999 AS SUBSCREEN .
SELECTION-SCREEN: BEGIN OF BLOCK bloco1 WITH FRAME TITLE TEXT-001 .
SELECT-OPTIONS s_carrid FOR spfli-carrid NO INTERVALS.
SELECT-OPTIONS s_connid FOR spfli-connid NO INTERVALS.
SELECT-OPTIONS s_cityfr FOR spfli-cityfrom NO INTERVALS.
SELECT-OPTIONS s_cityto FOR spfli-cityto NO INTERVALS.

SELECTION-SCREEN: END OF BLOCK bloco1.
SELECTION-SCREEN: END OF SCREEN 9999 .
```

Dentro da seção `z_controle_de_voos_pbo` vamos implementar o seguinte trecho de código:

```
*&-----*
*& Include      Z_CONTROLE_DE_VOOS_PBO
*&-----*

MODULE pbo OUTPUT.

DATA coluna LIKE LINE OF fieldcat .

SET PF-STATUS '1000'.

IF custom_container IS NOT BOUND .

  CREATE OBJECT custom_container
    EXPORTING
      container_name = 'CC_ALV'.

  CREATE OBJECT alv
    EXPORTING
      i_parent = custom_container.

  CALL FUNCTION 'LVC_FIELDCATALOG_MERGE'
    EXPORTING
      i_structure_name = 'SPFLI'
    CHANGING
      ct_fieldcat      = fieldcat.
```

```
LOOP AT fieldcat INTO coluna .
```

```
IF coluna-fieldname = 'DISTANCE'
  OR coluna-fieldname = 'DISTID'
  OR coluna-fieldname = 'FLTYPE'
  OR coluna-fieldname = 'PERIOD'.
```

```
coluna-no_out = 'X'.
MODIFY fieldcat FROM coluna .
```

```
ENDIF .
```

```
ENDLOOP .
```

```
CALL METHOD alv->set_table_for_first_display
  CHANGING
    it_outtab      = it_spfli      " Output Table
    it_fieldcatalog = fieldcat.    " Field Catalog
```

```
ENDIF .
```

```
CALL METHOD alv->refresh_table_display( ) .
```

```
ENDMODULE.
```

Dentro da seção `z_controle_de_voos_pai` vamos implementar a ação de seleção do nosso relatório junto com as ações dos botões de controle de tela da nossa aplicação:

```
*&-----*
*& Include      Z_CONTROLE_DE_VOOS_PAI
*&-----*
```

```
MODULE pai INPUT.
```

```
CASE sy-ucomm.
  WHEN 'VOLTAR' OR 'CANCELAR' OR 'SAIR'.
    LEAVE PROGRAM.
  WHEN 'EXEC'.
    SELECT *
      FROM spfli
     INTO TABLE it_spfli
    WHERE carrid IN s_carrid
      AND connid IN s_connid
      AND cityfrom IN s_cityfr
      AND cityto IN s_cityto .
ENDCASE.
```

```
ENDMODULE.
```

Código completo

```
PROGRAM z_controle_de_voos.
```

```
INCLUDE z_controle_de_voos_declaracoes.
```

```
INCLUDE z_controle_de_voos_selecao.
INCLUDE z_controle_de_voos_pbo.
INCLUDE z_controle_de_voos_pai.

*&-----*
*& Include          Z_CONTROLE_DE_VOOS_DECLARACOES
*&-----*

DATA spfli TYPE spfli .

DATA: custom_container TYPE REF TO cl_gui_custom_container,
      alv              TYPE REF TO cl_gui_alv_grid,
      fieldcat         TYPE lvc_t_fcat.

DATA it_spfli TYPE STANDARD TABLE OF spfli .

*&-----*
*& Include          Z_CONTROLE_DE_VOOS_SELECAO
*&-----*

SELECTION-SCREEN: BEGIN OF SCREEN 9999 AS SUBSCREEN .
SELECTION-SCREEN: BEGIN OF BLOCK bloco1 WITH FRAME TITLE TEXT-001 .

SELECT-OPTIONS s_carrid FOR spfli-carrid NO INTERVALS.
SELECT-OPTIONS s_connid FOR spfli-connid NO INTERVALS.
SELECT-OPTIONS s_cityfr FOR spfli-cityfrom NO INTERVALS.
SELECT-OPTIONS s_cityto FOR spfli-cityto NO INTERVALS.

SELECTION-SCREEN: END OF BLOCK bloco1.
SELECTION-SCREEN: END OF SCREEN 9999 .

*&-----*
*& Include          Z_CONTROLE_DE_VOOS_PBO
*&-----*

MODULE pbo OUTPUT.

  DATA coluna LIKE LINE OF fieldcat .

  SET PF-STATUS '1000'.

  IF custom_container IS NOT BOUND .

    CREATE OBJECT custom_container
      EXPORTING
        container_name = 'CC_ALV'.

    CREATE OBJECT alv
      EXPORTING
        i_parent = custom_container.

    CALL FUNCTION 'LVC_FIELDCATALOG_MERGE'
      EXPORTING
        i_structure_name = 'SPFLI'
      CHANGING
```

```
ct_fieldcat      = fieldcat.
```

```
LOOP AT fieldcat INTO coluna .
```

```
IF coluna-fieldname = 'DISTANCE'  
  OR coluna-fieldname = 'DISTID'  
    OR coluna-fieldname = 'FLTYPE'  
      OR coluna-fieldname = 'PERIOD'.
```

```
coluna-no_out = 'X'.  
MODIFY fieldcat FROM coluna .
```

```
ENDIF .
```

```
ENDLOOP .
```

```
CALL METHOD alv->set_table_for_first_display  
  CHANGING  
    it_outtab      = it_spfli      " Output Table  
    it_fieldcatalog = fieldcat.    " Field Catalog
```

```
ENDIF .
```

```
CALL METHOD alv->refresh_table_display( ) .
```

```
ENDMODULE.
```

```
*&-----*  
*& Include      Z_CONTROLE_DE_VOOS_PAI  
*&-----*
```

```
MODULE pai INPUT.
```

```
CASE sy-ucomm.  
  WHEN 'VOLTAR' OR 'CANCELAR' OR 'SAIR'.  
    LEAVE PROGRAM.  
  WHEN 'EXEC'.  
    SELECT *  
      FROM spfli  
      INTO TABLE it_spfli  
      WHERE carrid IN s_carrid  
        AND connid IN s_connid  
        AND cityfrom IN s_cityfr  
        AND cityto IN s_cityto .  
ENDCASE.
```

```
ENDMODULE.
```