

DIPLOMATURA EN PROGRAMACION ABAP  
MÓDULO 12: ABAP LIST VIEWER

Utilización de ALV  
(Abap List Viewer) para la  
creación y el formato de  
reportes avanzados con  
ABAP

## ALV

A continuación se detallan sus características más salientes.

### Introducción a ALV

SAP provee sets de módulos de funciones ALV (Abap List Viewer), que pueden utilizarse para mejorar el listado de salida de un reporte. Este juego de funciones ALV se utiliza para optimizar la lectura y funcionalidad de cualquier listado. En los casos en que los listados poseen columnas con más de 255 caracteres de longitud, estas funciones permiten la selección y disposición de columnas de un listado de salida, incluso permiten guardar distintas variantes de visualización de un mismo reporte de salida y asignar distintos criterios de ordenamiento.

Las funciones más comunes para el uso de ALV son:

1. REUSE\_ALV\_VARIANT\_DEFAULT\_GET
2. REUSE\_ALV\_VARIANT\_F4
3. REUSE\_ALV\_VARIANT\_EXISTENCE
4. REUSE\_ALV\_EVENTS\_GET
5. REUSE\_ALV\_COMMENTARY\_WRITE
6. REUSE\_ALV\_FIELDCATALOG\_MERGE
7. REUSE\_ALV\_LIST\_DISPLAY
8. REUSE\_ALV\_GRID\_DISPLAY
9. REUSE\_ALV\_POPUP\_TO\_SELECT

### ***Pasos para la utilización de las funciones ALV:***

#### ***Declaración de Datos:***

■ Estándar SAP type pools: SLIS , KKBLO .

■ Tablas estándar de SAP referenciadas al Type Pool:

```
SLIS_LAYOUT_ALV ,  
SLIS_T_FIELDCAT_ALV,  
SLIS_T_LISTHEADER,  
SLIS_T_EVENT,  
SLIS_SELFIELD.
```

- Tablas internas utilizadas en el programa, basadas en los tipos de tablas declaradas anteriormente:

```
DATA:      I_LAYOUT TYPE SLIS_LAYOUT_ALV,
           I_FIELDTAB TYPE SLIS_T_FIELDCAT_ALV,
           I_HEADING TYPE SLIS_T_LISTHEADER,
           I_EVENTS TYPE SLIS_T_EVENT.

TYPES:     KKBLO_SELFIELD TYPE SLIS_SELFIELD.
```

### ***Variante de visualización inicial (opcional):***

Las variantes de visualización pueden ser específicas de usuario o generales. El usuario puede definir la variante inicial para la salida del listado. La variante por default se puede definir en el siguiente módulo de función:

```
'REUSE_ALV_VARIANT_DEFAULT_GET'.
```

Ejemplo:

```
CALL FUNCTION 'REUSE_ALV_VARIANT_DEFAULT_GET'
  EXPORTING
    i_save      = variant save condition ( A=all, U = user-
specific )
  CHANGING
    cs_variant = internal table containing the program name (and
the default variant---optional)
  EXCEPTIONS
    not_found  = 2.
```

El usuario también puede elegir de un listado todas las variantes existentes utilizando el modulo de función 'REUSE\_ALV\_VARIANT\_F4'.

### ***Definir las características de Salida: Preparar el catálogo de los campos de visualización***

Un catálogo de campos se prepara a través de una tabla interna (I\_FIELDCAT) del tipo SLIS\_T\_FIELDCAT\_ALV.

El catálogo de campos contiene las descripciones de los campos de salida del listado, y es obligatorio para cada Listado de salida del tipo ALV para adicionar cualquier tipo de funcionalidad extra a ciertos campos, como ser columnas justificadas, cabeceras específicas de campos, hotspot, etc. Si no se especifican para cada campo, tendrán las características que SAP define por default. Los valores posibles son descriptos más abajo.

El catálogo de campos para la tabla de salida se crea a medida que se ejecuta el código del ALV. La creación puede ser completamente o parcialmente automatizada invocando al módulo REUSE\_ALV\_FIELDCATALOG\_MERGE.

El usuario puede utilizar otros parámetros opcionales para asignar atributos de salida a los campos del listado, que difieren de los default.

No es necesario construir un catálogo de campos bajo las siguientes condiciones:

1. La tabla interna utilizada para el listado de salida tiene la misma estructura que una tabla del Diccionario de ABAP, la cual es referida en la declaración de la tabla interna utilizando LIKE o INCLUDE STRUCTURE. En este caso los atributos de los distintos campos son tomados directamente de las tablas.
2. Todos los campos de la estructura se deben incluir en la salida del reporte.
3. El nombre de la estructura es pasado al ALV en el parámetro I\_STRUCTURE\_NAME del módulo de función REUSE\_ALV\_LIST\_DISPLAY.

Todos los valores ingresados en el catálogo son específicos de un campo en particular cuyo nombre es ingresado en el campo FIELDNAME de la estructura del campo del catálogo. El nombre de la tabla también es ingresado en el campo TABNAME de la estructura.

