



TEORÍA

Pantallas de selección

SAP ABAP Programación





Contenido

1.	Visión General	4
2.	Tareas	4
3.	Eventos de las pantallas de selección	5
4.	Tipos de pantallas de selección	5
4.1.	Pantallas de selección normales	5
4.2.	Subpantallas	6
5.	Ejemplos básicos	7
5.1.	Ejemplo 1 - Creación de campos	7
5.2.	Ejemplo 2 – Ubicar una pantalla	7
5.3.	Ejemplo 3 – Pestañas	8
6.	Elementos de la pantalla	10
6.1.	SKIP	11
6.2.	ULINE	11
6.3.	COMMENT	11
6.4.	PUSHBUTTON	12
6.5.	SCREEN-LINE	12
6.6.	SCREEN-BLOCK	13
6.7.	TABBED-BLOCK	13
6.8.	FUNCTION KEY	14
7.	Ejemplos complejos	15
7.1.	Ejemplo 1 – Radio buttons	15
7.2.	Ejemplo 2 – Push button	16
7.3.	Ejemplo 3 – Tratamiento de datos	17
7.4.	Ejemplo 4 – Radio buttons con sentencia de bifurcación	18
7.5.	Ejemplo 5 – Iconos	19
8.	Incluir objetos	21
8.1.	Incluir parámetros	21
8.2.	Incluir áreas de selección	21
8.3.	Incluir campos de salida	21
8.4.	Incluir pushbutton	21
8.5.	Incluir bloques	21
9.	Opciones ldb (base de datos lógica)	22
9.1.	Versiones de la pantalla de selección estándar	22
9.2.	Nodos para campos de selección	22
9.3.	Nodos para selecciones libres	23



10.	Parámetros	23
11.	SCREEN-OPTIONS.....	25
12.	VALUE-OPTIONS	26
13.	LDB-OPTIONS	28
14.	SELECT-OPTIONS	28



1. Visión General

Las pantallas de selección son pantallas especiales que pueden ser definidas en programas ejecutables, grupos de funciones y module pools. Las pantallas de selección son definidas en la sección de declaraciones globales de los mencionados programas ABAP con sentencias, sin usar el Screen Painter:

- SELECT-OPTIONS
- SELECTION-SCREEN
- PARAMETERS

Las pantallas de las pantallas de selección pueden contener subconjuntos de los elementos de pantalla de las pantallas generales.

Las pantallas de selección descansan en el mismo espacio-nombre de las pantallas de los programas. Adicionalmente, el número 1000 de dynpro está reservado para las pantallas de selección estándar. Aparte de las pantallas de selección estándar, podemos definir pantallas de selección independientes, que sólo pueden ser definidas en grupos de funciones y module pools.

Cuando un programa ABAP es activado, los componentes de la pantalla de selección del programa, es decir, pantallas con elementos de pantalla y lógica de proceso de pantalla, son automáticamente generados.

2. Tareas

Las pantallas de selección tienen dos misiones principales:

- Permiten a los usuarios introducir parámetros (valores únicos) y criterios de selección (intervalos de selección en forma tabular).
- Representan los programas de interfaz ejecutable, cuyos campos de entrada pueden ser suministrados por el programa llamante con SUBMIT.



3. Eventos de las pantallas de selección

No se pueden definir módulos de diálogo para las pantallas de selección en un programa ABAP. En su lugar, el entorno de proceso lanza eventos específicos durante los procesos PBO y PAI procesando la pantalla, que pueden ser capturados en los correspondientes bloques de eventos durante el proceso de pantalla de selección.

4. Tipos de pantallas de selección

- Como pantallas normales

```
SELECTION-SCREEN BEGIN OF SCREEN dynnr [TITLE title] [AS  
WINDOW].
```

```
...
```

```
SELECTION-SCREEN END OF SCREEN dynnr.
```

- Como subpantallas de las pantallas de selección

```
SELECTION-SCREEN BEGIN OF SCREEN dynnr AS SUBSCREEN [NO  
INTERVALS]
```

```
LEVEL n].
```

```
...
```

```
SELECTION-SCREEN END OF SCREEN dynnr.
```

4.1. Pantallas de selección normales

Estas sentencias pueden ser ejecutadas en la sección de declaraciones globales de programas ejecutables, grupos de funciones y module pools. Podemos crear una pantalla de selección aislada con el número de pantalla dynnr. Debemos especificar el número de pantalla directamente y debe comprender un máximo de cuatro dígitos.

Todas las sentencias PARAMETERS, SELECT-OPTIONS, y SELECTION-SCREEN que son ejecutadas dentro de estas sentencias definen los



elementos de pantalla para las pantallas de selección. No podemos definir más pantallas de selección dentro de la definición de una pantalla de selección.

Podemos usar TITLE para definir un título para la barra de títulos de nuestra pantalla de selección.

Podemos usar AS WINDOW para definir una pantalla de selección aislada para mostrarse en un cuadro de diálogo modal. El marco actual de la ventana no está definido hasta que no sea accedido por CALL SELECTION-SCREEN. AS WINDOW también mostrará advertencias y mensajes de error que ocurran durante el proceso de un evento de pantalla de selección como un cuadro de diálogo modal.

