UNIDAD

10

DIPLOMATURA EN PROGRAMACION ABAP

MÓDULO 10: PROGRAMACIÓN DE DIÁLOGO. MODULE POOL

Programación de Diálogo y Creación de Module Pools con ABAP

Programación de Diálogo

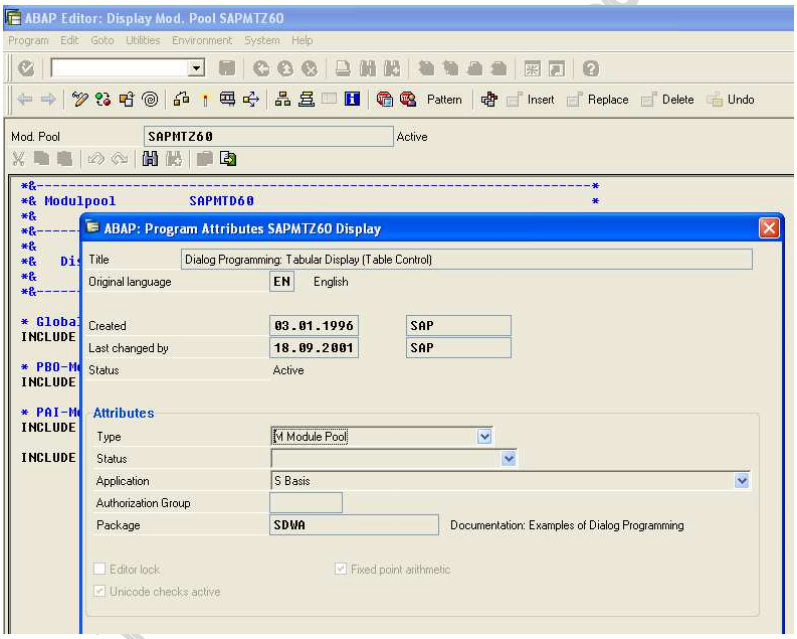
Este módulo detalla los pasos para crear programas de diálogo del tipo *module pool* en *ABAP*.

Module Pool

Pasos para su creación

Los pasos para la creación de un module pool son los siguientes:

- Crear el programa a través de la transacción SE38, de tipo module pool:

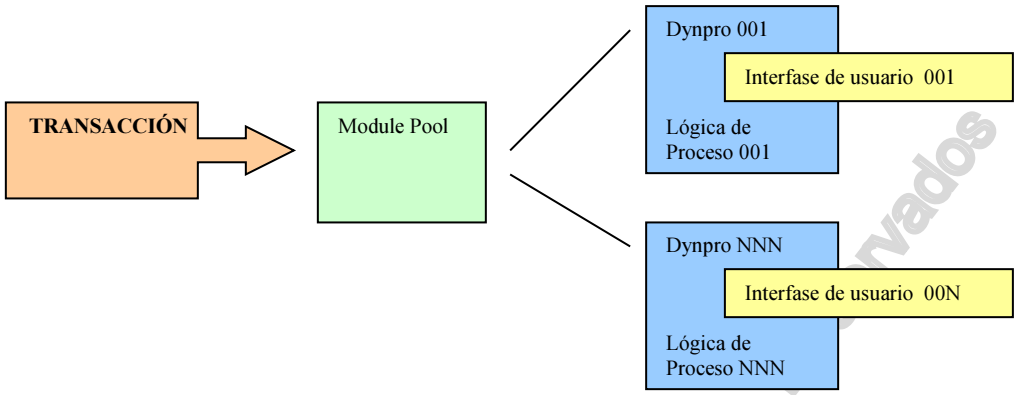


- Crear las pantallas a través de la transacción SE51 (Screen Painter). Cada una de ellas se denomina dynpro en SAP (dynamic program).

- Crear el diseño de la barra de tareas de la pantalla y asignar un título a través de la transacción SE41 (Menú Painter).