**Los posibles atributos son:**

- Row\_pos (posición de fila): Sólo es relevante si el listado de salida debe ser de múltiples líneas (dos o tres líneas por default). Este atributo puede ser utilizado para generar un cierto nivel de alineación en la salida. Valores posibles 0-3.
- Col\_pos (posición de columna): Este parámetro es relevante cuando los campos de la salida tienen distintas posiciones que la secuencia definida en la tabla interna para los datos de salida. El orden de las columnas puede modificarse interactivamente por el usuario. Si este parámetro contiene un valor inicial para todos los campos del catálogo, las columnas aparecen en la secuencia en que fueron definidos en la tabla interna. Valores posibles: 0-60.
- Fieldname (Etiqueta del Campo): contiene el nombre del campo de la tabla interna al cual se le asociarán todos los atributos definidos en el catálogo. Valores posibles: nombre del campo de la tabla interna de salida. Es un parámetro obligatorio.
- Tabname (nombre de la tabla interna que contiene los datos de salida): Nombre de la tabla interna de salida que contiene los campos definidos en el campo FIELDNAME. Valores posibles: nombre de la tabla interna de salida o SPACE.
- Ref\_fieldname (campo de referencia): nombre de un campo del diccionario de datos al cual el campo asociado de la tabla de salida, tomará sus atributos. Este parámetro se

utiliza cuando el campo de la tabla de salida tiene un nombre distinto al definido en el Diccionario de Datos. Si los campos son idénticos, referenciar la estructura o tabla del diccionario de datos en el campo del catálogo FIELD CAT-REF\_TABNAME. Valores posibles : SPACE o campo del diccionario de datos.

- Ref\_tabname (Nombre de la estructura o tabla de referencia): Estructura o nombre de tabla del diccionario de datos del campo del catálogo. Este parámetro es utilizado sólo cuando un campo de la tabla interna de salida tiene una tabla de referencia definido en el diccionario de datos. Valores posibles: SPACE o nombre de la estructura o tabla del diccionario de datos.

#### **Link a unidades de Monedas**

- Cfieldname (Unidad de Moneda del Campo): Es utilizado para campos de tipo importes en monedas. Este parámetro contiene el nombre del campo de la tabla interna de salida que tiene asociado una unidad de moneda y que hace referencia a un campo de importe definido en el FIELD CAT-FIELDNAME. El campo en el FIELD CAT-CFIELDNAME debe tener su propia entrada en el catálogo. Valores posibles: SPACE o nombre del campo de la tabla interna de salida.
- Ctablename (Unidad de Moneda del Campo definido en la tabla interna de salida): nombre del campo de la tabla interna que contiene el FIELD CAT-CFIELDNAME. Valores posibles: SPACE o nombre del campo de la tabla interna de salida.

#### **Link a unidades de Medidas**

- Qfieldname (Unidad de Medida del Campo): Es utilizado para campos de tipo cantidades. Este parámetro contiene el nombre del campo de la tabla interna de salida que tiene asociado una unidad de medida y que hace referencia a un campo de cantidad definido en el FIELD CAT-FIELDNAME. El campo en el FIELD CAT-CFIELDNAME debe tener su propia entrada en el catálogo. Valores posibles: SPACE o nombre del campo de la tabla interna de salida.
- Qtablename (Unidad de Medida del Campo definido en la tabla interna de salida): nombre del campo de la tabla interna que contiene el FIELD CAT-CFIELDNAME. Valores posibles: SPACE o nombre del campo de la tabla interna de salida
- Outputlen (ancho de columna): se utiliza cuando se quiere definir una longitud de salida diferente a la definida en la tabla interna de salida de datos. Valores posibles: 0 (se toma la longitud de salida definida en el campo) o N (n caracteres de salida).
- Key (Columna Clave): por default se definen como campos claves aquellos que ya existen como campos claves en el diccionario de datos. Este parámetro permite definir qué columnas del reporte deberán permanecer fijas y que no serán desplazadas u ocultas. Para utilizarlo, el parámetro FIELD CAT-NO\_OUT debe tener valor SPACE. Valores posibles: SPACE, 'X'.

- **Key\_sel** (Columna que puede ocultarse): Este parámetro se utiliza en conjunto con aquellos campos que tienen el campo FIELD CAT-KEY = 'X'. Utilizando este parámetro, la clave puede ocultarse interactivamente. La secuencia de campos claves no puede ser modificada por el usuario. La salida es controlada por el campo FIELD CAT-NO\_OUT parámetro utilizado para campos que no son claves en el listado de salida. Valores posibles: SPACE, 'X'.
- **No\_out** (Campo del Listado de salida): este parámetro permite remover algunos campos en el listado inicial. Sin embargo el usuario puede seleccionarlo desde la variante de visualización del listado. Valores posibles: SPACE, 'X' (no muestra el campo en el listado inicial).
- **Emphasize** (resaltar columnas con color): Valores posibles: SPACE, 'X' or 'Cxyz' (x:'1'-'9'; y,z: '0'=off, '1'=on).
  - 'X' = la columna es resaltada con un color default.
  - 'Cxyz' = la columna se muestra con un color codificado:
    - C: Color (obligatorio)
    - X: código de Color
    - Y: en negrita
    - Z: color inverso.
- **Hotspot** (columna como hot spot): este parámetro permite definir un hot spot. Activa los eventos PICK o F2 cuando el cursor está sobre el campo. Valores posibles: SPACE, 'X' (muestra la columna como un hot spot).
- **Fix\_column** (Columna fija): Permite definir qué columnas no podrán ser modificadas y que permanecerán fijas en el listado de salida. Todas las columnas fijas deben tener este parámetro comenzando desde la izquierda. El usuario puede cambiar el orden de las columnas fijas interactivamente. Valores posibles: SPACE, 'X'.
- **Do\_sum** (sumarizar los valores de la columna): El usuario también puede activar esta funcionalidad interactivamente. Valores posibles: SPACE, 'X' (se sumariza el campo definido de todos los registros de la tabla interna).
- **No\_sum** (sumatoria prohibida): aunque el campo sea numérico o de importe, al setear este parámetro se prohíbe la sumarización del campo. Valores posibles: SPACE, 'X'.
- **Icon** (icono): el campo de la tabla interna de salida debe contener strings que representen códigos de íconos. Valores posibles: SPACE, 'X' (convierte el contenido del campo a un código de icono válido).
- **Symbol** (símbolo): el campo de la tabla interna de salida debe contener strings que representen códigos de símbolos. Valores posibles: SPACE, 'X' (convierte el contenido del campo a un código de símbolo válido).

