



**LOGALI**

# SOLUCIÓN EJERCICIO INCLUDES

---

SAP ABAP Programación

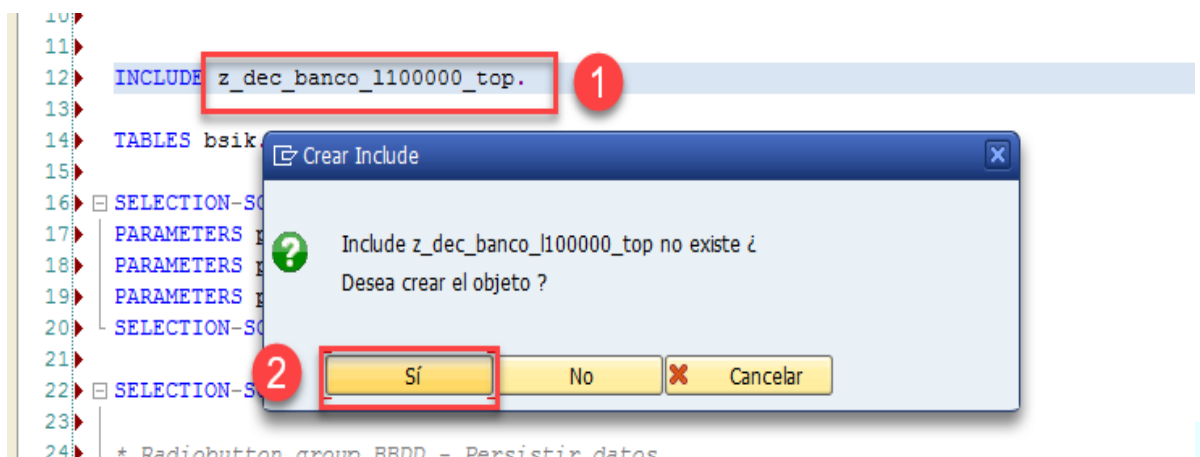




1. En la transacción del Editor ABAP modifique el programa Z\_DEC\_BANCO\_USUARIOSAP\*.
2. Añada la sentencia del include que va a contener las variables globales.

```
Report Z_DEC_BANCO_L100000 Activo(tratado)
1  *&-----*
2  *& Report Z_DEC_BANCO_L100000
3  *&
4  *&-----*
5  *&
6  *&
7  *&-----*
8
9  REPORT z_dec_banco_l100000.
10
11
12  INCLUDE z_dec_banco_l100000_top.
13
```

3. Con doble clic sobre el nombre del include Z\_DEC\_BANCO\_USUARIOSAP\_TOP cree el objeto asignando el paquete de desarrollo y la orden de transporte.





4. Mueva el código que corresponde a la declaración de las variables globales en el include creado.

```
*&-----*  
-----*  
*& Include          Z_DEC_BANCO_L100000_TOP  
*&-----*  
-----*
```

```
TABLES bsik.
```

```
DATA gwa_inform_banco TYPE zib_l100000.
```

5. Active el include y el programa principal.
6. Repita el mismo paso creando el include que va contener los elementos de la pantalla de selección agrupando el objeto en el paquete de desarrollo y asignando la orden de transporte.

```
1  *&-----*  
2  *& Report  Z_DEC_BANCO_L100000  
3  *&  
4  *-----*  
5  *&  
6  *-----*  
7  *-----*  
8  
9  REPORT z_dec_banco_l100000.  
10  
11  
12  INCLUDE z_dec_banco_l100000_top.  
13  
14  INCLUDE z_dec_banco_l100000_sel.
```

7. Mueva el código que corresponde a los elementos de la vista en el include creado.

```
*&-----*  
-----*  
*& Include          Z_DEC_BANCO_L100000_SEL  
*&-----*  
-----*  
  
SELECTION-SCREEN BEGIN OF BLOCK block1 WITH FRAME TITLE text-b01.  
PARAMETERS p_soc    TYPE bukrs      OBLIGATORY. " Sociedad  
PARAMETERS p_ejer   TYPE n LENGTH 4 OBLIGATORY. " Ejercicio  
PARAMETERS p_perio  TYPE n LENGTH 2 OBLIGATORY. " Período contable  
SELECTION-SCREEN END OF BLOCK block1.
```



```

SELECTION-SCREEN BEGIN OF BLOCK block4 WITH FRAME TITLE text-b04.

* Radiobutton group BBDD - Persistir datos
PARAMETERS: p_create RADIOBUTTON GROUP bbdd, " Create
             p_read   RADIOBUTTON GROUP bbdd, " Read
             p_update RADIOBUTTON GROUP bbdd, " Update
             p_delete RADIOBUTTON GROUP bbdd, " Delete
             p_modify RADIOBUTTON GROUP bbdd. " Modify

SELECTION-SCREEN END OF BLOCK block4.

*****
*****Tipo de ejecución*****
*****
SELECTION-SCREEN BEGIN OF BLOCK block2 WITH FRAME TITLE text-b02.
* Radiobutton group BUPA - Business Partner
PARAMETERS: p_acre RADIOBUTTON GROUP bupa, " Ejecución para Acreedo
res
             p_deud RADIOBUTTON GROUP bupa. " Ejecución para Deudore
s

SELECTION-SCREEN SKIP.

SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS p_test TYPE c AS CHECKBOX. " Test
SELECTION-SCREEN COMMENT (10) c_test.
PARAMETERS p_notif TYPE c AS CHECKBOX DEFAULT 'X'. " Notificar Mana
ger
SELECTION-SCREEN COMMENT (20) c_notif.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN END OF BLOCK block2.

SELECTION-SCREEN SKIP.

*****
*****Documentos a declarar*****
*****
SELECTION-SCREEN BEGIN OF BLOCK block3 WITH FRAME TITLE text-b03.
* Número de cuenta del proveedor o acreedor
SELECT-OPTIONS s_cuenta FOR bsik-lifnr.

* Clase de operación en cuenta de mayor especial
SELECT-OPTIONS s_clase FOR bsik-umsks.

* Fecha de la compensación
SELECT-OPTIONS s_fecha FOR bsik-augdt.

* Número de un documento contable
SELECT-OPTIONS s_doc FOR bsik-belnr.
SELECTION-SCREEN END OF BLOCK block3.

