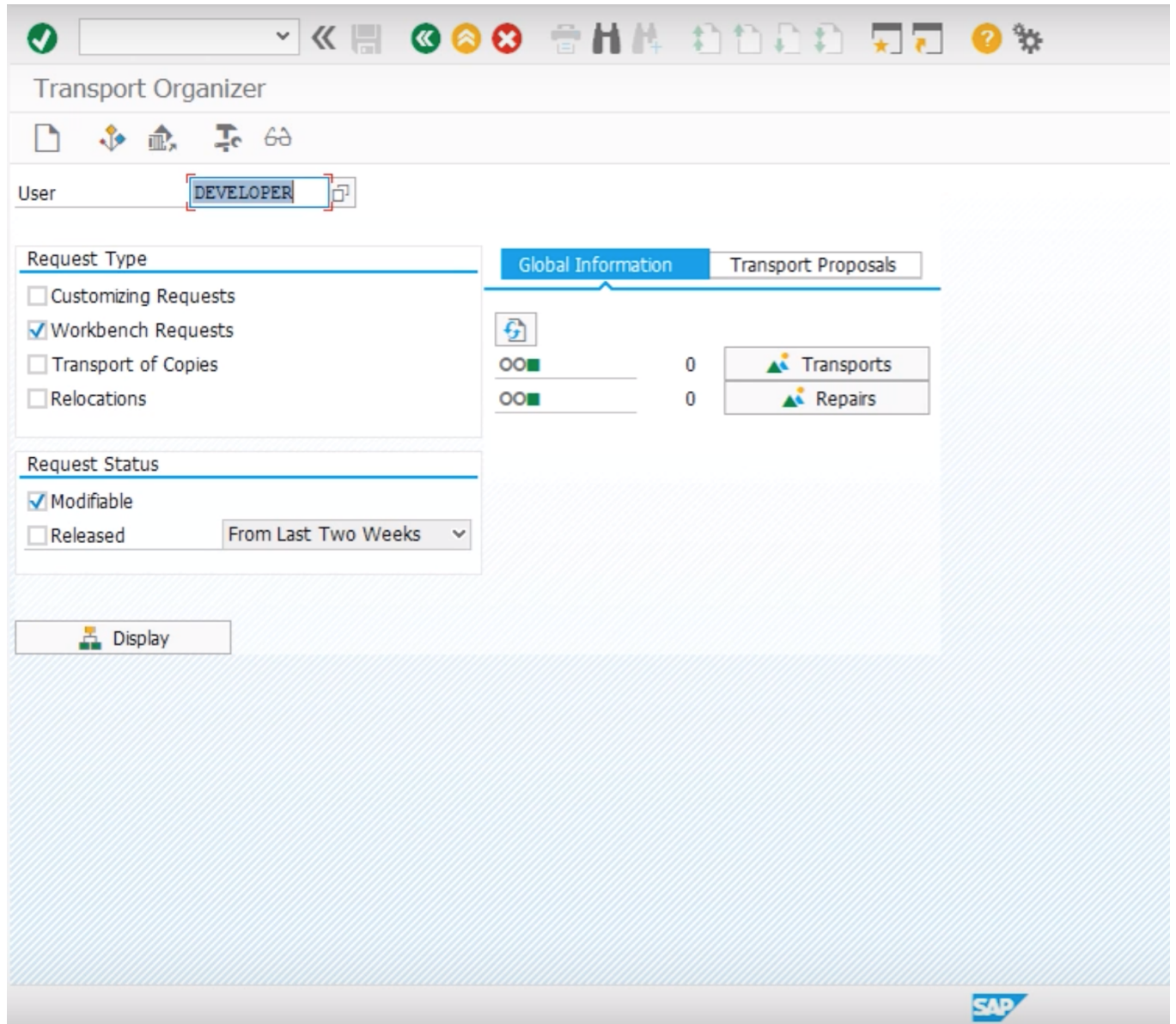


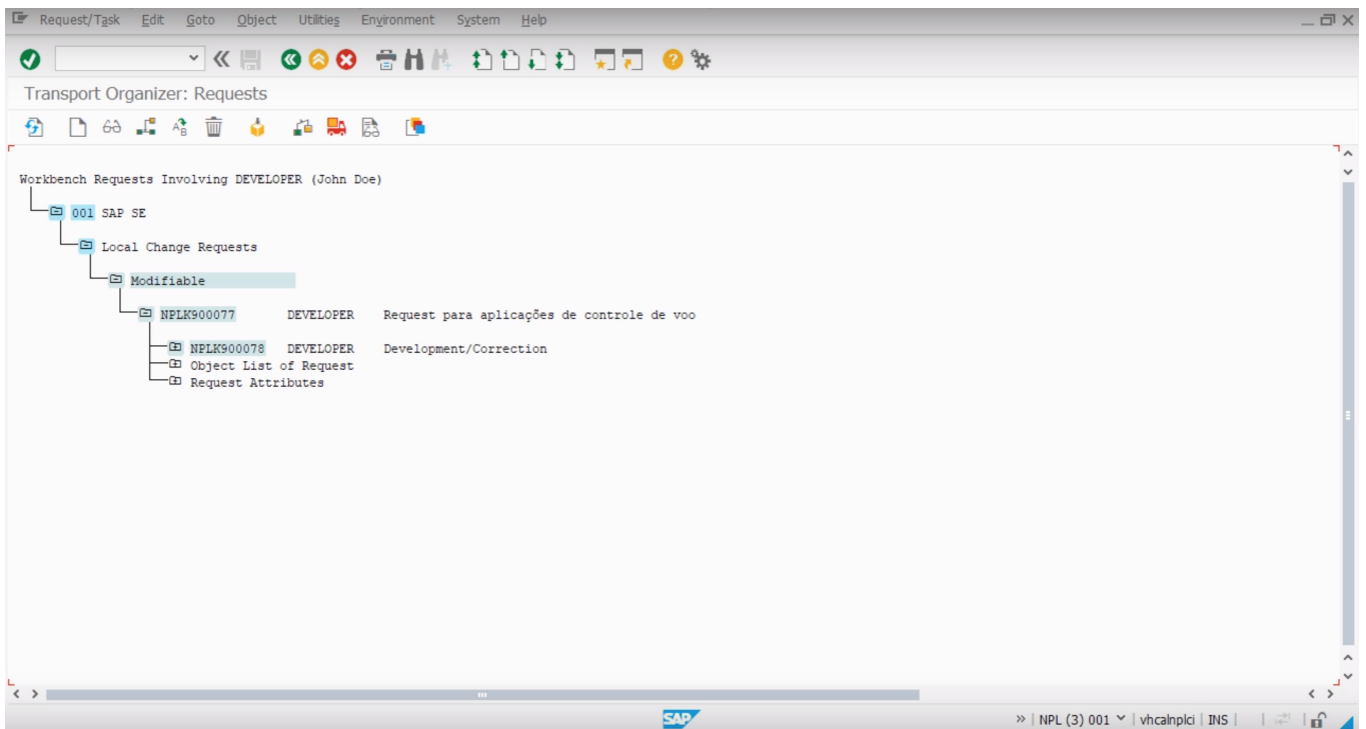
Faça o que eu fiz na aula

Agora vamos liberar a nossa aplicação para o próximo ambiente. Para isso, vamos