

Faça o que eu fiz na aula

Nosso código cresceu e novamente começou a ficar complicado de realizar novas implementações ou manutenções. Então o que vamos fazer agora é refatorar o nosso código.

Para isso, vamos implementar um novo include apenas para o `FORM`. Assim o nosso código ficará da seguinte maneira:

```
*&-----*
*& Module Pool      Z_CONTROLE_DE_VOOS
*&-----*
*&
*&-----*
PROGRAM z_controle_de_voos.

INCLUDE z_controle_de_voos_declaracoes.
INCLUDE z_controle_de_voos_selecao.
INCLUDE z_controle_de_voos_performs.
INCLUDE z_controle_de_voos_pbo.
INCLUDE z_controle_de_voos_pai.

*&-----*
*& 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-arrrtime = wa_relatorio_atrasos-arrrtime + wa_relatorio_atrasos-atraso
            MODIFY relatorio_atrasos FROM wa_relatorio_atrasos .

        ENDIF .

    ENDLOOP .
ENDIF .

ENDFORM .

FORM insere_atraso_voo .

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

    LOOP AT relatorio_atrasos INTO wa_spfli .

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

        INSERT zspfli_atrasos FROM wa_spfliatraso .

    IF sy-subrc = 0 .

        MESSAGE 'Atraso registrado com sucesso' TYPE 'S' .

    ELSE .

        MESSAGE 'Erro ao registrar atraso no banco de dados' TYPE 'S' DISPLAY LIKE 'E' .

    ENDIF .

ENDLOOP .

ENDFORM.

```

```

*&-----*
*& 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'.
```

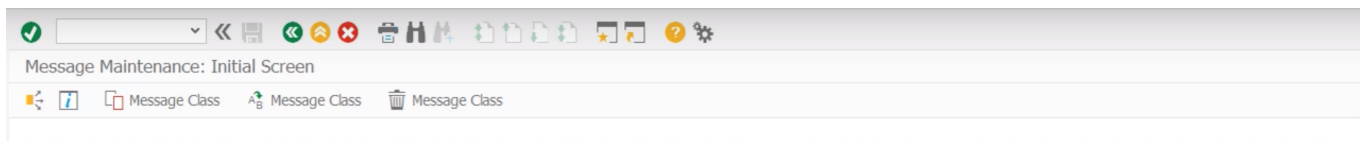
```
    PERFORM insere_atraso_voo .
```

```
ENDCASE.
```

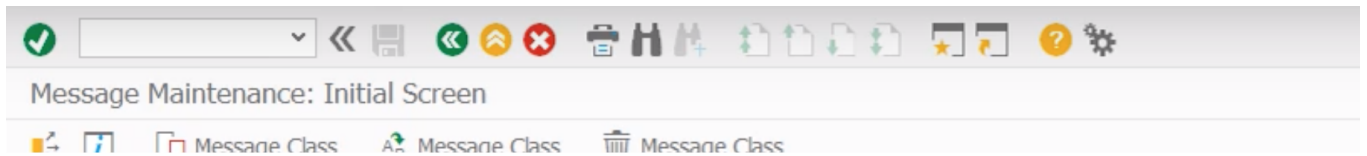
```
ENDMODULE.
```

Depois dessa refatoração, o nosso código fica mais simples de ler e realizar manutenções. Entretanto, um ponto que ainda não está adequado são as mensagens de forma hard code.

Então, vamos criar uma classe de mensagens e assim resolver este problema. Primeiro, vamos acessar a transação **SE91**.



Vamos criar a nossa classe de mensagem com o nome `ZMSG_CONTROLE_DE_VOO`.



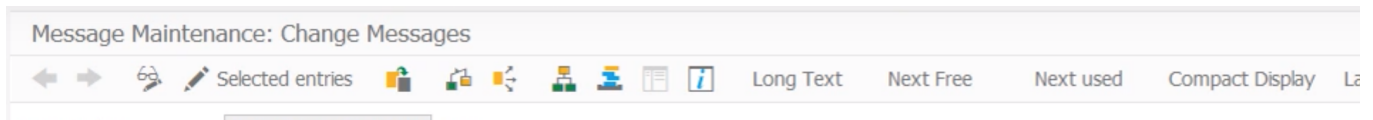
Após clicar no botão **Create** da etapa anterior, seremos redirecionados a uma nova janela, onde iremos informar o pacote que utilizaremos para salvar este novo objeto.

The screenshot shows the SAP Message Maintenance: Change Message Class dialog. The 'Message class' field is set to 'ZMSG_CONTROLE_DE_VOO' with a 'New' status. The 'Attributes' tab is selected. A sub-dialog titled 'Create Object Directory Entry' is open, showing the 'Object' field with 'R3TR MSAG ZMSG_CONTROLE_DE_VOO'. The 'Attributes' section of the sub-dialog includes: 'Package' (Z_APLICACAO_CONTROLE_VOOS), 'Person Responsible' (DEVELOPER), 'Original System' (NPL), 'Original language' (EN English), and 'Created On' (empty). The sub-dialog has buttons for 'Local Object', 'Lock Overview', and a close button.

Em seguida, devemos informar qual a ordem de transporte que será utilizada.

The screenshot shows the SAP Message Maintenance: Change Message Class dialog. The 'Message class' field is set to 'ZMSG_CONTROLE_DE_VOO' with a 'New' status. The 'Attributes' tab is selected. The sub-dialog 'Create Object Directory Entry' is not visible in this view.

Então podemos navegar para a aba **Messages**.



Vamos cadastrar as duas mensagens que estamos utilizando em nossa aplicação e então clicar no botão de salvar. Em seguida, teremos uma mensagem de sucesso.



Voltando ao código, iremos implementar o seguinte trecho de código que vai utilizar a classe de mensagens que acabamos de criar em vez das mensagens hard code.

```
*&-----*
*& Include      Z_CONTROLE_DE_VOOS_PERFORMS
*&-----*
```

```
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.

```

E referente à melhoria da nossa aplicação, temos um último ponto que está relacionado ao controle de elementos da tela. Não é adequado que o campo de **ATRASSO** fique sempre disponível para inserção de informações, sendo que informações podem ser inseridas somente após a seleção do relatório.

Vamos, primeiro, alterar o grupo dos três elementos gráficos que desejamos que iniciem seus estados de forma inativa.

Vamos abrir a ferramenta **Screen Painter** da janela 1000.

Com um clique duplo no elemento gráfico, seremos apresentados a uma ferramenta à nossa direita que controla os atributos dos elementos gráficos.

Vamos então inserir o valor **GP1** para os elementos gráficos: **Label**, **Text Box** e **Botão**.

Alterados os atributos, vamos, por meio do código, controlar os campos relacionado a **ATRASSO**. Para isso vamos implementar o seguinte trecho de código:

```

*&-----*
*& Include          Z_CONTROLE_DE_VOOS_PERFORMS
*&-----*

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 .

```

Então, como última melhoria, vamos cuidar para que sempre que ocorra a inserção de atraso no banco de dados o relatório seja atualizado automaticamente. Para fazer isso, insira o seguinte trecho de código:

```

*&-----*
*& 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 .

```

Código completo

```

PROGRAM z_controle_de_voos.

INCLUDE z_controle_de_voos_declaracoes.
INCLUDE z_controle_de_voos_selecao.
INCLUDE z_controle_de_voos_performs.
INCLUDE z_controle_de_voos_pbo.
INCLUDE z_controle_de_voos_pai.

*&-----*
*& Include          Z_CONTROLE_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_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_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-arrrtime = wa_relatorio_atrasos-arrrtime + 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.
```