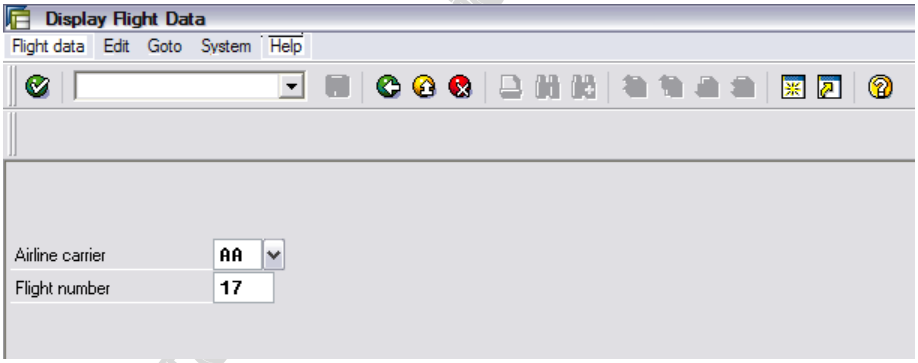
- Diseñar y desarrollar la lógica de proceso de las dynpros. Insertar el código correspondiente a los módulos PBO (Process Before Output) y PAI (Process Alter Input). En el módulo PBO se debe asociar la barra de tareas creado en el punto anterior.

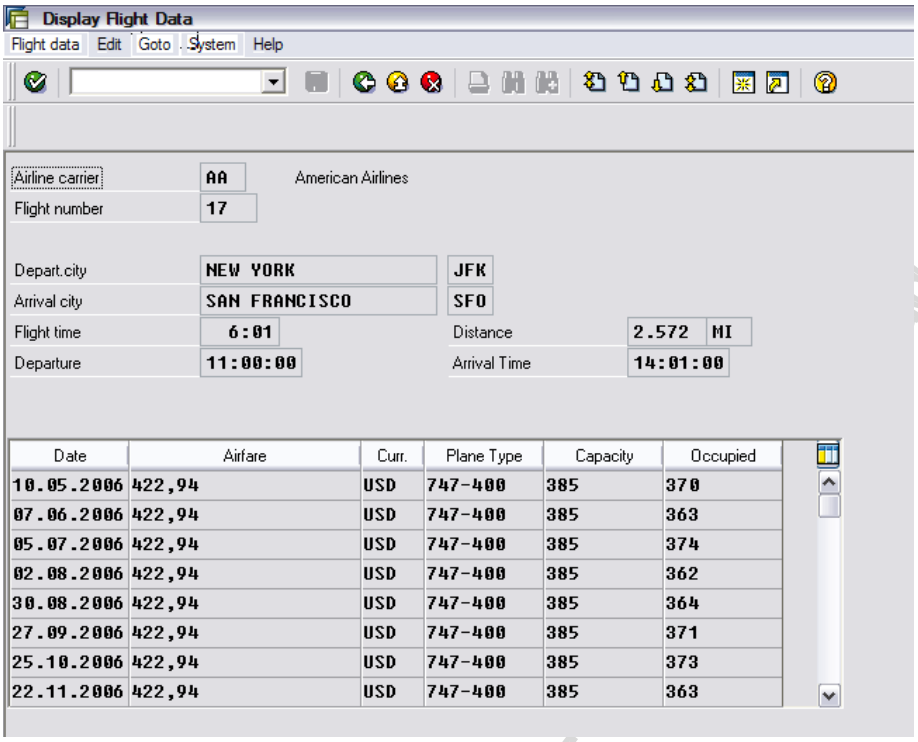
- Crear una transacción a través de la transacción SE93 que direccione la llamada al module pool creado.



Para ver cómo se estructura un module pool en SAP, seguiremos el siguiente ejemplo:

- Module Pool: SAPMTZ60
- Gui-Status: TD0100/ TD0200
- Dynpros: 0100-0200
- Código de transacción: TZ60