acessar a transação SE10.

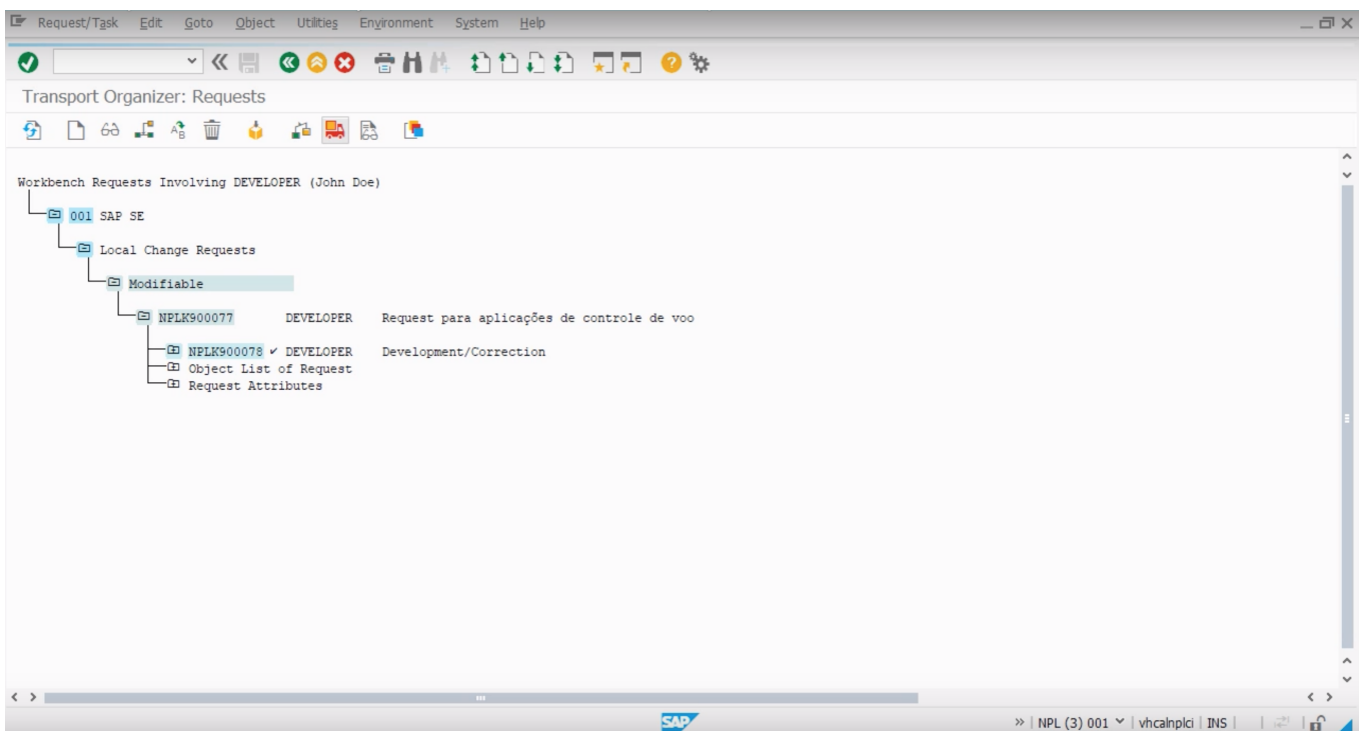
Vamos acessar a nossa request clicando em **Display** com os campos **Workbench** e **Modifiable** selecionados.



