

Módulo de Función

Una función es una porción de código como la subrutina, pero se genera de distinto modo, es independiente del reporte, y puede ser invocada por cualquier reporte. Tiene también parámetros de entrada y de salida. Veámosla en forma práctica.

Ejemplo:

```
call function 'Z_VUELOS'
  exporting
    seatsmax      = itab_flight-seatsmax
    seatsocc      = itab_flight-seatsocc
  importing
    percentage    = itab_list-capacity
  exceptions
    division_by_zero = 1
    occ_gt_max      = 2.

  case sy-subrc.
  when 1.
    write: text-zero, itab_flight-carrid, itab_flight-connid.
  when 2.
    write: text-max, itab_flight-carrid, itab_flight-connid.
  when 0.
    itab_list-number = 1.

    move-corresponding: itab_spfli to itab_list,
    move itab_list to itab_flight.

  endcase.
```

Aquí vemos que invocamos a una función llamada Z_VUELOS. Le pasamos como parámetros de entrada itab_flight-seatsmax e itab_flight-seatsocc. Recibiremos el resultado (percentage) en itab_list-capacity. Vemos también que hay un manejo de excepciones, que entenderemos enseguida cuando veamos el código de la función.

Por ahora diremos que si el valor del sy-subrc es 1, significa que la cantidad de asientos total es cero, por lo que imprimiremos el registro para lo cual se produce esto, y no lo tendremos en cuenta en el cálculo total. Si el sy-subrc es 2, significa que la cantidad de asientos total es menor que la de asientos ocupados, lo cual también es un error, por lo que imprimimos el registro para el cual se da este caso, y tampoco lo tenemos en cuenta para el cálculo total.

Entonces creamos la función para ir comprendiendo mejor los conceptos.

La transacción para gestionar funciones es la **SE37**.

Cuando creamos nuestro modulo de funciones, deberemos hacerlo pertenecer a un grupo de funciones, por lo cual, previamente crearemos el

Por ejemplo, al declarar variables globales en un modulo de funciones, las mismas se compartirán con los demás módulos de función que integren el grupo de funciones.

Creación de grupo de funcionesTransacción SE80

Para crear el grupo de funciones, usaremos la transacción **SE80**, que es el Object Navigator.