Ejemplo de Module Pool

Para ejemplificar lo expuesto, vamos a proceder a la visualización del programa SAPMTZ60 a través de la SE38:

```
*&---------------------------------------------------------------------*
*& Modulpool         SAPMTD60                                          *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&   Display DATA of table SPFLI                                       *
*&                                                                     *
*&---------------------------------------------------------------------*
* Globale Daten
INCLUDE MTZ60TOP.

* PBO-Module
INCLUDE MTZ60O01.

* PAI-Module
INCLUDE MTZ60I01.

INCLUDE MTZ60F01.
```

```
*---------------------------------------------------------------------*
*   INCLUDE MTD40TOP                                                  *
*---------------------------------------------------------------------*
PROGRAM  SAPMTZ60 MESSAGE-ID A&.
```

```
  TABLES: SFLIGHT,                          " Flüge
          SPFLI,                            " Flugdaten
          SAIRPORT,                         " Prüfung: Flughafen
          SCARR.                            " Fluggesellschaft

  DATA: OK_CODE(4),
        RCODE(5),
        LINE_COUNT TYPE I.

  CONTROLS: FLIGHTS TYPE TABLEVIEW USING SCREEN 200.

  DATA INT_FLIGHTS LIKE SFLIGHT OCCURS 1 WITH HEADER LINE.
```

```
*---------------------------------------------------------------------*
*   INCLUDE MTD40O01                                                  *
*---------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Module  STATUS_0100  OUTPUT
*&---------------------------------------------------------------------*
*       text                                                          *
*---------------------------------------------------------------------*
MODULE STATUS_0100 OUTPUT.
  SET PF-STATUS 'TD0100'.
  SET TITLEBAR '100'.
ENDMODULE.                 " STATUS_0100  OUTPUT

MODULE STATUS_0200 OUTPUT.
  SET PF-STATUS 'TD0200'.
  SET TITLEBAR '100'.
ENDMODULE.                 " STATUS_0100  OUTPUT

*&---------------------------------------------------------------------*
*&      Module  DISPLAY_FLIGHTS  OUTPUT
*&---------------------------------------------------------------------*
MODULE DISPLAY_FLIGHTS OUTPUT.
  SFLIGHT-FLDATE    = INT_FLIGHTS-FLDATE.
  SFLIGHT-PRICE     = INT_FLIGHTS-PRICE.
  SFLIGHT-CURRENCY  = INT_FLIGHTS-CURRENCY.
  SFLIGHT-PLANETYPE = INT_FLIGHTS-PLANETYPE.
  SFLIGHT-SEATSMAX  = INT_FLIGHTS-SEATSMAX.
  SFLIGHT-SEATSOCC  = INT_FLIGHTS-SEATSOCC.
ENDMODULE.                 " DISPLAY_FLIGHTS  OUTPUT

*---------------------------------------------------------------------*
*   INCLUDE MTD40I01                                                  *
*---------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
*       Fetch data from SPFL or exit from transaction                 *
*---------------------------------------------------------------------*
MODULE USER_COMMAND_0100 INPUT.
  CASE OK_CODE.
    WHEN SPACE.
      AUTHORITY-CHECK OBJECT 'S_CARRID'
            ID 'CARRID' FIELD 'LH'
```

```
            ID 'ACTVT' FIELD '*'.
      IF SY-SUBRC NE 0. Message E009. ENDIF.
      SELECT SINGLE * FROM  SPFLI
             WHERE  CARRID      = SFLIGHT-CARRID
             AND    CONNID      = SFLIGHT-CONNID.
      IF SY-SUBRC NE 0.
         MESSAGE E005 WITH SFLIGHT-CARRID SFLIGHT-CONNID.
      ENDIF.
      SELECT * FROM SFLIGHT INTO TABLE INT_FLIGHTS
                            WHERE  CARRID = SFLIGHT-CARRID
                            AND    CONNID = SFLIGHT-CONNID.
      IF SY-SUBRC NE 0.
         MESSAGE E006 WITH SFLIGHT-CARRID SFLIGHT-CONNID.
      ENDIF.
      FLIGHTS-TOP_LINE = 1.
      DESCRIBE TABLE INT_FLIGHTS LINES FLIGHTS-LINES.
      CLEAR OK_CODE.
    WHEN 'CANC'.
      CLEAR OK_CODE.
      SET SCREEN 0. LEAVE SCREEN.
    WHEN 'EXIT'.
      CLEAR OK_CODE.
      SET SCREEN 0. LEAVE SCREEN.
    WHEN 'BACK'.
      CLEAR OK_CODE.
      SET SCREEN 0. LEAVE SCREEN.
  ENDCASE.
ENDMODULE.                 " USER_COMMAND_0100  INPUT

MODULE USER_COMMAND_0200 INPUT.
  CASE OK_CODE.
    WHEN 'CANC'.
      CLEAR OK_CODE.
      SET SCREEN 100. LEAVE SCREEN.
    WHEN 'EXIT'.
      CLEAR OK_CODE.
      SET SCREEN 0. LEAVE SCREEN.
    WHEN 'BACK'.
      CLEAR OK_CODE.
      SET SCREEN 100. LEAVE SCREEN.
    WHEN 'NEW'.
      CLEAR OK_CODE.
      SET SCREEN 100. LEAVE SCREEN.
    WHEN 'P--'.
      CLEAR OK_CODE.
      PERFORM PAGING USING 'P--'.
    WHEN 'P-'.
      CLEAR OK_CODE.
      PERFORM PAGING USING 'P-'.
    WHEN 'P+'.
      CLEAR OK_CODE.
      PERFORM PAGING USING 'P+'.
    WHEN 'P++'.
      CLEAR OK_CODE.
      PERFORM PAGING USING 'P++'.
  ENDCASE.
ENDMODULE.                 " USER_COMMAND_0100  INPUT
```

```
*&---------------------------------------------------------------------*
*&      Module  EXIT  INPUT
*&---------------------------------------------------------------------*
*       text                                                          *
*---------------------------------------------------------------------*
MODULE EXIT_0100 INPUT.
  CASE OK_CODE.
    WHEN 'CANC'.
      CLEAR OK_CODE.
      SET SCREEN 0. LEAVE SCREEN.
    WHEN 'EXIT'.
      CLEAR OK_CODE.
      SET SCREEN 0. LEAVE SCREEN.
  ENDCASE.
ENDMODULE.                 " EXIT  INPUT

MODULE READ_TEXT_0100 INPUT.
  SELECT SINGLE * FROM SCARR WHERE CARRID = SFLIGHT-CARRID.
ENDMODULE.                 " READ_TEXT_0100  INPUT

*&---------------------------------------------------------------------*
*&      Module  SET_LINE_COUNT  INPUT
*&---------------------------------------------------------------------*
*       text                                                          *
*---------------------------------------------------------------------*
MODULE SET_LINE_COUNT INPUT.
  LINE_COUNT = SY-LOOPC.
ENDMODULE.                 " SET_LINE_COUNT  INPUT
```

```
*---------------------------------------------------------------------
***INCLUDE MTD60F01 .
*---------------------------------------------------------------------
*&---------------------------------------------------------------------*
*&      Form  PAGING
*&---------------------------------------------------------------------*
*       text                                                          *
*---------------------------------------------------------------------*
FORM PAGING USING CODE.
  DATA: I TYPE I,
        J TYPE I.

  CASE CODE.
    WHEN 'P--'. FLIGHTS-TOP_LINE = 1.
    WHEN 'P-'.  FLIGHTS-TOP_LINE = FLIGHTS-TOP_LINE - LINE_COUNT.
                IF FLIGHTS-TOP_LINE LE 0. FLIGHTS-TOP_LINE = 1. ENDIF.
    WHEN 'P+'.  I = FLIGHTS-TOP_LINE + LINE_COUNT.
                J = FLIGHTS-LINES - LINE_COUNT + 1.
                IF J LE 0. J = 1. ENDIF.
                IF I LE J. FLIGHTS-TOP_LINE = I.
                ELSE.      FLIGHTS-TOP_LINE = J.
                ENDIF.
    WHEN 'P++'. FLIGHTS-TOP_LINE = FLIGHTS-LINES - LINE_COUNT + 1.
                IF FLIGHTS-TOP_LINE LE 0. FLIGHTS-TOP_LINE = 1. ENDIF.
  ENDCASE.
ENDFORM.                   " PAGING
```