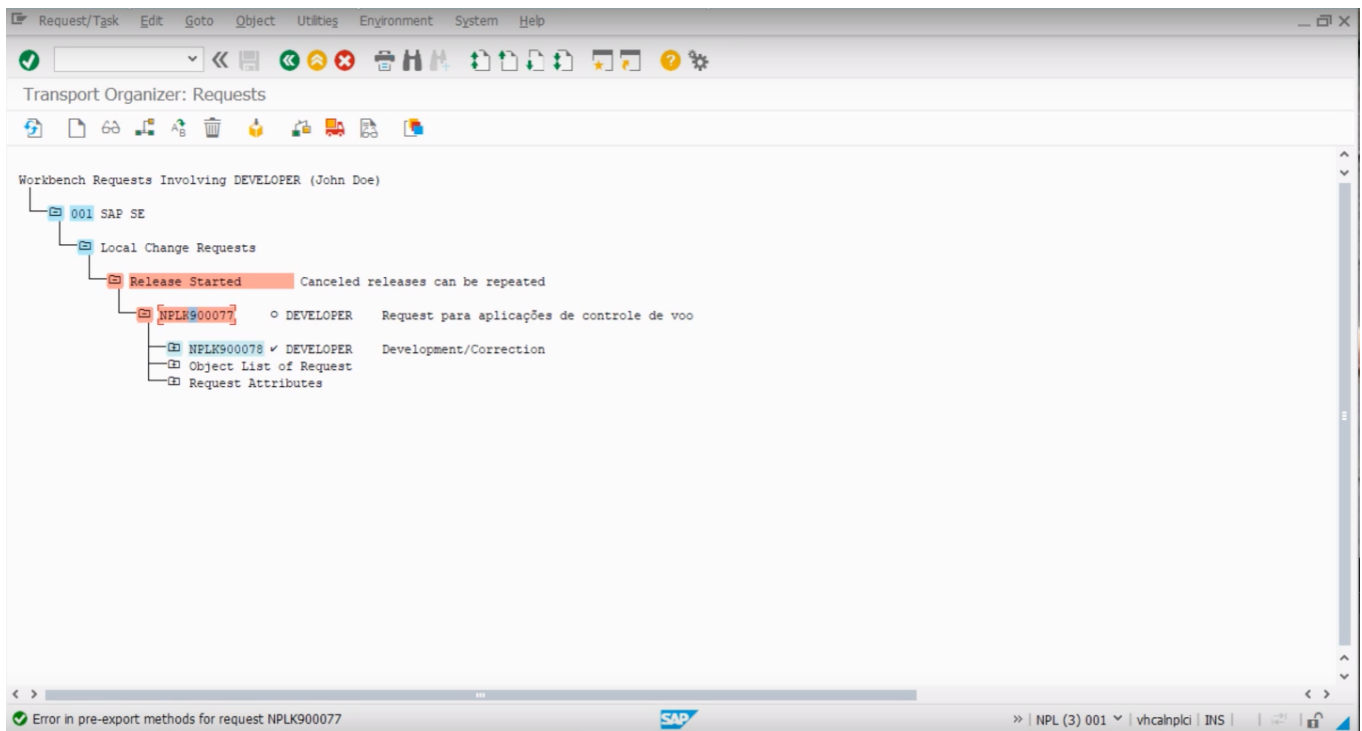
Teremos acesso a nossa request e task com o estado de **Modifiable**, e isso significa que, para utilizar a aplicação que acabamos desenvolver no ambiente de qualidade ou produção, precisamos liberá-la.