El número de pantalla especificado dynnr no debe haber sido asignado previamente a pantallas o pantallas de selección ya existentes.

4.2. Subpantallas

Estas sentencias sirven también para crear pantallas de selección, pero aquí éstas son definidas como subpantallas de pantallas. Como todas las subpantallas de pantallas, las pantallas de selección creadas así pueden ser incluidas en otras pantallas o pantallas de selección, o en áreas de subpantallas y elementos de página. Sin embargo, no pueden ser accedidas con CALL SELECTIN-SCREEN. Deben ser integradas dentro de pantallas de selección usando la adición TABBED BLOCK en la sentencia SELECTION-SCREEN.



5. Ejemplos básicos

5.1. Ejemplo 1 - Creación de campos

The screenshot shows a SAP Selection Screen window. The title bar reads "Selection Screen". Below the title bar is a green checkmark icon. The main area contains a text input field with the label "NOMBRE" to its left. The input field is highlighted with a yellow background and a red border.

Código Fuente:

```
SELECTION-SCREEN BEGIN OF SCREEN 0100 TITLE titulo.  
PARAMETERS nombre TYPE sy-uname.  
SELECTION-SCREEN END OF SCREEN 0100.  
  
INITIALIZATION.  
    titulo = 'Selection Screen'.  
  
START-OF-SELECTION.  
    CALL SELECTION-SCREEN '0100'.  
    WRITE: 'hola', nombre.
```

5.2. Ejemplo 2 – Ubicar una pantalla

This screenshot is identical to the one in the previous block, showing a SAP Selection Screen window with the title "Selection Screen", a green checkmark icon, and a text input field labeled "NOMBRE".



```
SELECTION-SCREEN BEGIN OF SCREEN 0100 TITLE Título.  
  PARAMETERS Nombre TYPE sy-uname.  
SELECTION-SCREEN END OF SCREEN 0100.
```

```
INITIALIZATION.  
  Título = 'Selection Screen Example'.
```

```
START-OF-SELECTION.  
  CALL SELECTION-SCREEN '0100' STARTING AT 10 10.  
  write: 'hola', Nombre.
```

5.3. Ejemplo 3 – Pestañas

The screenshot shows a SAP selection screen with two tabs at the top: 'Datos del Profesor' and 'Datos del Alumno'. The 'Datos del Alumno' tab is selected. Below the tabs, there are three input fields labeled P1, P2, and P3. The P1 field is highlighted with a yellow background and a red border, indicating it is the current focus. The rest of the screen is empty.

```
SELECTION-SCREEN BEGIN OF SCREEN 100 AS SUBSCREEN.  
PARAMETERS: p1 TYPE c LENGTH 10,  
             p2 TYPE c LENGTH 10,  
             p3 TYPE c LENGTH 10.  
SELECTION-SCREEN END OF SCREEN 100.
```

```
SELECTION-SCREEN BEGIN OF SCREEN 200 AS SUBSCREEN.  
PARAMETERS: q1 TYPE c LENGTH 10,  
             q2 TYPE c LENGTH 10,  
             q3 TYPE c LENGTH 10.  
SELECTION-SCREEN END OF SCREEN 200.
```

```
SELECTION-SCREEN: BEGIN OF TABBED BLOCK mytab FOR 10 LINES,  
                  TAB (20) button1 USER-COMMAND push1  
                      DEFAULT SCREEN 100,  
                  TAB (20) button2 USER-COMMAND push2  
                      DEFAULT SCREEN 200,  
                  END OF BLOCK mytab.
```

```
INITIALIZATION.
```




```
button1 = 'Datos del Profesor'.  
button2 = 'Datos del Alumno'.  
  
START-OF-SELECTION.  
  WRITE: / 'Nombre del profesor: ',    p1,  
         / 'Apellido1 del profesor: ', p2,  
         / 'Apellido2 del profesor: ', p3.  
  ULINE.  
  WRITE: / 'Nombre del alumno: ',      q1,  
         / 'Apellido1 del alumno: ',   q2,  
         / 'Apellido2 del alumno: ',   q3.
```



6. Elementos de la pantalla

Crearemos los campos de entrada de la pantalla de selección usando los comandos **PARAMETERS** y **SELECT-OPTIONS**. Por defecto, cada uno de ellos crea su propia fila en la pantalla de selección. Las variantes del comando **SELECTION-SCREEN** vista anteriormente permiten crear pantallas de selección diferentes. Con ellas, podemos definir otros elementos de pantalla e influir en el orden de los elementos de la pantalla.

Algún elemento de pantalla de una pantalla de selección puede ser asignado usando la adición **MODIF ID** modification groups. Los grupos de modificaciones están enlazados a la columna group1 en la tabla de sistema screen. Los grupos de modificaciones que son asignados a las columnas group2 y group3 de la tabla de sistema screen son establecidos por el sistema cuando una pantalla de selección es generada. El group4 es sólo para uso interno.

El group2 contiene el valor "DBS" para los elementos de pantalla que están definidos en la base de datos lógica.