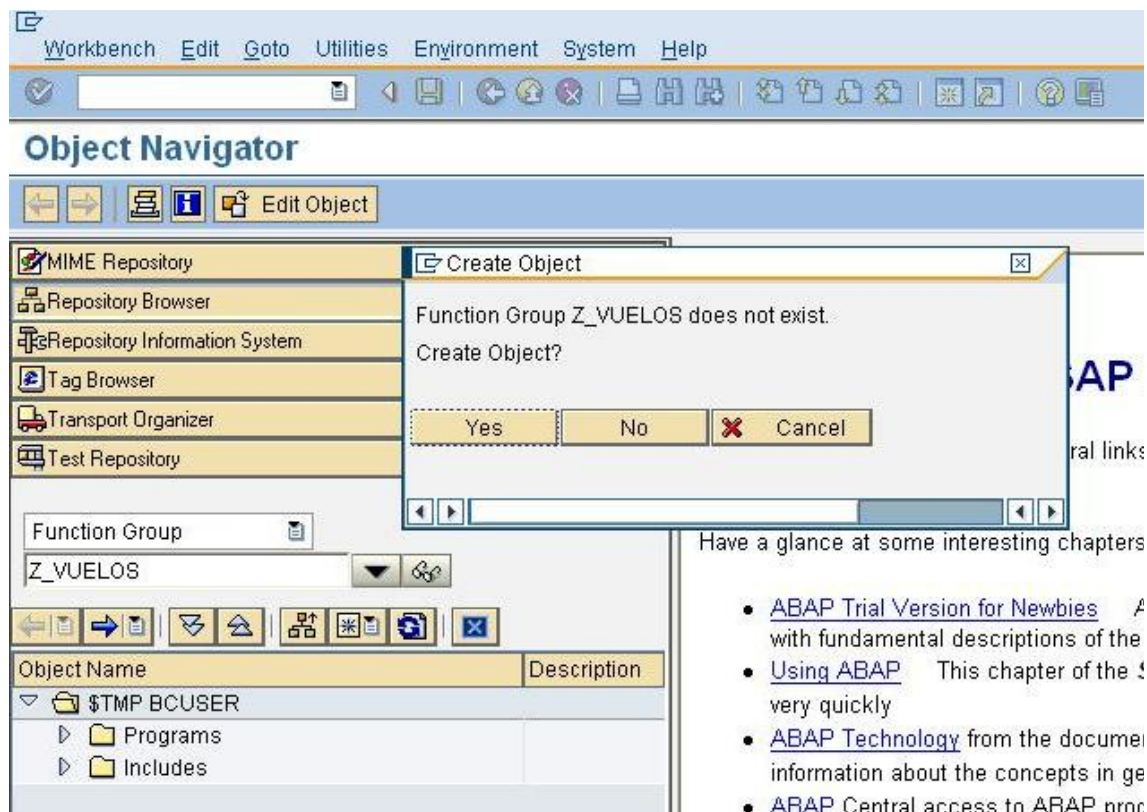


FIG.1- GRUPO DE FUNCIONES

Tal como vemos en la FIGURA 1, seleccionaremos como objeto para crear FUNCTION GROUP y debajo, ingresaremos el nombre que tendrá nuestro grupo de funciones, Z VUELOS.

Cuando presionemos enter, aparecerá el mensaje que podemos observar en la figura, indicando que el grupo de funciones no existe. Presionamos Yes para crearlo. Aparecerá una nueva ventana como podemos observar en la figura 2.

Colocamos una descripción en short text y presionamos Save. Como podemos ver en la figura 3, se ha creado el grupo de funciones, con dos includes. Y a la derecha, en dicha figura, vemos el editor del primer include, en donde podremos ingresar código.

Podemos observar también en esta parte de la figura, que el incluye está inactivo, por lo cual procederemos a activarlo (igual que como hacíamos en la

transacción SE38). Como podemos ver en la figura 4, aparecerán dos objetos para activar.
Seleccionamos los dos y los activamos.