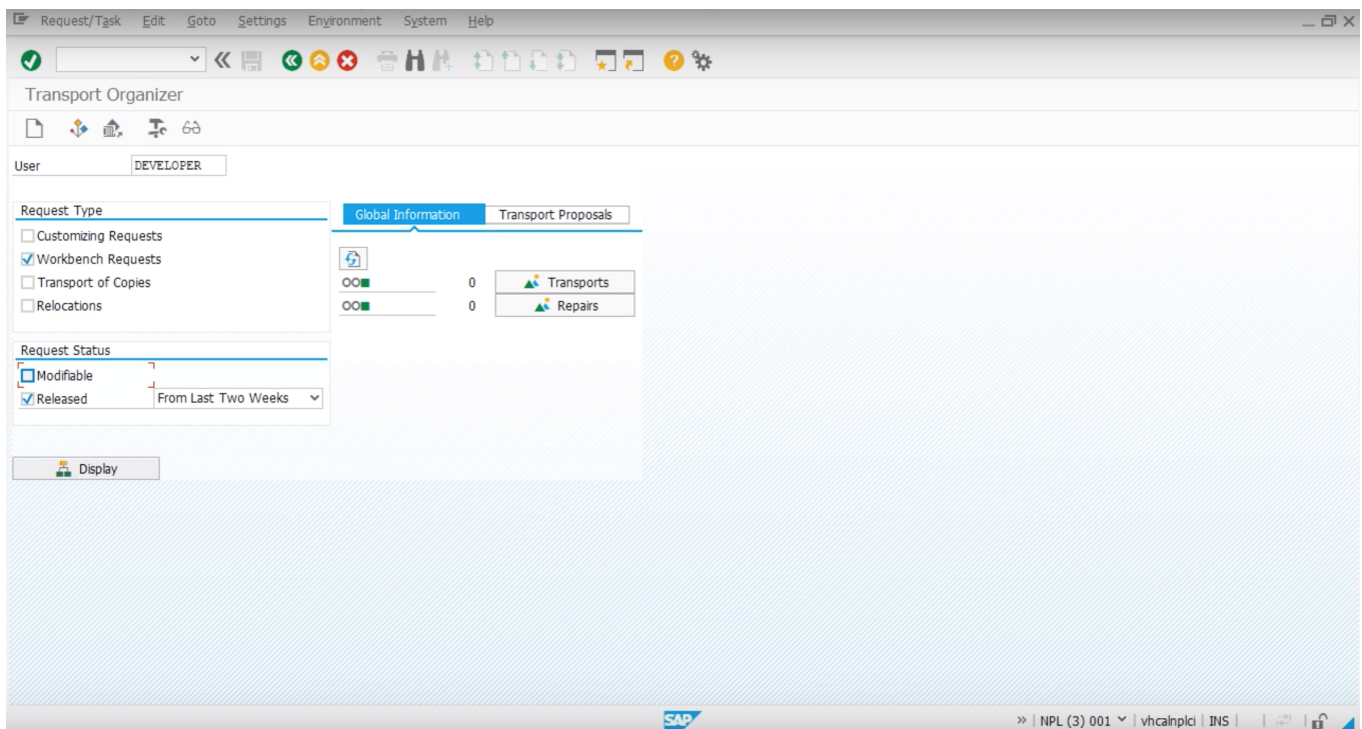
Primeiro, vamos liberar a nossa task. Para isso, é necessário posicionar o cursor sobre a task e então clicar no ícone de caminhão.