- **Just (Justificado):** permite justificar el contenido de un campo de tipo C o NUMC. Valores posibles: SPACE, 'R', 'L', y 'C'.
  - ' ' = justificación por default de acuerdo al tipo definido del campo de salida de la tabla interna.
  - 'R' = justificado a la derecha.
  - 'L' = justificado a la izquierda.
  - 'C' = centrado.
- **Lzero (con ceros):** este parámetro sólo válido para campos NUMC, muestra los campos con valor cero, es decir que no se visualizan en blanco. Valores posibles: SPACE, 'X'.
- **No\_sign (sin signos +/-):** permite suprimir el signo del campo en la tabla de salida. Valores posibles: SPACE, 'X'.
- **No\_zero (sin ceros):** este parámetro sólo válido para campos de valor, no muestra los campos con valor cero, es decir que se visualizan en blanco. Valores posibles: SPACE, 'X'.
- **Edit\_mask (formato del campo):** Se aplican los formatos de salida, los mismos que se definen con la sentencia write. Valores posibles: SPACE, template.

Los siguientes parámetros permiten customizar los textos de los encabezados de las columnas del listado de salida. Los textos son tomados del Diccionario de Datos para los campos con una referencia a un campo del Diccionario. También los parámetros de textos pueden ser definidos y se pueden ignorar los definidos en el Diccionario de Datos. Si el usuario modifica el ancho de una columna interactivamente, se utilizará el texto del encabezado de la columna que se ajuste al ancho definido:

“La etiqueta de campo Grande”, es utilizada en las opciones de definición de la variante de visualización del ALV.

- **seltext\_l (Etiqueta de campo grande)**
- **seltext\_m (Etiqueta de campo mediana)**
- **seltext\_s (Etiqueta de campo corta)**
- **reptext\_ddic(encabezado)** para utilizar la descripción del elemento de dato del Diccionario de datos.
- **Ddictxt (texto específico) :** Se utiliza este parámetro para definir qué tipo de etiqueta se utilizará de acuerdo a la definición del campo en el Diccionario de Datos. Valores posibles: SPACE, 'L', 'M', y 'S'.

### **Código de Ejemplo:**

Supongamos que I\_PO\_DETAILS es una tabla interna que contiene dos campos EBELN (número de Orden de Compra) y EBELP (ítem de la orden de compra).

\*estructura que contendrá los parámetros definidos.

```
DATA: L_FIELDCAT TYPE SLIS_FIELDCAT_ALV.
```

\*Para un campo en particular del catálogo de campos

\*En este caso el campo Orden de Compra se convierte

\*en clave del listado,

\*fijo, no desplazable y hot spot.

```
CLEAR L_FIELDCAT.
L_FIELDCAT-TABNAME   = 'I_PO_DETAILS'.
L_FIELDCAT-FIELDNAME = 'EBELN'.
L_FIELDCAT-KEY       = 'X'.
L_FIELDCAT-HOTSPOT   = 'X'.
L_FIELDCAT-SELTEXT_S = 'No O.C.'.
L_FIELDCAT-SELTEXT_M = 'No. Orden C.'.
L_FIELDCAT-SELTEXT_L = 'No de Orden de Compra'.
```

```
APPEND L_FIELDCAT TO I_FIELDTAB.
```

```
CLEAR L_FIELDCAT
```

\*El campo ítem es clave, resaltado y fijo.

```
L_FIELDCAT-TABNAME   = 'I_PO_DETAILS'.
L_FIELDCAT-FIELDNAME = 'EBELN'.
L_FIELDCAT-KEY       = 'X'.
L_FIELDCAT-SELTEXT_S = 'No. Item'.
L_FIELDCAT-SELTEXT_M = 'No. Item`.
L_FIELDCAT-SELTEXT_L = 'Nro. Item OC'.
APPEND L_FIELDCAT TO I_FIELDTAB.
CLEAR L_FIELDCAT
```

De esta forma, el catálogo está listo para el listado de Salida.