FIG.2- GRUPO DE FUNCIONES

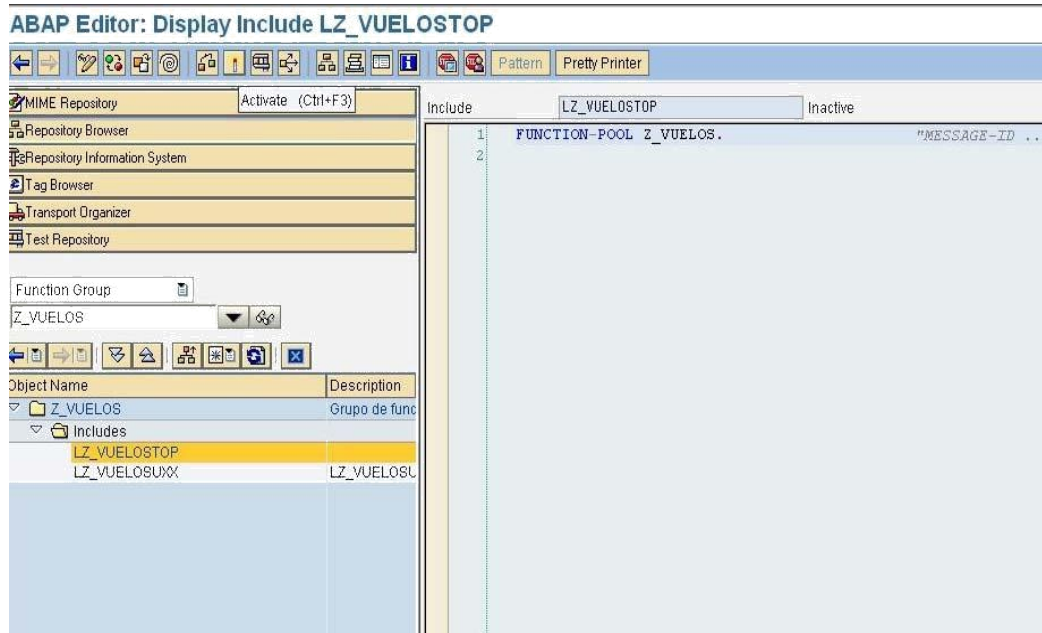


FIG. 3- GRUPO DE FUNCIONES

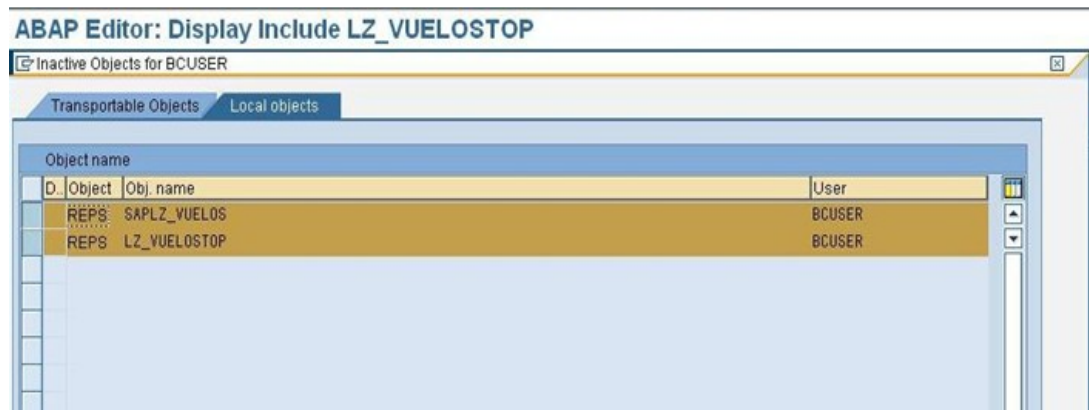


FIG.4- ACTIVACION DEL GRUPO DE FUNCIONES

Ahora estamos en condiciones de crear nuestro modulo de funciones.

Creación de módulo de funciones

Ejecutamos la transacción SE37.

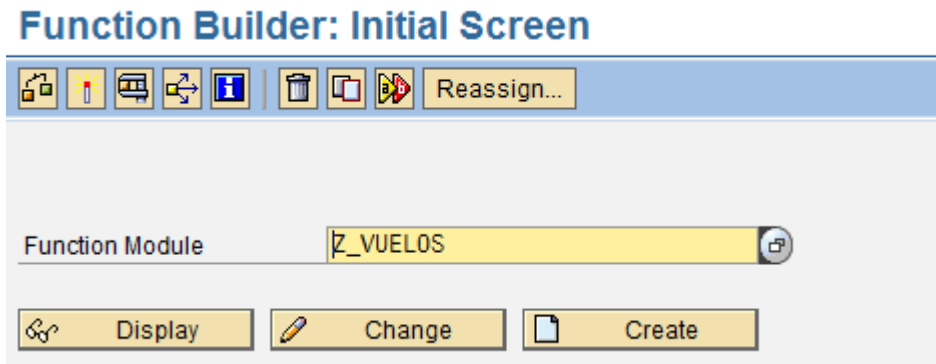


FIG.5- CREACION DE MODULO DE FUNCIONES

En nombre colocamos Z_VUELOS y presionamos Create.

Nos aparecerá una nueva ventana como la de la figura 6. En Function group colocamos el nombre del grupo de funciones que creamos en el paso anterior, y debajo una breve descripción.

Como podemos observar en la figura 8, se ha creado un nuevo modulo de funciones, y ahora debemos configurar los parámetros de entrada, salida, excepciones y código. Comenzamos con los parámetros de entrada. Colocamos los datos que observamos en la figura 7.

Para el parámetro de salida, vamos a la solapa Export, e ingresamos los datos tal como indica la figura 8. Idem, para las excepciones, vamos a la solapa correspondiente y repetimos el proceso, como se ve en la fig.9.