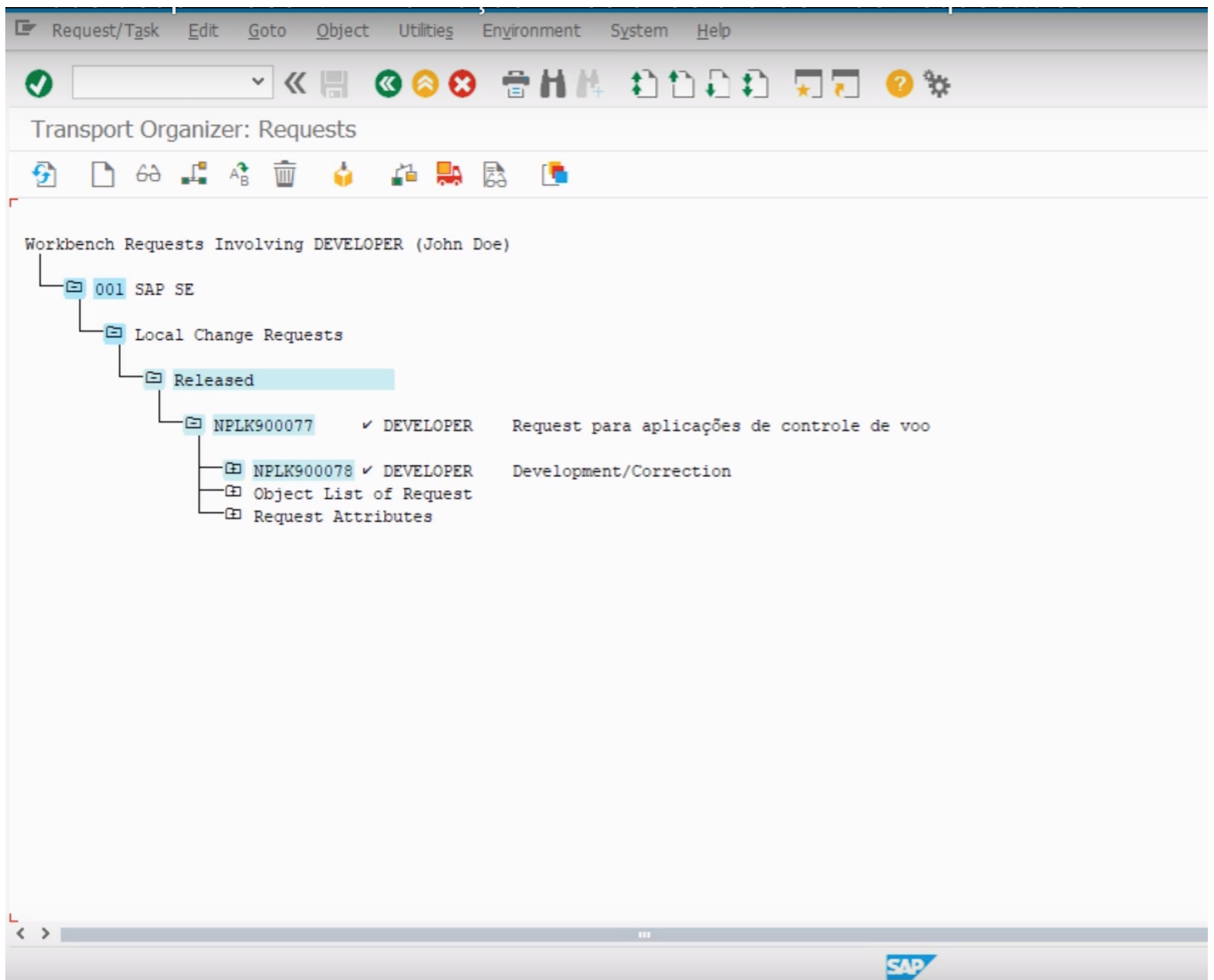
Depois de a task ser liberada, podemos liberar a request. Para isso posicione o cursor sobre a request, e então clique no caminhão novamente. Então, o estado da request será alterado, e teremos a mensagem informando que foi iniciada a liberação da request.



Vamos pressionar **F3** e voltar à tela anterior. Então vamos alterar para buscar por requests liberadas recentemente. Para isso, selecione a opção **Workbench Requests e Released**.

