

Ejercicios con Eventos

Parte A:

Conceptos a integrar:

Work Areas
Tablas Internas
Parámetros
Select Options
Eventos(Initialization, At Selection Screen, At Line-Selection)
Subrutinas (Performs)
Clase de Mensajes
Sentencia SELECT

Enunciado General:

El usuario quiere un reporte de todas las conexiones dada una línea aérea (Elegida por un parámetro PARAMETERS) y dado un rango de conexiones (Elegido por un rango de conexiones SELECT-OPTIONS).

(Sabemos que una línea aérea tiene muchas conexiones, lo vemos en la tabla SPFLI)

Una vez impreso el listado en pantalla de todas las conexiones para la línea aérea elegida, el usuario desea hacer doble clic en una conexión del listado y obtener otro listado con todos los vuelos de esa conexión (de la tabla SFLIGHT).

Restricciones:

El parámetro de la línea AEREA deberá tener por default el valor LH y debe ser ingresado en el evento INITIALIZATION.

Validar que el usuario pueda ejecutar el reporte solo con el valor LH, si cambia el valor por cualquier otro <> a LH, informar el error por pantalla! (eso se hace en el evento AT SELECTION-SCREEN y el error se informa con un MESSAGE EXXX(ZUXX_MSG de mensaje).

Vayamos Paso a Paso, resolviendo de a una cosa a la vez:

Paso 1) Definamos un parámetro de entrada con la sentencia:

Parameters: P_Carrid Type Scarr-Carrid.

Luego definamos una variable de tipo rango para elegir las conexiones de dicha aerolínea:

Select-Options: S_connid for wa_spfli-connid. *(Definir la Wa_spfli antes del Parameter)*

Paso 2) Nos piden llenar por default el parámetro P_carrid con el valor LH desde el evento INITIALIZATION.

INITIALIZATION.

P_carrid = 'LH'.

Probar que el programa hasta aquí funcione bien.

Paso 3) Nos pide que validemos que el reporte se ejecute solo si el P_carrid es = LH, como es una validación de parámetros lo validaremos en el evento AT SELECTION-SCREEN y lo pondremos debajo del INITIALIZATION.

AT SELECTION-SCREEN.

```
If p_carrid <> 'LH'.  
    Message E001( ZUXX_MSG ).  
Endif.
```

Aquí pusimos un message de tipo E para que sea de ERROR y no deje avanzar con el reporte hasta que el usuario no deje el parámetro CARRID con valor LH. (Reacuérdesse que el message E en el evento AT SELECTION-SCREEN le da chance al usuario a cambiar los parámetros y poder ejecutar nuevamente el reporte)

El programa así no va a compilar hasta que no creemos nuestra clase de mensaje desde la SE91, hágalo desde otra sesión y luego **PRUEBE EL PROGRAMA PARA VEO COMO FUNCIONA HASTA ACA**

Paso 4) Ahora debemos programar debajo del evento START-OF-SELECTION debido a que tenemos los otros eventos arriba.

Debemos Buscar primero todas las conexiones elegidas por el usuario e imprimirlas en pantalla.

Acá vamos a ser claros con el código con lo cual haremos 2 Subrutinas, una para buscar la información y otra para listarla en pantalla.

PERFORM F_BUSCAR_CONEXIONES. (Realizar los PERFORMS sin PARAMETROS)
PERFORM F_IMPRIMIR_CONEXIONES. (Realizar los PERFORMS sin PARAMETROS)

Realizar el Select correspondiente en F_BUSCAR_CONEXIONES e IMPRIMIR la tabla interna de la búsqueda en la rutina F_IMPRIMIR_CONEXIONES.

COMPROBAR QUE EL PROGRAME FUNCIONE HASTA ACA

Paso 5) (Esperar que el Instructor realice una DEMO del AT LINE-SELECTION) luego de la explicación ud. Puede hacer este punto.

Ahora agregar la funcionalidad del AT LINE-SELECTION para que cuando el usuario haga doble clic en una línea le traiga todos los vuelos de esa CONEXIÓN.

Puede ver el siguiente del uso de HIDE con el evento AT LINE-SELECTION:

DATA: campo (3) TYPE c.

START-OF-SELECTION.

```

WRITE: / 'primera linea'.
campo = '111'.

HIDE campo. " asociamos valor 111 a la linea escrita anteriormente.

WRITE: / 'segunda linea'.
campo = '222'."

HIDE campo. " asociamos valor 222 a la linea escrita anteriormente.

END-OF-SELECTION.

AT LINE-SELECTION.
  WRITE: / campo. " campo valdrá 111 si se selecciona la primera línea
           " o valdrá 222 si se selecciona la segunda

```

Parte B:

Listas - Subrutinas - Eventos

1) Declarar un tipo estructura con los siguientes campos:

```

carrid TYPE sflight-carrid,
connid
fdate
seatsmax
seatsocc
seatsocc_b
seatsmax_f
seatsocc_f
percentage(2) TYPE p DECIMALS 2,

```

Cada uno de estos tiene su tipo con el mismo nombre en la tabla del DD sflight.