El group3 puede contener los valores de la siguiente tabla:

Abreviatura	Significado del elemento de pantalla
BLK	Marco de título de un bloque
COF	Campo de salida que es enlazado con un parámetro o criterio de selección con la adición FOR FIELD.
COM	Campo de salida que no está enlazado a un parámetro o criterio de selección.
HGH	Campo de entrada para el límite superior de intervalo de un criterio de selección.
ISX	Campo de entrada de un parámetro que está enlazado a una búsqueda de ayuda mediante la adición de AS SEARCH PATTERN.
LOW	Campo de entrada para el límite inferior de un criterio de selección.
OPU	Icono para la opción de selección de un criterio de selección.
PAR	Campo de entrada para un parámetro.
PBU	Push button.
TAB	Título de pestaña
TOT	Campo de salida para texto en frente de un campo de entrada para un parámetro o el límite inferior de un intervalo de un criterio de selección.
TST	Tabstrip.
TXT	Campo de salida para texto en frente de un campo de entrada de un parámetro del límite inferior de un intervalo para un criterio de selección.
ULI	Raya horizontal.
VPU	Pushbutton para una selección múltiple de un criterio de selección.

La anchura máxima de una pantalla de selección es de 83 columnas. Si se excediera, la pantalla se cortarían o no sería generada. En el caso de bloques con marco, la anchura máxima se reduce por el ancho del marco.



Todos los elementos de pantalla que pueden ser generados usando el comando SELECTION-SCREEN se localizan en un nombre-espacio, incluso si fueron creados en pantallas de selección diferentes

6.1. SKIP

```
SELECTION-SCREEN SKIP [n] [ldb_additions].
```

Esta sentencia crea n líneas vacías en la pantalla de selección actual entre las líneas ya rellenas. Si no especificamos n, sólo se creará una línea vacía.

6.2. ULINE

```
SELECTION-SCREEN ULINE [[/][pos](len)] [MODIF ID modid]  
[ldb_additions].
```

Esta sentencia crea una línea horizontal de la pantalla de selección actual. Si no especificamos el formato [[/][pos](len)] el sistema creará en todo lo ancho debajo de las líneas ya rellenas.

6.3. COMMENT

```
SELECTION-SCREEN COMMENT [[/][pos](len)  
{text|{[text] FOR FIELD sel}}  
[VISIBLE LENGTH vlen]  
[MODIF ID modid]  
[ldb_additions].
```

Esta sentencia crea un campo de salida en la pantalla de selección actual y entra el contenido de text en este campo. Podemos usar TEXT-XXX en text.

Si usamos FOR FIELD sel el campo de salida está enlazado a un parámetro o criterio de selección sel. del mismo programa definido por PARAMETERS o SELECT-OPTIONS. Su nombre sel debe ser especificado directamente. Cuando este enlace sea hecho, el campo de ayuda para sel se muestra



cuando el usuario selecciona el campo de salida usando las teclas de funciones F1 y F4.

MODIF ID asigna el campo de salida al grupo de modificaciones modif, que es asignado a la columna group1 de la tabla de sistema screen. Por lo tanto, puede ser modificado usando la sentencia MODIFY SCREEN antes de que la pantalla de selección sea mostrada.

6.4. PUSHBUTTON

```
SELECTION-SCREEN PUSHBUTTON [/][pos](len) button_text  
USER-COMMAND fcode  
[VISIBLE LENGTH vlen]  
[MODIF ID modid]  
[ldb_additions].
```

Esta sentencia crea un pushbutton en la pantalla de selección actual. El texto en el botón está determinado por el contenido de button_text.

Puedes usar el módulo de funciones ICON_CREATE para asignar un icono, un texto de información rápida y un texto correspondiente al botón. Para hacer esto es necesario especificar una longitud len adecuada para el botón por lo que el icono puede ser mostrado internamente y ajustado la longitud visible con VISIBLE LENGTH.

Una vez que el bloque de eventos en AT SELECTION-SCREEN ha sido procesado, el sistema suele volver a mostrar la ventana de selección. Para salir del procesamiento de la pantalla de selección y continuar la ejecución del programa, sólo podemos elegir Ejecutar o Cancelar. Esto significa que los pushbuttons en las pantallas de selección están pensadas para modificaciones dinámicas más que para controlar el programa.

6.5. SCREEN-LINE

```
SELECTION-SCREEN BEGIN OF LINE.
```

```
...
```



```
[SELECTION-SCREEN POSITION pos [ldb_additions]].
```

```
...
```

```
SELECTION-SCREEN END OF LINE.
```

La primera y última sentencias definen una nueva línea debajo de los elementos existentes en la pantalla de selección.

Todos los elementos de pantalla dentro de estas sentencias que estén definidos con las sentencias PARAMETERS, SELECT-OPTIONS y SELECTION-SCREEN son colocados en esta línea uno después del otro sin separación.

6.6. SCREEN-BLOCK

```
SELECTION-SCREEN BEGIN OF BLOCK block
```

```
[WITH FRAME [TITLE title]]
```

```
[NO INTERVALS].
```

```
...
```

```
SELECTION-SCREEN END OF BLOCK block.
```

Estas sentencias definen un bloque con el nombre block en la pantalla de selección actual. Debemos especificar el nombre block directamente y debe contener un máximo de 20 caracteres.

Todos los elementos de pantalla definidos con las sentencias PARAMETERS, SELECT-OPTIONS, y SELECTION-SCREEN entre estas sentencias forman parte del bloque block. Se pueden crear bloques adicionales dentro de la definición del bloque.