### ***Tablas de Eventos:***

El próximo paso es construir una tabla de eventos, utilizadas por las acciones del usuario o por la definición de eventos en el programa, por ejemplo top of page, end of page etc.



Una lista de posibles eventos se almacena en una tabla de eventos (I\_EVENTS) y son capturados por el módulo de función **REUSE\_ALV\_EVENT\_NAMES\_GET**. La tabla de resultados de este módulo contiene todos los eventos que se han realizado.

El módulo de función contiene los siguientes parámetros import/export:

IMPORTING PARAMETERS: I\_LIST\_TYPE

Este parámetro puede tomar los valores 0-4.

El parámetro I\_LIST\_TYPE es del tipo TYPE SLIS\_LIST\_TYPE y su valor DEFAULT es 0.

EXPORTING PARAMETERS: I\_EVENTS table.

Esta tabla es del tipo TYPE SLIS\_T\_EVENT y devuelve al programa el nombre de todos los eventos posibles. La estructura de la tabla contiene los campos:

I\_EVENTS-NAME: nombre del evento invocado.

I\_EVENTS-FORM: nombre de la rutina del programa de control “FORM” que debe ser invocado al momento de detectar el evento.

Sólo los eventos con una rutina asociada son procesados.

La tabla I\_EVENTS devuelve valores con las siguientes constantes:

- Slis\_ev\_item\_data\_expand**                      **TYPE**                      **slis\_formname**                      **VALUE**  
**'ITEM\_DATA\_EXPAND'**. Es relevante sólo para listados con sublistados que utilizan el parámetro IS\_LAYOUT-EXPAND\_FIELDNAME de la estructura IS\_LAYOUT. Este evento permite detectar si un registro cabecera ha sido expandido por el usuario, el detalle se pasará a una tabla interna de detalle (ITEM).
- Slis\_ev\_reprep\_sel\_modify**                      **TYPE**                      **slis\_formname**                      **VALUE**  
**'REPREP\_SEL\_MODIFY'**. RS\_SELFIELD-TABINDEX contiene el índice de la tabla cabecera para el cual se mostrarán los ítems de detalle que se resguardarán en la tabla de salida de detalle (T\_OUTTAB\_SLAVE). RFLG\_ALL es pasado con el valor 'X' si el usuario quiere visualizar los ítems. La aplicación debe asegurarse que no existen registros duplicados en la tabla de detalle. El campo RS\_SELFIELD es inicial en este caso.
- Slis\_ev\_caller\_exit\_at\_start** **TYPE** **slis\_formname** **VALUE** **'CALLER\_EXIT'**. Es invocado al comienzo del módulo de función para realizar seteos especiales, no se utiliza con frecuencia.
- Slis\_ev\_user\_command** **TYPE** **slis\_formname** **VALUE** **'USER\_COMMAND'**. Es un evento comúnmente utilizado. La rutina “FORM” puede incluso ser pasada directamente al módulo de función pasando el comando de usuario en el parámetro IMPORTING I\_CALLBACK\_USER\_COMMAND.

5. **Slis\_ev\_top\_of\_page** TYPE **slis\_formname** VALUE **'TOP\_OF\_PAGE'**. Equivalente al evento de procesamiento de listas TOP-OF-PAGE.
6. **Slis\_ev\_top\_of\_coverpage** TYPE **slis\_formname** VALUE **'TOP\_OF\_COVERPAGE'**. La información seleccionada y una cobertura del listado se muestran en el listado de salida en páginas separadas.
7. **Slis\_ev\_end\_of\_coverpage** TYPE **slis\_formname** VALUE **'END\_OF\_COVERPAGE'**. Análogamente al evento TOP\_OF\_COVERPAGE el usuario puede adicionar otra información a la salida del ALV.
8. **Slis\_ev\_pf\_status\_set** TYPE **slis\_formname** VALUE **'PF\_STATUS\_SET'**. Si una barra de tareas del listado debe ser customizada se realiza a través de la rutina asignada a este evento. Los códigos de función del ALV se pasan a través del parámetro RT\_EXTAB. Esta tabla debe ser pasada con el comando SET PF-STATUS. El estatus STANDARD del grupo de función SALV debe ser utilizado como un template para los status específicos de usuario. Como es un evento de retorno comúnmente usado, su rutina de procesamiento también puede ser pasado al módulo de función en el parámetro de función IMPORTING I\_CALLBACK\_PF\_STATUS\_SET.
9. **Slis\_ev\_list\_modify** TYPE **slis\_formname** VALUE **'LIST\_MODIFY'**. Permite modificar los valores de un listado de salida.  

```
LIST_MODIFY USING R_TABNAME TYPE SLIS_TABNAME  
                 R_INDEX LIKE SY-TABIX  
                 R_INDEX_ITEM LIKE SY-TABIX  
                 R_INDEX_SUM LIKE SY-TABIX.
```
10. **Slis\_ev\_top\_of\_list** TYPE **slis\_formname** VALUE **'TOP\_OF\_LIST'**. La información contenida se mostrará al inicio de la lista.
11. **Slis\_ev\_top\_of\_list** TYPE **slis\_formname** VALUE **'TOP\_OF\_LIST'**. Se muestra información de salida al comienzo del listado.
12. **Slis\_ev\_end\_of\_page** TYPE **slis\_formname** VALUE **'END\_OF\_PAGE'**. Se muestra información de salida al término de cada página del listado.
13. **Slis\_ev\_end\_of\_list** TYPE **slis\_formname** VALUE **'END\_OF\_LIST'**. Se muestra información de salida al término del listado.
14. **Slis\_ev\_after\_line\_output** TYPE **slis\_formname** VALUE **'AFTER\_LINE\_OUTPUT'**. Permite generar información de salida después de cada línea del listado. Sólo debe usarse en casos justificados, ya que no es una solución muy performante.