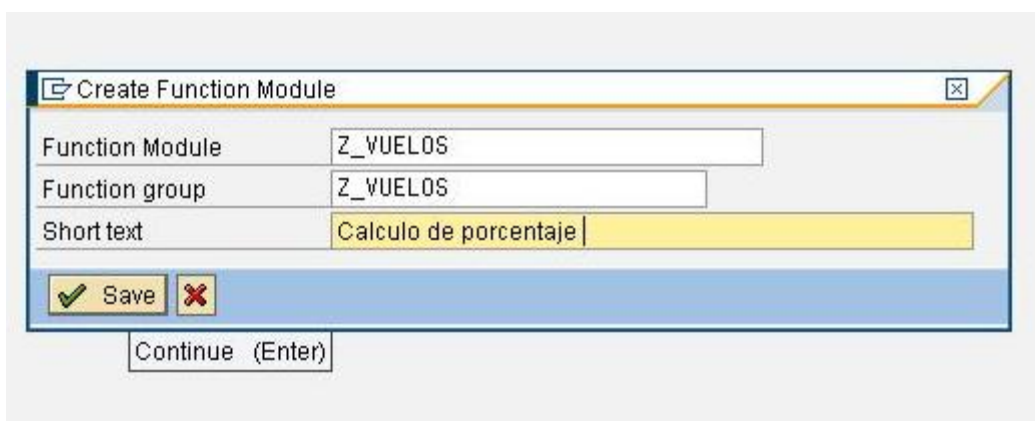


FIG.6- MODULO DE FUNCIONES

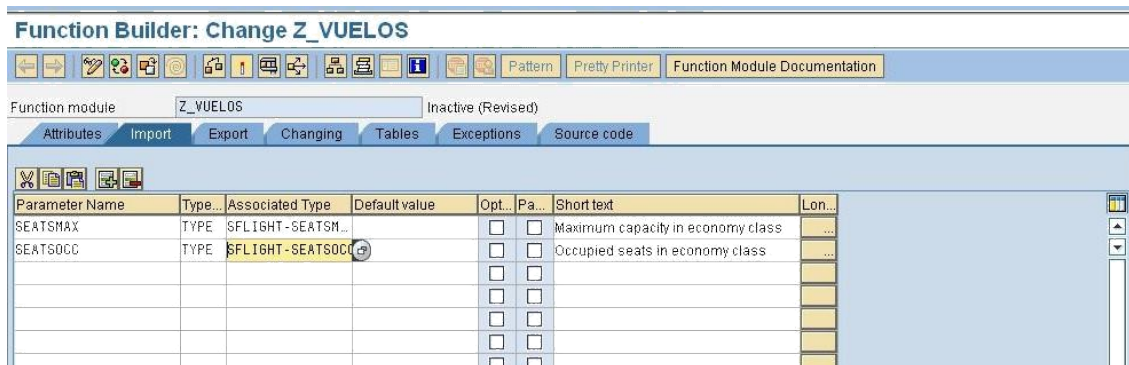


FIG.7- MODULO DE FUNCIONES-IMPORT

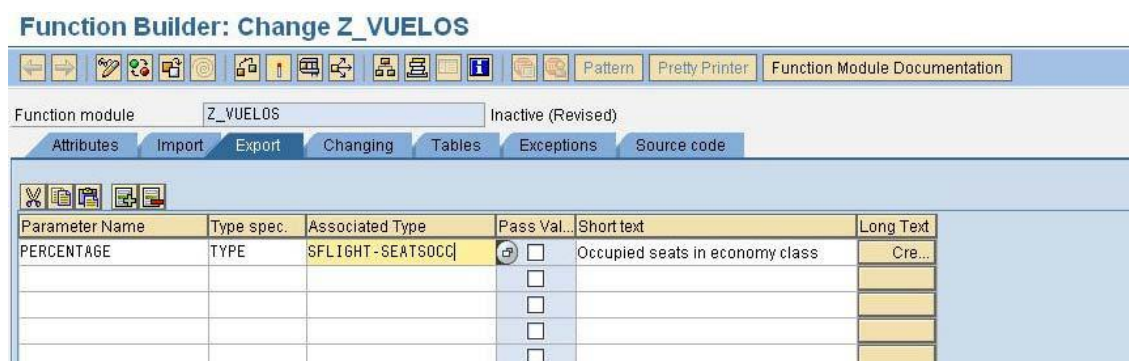


FIG.8- MODULO DE FUNCIONES-EXPORT

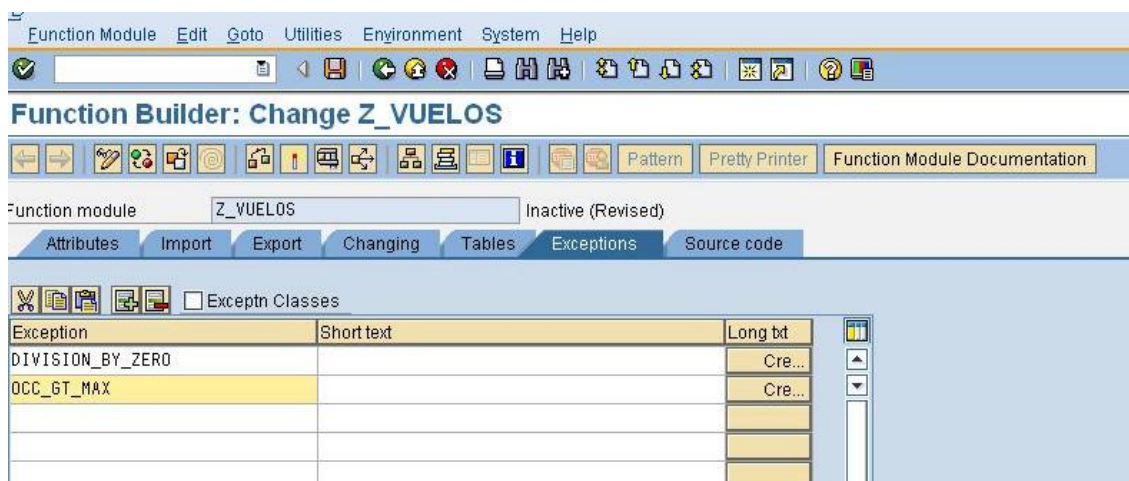


FIG.9- MODULO DE FUNCIONES-EXCEPTIONS

Finalmente, iremos a la solapa de código, e ingresaremos el mismo (Fig. 10).

```
IF SEATSMAX = 0.
  RAISE DIVISION_BY_ZERO.
ELSEIF SEATSOCC > SEATSMAX.
  RAISE OCC_GT_MAX.
ENDIF.
```

```
PERCENTAGE = ( SEATSMAX - SEATSOCC ) * 100 / SEATSMAX.
```


Una vez escrito el código, activamos.

Ahora, si en el reporte Z_FUNCT nos posicionamos en 'call function 'Z_VUELOS' y hacemos doble click sobre el nombre del modulo de funciones (Z_VUELOS), automáticamente navegaremos a la transacción que nos mostrara el modulo de funciones que acabamos de crear.

Si ejecutamos el reporte, veremos que el resultado es el mismo que se obtiene al usar subrutinas.