Continuamos luego con la visualización del programa SAPMTZ60 (dynpro 0100) a través de la SE51:

```
PROCESS BEFORE OUTPUT.
  MODULE STATUS_0100.
*
PROCESS AFTER INPUT.
  MODULE EXIT_0100 AT EXIT-COMMAND.
*
  FIELD SFLIGHT-CARRID
        MODULE READ_TEXT_0100 ON REQUEST.
*
  MODULE USER_COMMAND_0100.
```

A colación de ellos, proseguimos con la visualización del programa SAPMTZ60 (dynpro0200) a través de la SE51:

```
PROCESS BEFORE OUTPUT.
  MODULE STATUS_0200.

  LOOP AT INT_FLIGHTS WITH CONTROL FLIGHTS CURSOR FLIGHTS-TOP_LINE.
    MODULE DISPLAY_FLIGHTS.
  ENDLOOP.
*
PROCESS AFTER INPUT.
*
  LOOP AT INT_FLIGHTS.
    MODULE SET_LINE_COUNT.
  ENDLOOP.
  MODULE USER_COMMAND_0200.
```

A continuación, en la próxima Unidad, veremos todos los detalles y funcionalidades del Screen Painter y del Menú Painter. En tanto, los invitamos a realizar las actividades de la Unidad actual.