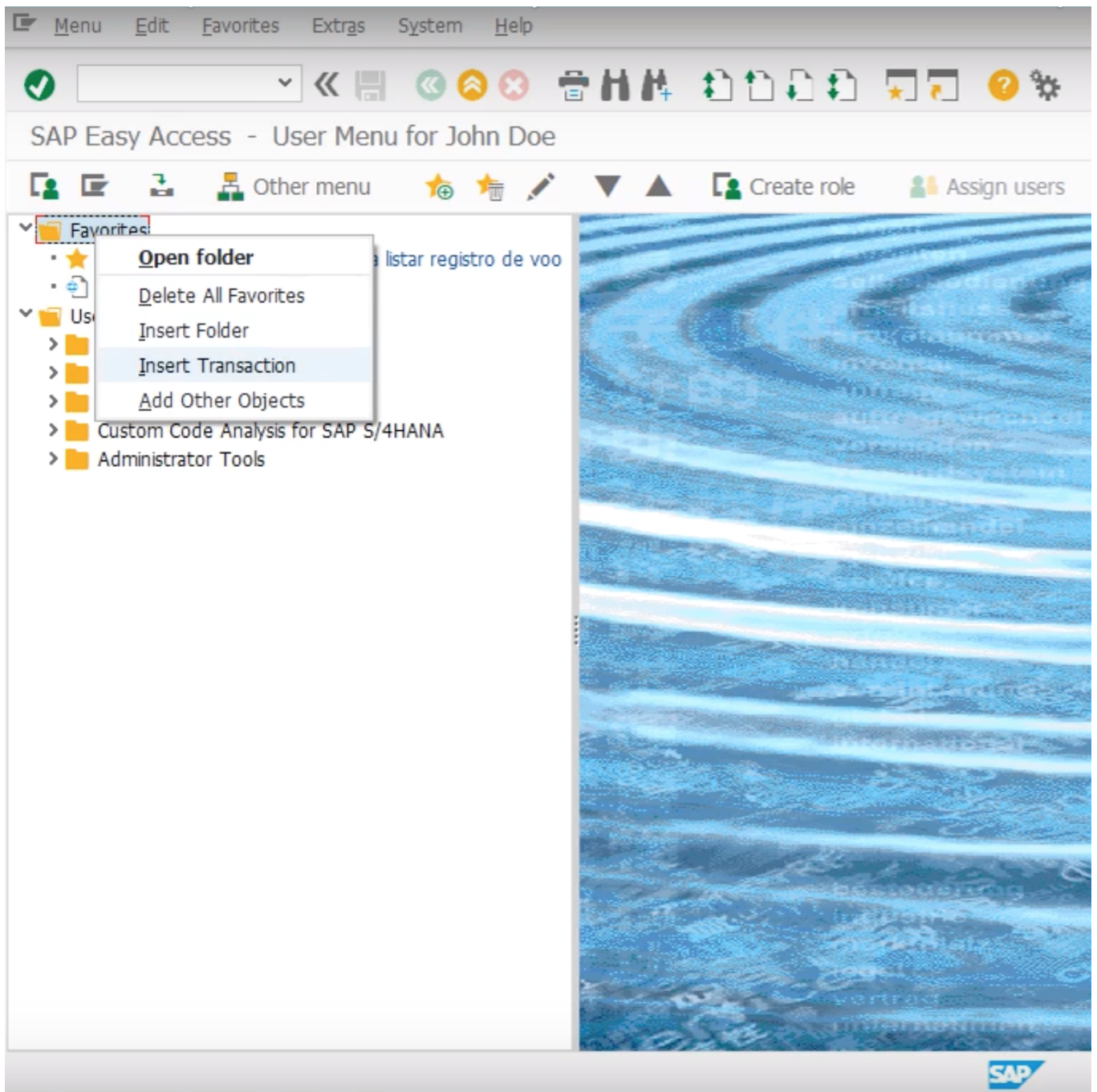


Teremos então acesso à request liberada e sem avisos.