6.7. TABBED-BLOCK

```
SELECTION-SCREEN BEGIN OF TABBED BLOCK tblock FOR n LINES.
```

```
...
```

```
[SELECTION-SCREEN TAB (len) tab USER-COMMAND fcode
```

```
[DEFAULT [PROGRAM prog] SCREEN dynnr].]
```



...

SELECTION-SCREEN END OF BLOCK tblock.

La primera y última sentencias definen un área tabstrip con el nombre tblock en la pantalla de selección actual. El nombre tblock debe ser especificado directamente y puede contener un máximo de 16 caracteres. El número de líneas en el área tabstrip está determinado por un número n, que debe ser especificado directamente y puede contener un máximo de tres caracteres, pero no debe exceder de 197.

Dentro de las sentencias que definen un área tabstrip, sólo pueden estar las sentencias SELECTION-SCREEN con una adición TAB, y sólo puede ser usado en esta localización. Estas sentencias definen títulos tab con el nombre tab y la longitud len. Los nombres tab deben ser especificados directamente y pueden contener un máximo de 8 caracteres. Las longitudes len deben ser especificados directamente como números positivos con un máximo de dos caracteres, que no debe exceder de 79. Si el ancho de todos los títulos tab es mayor que el ancho del área, el sistema automáticamente establece una barra de desplazamiento así que podremos acceder a todas las páginas tab.

El sistema crea automáticamente un tipo de variable global c del mismo nombre y con una longitud de 83 para cada título tab. El contenido de las variables es mostrado como una etiqueta para el título tab en una pantalla de selección.

Si no se incluye una sentencia SELECTION-SCREEN con una adición TAB dentro de las sentencias de definición de las áreas tab, un área tab será definido sin un título tab. Este caso especial actúa como una definición de un área de subpantalla para la integración de un único subscreendynpro de la pantalla de selección. El dynpro de subpantalla es asignado dinámicamente.

6.8. FUNCTION KEY

SELECTION-SCREEN FUNCTION KEY n [ldb_additions].



En el estado GUI de la pantalla de selección establecida por el sistema, la barra de herramientas de aplicaciones contiene cinco pushbuttons inactivos, a los que los códigos de función FC01 a FC05 son asignados. Esta sentencia activa el pushbutton del código de función FC0n, por lo que un valor entre 1 y 5 debe ser metido en n.

Para activar el uso de los pushbuttons, la sentencia TABLES debe ser usado para declarar un interfaz de área de trabajo de la estructura SSCRFIELDS desde el diccionario ABAP.

Si se asigna un texto al componente functxt_On del área de interfaz sscrfields antes de que se llame a la pantalla de selección, este texto es mostrado en un pushbutton. De otro modo, el pushbutton no contendrá ningún texto.

Cuando el usuario elige un pushbutton en la barra de herramientas, el entorno de ejecución levanta un evento AT SELECTION-SCREEN y el código de la función correspondiente es transferido al componente comm del interfaz de un área de trabajo sscrfields.

7. Ejemplos complejos

7.1. Ejemplo 1 – Radio buttons

The screenshot shows a SAP Selection Screen titled 'Pantalla de Selección'. Below the title bar, there is a horizontal line and a prompt 'Seleccione una opción'. Underneath the prompt, there are three radio buttons labeled R1, R2, and R3. The R3 radio button is selected, indicated by a filled circle. Below the radio buttons, there is another horizontal line.

Código Fuente:

```
SELECTION-SCREEN COMMENT /1(50) comm1 MODIF ID mg1.  
SELECTION-SCREEN ULINE.  
SELECTION-SCREEN SKIP.  
  
SELECTION-SCREEN COMMENT /1(30) comm2.  
SELECTION-SCREEN ULINE /1(50).
```



```
PARAMETERS: r1 RADIOBUTTON GROUP rad1,  
            r2 RADIOBUTTON GROUP rad1,  
            r3 RADIOBUTTON GROUP rad1.  
SELECTION-SCREEN ULINE /1(50).  
  
AT SELECTION-SCREEN OUTPUT.  
  comm1 ='Pantalla de Selección'.  
  comm2 ='Seleccione una opción'.  
  LOOP AT SCREEN.  
    IF screen-group1 = 'MG1'.  
      screen-intensified = '1'.  
      MODIFY SCREEN.  
    ENDIF.  
  ENDLOOP.  
  
START-OF-SELECTION.  
  CASE 'X'.  
    WHEN r1.  
      WRITE 'Primera opción'.  
    WHEN r2.  
      WRITE 'Segunda opción'.  
    WHEN r3.  
      WRITE 'Tercera opción'.  
  ENDCASE.
```

7.2. Ejemplo 2 – Push button



Código Fuente:

```
TABLES sscrfields.  
TYPE-POOLS icon.  
  
SELECTION-SCREEN:  
  BEGIN OF SCREEN 0100 AS WINDOW TITLE title,  
    PUSHBUTTON 2(10)  but1 USER-COMMAND cli1,  
    PUSHBUTTON 12(40) but2 USER-COMMAND cli2  
                                VISIBLE LENGTH 10,  
  END OF SCREEN 0100.  
  
AT SELECTION-SCREEN.
```