2) Declarar una work area del tipo antes creado y otra de tipo sbook.

3) Declarar 2 tablas internas de los tipos de estructuras declarados anteriormente.

4) Pantalla:

Declarar un parameter de tipo S_CARR_ID y
Declarar un select-options del tipo wa-connid.

5) Hacer un select a la tabla sflight filtrando con los parámetros de selección declarados anteriormente, colocando el resultado en la tabla interna también declarada anteriormente.

6) Por cada loop endloop imprimir por pantalla la siguiente información:

En el campo de la estructura percentage calcular el porcentaje de ocupación del avión. $(\text{seatsocc_b} + \text{seatsmax_f} + \text{seatsocc_f}) / \text{seatsmax}$
Imprimir: carrid, connid, fdate, seatsocc, seatsmax, percentage, '%'.

7) Y guardar con HIDE la siguiente info: carrid, connid, fdate

8) Si en el select no se encontró ningún dato, imprimir un message advirtiendo eso.

9) Si el usuario realizo un doble click sobre una línea imprimir la siguiente información:

a. La guardada anteriormente con hide más:

b. Realizar un select a la tabla sbook, filtrada por carrid, connid y fldate. Guardar dicha información en la tabla interna del tipo sbook creada anteriormente.

c. Imprimir los siguientes datos loopeando la tabla

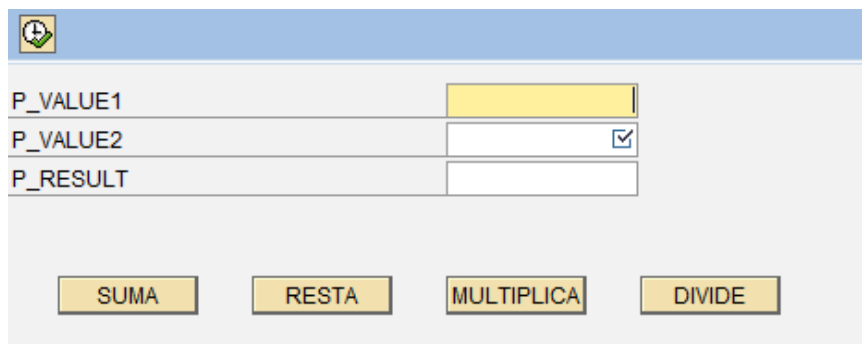
bookid, customid, custtype, class, order_date, smoker, cancelled.

Nota: Deberá usar Subrutinas para los Eventos a utilizar solo para la parte B.

Parte C:

Creación de botones pulsadores en la pantalla de selección

Crear 4 botones -> PUSHBUTTON y sus respectivos elementos de texto como se muestra a continuación.



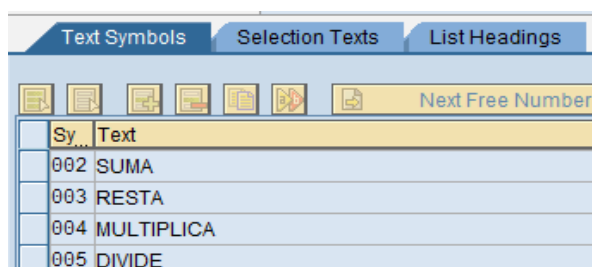
Definir 3 parámetros de selección, 2 de ellos obligatorios para poder realizar las distintas operaciones aritméticas y un tercer parámetro donde pueda mostrar el resultado de cada operación cuando se oprime el Botón correspondiente.

Para poder crear los Botones utilice la estructura sscrfields para poder definirlos de la siguiente manera:

```
TABLES: sscrfields.
```

```
SELECTION-SCREEN PUSHBUTTON /5(10) text-002 USER-COMMAND pb1.
```

El elemento de texto -> text-002 debe contener el nombre de la operación:



Sy...	Text
002	SUMA
003	RESTA
004	MULTIPLICA
005	DIVIDE

Deberá encontrar el evento apropiado para poder realizar la operación correspondiente a cada Botón y que muestre el resultado en el tercer parámetro.

En el caso de la operación de DIVIDE deben controlar que el segundo parámetro sea distinto de 0 mostrando un mensaje que no se puede dividir por 0.

Sugerencia: El mensaje puede ser del tipo I y los tipos de Datos de los parámetros deberían ser tipo P decimals 3.