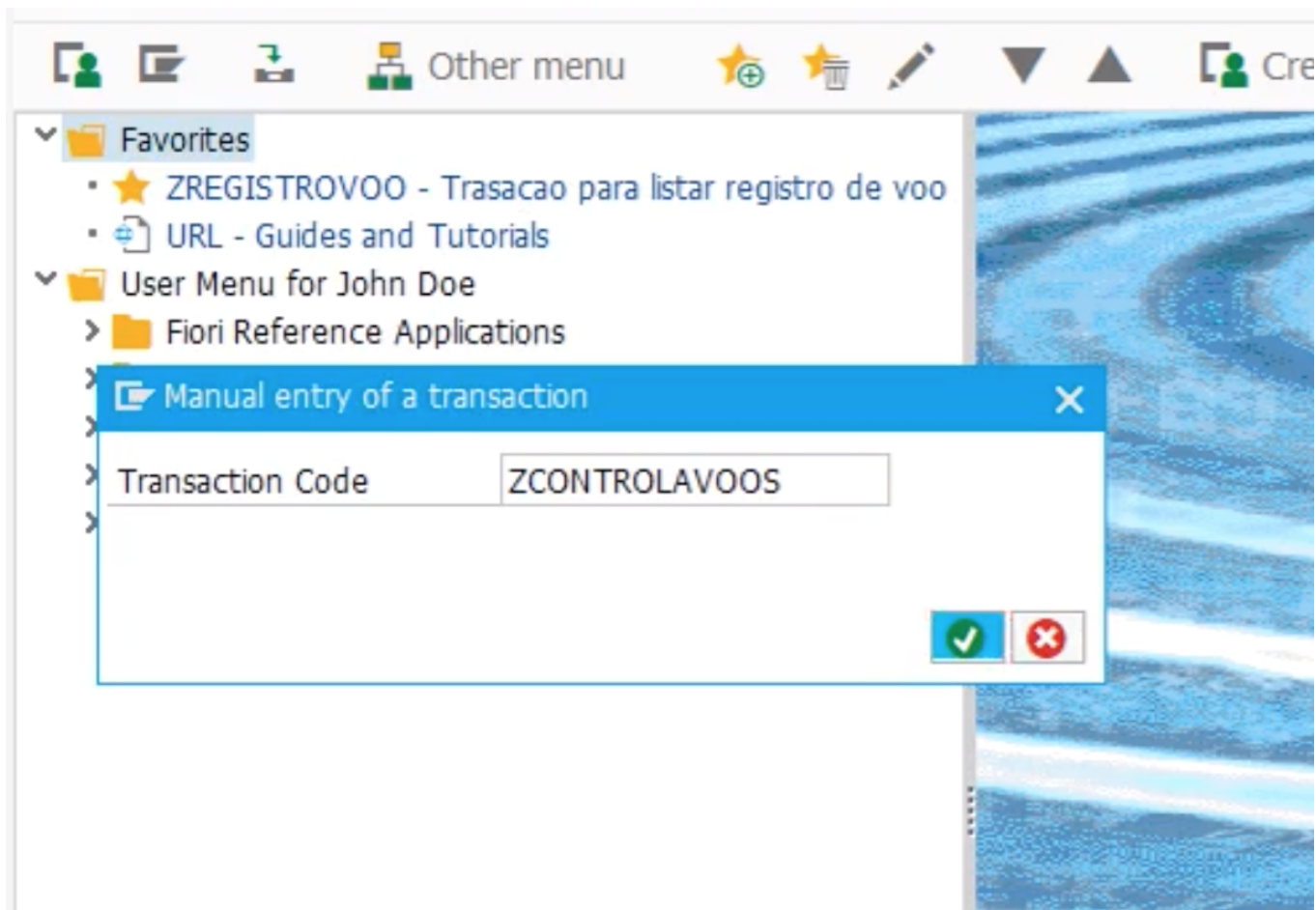


Com a ordem de transporte liberada, vamos incluir a transação da nossa aplicação no menu inicial do usuário.

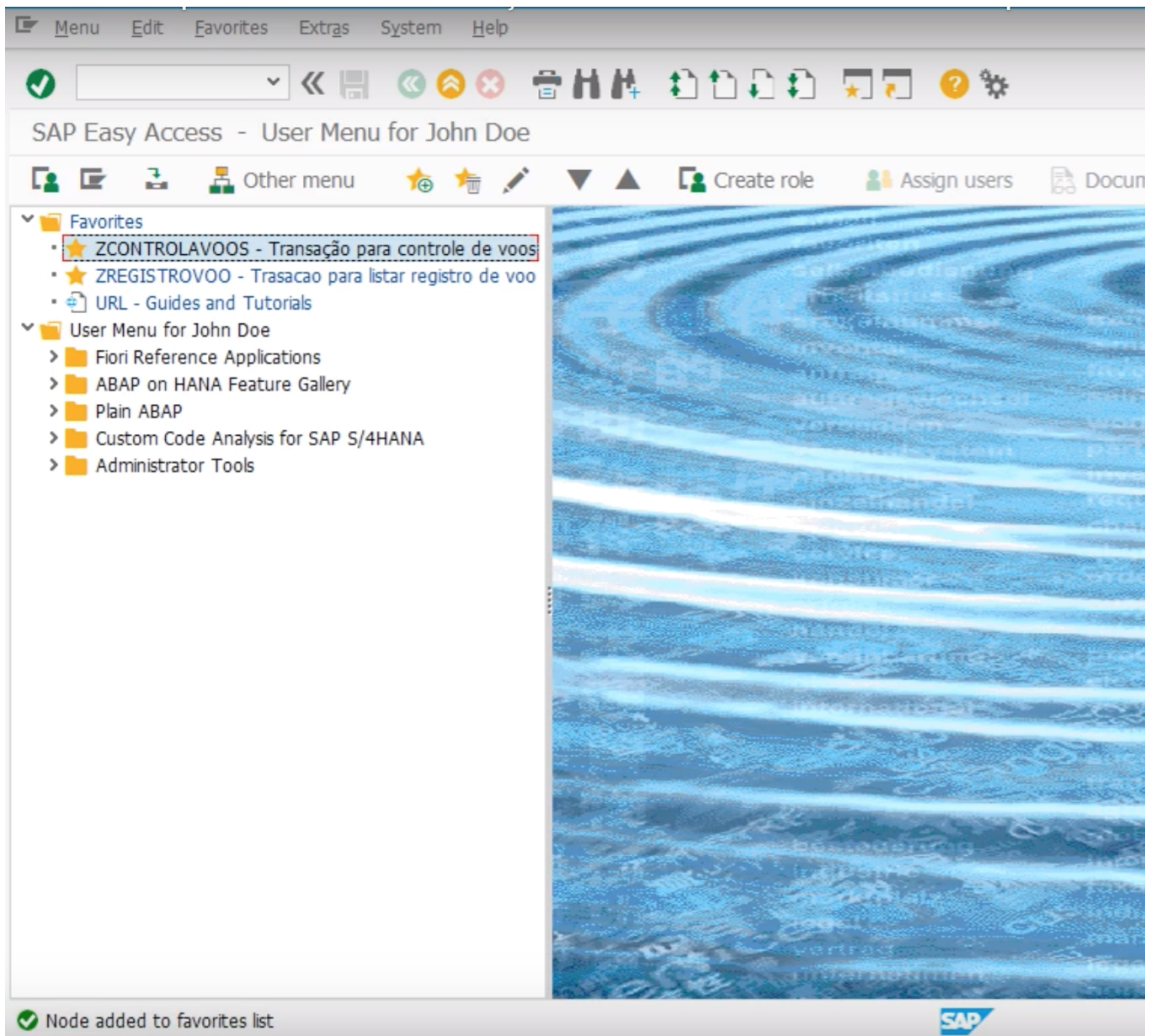
Para isso, vamos navegar pela tela inicial do SAP, clicar com o botão secundário do mouse na pasta **Favorites** e pressionar a opção **Insert Transaction**.