* El nombre del comando de usuario debe ir en Mayúsculas

```
CASE sscrfields.
  WHEN 'CLI1'.
    MESSAGE 'Has pulado el botón 1' type 'I'.
  WHEN 'CLI2'.
    MESSAGE 'Has pulado el botón 2' type 'I'.
ENDCASE.
```

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

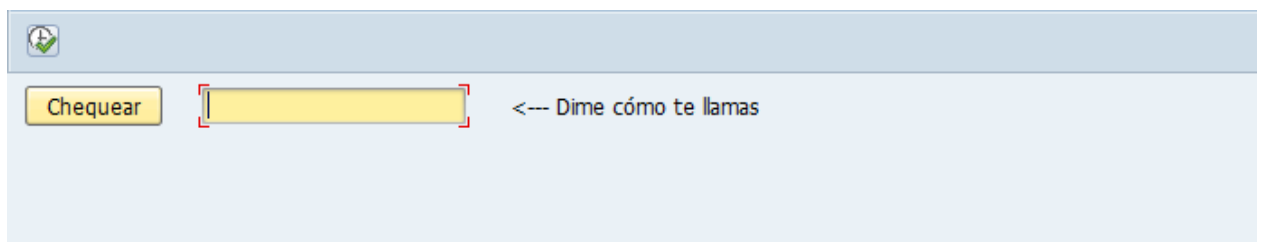
```
title  = 'Pantalla de selección - Push button'.
but1   = 'Button 1'.
```

```
CALL FUNCTION 'ICON_CREATE'
  EXPORTING
    name  = ICON_ANNOTATION
    text  = 'Button 2'
    info  = 'Ejemplo de Quickinfo'
  IMPORTING
    RESULT = but2
  EXCEPTIONS
    OTHERS = 0.
```

* Con STARTING AT 10 10 crea una ventana modal

```
CALL SELECTION-SCREEN '0100'.
```

7.3. Ejemplo 3 – Tratamiento de datos



Código Fuente:

```
TABLES sscrfields.
DATA: vcsaludo TYPE string.

SELECTION-SCREEN: BEGIN OF LINE,
                  PUSHBUTTON 2(10) pbboton USER-COMMAND fucodigo,
                  POSITION 16.
PARAMETERS nombre TYPE c LENGTH 20.
SELECTION-SCREEN: COMMENT 40(40) text,
                  END OF LINE.
```



```

AT SELECTION-SCREEN.
CASE sscrfields.
  WHEN 'FUCODIGO'. " Esto tiene que ir en mayúsculas
    IF nombre IS INITIAL.
      MESSAGE 'Tienes que rellenar el campo' TYPE 'E'.
    ELSE.
      CONCATENATE 'hola' nombre INTO vcsaludo SEPARATED BY ' '.
      MESSAGE vcsaludo TYPE 'I'.
    ENDIF.
  ENDCASE.

INITIALIZATION.
  pbboton = 'Chequear'.
  text    = '<--- Dime cómo te llamas'.

START-OF-SELECTION.
  IF nombre IS NOT INITIAL.
    CONCATENATE 'hola' nombre INTO vcsaludo SEPARATED BY ' '.
    WRITE vcsaludo.
  ELSE.
    MESSAGE 'Tienes que rellenar el campo' TYPE 'E'.
  ENDIF.

```

7.4. Ejemplo 4 – Radio buttons con sentencia de bifurcación

Código Fuente:

```

SELECTION-SCREEN BEGIN OF BLOCK rad1
                        WITH FRAME TITLE title.
PARAMETERS: r1 RADIOBUTTON GROUP rad1,
             r2 RADIOBUTTON GROUP rad1,
             r3 RADIOBUTTON GROUP rad1.
SELECTION-SCREEN END OF BLOCK rad1.

INITIALIZATION.
  title = 'Qué color es el que más te gusta'.

START-OF-SELECTION.

CASE 'X'.
  WHEN r1.
    WRITE 'Te gustan el rojo'.
  WHEN r2.
    WRITE 'Te gustan el azul'.

```



```

    WHEN r3.
        WRITE 'Te gustan el verde'.
    ENDCASE.

```

7.5. Ejemplo 5 – Iconos

Teléfono Transporte	
Teléfono	912345678
Medio de transporte	Avion

Código Fuente:

```

TYPE-POOLS icon.
TABLES sscrfields.
DATA: functxt1 TYPE smp_dyntxt,
      functxt2 TYPE smp_dyntxt.

PARAMETERS: p_telef(9) TYPE c,
             p_trans(12) TYPE c.
SELECTION-SCREEN: FUNCTION KEY 1,
                  FUNCTION KEY 2.

INITIALIZATION.
    functxt1-icon_id   = icon_phone.
    functxt1-quickinfo = 'Teléfono predeterminado'.
    functxt1-icon_text = 'Teléfono'.
    sscrfields-functxt_01 = functxt1.
    functxt2-icon_id   = icon_transport.
    functxt2-quickinfo = 'Medio de transporte predeterminado'.
    functxt2-icon_text = 'Transporte'.
    sscrfields-functxt_02 = functxt2.

AT SELECTION-SCREEN.
    CASE sscrfields-ucomm.
        WHEN 'FC01'.
            p_telef = '912345678'.
        WHEN 'FC02'.
            p_trans = 'Avion'.
        WHEN OTHERS.
            "...
    ENDCASE.

START-OF-SELECTION.
    WRITE: 'Tu teléfono es:', p_telef.
    WRITE: / 'Tu transporte :', p_trans.