```



8. Active el include y el programa principal.

9. El código del programa principal.

```
*&-----*
*-----*
*& Report  Z_DEC_BANCO_L100000
*&
*&-----*
*-----*
*&
*&
*&-----*
*-----*

REPORT z_dec_banco_l100000.

INCLUDE z_dec_banco_l100000_top.

INCLUDE z_dec_banco_l100000_sel.

INITIALIZATION.

    c_test  = text-c01. " Test
    c_notif = text-c02. " Notificar Manager

AT SELECTION-SCREEN ON p_perio.

    IF p_perio LT 1 OR p_perio GT 12.
        MESSAGE e001(z_l100000).
    ENDIF.

START-OF-SELECTION.

    CASE abap_true.

        WHEN p_create. " Create

* Sociedad
        gwa_inform_banco-sociedad = p_soc.
* Ejercicio
        gwa_inform_banco-ejercicio = p_ejer.
* Período contable
        gwa_inform_banco-per_cont = p_perio.

* Acreedor
        IF p_acre IS NOT INITIAL.
            gwa_inform_banco-tipo_ejec = 'A'.
* Deudor
        ELSEIF p_deud IS NOT INITIAL.
            gwa_inform_banco-tipo_ejec = 'D'.
```



```
ENDIF.

* Usuario responsable
gwa_inform_banco-usuario_res = sy-uname.

* Fecha de ejecución
gwa_inform_banco-fecha_eje = sy-datum.

* INSERT en base de datos
INSERT zib_l100000 FROM gwa_inform_banco.

* Mensajes de información
IF sy-subrc EQ 0.
    MESSAGE i002(z_l100000).
ELSE.
    MESSAGE i003(z_l100000).
ENDIF.

WHEN p_read. " Read

SELECT SINGLE * FROM zib_l100000
    INTO gwa_inform_banco
    WHERE sociedad EQ p_soc
        AND ejercicio EQ p_ejer
        AND per_cont EQ p_perio.

IF sy-subrc EQ 0.

    WRITE: / 'Sociedad = ',
            gwa_inform_banco-sociedad,
            / 'Ejercicio = ',
            gwa_inform_banco-ejercicio,
            / 'Período contable = ',
            gwa_inform_banco-per_cont,
            / 'Tipo ejecución = ',
            gwa_inform_banco-tipo_ejec,
            / 'Usuario responsable = ',
            gwa_inform_banco-usuario_res,
            / 'Fecha de ejecución = ',
            gwa_inform_banco-fecha_eje MM/DD/YYYY.

ELSE.
    MESSAGE i004(z_l100000).
ENDIF.

WHEN p_update. " Update

CASE abap_true.
    WHEN p_acre.
        * Update tipo ejecución con Acreedor
        UPDATE zib_l100000 SET tipo_ejec = 'A'
        WHERE sociedad EQ p_soc
            AND ejercicio EQ p_ejer
            AND per_cont EQ p_perio.

    WHEN p_deud.
        * Update tipo ejecución con Deudor
```



```
        UPDATE zib_l100000 SET tipo_ejec = 'D'
        WHERE sociedad EQ p_soc
        AND ejercicio EQ p_ejer
        AND per_cont EQ p_perio.
    ENDCASE.

    IF sy-subrc EQ 0.
        MESSAGE i005(z_l100000).
    ELSE.
        MESSAGE i006(z_l100000).
    ENDIF.

    WHEN p_delete. " Delete

* Eliminación del registro
* de base de datos
    DELETE FROM zib_l100000
    WHERE sociedad EQ p_soc
    AND ejercicio EQ p_ejer
    AND per_cont EQ p_perio.

    IF sy-subrc EQ 0.
        MESSAGE i007(z_l100000).
    ELSE.
        MESSAGE i008(z_l100000).
    ENDIF.

    WHEN p_modify. " Modify

* Sociedad
    gwa_inform_banco-sociedad = p_soc.
* Ejercicio
    gwa_inform_banco-ejercicio = p_ejer.
* Período contable
    gwa_inform_banco-per_cont = p_perio.

* Acreedor
    IF p_acre IS NOT INITIAL.
        gwa_inform_banco-tipo_ejec = 'A'.
* Deudor
    ELSEIF p_deud IS NOT INITIAL.
        gwa_inform_banco-tipo_ejec = 'D'.
    ENDIF.

* Usuario responsable
    gwa_inform_banco-usuario_res = sy-uname.

* Fecha de ejecución
    gwa_inform_banco-fecha_eje = sy-datum.

* MODIFY en base de datos
    MODIFY zib_l100000 FROM gwa_inform_banco.

* Mensajes de información
    IF sy-subrc EQ 0.
        MESSAGE i009(z_l100000).
```



```
ELSE.  
    MESSAGE i010(z_1100000).  
ENDIF.  
  
WHEN OTHERS.  
  
ENDCASE.
```