Um popup será aberto para que seja informado qual a transação desejamos inserir no menu. Vamos inserir ZCONTROLAV00S e confirmar.



Após a confirmação, teremos a nossa transação disponível do menu inicial do SAP.



** Código completo **

```
*&-----*
*& Include      Z_CONTROL_DE_VOOS_DECLARACOES
*&-----*

TABLES zspfli_atrasos .
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 relatorio_atrasos TYPE STANDARD TABLE OF zspfli_atrasos_estrutura .

*&-----*
*& Include      Z_CONTROL_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_PERFORMS
*&-----*
```

FORM executa_selecao .

IF zspfli_atrasos-dia_do_voo IS INITIAL .

```
  SELECT carrid, connid, countryfr,
         cityfrom, airpfrom, countryto,
         cityto, airpto, spfli~fltime, deptime,
         arrtime
  FROM spfli
  INTO TABLE @relatorio_atrasos
  WHERE carrid IN @s_carrid
     AND connid IN @s_connid
     AND cityfrom IN @s_cityfr
     AND cityto IN @s_cityto.
```

ELSE .

```
  SELECT spfli~carrid, spfli~connid, spfli~countryfr,
         spfli~cityfrom, spfli~airpfrom, spfli~countryto,
         spfli~cityto, spfli~airpto, spfli~fltime, spfli~deptime,
         spfli~arrtime, zspfli_atrasos~dia_do_voo, zspfli_atrasos~atraso
  FROM spfli AS spfli
  INNER JOIN zspfli_atrasos AS zspfli_atrasos
  ON spfli~carrid = zspfli_atrasos~carrid
     AND spfli~connid = zspfli_atrasos~connid
  INTO TABLE @relatorio_atrasos
  WHERE spfli~carrid IN @s_carrid
     AND spfli~connid IN @s_connid
     AND spfli~cityfrom IN @s_cityfr
     AND spfli~cityto IN @s_cityto
     AND zspfli_atrasos~dia_do_voo = @zspfli_atrasos-dia_do_voo.
```

IF sy-subrc = 0 .

DATA wa_relatorio_atrasos LIKE LINE OF relatorio_atrasos .

LOOP AT relatorio_atrasos INTO wa_relatorio_atrasos .

IF wa_relatorio_atrasos-atraso IS NOT INITIAL .

```
  wa_relatorio_atrasos-arrtime = wa_relatorio_atrasos-arrtime + wa_relatorio_atrasos-atraso
  MODIFY relatorio_atrasos FROM wa_relatorio_atrasos .
```

```
ENDIF .

ENDLOOP .
ENDIF .
ENDIF .

ENDFORM .

FORM insere_atraso_voo .

DATA: wa_spfli          LIKE LINE OF relatorio_atrasos,
      wa_spfli_atrasos TYPE zspfli_atrasos.

LOOP AT relatorio_atrasos INTO wa_spfli .

    wa_spfli_atrasos-carrid = wa_spfli-carrid .
    wa_spfli_atrasos-connid = wa_spfli-connid .
    wa_spfli_atrasos-dia_do_voo = zspfli_atrasos-dia_do_voo .
    wa_spfli_atrasos-atraso = zspfli_atrasos-atraso .

    INSERT zspfli_atrasos FROM wa_spfli_atrasos .

    IF sy-subrc = 0 .
        MESSAGE s000(zmsg_controle_de_voo) . .
    ELSE .
        MESSAGE s001(zmsg_controle_de_voo) DISPLAY LIKE 'E'.
    ENDIF .

ENDLOOP .

ENDFORM.

FORM controla_campos .

    IF relatorio_atrasos IS NOT INITIAL .

        LOOP AT SCREEN .

            IF screen-group1 = 'GP1' .

                screen-input = 1 .
                MODIFY SCREEN .

            ENDIF .

        ENDLOOP .

    ELSE .

        LOOP AT SCREEN .

            IF screen-group1 = 'GP1' .

                screen-input = 0 .
                MODIFY SCREEN .
```

```
ENDIF .

ENDLOOP .

ENDIF .

ENDFORM .

*&-----*
*& 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 = 'ZSPFLI_ATRASOS ESTRUTURA'
      CHANGING
        ct_fieldcat      = fieldcat.

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

  ENDIF .

  CALL METHOD alv->refresh_table_display( ) .

  PERFORM controla_campos .

ENDMODULE.

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

MODULE pai INPUT.
```

```
CASE sy-ucomm .  
  
    WHEN 'VOLTAR' OR 'CANCELAR' OR 'SAIR' .  
  
        LEAVE PROGRAM .  
  
    WHEN 'EXEC' .  
  
        PERFORM executa_selecao .  
  
    WHEN 'BTN_ATRASO' .  
  
        PERFORM insere_atraso_voo .  
        PERFORM executa_selecao .  
  
ENDCASE.  
  
ENDMODULE.
```