```



Los iconos están disponibles en el sistema en la tabla ICON:

Entrada en tabla Iratar Pasar a Opciones Utilidades Entorno Sistema Ayuda

SE11

Data Browser: Tabla ICON 200 aciertos

Tabla: ICON
Campos visualiz.: 15 De 15 Columnas clave fijas: 1 AnchoLista 0250

	ID	NAME	OLENG	BUTTON	STATUS	MESSAGE	FUNCTION	TEXTFIELD	INTERNAL
		ICON_DUMMY	2	X	X	X	X	X	
		ICON_CHECKED	2		X				
		ICON_INCOMPLETE	2		X				
		ICON_FAILURE	2		X		X		
		ICON_POSITIVE	2		X				
		ICON_NEGATIVE	2		X				
		ICON_LOCKED	2	X	X		X		
		ICON_UNLOCKED	2	X	X		X		
		ICON_GREEN_LIGHT	4		X				
		ICON_YELLOW_LIGHT	4		X				
		ICON_RED_LIGHT	4		X				
		ICON_TOTAL_LEFT	2	X			X		
		ICON_TOTAL_RIGHT	2	X			X		
		ICON_COLUMN_LEFT	2	X			X		
		ICON_COLUMN_RIGHT	2	X			X		



8. Incluir objetos

8.1. Incluir parámetros

```
SELECTION-SCREEN INCLUDE PARAMETERS para
                                [OBLIGATORY [OFF]]
                                [MODIF ID modid]
                                [ID id].
```

8.2. Incluir áreas de selección

```
SELECTION-SCREEN INCLUDE SELECT-OPTIONS selcrit
                                [OBLIGATORY [OFF]]
                                [NO INTERVALS [OFF]]
                                [NO-EXTENSIONS [OFF]]
                                [MODIF ID modid]
                                [ID id].
```

8.3. Incluir campos de salida

```
SELECTION-SCREEN INCLUDE COMMENT [/][pos](len) text
                                [FOR FIELD sel]
                                [MODIF ID modid]
                                [ID id].
```

8.4. Incluir pushbutton

```
SELECTION-SCREEN INCLUDE PUSHBUTTON [/][pos](len) button_text
                                [USER-COMMAND ucom]
                                [MODIF ID modid]
                                [ID id].
```

8.5. Incluir bloques



```
SELECTION-SCREEN INCLUDE BLOCKS block [ID id].
```

Estas variantes de la sentencia SELECTION-SCREEN permiten que elementos que ya han sido creados en otras pantallas de selección en el mismo programa sean incluidos en una pantalla de selección. Cualquier parámetro, criterio de selección, campo de salida, pushbutton, o todo el bloque entero puede ser transferido. Pueden ser incluidos en toda y desde todas las pantallas de selección de los elementos del programa. Ningún elemento puede ser incluido dentro de la misma pantalla de selección. Un elemento no puede ser usado más de una vez en la misma pantalla de selección. La definición de la pantalla de selección desde la que los elementos son transferidos deben ser organizados antes de la pantalla de selección actual.

9. Opciones ldb (base de datos lógica)

9.1. Versiones de la pantalla de selección estándar

```
SELECTION-SCREEN BEGIN OF VERSION vers text.  
  
...  
SELECTION-SCREEN EXCLUDE {PARAMETERS para}  
    | {SELECT-OPTIONS selcrit}  
    | {RADIOBUTTON GROUPS radi}  
    | {BLOCKS block}  
    | {IDS id}.  
  
...  
SELECTION-SCREEN END OF VERSION vers.
```

9.2. Nodos para campos de selección

```
SELECTION-SCREEN FIELD SELECTION  
    FOR {NODE|TABLE} node [ID id].
```



9.3. Nodos para selecciones libres

SELECTION-SCREEN DYNAMIC SELECTIONS

FOR {NODE|TABLE} node [ID id].

Una pantalla de selección estándar puede ser definida en una base de datos lógica. La pantalla de selección estándar de una base de datos lógica está agrupada junto con la pantalla de selección estándar de un programa ejecutable que está enlazado a la base de datos lógica, para formar una pantalla de selección estándar compartida.

Definimos una pantalla de selección estándar de una base de datos lógica en el include selección de una base de datos lógica. El nombre de este programa include es DBldbSEL, donde ldb es el nombre de la base de datos lógica. Los elementos de la pantalla de selección estándar de una base de datos lógica están definidos usando los comandos usuales: PARAMETERS, SELECT-OPTIONS y SELECTION-SCREEN. Las variantes mencionadas arriba del comando SELECTION-SCREEN están pensadas para el uso en bases de datos lógicas y pueden ser usadas sólo en el include selección.

10. Parámetros

```
PARAMETERS {para[{len}]}|{para [LENGTH len]}  
            [type_options]  
            [screen_options]  
            [value_options]  
            [ldb_options].
```

La declaración de un parámetro para de longitud len. Los parámetros están compuestos de una pantalla de selección que está asignada a un objeto de datos elemental global en el programa ABAP y un campo de entrada en la pantalla de selección.

