

# Practica con ejemplos – Selection Screens

Escribe un programa que ...

- 1) Contenga un select-options con valores numéricos e imprima el resultado de multiplicar cada número dentro del rango por 3.

Solución:

```
REPORT z_abap1.
```

```
DATA v_number TYPE i.
```

```
SELECT-OPTIONS s_number FOR v_number NO-EXTENSION.
```

```
START-OF-SELECTION.
```

```
DATA v_difference TYPE i.
```

```
DATA v_multiplication_result TYPE i.
```

```
v_difference = s_number-high - s_number-low + 1.
```

```
DO v_difference TIMES.
```

```
    v_multiplication_result = ( s_number-low + sy-index - 1 ) * 3.
```

```
    WRITE v_multiplication_result.
```

```
    NEW-LINE.
```

```
ENDDO.
```

```
END-OF-SELECTION.
```

- 2) Contenga un select-options con valores numéricos e imprima todos los criterios de búsqueda separados ",", "

Solución:

```
REPORT z_abap2.
```

```
DATA v_number TYPE i.
```

```
SELECT-OPTIONS s_number FOR v_number.
```

```
START-OF-SELECTION.
```

```
LOOP AT s_number.
```

```
    IF s_number-sign = 'I'.
```

```
        WRITE: 'Include'.
```

```
    ELSE.
```

```
        WRITE: 'Exclude'.
```

```
    ENDIF.
```

```

CASE s_number-option.
  WHEN 'EQ'.
    WRITE: 'Equal', s_number-low.
  WHEN 'NE'.
    WRITE: 'Not equal', s_number-low.
  WHEN 'LT'.
    WRITE: 'Less than', s_number-low.
  WHEN 'LE'.
    WRITE: 'Less or equal', s_number-low.
  WHEN 'GT'.
    WRITE: 'Greater than', s_number-low.
  WHEN 'GE'.
    WRITE: 'Greater or equal', s_number-low.
  WHEN 'BT'.
    WRITE: 'between', s_number-low, ' and ', s_number-high.
  WHEN 'BT'.
    WRITE: 'not between', s_number-low, ' and ', s_number-high. WHEN 'CP'.
    "Los patrones se usan en los select-options de caracteres y cadenas"
    WRITE: 'Contains pattern', s_number-low.
  WHEN 'NP'.
    WRITE: 'not the pattern', s_number-low.
ENDCASE.

NEW-LINE.

ENDLOOP.

END-OF-SELECTION.

```

3) Declare un select-options para valores numéricos sin múltiples rangos. Luego, valide si se ingresa un rango mayor a 100 y, de ser así, muestre un mensaje de error "El rango es muy grande".

**Debe proponer una solución**

4) Declare un parámetro como un listbox que contiene todos los códigos de la línea aérea de la tabla SCARR.

Solución:

```

REPORT z_abap4.

PARAMETER p_list TYPE scarr-carrid AS LISTBOX VISIBLE LENGTH 20.

```

5) Declare tres parámetros como checkboxes. Cada uno de ellos representará una clase de vuelo diferente (first, business and economy).

**Debe proponer una solución**

6) Declare tres parámetros como radio buttons. Cada uno de ellos representará una clase de vuelo diferente (first, business and economy).

Solución:

**REPORT** z\_abap6.

**PARAMETER** p\_first **RADIOBUTTON GROUP** grp1.

**PARAMETER** p\_busin **RADIOBUTTON GROUP** grp1.

**PARAMETER** p\_econo **LIKE** p\_first **RADIOBUTTON GROUP** grp1.

7) Declara tres parámetros como checkboxes. El primero siempre se debe verificar una vez que se inicia el programa. Además, si el día actual está entre 1 y 20, los otros checkboxes también deben verificarse una vez que se inicie el programa.

Debe proponer una solución

8) Declara tres radio buttons y un campo de entrada. Si se selecciona un radio button, el campo de entrada debe borrarse. Nota: el campo debe borrarse tan pronto como se seleccione cualquier botón de opción y no después de que se ejecute el programa.

Solución:

**REPORT** z\_abap8.

**PARAMETER** p\_first **RADIOBUTTON GROUP** grp1 **USER-COMMAND** action.

**PARAMETER** p\_busin **RADIOBUTTON GROUP** grp1.

**PARAMETER** p\_econo **LIKE** p\_first **RADIOBUTTON GROUP** grp1.

**PARAMETER** p\_input **TYPE** string.

**AT SELECTION-SCREEN.**

**IF** sy-ucomm = 'ACTION'.

**CLEAR** p\_input.

**ENDIF.**

9) Declare tres radio buttons y dos campos de entrada. Si se selecciona el primer radio button, ambos campos de entrada se deben mostrar y estar listos para la entrada. Si se elige el segundo, el primer campo de entrada debe ser obligatorio y el segundo debe estar bloqueado para la entrada. Si se elige el último botón de opción, ambos campos de entrada no se deben mostrar en la pantalla.