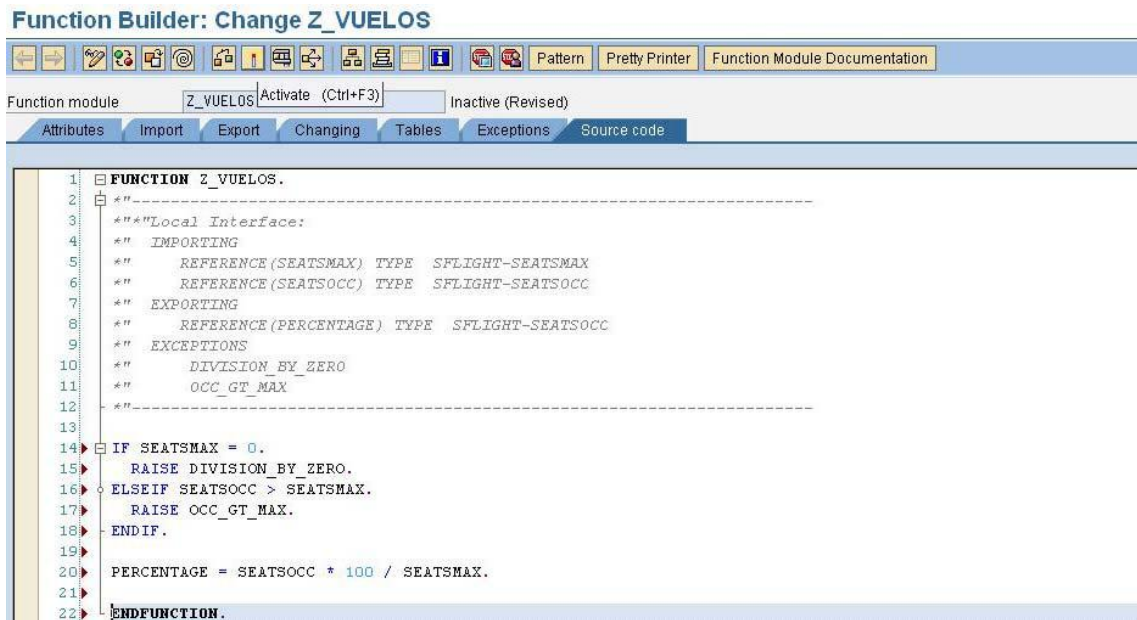


FIG.10- MODULO DE FUNCIONES-CODIGO

Si tuviéramos que escribir nuestro código desde cero, para incorporar la invocación a una función, podemos hacerlo así:

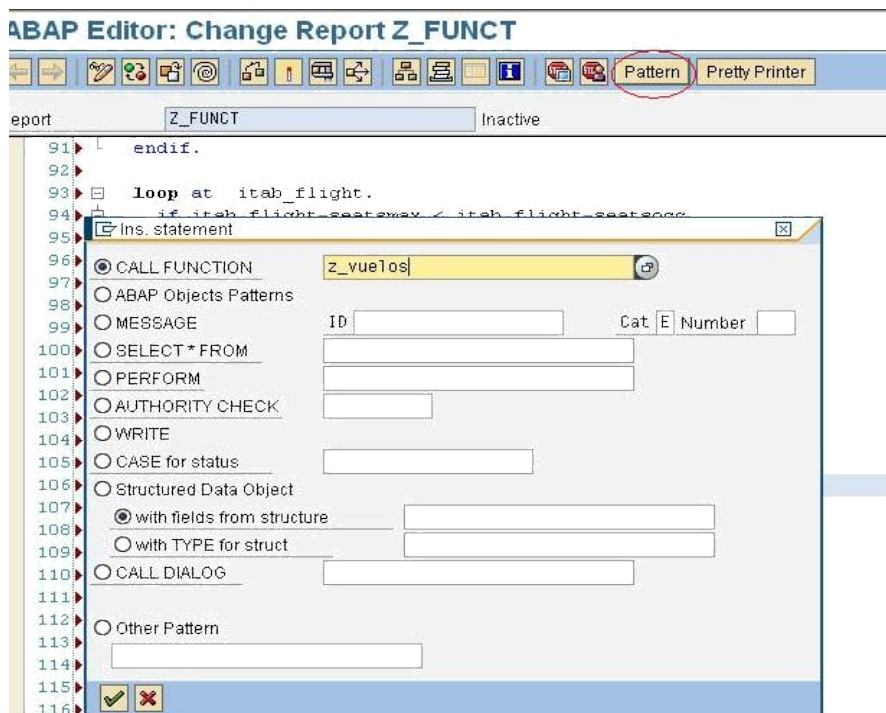


FIG. 11- CALL FUNCTION

Tal como muestra la figura 11, presionamos el botón PATTERN, y aparecerá una ventana para insertar diversos tipo de objetos. Seleccionamos CALL FUNCTION, ingresamos el nombre (o lo buscamos con F4) y presionamos la tecla de ok. Veremos que, en el editor aparece la invocación a la función, con la ventaja ya figuran todos los parámetros, para evitar los errores que podrían aparecer si la escribiéramos a mano.