El nombre del parámetro debe contener un máximo de ocho caracteres. Esta sentencia está permitida en las declaraciones globales de un programa



ejecutable, grupos de funciones y module pools. En grupos de funciones y module pools sólo está permitido dentro de la definición de una pantalla de selección independiente. En los programas ejecutables está automáticamente asignado a las pantallas de selección estándar.

La longitud len sólo puede ser especificada si el tipo de datos especificado en type_options es genérico según la longitud (c, n, p, x). La longitud len debe ser especificada como un literal o una constante numérica dentro del correspondiente rango de intervalos. Si no se especifica len, la longitud se establece en 1 para un tipo de datos genérico, o a la longitud del tipo de datos. Desde la Release 6.10, es posible especificar la longitud con la adición LENGTH.

En detalle, la sentencia PARAMETERS tiene el siguiente efecto:

Declara una variable global para de longitud determinada en el programa. El tipo del objeto de datos está especificado en type_options.

En la pantalla de selección actual, un campo de entrada con el mismo nombre y tipo de datos externo conveniente es creado en una nueva línea en la posición 35. La longitud del campo de entrada está alineada con la longitud del parámetro. La longitud máxima será de 132, mientras que la visible está entre 39 y 45.

Antes de que se envíe la pantalla de selección, el contenido de para es transportado al campo de entrada en la pantalla de selección y puede ser ejecutada una rutina de conversión.

Si los parámetros son definidos en un include de selección en la base de datos lógica, las adiciones ldb_options son necesarias o posibles.

Tipos:

```
... { TYPE type [DECIMALS dec] }  
  | { LIKE dobj }  
  | { LIKE (name) } ... .
```

Aquí definimos el tipo de datos del parámetro. Si no se especifica esta sección, el parámetro tendrá el tipo c. El tipo de datos puede ser definido desde una referencia estática a un tipo de datos existente type, pasando por una referencia estática a un objeto de datos dobj, o una referencia dinámica a un tipo de datos desde el diccionario ABAP en name.



11. SCREEN-OPTIONS

```
... { {[OBLIGATORY|NO-DISPLAY] [VISIBLE LENGTH vlen]}  
    | {AS CHECKBOX [USER-COMMAND fcode]}  
    | {RADIOBUTTON GROUP group [USER-COMMAND fcode]}  
    | {AS LISTBOX VISIBLE LENGTH vlen [USER-COMMAND fcode]}  
        [OBLIGATORY]} }  
  
[MODIF ID modid] ... .
```

Extras:

1. ... OBLIGATORY
2. ... NO-DISPLAY
3. ... VISIBLE LENGTH vlen
4. ... AS CHECKBOX [USER-COMMAND fcode]
5. ... RADIOBUTTON GROUP group [USER-COMMAND fcode]
6. ... AS LISTBOX VISIBLE LENGTH vlen [USER-COMMAND fcode]
7. ... MODIF ID modid

Estas adiciones pueden usarse para declarar el campo de entrada de tipo obligatorio, ocultar un campo en la pantalla de selección, y determinar la longitud visible del campo. El campo de entrada puede ser un checkbox, un radio button, o una lista desplegable. Todos los elementos de pantalla de un parámetro pueden ser asignados a un grupo de modificaciones.

OBLIGATORY. Indica qué campo es obligatorio. Si no se ingresa ningún dato en este campo, no se podrá ejecutar, tan sólo usar las funciones Atrás, Salir, o Cancelar.

NO-DISPLAY. Impide generar elementos de pantalla para el parámetro en la pantalla de selección. En un programa ejecutable, un parámetro de este tipo se usa exclusivamente en una interfaz definida por una pantalla de selección. Este campo podrá tener cualquier tipo de datos excepto por referencias. Sólo se les podrá suministrar valores usando la adición WITH con la sentencia SUBMIT. No está restringido a los 132 caracteres de los parámetros de campos de entrada.



VISIBLE LENGTH vlen. Define la longitud visible de un campo de entrada en vlen.

AS CHECKBOX [USER-COMMAND fcode]. Indica que el campo de entrada será de tipo checkbox con la correspondiente descripción a su derecha. Un checkbox está seleccionado sólo si tiene valor "X" o "x". El parámetro deberá ser definido como tipo c de longitud 1.

Podemos usar USER-COMMAND para asignar un código de función fcode al parámetro. Para evaluar el código de función, debemos declarar una interfaz de área de trabajo de la estructura SSCRFIELDS del diccionario ABAP usando la sentencia TABLES. Cuando el usuario seleccione el checkbox, se lanza un evento AT SELECTION-SCREEN y se transfiere el código de función fcode al componente ucomm del interfaz del área de trabajo sscrfields.

RADIOBUTTON GROUP group [USER-COMMAND fcode]. Indica que el campo de entrada es mostrado como un radio button, y el campo de salida es mostrado a su derecha. Estará seleccionado sólo si su valor fuera "x" o "X".

AS LISTBOX VISIBLE LENGTH vlen [USER-COMMAND fcode]. Genera una lista de selección en el campo de entrada. Si el parámetro fuera creado mediante un tipo de datos del diccionario ABAP, y el tipo de datos fuera enlazado a la ayuda del diccionario, la primera columna de la ayuda se muestra en la lista desplegable. De otro modo, sólo se muestra una sola línea con el valor actual del parámetro.