Debe proponer una solución

10) Declare cuatro parámetros. Los dos primeros deben ser un tipo caracter y los dos últimos de tipo numérico. Separe cada par en la pantalla de selección usando bloques

de pantalla de selección. Ambos bloques deben contener un marco por lo que es posible ver la separación entre ellos.

#### Debe proponer una solución

11) Declare cuatro parámetros. Los dos primeros deben ser un tipo de carácter y los dos últimos de tipo numérico. Separe cada par en la pantalla de selección usando bloques de pantalla de selección. Ambos bloques deben contener un marco por lo que es posible ver la separación entre ellos. Cada cuadro debe tener un título. Además, defina un texto para cada etiqueta de parámetro usando elementos de texto.

#### Debe proponer una solución

12) Declare un botón dentro de una pantalla de selección y muestre un mensaje de información cuando se presione.

Solución:

```
REPORT z_abap9.
```

```
SELECTION-SCREEN PUSHBUTTON 10(8) text-001 USER-COMMAND press. "text-001 =  
'PushButton'
```

```
AT SELECTION-SCREEN.
```

```
IF sy-ucomm = 'PRESS'.  
    MESSAGE 'El botón fue presionado' TYPE 'I'.  
ENDIF.
```

13) Crea un bloque con 3 pestañas. Cada uno de ellos debe tener un contenido diferente.

Solución:

```
REPORT z_abap10.
```

```
SELECTION-SCREEN BEGIN OF SCREEN 1 AS SUBSCREEN.
```

```
PARAMETER p_1 TYPE string.
```

```
SELECTION-SCREEN END OF SCREEN 1.
```

```
SELECTION-SCREEN BEGIN OF SCREEN 2 AS SUBSCREEN.
```

```
PARAMETER p_2 TYPE d.
```

```
SELECTION-SCREEN END OF SCREEN 2.
```

```
SELECTION-SCREEN BEGIN OF SCREEN 3 AS SUBSCREEN.
```

```
PARAMETER p_3 TYPE t.
```

**SELECTION-SCREEN END OF SCREEN 3.**

**SELECTION-SCREEN BEGIN OF TABBED BLOCK tb FOR 10 LINES.**

**SELECTION-SCREEN TAB (10) tab1 USER-COMMAND tab1\_pressed DEFAULT SCREEN 1.**

**SELECTION-SCREEN TAB (10) tab2 USER-COMMAND tab2\_pressed DEFAULT SCREEN 2.**

**SELECTION-SCREEN TAB (10) tab3 USER-COMMAND tab3\_pressed DEFAULT SCREEN 3.**

**SELECTION-SCREEN END OF BLOCK tb.**

**INITIALIZATION.**

```
tab1 = 'String'.  
tab2 = 'Date'.  
tab3 = 'Time'.
```

14) Declare tres parámetros. Debe haber una línea horizontal que separe los dos primeros y una línea en blanco que separe los dos últimos.

Solución:

**REPORT** z\_abap11.

**PARAMETERS** p\_1.

**SELECTION-SCREEN ULINE.**

**PARAMETERS** p\_2 TYPE i.

**SELECTION-SCREEN SKIP** 1.

**PARAMETERS** p\_3 TYPE d.

15) Declare una pantalla de selección con 8 parámetros y 3 select-options de su elección. Ejecute su programa y guarde una variante para que se refiera a su selección en el momento que desee.

Solución:

**REPORT** z\_abap12.

**SELECTION-SCREEN BEGIN OF BLOCK b01 WITH FRAME TITLE** v\_person.

**PARAMETERS:** p\_name TYPE string LOWER CASE,  
p\_born TYPE d,

p\_last TYPE t.

SELECTION-SCREEN SKIP 1.

PARAMETERS p\_male RADIOBUTTON GROUP gend.

PARAMETERS p\_female RADIOBUTTON GROUP gend.

SELECTION-SCREEN END OF BLOCK b01.

SELECTION-SCREEN SKIP 1.

SELECTION-SCREEN BEGIN OF BLOCK b02 WITH FRAME TITLE v\_autho.

PARAMETER p\_admin AS CHECKBOX.

PARAMETER p\_user AS CHECKBOX.

PARAMETER p\_guest AS CHECKBOX.

SELECTION-SCREEN END OF BLOCK b02.

SELECTION-SCREEN SKIP 1.

DATA v\_number TYPE i.

DATA v\_number\_2 TYPE i.

DATA v\_favourite TYPE p LENGTH 10.

SELECTION-SCREEN BEGIN OF BLOCK b03 WITH FRAME TITLE v\_number.

SELECT-OPTIONS: s\_numb FOR v\_number,  
s\_numb2 FOR v\_number\_2 NO INTERVALS,  
s\_favour FOR v\_favourite NO-EXTENSION.

SELECTION-SCREEN END OF BLOCK b03.

INITIALIZATION.

v\_person = 'Personal Data'.

v\_autho = 'Authorization'.

v\_number = 'Numbers'.