Pruebas de módulos de funciones

Veamos ahora un poco las pruebas individuales que podemos realizar sobre los módulos de funciones. Si queremos probar el funcionamiento de un modulo de funciones realizado por nosotros, o necesitamos saber si alguno de los estándar de SAP sirve para lo que necesitamos, no es necesario escribir un report e invocarlo desde aquí. Podemos ejecutar los módulos de funciones en forma individual. Veamos como.

Ingresamos a la transacción SE37, y escribimos la función Z_VUELOS. Si presionamos Display, ingresaremos a visualizar el modulo. Ahora presionemos F8, para ejecutar. Aparecerá una pantalla como la de la figura 12.

Parameter	Value
SEATSMAX	1000
SEATSOCC	200

FIG. 12- TEST MODULO DE FUNCIONES

Ingresamos los valores de la figura 12 para los parámetros de entrada, y ejecutamos (F8 o el primer botón, a la izquierda del de Debugging).

Parameter	Value
SEATSMAX	1.000
SEATSOCC	200

Parameter	Value
PERCENTAGE	20

FIG. 13- TEST MODULO DE FUNCIONES

Como vemos en la figura 13, aparecerá la pantalla de resultado. En la parte superior vemos los valores que ingresamos para los parámetros de entrada, y en la parte inferior, PERCENTAGE, el valor que devuelve la función.

Si tuviéramos una función con gran cantidad de parámetros de entrada, o tablas, y quisiéramos dejar grabada la variante de ejecución, hacemos lo siguiente:

Cuando ingresamos los parámetros de entrada, antes de ejecutar, presionamos el botón de guardar