### **15. Slis\_ev\_before\_line\_output TYPE slis\_formname VALUE**

**'BEFORE\_LINE\_OUTPUT'**. Permite generar información de salida antes de cada línea del listado. Sólo debe usarse en casos justificados, ya que no es una solución muy performante.

Ahora la tabla de eventos I\_EVENTS está seteada con las constantes deseadas. Si el evento deseado se encuentra en esta tabla, entonces el campo correspondiente al FORM NAME es ejecutado con la rutina que contiene el tratamiento para el correspondiente evento.

#### **Otro ejemplo :**

```
FORMNAME_TOP_OF_PAGE TYPE SLIS_FORMNAME VALUE 'TOP_OF_PAGE',  
FORMNAME_END_OF_PAGE TYPE SLIS_FORMNAME VALUE 'END_OF_PAGE',  
FORMNAME_USER_COMMAND TYPE SLIS_FORMNAME VALUE 'USER_COMMAND'.
```

```
DATA: L_I_EVENT TYPE SLIS_ALV_EVENT.
```

```
CALL FUNCTION 'REUSE_ALV_EVENTS_GET'  
  EXPORTING  
    I_LIST_TYPE = 0  
  IMPORTING  
    ET_EVENTS   = I_EVENTS.
```

```
READ TABLE I_EVENTS WITH KEY NAME = SLIS_EV_TOP_OF_PAGE  
  INTO L_I_EVENT.
```

```
IF SY-SUBRC = 0.  
  MOVE FORMNAME_TOP_OF_PAGE TO L_I_EVENT-FORM.  
  APPEND L_I_EVENT TO I_EVENTS.  
ENDIF.
```

```
READ TABLE I_EVENTS WITH KEY NAME = SLIS_EV_END_OF_PAGE  
  INTO L_I_EVENT.
```

```
IF SY-SUBRC = 0.  
  MOVE FORMNAME_END_OF_PAGE TO L_I_EVENT-FORM.  
  APPEND L_I_EVENT TO I_EVENTS.  
ENDIF.
```

```
CLEAR L_I_EVENT.  
READ TABLE I_EVENTS WITH KEY NAME = SLIS_EV_USER_COMMAND  
  INTO L_I_EVENT.
```

```
IF SY-SUBRC = 0.  
  MOVE FORMNAME_USER_COMMAND TO L_I_EVENT-FORM.  
  APPEND L_I_EVENT TO I_EVENTS.  
ENDIF.
```

Este código prepara la tabla de eventos para el programa que contiene el tratamiento del ALV. Entonces, el programa contendrá tres rutinas para tratar estos eventos:

1. FORM TOP\_OF\_PAGE: Esta rutina contendrá el encabezado del listado para cada nueva página que se imprima.  
Utilizando el módulo de función 'REUSE\_ALV\_COMMENTARY\_WRITE', la tabla interna que contenga los encabezados para el evento TOP OF PAGE puede ser transferida al listado de salida. Así mismo, se puede incluir un logo específico al reporte utilizando esta función.
2. FORM END\_OF\_PAGE : Esta rutina contiene todo el tratamiento para el evento END OF PAGE.
3. FORM USER\_COMMAND : Esta rutina contendrá todo el bloque de sentencias para el tratamientos de los comandos de usuario, por ejemplo "Seleccionar Línea", "Modificar", etc.

### ***Layout del Listado (type slis\_layout\_alv\_spec)***

El layout del reporte se construye con la tabla interna I\_LAYOUT.

Los parámetros que se utilizan son:

- Opciones de visualización.
- Excepciones.
- Totales.
- Interacciones.
- Detalle de pantalla.
- Mostrar variantes (solo para listados con sublistados).
- Colores.
- Otros

La tabla de layout es del tipo type slis\_layout\_alv\_spec y tiene los campos descriptos en los puntos siguientes.

### ***Opciones de Visualización:***

- Colwidth\_optimize (1) TYPE c: permite ajustar el ancho de las columnas a su contenido. Valores posibles: SPACE, 'X'.
- No\_colhead(1) TYPE c : este parámetro permite suprimir los encabezados de las columnas. Valores posibles: SPACE, 'X'.
- No\_hotspot(1) TYPE c : los encabezados de las columnas no se muestran como hot spots. Valores posibles: SPACE, 'X'.