MODIF ID modid. Asigna todos los parámetros de elementos de pantalla al grupo de modificaciones modid, que es asignado a la columna group1 de la tabla de sistema screen. Esto los activa para ser modificados antes de que la pantalla de selección sea mostrada, usando la sentencia MODIFY SCREEN. El nombre del grupo de modificaciones group modid debe ser directamente especificado y puede tener una longitud máxima de tres caracteres.

12. VALUE-OPTIONS

```
... [DEFAULT val]  
[LOWER CASE]
```



```
[MATCHCODE OBJECT search_help]
[MEMORY ID pid]
[VALUE CHECK] ... .
```

Extras:

1. ... *DEFAULT val*
2. ... *LOWER CASE*
3. ... *MATCHCODE OBJECT search_help*
4. ... *MEMORY ID pid*
5. ... *VALUE CHECK*

Con estas adiciones podemos definir un valor de inicio que permita minúsculas, una ayuda de búsqueda, parámetros SPA/GPA, o la ejecución de un check contra un listado de valores.

DEFAULT val. Así se crea un valor inicial para el contenido del parámetro para. El valor inicial var puede ser especificado como un literal o como un objeto de datos ya definido.

LOWER CASE. Convierte el contenido de un campo a mayúsculas cuando el campo de entrada de una pantalla de selección es transportado al objeto de datos para en el programa y viceversa.

MATCHCODE OBJECT search_help. Enlaza el campo de entrada de un parámetro a la ayuda de búsqueda search_help del diccionario. El nombre de la ayuda de búsqueda debe ser ingresado directamente.

MEMORY ID pid.

Enlaza el campo de entrada del parámetro win un parámetro SPA/GPA en la memoria de SAP. El nombre del parámetro SPA/GPA debe ser especificado directamente con un máximo de 20 caracteres.

El campo de entrada es rellenado cuando la pantalla de selección es llamada con el valor actual del parámetro SPA/GPA en la memoria de SAP, el objeto de datos para provisto es inicial después del proceso del evento AT SELECTION-SCREEN OUTPUT. De otro modo, el valor de para es mostrado. Si hay una acción del usuario en la pantalla de selección, el contenido del campo de entrada es asignado al parámetro SPA/GPA en la memoria de SAP. Si no existiera parámetro SPA/GPA para el nombre especificado, se crearía.

La adición DEFAULT desactiva la adición MEMORY ID.



VALUE CHECK.

Esta adición sólo puede ser especificada si el tipo de parámetro está definido como referencia a un tipo de datos del diccionario.

El contenido del campo de entrada es chequeado contra los valores fijos posiblemente definidos en el dominio de un tipo de datos. Si el tipo de datos es una componente

de una clave foránea de una tabla, se ejecutaría un chequeo contra esa tabla. Si el chequeo no tuviera éxito, se mostraría un mensaje de error en la línea de estado de la pantalla de selección. Si el programa fue llamado mediante un SUBMIT sin reflejo en la pantalla de selección, se mostraría en caso de error.

El chequeo contra una tabla se ejecutaría aunque el campo estuviera vacío, por lo que sería recomendable definirlo como obligatorio al mismo tiempo.

13. LDB-OPTIONS

```
... FOR {TABLE|NODE} node  
    [HELP-REQUEST]  
    [VALUE-REQUEST]  
    [AS SEARCH PATTERN] ... .
```

Extras:

1. ... FOR {TABLE|NODE} node
2. ... HELP-REQUEST
3. ... VALUE-REQUEST
4. ... AS SEARCH PATTERN

14. SELECT-OPTIONS

```
SELECT-OPTIONS selcrit FOR {dobj| (name)}  
    [screen_options]  
    [value_options]  
    [ldb_options].
```



Con esto declaramos un criterio de selección `selcrit` para un objeto de datos `dobj` o un tipo especificado en `name`. El criterio de selección son componentes de una pantalla de selección, a la que se asigna una tabla de selección en un programa ABAP, así como dos campos de entrada y un `pushbutton` para selecciones múltiples.

La sentencia `SELECT-OPTIONS` declara una tabla de selección en el programa con el nombre `selcrit`. Una tabla de selección es una tabla estándar interna con línea de cabecera y clave estándar, donde podemos guardar múltiples condiciones lógicas. El contenido de las pantallas de selección puede ser analizado en una expresión lógica en la expresión `WHERE` de Open SQL. Las tablas de selección tienen las siguientes cuatro columnas:

Signo de tipo `c` y longitud 1. Su valor determina para cada fila si el resultado de la condición formulada en la columna está incluido o excluido en el conjunto de resultados para todas las filas. Los valores evaluables son "I" para incluir y "E" para excluir.

Option de tipo `c` y longitud 2. Contiene la opción de selección para la condición de la fila en el formulario de operadores lógicos. Los operadores analizables son "EQ", "NE", "GE", "LE", "LT", "CP" y "NP" si la columna `high` está vacía, y "BT", "NB" si no lo está.

Low del tipo de datos definidos después de FOR.

High del tipo de datos definidos después de FOR.