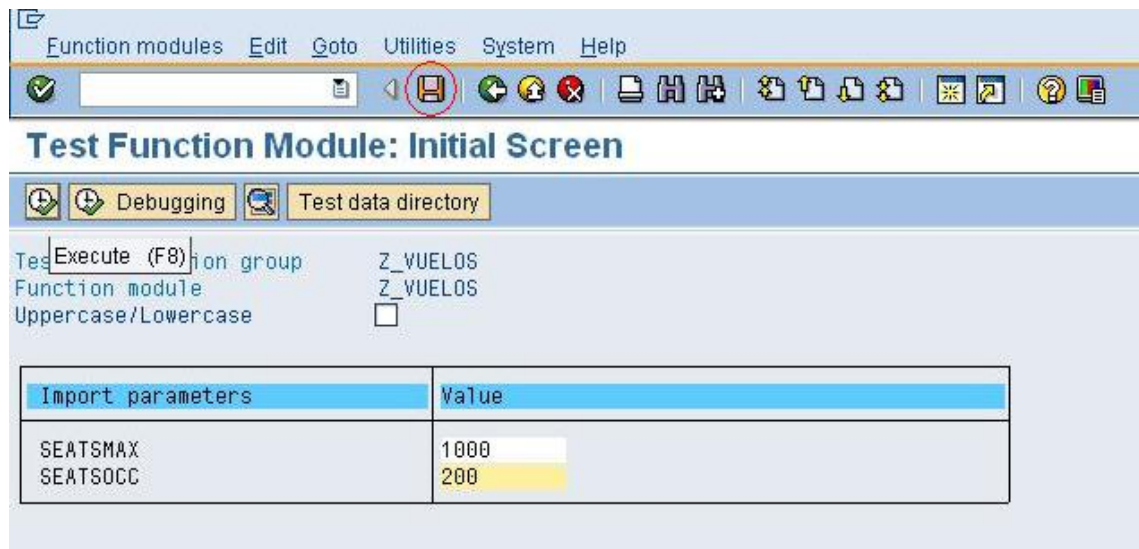


FIG. 14- GENERACION JUEGO DE DATOS

Aparecerá una ventana como se ve en la figura 15

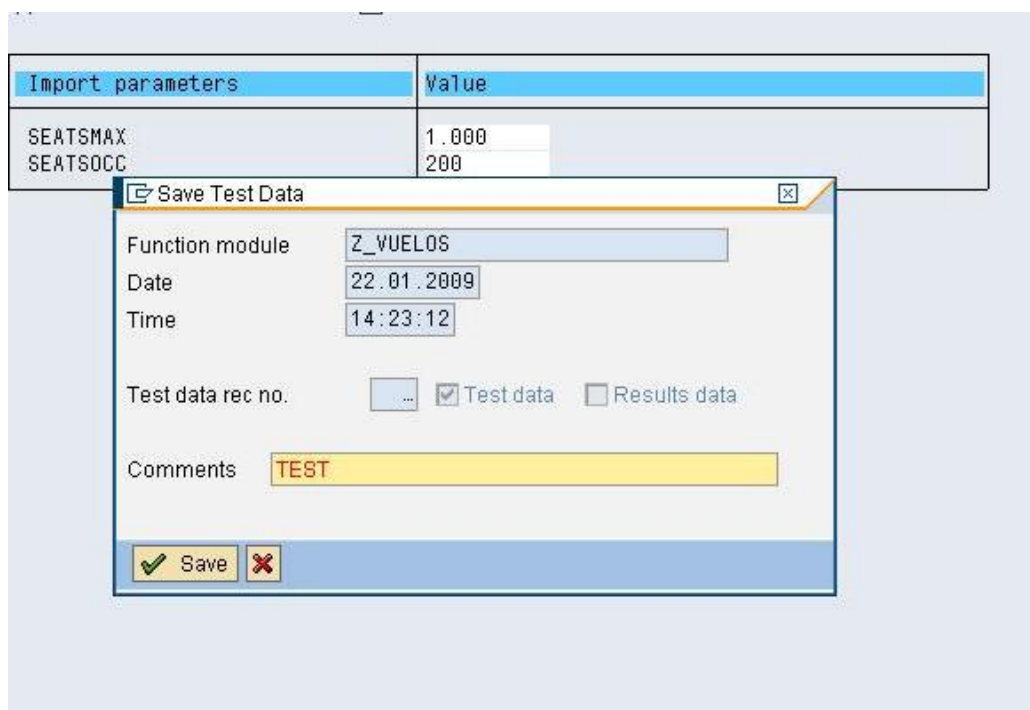


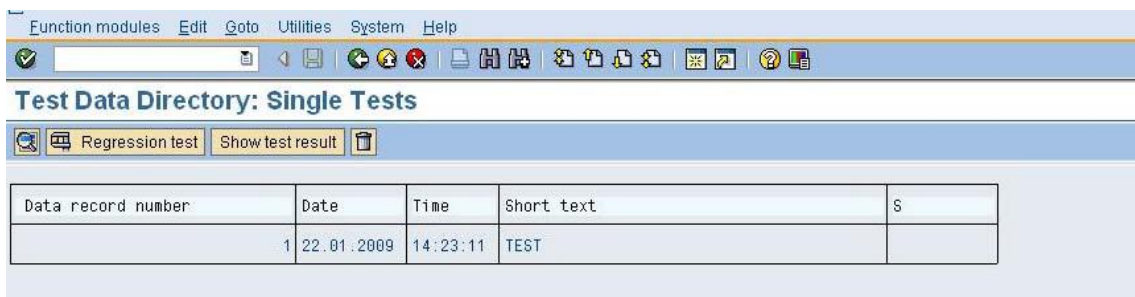
FIG. 15- GENERACION JUEGO DE DATOS - COMMENTS

En Comments ingresamos un texto cualquiera, con el que identificaremos el nombre del juego de datos.

Entonces, al momento de ingresar los valores de los parámetros de entrada, si presionamos el botón Test data directory, iremos a una pantalla, que nos mostrara un listado con todos los juegos de datos almacenados. Haciendo doble click sobre cualquiera de ellos (en nuestro caso tendremos solo uno), aparecerán automáticamente los valores de los parámetros de entrada cargados.



FIG. 16- RECUPERACION DE JUEGO DE DATOS



Ejemplo de Modulo de funcion utilizando tablas internas:

MF: READ_SPFLI_INTO_TABLE

Reporte que lo utiliza: DEMO_MOD_TECH_FB_READ_SPFLI

La función también se puede testear desde la SE37