- Zebra(1) TYPE c : el reporte de salida se muestra con columnas delineadas. Valores posibles: SPACE, 'X'.
- No\_vline(1) TYPE c : el reporte de salida se muestra con columnas separadas por espacios en blanco. Valores posibles: SPACE, 'X'.
- No\_min\_linesize(1) TYPE c : El ancho del reporte de salida es el mismo al definido. Valores posibles: SPACE (el ancho de salida es de 80 o el valor definido en el parámetro MIN\_LINESIZE, 'X' (el ancho de la línea es igual al ancho del listado).
- Min\_linesize LIKE sy-linsz: El reporte se salida contiene el mínimo ancho de línea posible. El parámetro no\_min\_linesize debe ser igual a SPACE. Valores posibles: 0, 10 – 250.
- Max\_linesize LIKE sy-linsz: Por default el valor máximo es 250. Para modificar este valor, se utiliza este parámetro. Valores posibles: 0, 80 - 1020
- Window\_titlebar LIKE rsmpe-tittext: permite setear un título al listado de salida.
- No\_uline\_hs(1) TYPE c: El encabezado no se muestra subrayado.

### **Excepciones**

- Lights\_fieldname TYPE slis\_fieldname: Un campo de la tabla de salida contiene los códigos de excepciones que deberán mostrarse: Código del campo de la tabla de salida:
  - '1' = semáforo rojo
  - '2' = semáforo amarillo
  - '3' = semáforo verde.Valores posibles SPACE, nombre del campo de la tabla interna de salida.
- Lights\_tabname TYPE slis\_tabname: nombre de la tabla interna de salida que contiene el campo con el parámetro LIGHTS\_FIELDNAME. Si LIGHTS\_FIELDNAME contiene algún valor, este campo además debe ser utilizado en listas con sublistas. Valores posibles: SPACE, nombre de la tabla interna de salida.
- Lights\_rollname LIKE dfies-rollname: permite desplegar la documentación asociada al elemento de dato utilizado. Valores posibles: SPACE, nombre del elemento de datos del diccionario.
- Lights\_condense(1) TYPE c : Si un ítem del listado tiene un campo con un “semáforo en rojo”, cada uno de los subtotales que incluya este ítem también será desplegado con un semáforo en rojo”. Valores posibles: SPACE, 'X'.

### **Sumarización**

- No\_sumchoice(1) TYPE c : solo se permite la sumarización a través del catálogo de campos y nos por la acción del usuario en el listado de salida. Valores posibles SPACE, 'X'.
- No\_totalline(1) TYPE c : remueve la opción de tener totales luego de los subtotales. Valores posibles SPACE, 'X'.
- No\_subchoice(1) TYPE c : No se le permite al usuario subtotalizar campos en forma interactiva. Valores posibles: SPACE, 'X'.
- No\_subtotals(1) TYPE c : No se permite mostrar líneas de subtotales. Valores posibles: SPACE, 'X'.
- Numc\_sum(1) TYPE c : Sólo se pueden totalizar los campos de tipo NUMC.
- No\_unit\_splitting TYPE c: No se realiza una apertura de totales por unidades de medida.
- Totals\_before\_items TYPE c: solo se muestran los totales luego de las líneas de detalle.
- Totals\_only(1) TYPE c : Sólo se muestran las líneas de totales. Valores posibles: SPACE, 'X'.
- Totals\_text(60) TYPE c : Permite incluir textos en la primer columna de las líneas de totales. Valores posibles: SPACE, o string de 60 caracteres.
- Subtotals\_text(60) TYPE c : Permite incluir textos en la primer columna de las líneas de subtotales. Valores posibles: SPACE, o string de 60 caracteres.

### **Interacciones**

- Box\_fieldname TYPE slis\_fieldname: Muestra un checkbox en el campo del listado de salida. Valores posibles: SPACE, nombre del campo de la tabla interna de salida.
- Box\_tabname TYPE slis\_tabname: Nombre de la tabla interna que contiene el campo con el parámetro BOX\_FIELDNAME. Valores posibles: SPACE, nombre de la tabla interna.
- Box\_rollname LIKE dd03p-rollname: definición semántica para el checkbox.
- Expand\_fieldname TYPE slis\_fieldname: Flag de expansión para el campo. El usuario puede ocultar o mostrar ítems cliqueando sobre el símbolo de carpeta (hotspot). La expansión de ítems puede ser controlada por el evento 'ITEM\_DATA\_EXPAND'.

- Hotspot\_fieldname TYPE slis\_fieldname: Permite convertir a un campo en hotspot.
- No\_input(1) TYPE c : Los campos son solo de visualización. Valores posibles: SPACE, 'X'.
- Confirmation\_prompt: Habilita una ventana de confirmación para las funciones “back”, “Exit” o “Cancel”. Valores posibles: SPACE, 'X': para las funciones 'Back (F03)', 'Exit (F15)' o 'Cancel (F12)'.
- Key\_hotspot(1) TYPE c : Muestra los campos con valor (FIELDCAT-KEY = 'X') como hotspots. Valores posibles: SPACE, 'X'.
- No\_scrolling(1) TYPE c : no permite desplazamientos hacia la derecha. Valores posibles: SPACE, 'X'.
- Expand\_all(1) TYPE c : Expande todas las posiciones.

#### ***Pantalla:***

- Detail\_popup(1) TYPE c : muestra el reporte en una ventana emergente, sin top-of-page. Valores posibles: SPACE, 'X'.
- Detail\_initial\_lines(1) TYPE c : Muestra todas las líneas, aunque no tengan valores. Valores posibles: SPACE, 'X'.
- Detail\_titlebar(30) type c : muestra un título para el listado. Valores posibles: SPACE, string de 30 caracteres.

#### ***Variantes de Visualización:***

- Header\_text (20) TYPE c: Texto para una lista principal con sublistas. Valores posibles: SPACE, string de 20 caracteres.
- Item\_text(20) TYPE c : Texto para las sublistas. Valores posibles: SPACE, string de 20 caracteres.
- Default\_item(1) TYPE c : La sublista se muestra directamente. Valores posibles: SPACE, 'X'.

## **Colores**

- **Info\_fieldname TYPE slis\_fieldname:** Permite setear colores individuales a los registros de la tabla interna de salida. Valores posibles: SPACE, nombre del campo de la tabla interna de salida. El contenido del campo debe ser de CHAR 03. El contenido debe ser: 'Cxy':
  - C = color (todos los códigos deben comenzar con 'C')
  - X = número de color ('1'-'9')
  - Y = negrita ('0' = off, '1' = on)
- **Coltab\_fieldname TYPE slis\_fieldname:** Las columnas pueden ser coloreadas individualmente. Asignar el nombre del campo de la tabla interna a este parámetro.

## **Código Ejemplo**

```
I_LAYOUT-f2code          = ws_fcode.
I_LAYOUT-zebra            = 'X'.
I_LAYOUT-colwidth_optimize = 'X'.
I_LAYOUT-no_keyfix        = 'X'.
I_LAYOUT-get_selinfos     = 'X'.
I_LAYOUT-no_hotspot       = 'X'.
I_LAYOUT-no_input         = 'X'.
I_LAYOUT-hotspot_fieldname = FIELDNAME.
I_LAYOUT-no_input         = 'X'.
I_LAYOUT-no_vline         = 'X'.
I_LAYOUT-no_colhead       = ' '.
I_LAYOUT-lights_condense  = ' '.
I_LAYOUT-totals_text      = ' '.
I_LAYOUT-subtotals_text    = ' '.
I_LAYOUT-totals_only      = ' '.
I_LAYOUT-key_hotspot      = 'X'.
I_LAYOUT-detail_popup     = 'X'.
I_LAYOUT-group_change_edit = 'X'.
I_LAYOUT-GROUP_BUTTONS    = 'X'.
```

## ***Datos de Salida del ALV-Sublistas:***

Este paso permite obtener la información correspondiente al sublistado del listado de salida principal y para ello es necesario setear el parámetro LAYOUT-GET\_SELINFO de la estructura de IMPORTING.

Los parámetros que se deben pasar a la tabla IS\_SELHIDE son:



- **mode:**
  - 'R' = Sólo los registros pasados a la tabla IS\_SEL\_HIDE-T\_ENTRIES se muestran en una ventana emergente.
  - 'S' = los datos de la pantalla de salida del listado ALV son reemplazados por el contenido de la tabla interna IS\_SEL\_HIDE-T\_ENTRIES (no se muestran en una ventana emergente).
- **t\_entries:** tabla con la información de salida.
- **t\_entries-mode:**
  - 'A' = muestra la información de selección para el registro tratado en una ventana emergente.
  - 'D' = no muestra la información de selección en una ventana emergente (no se muestran criterios de filtración o parámetros)
- **t\_entries-selname:** (sólo para t\_entries-mode = 'D'): nombre del select option o parameter.

Las siguientes tablas sólo son utilizadas para **t\_entries-mode = 'A'**. Contienen la información que será adicionada al listado de salida.

- **t\_entries-field:** nombre del campo del DDIC para el cual la información de selección es desplegada.
- **t\_entries-table:** nombre de la tabla del DDIC del campo t\_entries-field.
- **t\_entries-stext:** nombre del campo en la ventana emergente. Si los parámetros t\_entries-field y t\_entries-table contienen valores, el texto es tomado directamente del DDIC.
- **t\_entries-valuf:** condición de selección para los valores del 'from' (formato externo).
- **t\_entries-valut** condición de selección para los valores del 'to' (formato externo).
- **t\_entries-sign0:** (I)nclusive (E)xclusive.
- **t\_entries-option:** todos los valores de las opciones de selección para el campo Option permitidos.

### ***Criterio de Ordenamiento de los datos de Salida***

La tabla IT\_SORT se despliega con los criterios de ordenamiento para los distintos campos, en la misma además se puede determinar la forma en que se puede subtotalizar la lista básica.

La tabla interna tiene los siguientes campos:

- spos : Secuencia de ordenamiento
- fieldname: nombre del campo de la tabla interna de salida.
- tabname: Sólo para sublistas, se corresponde con el nombre de la tabla interna.
- up : 'X' = ordenamiento ascendente
- down : 'X' = ordenamiento descendente
- subtot : 'X' = agrupar por subtotales.
- group: '\*' = permite generar un cambio de página con cada grupo de subtotal , 'UL' = genera una línea de separación con cada cambio de grupo.

### ***Paso Final-Preparación del Listado de Salida***

El paso final para la salida del reporte es la invocación de los siguientes módulos de función ALV:

1. REUSE\_ALV\_FIELDCATALOG\_MERGE
2. REUSE\_ALV\_LIST\_DISPLAY

La primera función se utiliza para pasar el catálogo de campos al reporte de salida y asociarlo con los datos contenidos en la tabla interna de salida.

```
FUNCTION reuse_alv_fieldcatalog_merge.
* "-----
* "Lokale Schnittstelle:
* " IMPORTING
* "          VALUE(I_PROGRAM_NAME) LIKE SY-REPID OPTIONAL
* "          VALUE(I_INTERNAL_TABNAME) TYPE SLIS_TABNAME
OPTIONAL
* "          VALUE(I_STRUCTURE_NAME) LIKE DD02L-TABNAME
OPTIONAL
* "          VALUE(I_CLIENT_NEVER_DISPLAY) TYPE
SLIS_CHAR_1
* "          DEFAULT 'X'
* "          VALUE(I_INCLNAME) LIKE TRDIR-NAME OPTIONAL
* " CHANGING
* "          VALUE(CT_FIELDCAT) TYPE SLIS_T_FIELDCAT_ALV
* " EXCEPTIONS
* "          INCONSISTENT_INTERFACE
* "          PROGRAM_ERROR
* "-----
```

### Parámetros IMPORT:

- I\_PROGRAM\_NAME: nombre del programa donde se declara y se trata la tabla interna de salida.
- I\_INTERNAL\_TABNAME: nombre de la tabla interna de salida.
- I\_STRUCTURE\_NAME: estructura de la tabla interna.
- I\_CLIENT\_NEVER\_DISPL: no muestra el contenido de los campos "MANDT", default "X".
- I\_INCLNAME: declaración de datos de estructuras de tipo include.

### Parámetros CHANGING

- CT\_FIELDCAT: catálogo de campos y descripciones de campos.

La segunda función se utiliza para mostrar el contenido de la tabla interna de salida:

```
FUNCTION reuse_alv_list_display.  
* "-----  
-----  
* "Lokale Schnittstelle:  
* " IMPORTING  
* "          VALUE(I_INTERFACE_CHECK) DEFAULT SPACE  
* "          VALUE(I_CALLBACK_PROGRAM) LIKE SY-REPID  
DEFAULT SPACE  
* "          VALUE(I_CALLBACK_PF_STATUS_SET) TYPE  
SLIS_FORMNAME  
* "          DEFAULT SPACE  
* "          VALUE(I_CALLBACK_USER_COMMAND) TYPE  
SLIS_FORMNAME  
* "          DEFAULT SPACE  
* "          VALUE(I_STRUCTURE_NAME) LIKE DD02L-TABNAME  
OPTIONAL  
* "          VALUE(IS_LAYOUT) TYPE SLIS_LAYOUT_ALV  
OPTIONAL  
* "          VALUE(IT_FIELDCAT) TYPE SLIS_T_FIELDCAT_ALV  
OPTIONAL  
* "          VALUE(IT_EXCLUDING) TYPE SLIS_T_EXTAB  
OPTIONAL  
* "          VALUE(IT_SPECIAL_GROUPS) TYPE  
SLIS_T_SP_GROUP_ALV  
* "          OPTIONAL  
* "          VALUE(IT_SORT) TYPE SLIS_T_SORTINFO_ALV  
OPTIONAL  
* "          VALUE(IT_FILTER) TYPE SLIS_T_FILTER_ALV  
OPTIONAL
```

```
*"          VALUE(IS_SEL_HIDE) TYPE  SLIS_SEL_HIDE_ALV
OPTIONAL
*"          VALUE(I_DEFAULT) DEFAULT 'X'
"          VALUE(I_SAVE) DEFAULT SPACE
"          VALUE(IS_VARIANT) LIKE  DISVARIANT
"                      STRUCTURE  DISVARIANT DEFAULT
SPACE
"          VALUE(IT_EVENTS) TYPE  SLIS_T_EVENT OPTIONAL
"          VALUE(IT_EVENT_EXIT) TYPE  SLIS_T_EVENT_EXIT
OPTIONAL
"          VALUE(IS_PRINT) TYPE  SLIS_PRINT_ALV OPTIONAL
"          VALUE(IS_REPREP_ID) TYPE  SLIS_REPREP_ID
OPTIONAL
"          VALUE(I_SCREEN_START_COLUMN) DEFAULT 0
"          VALUE(I_SCREEN_START_LINE) DEFAULT 0
"          VALUE(I_SCREEN_END_COLUMN) DEFAULT 0
"          VALUE(I_SCREEN_END_LINE) DEFAULT 0
"          EXPORTING
"          VALUE(E_EXIT_CAUSED_BY_CALLER)
"          VALUE(ES_EXIT_CAUSED_BY_USER) TYPE
SLIS_EXIT_BY_USER
"          TABLES
"          T_OUTTAB
"          EXCEPTIONS
"          PROGRAM_ERROR
```

**Parámetros IMPORT:**

- I\_INTERFACE\_CHECK: se genera un log de salida de verificación de generación de datos.
- I\_CALLBACK\_PROGRAM: Nombre del programa que invoca a la función.
- I\_CALLBACK\_PF\_STATUS\_SET: setear la rutina EXIT del status del listado (barra de tareas).
- I\_CALLBACK\_USER\_COMMAND: Rutina para el tratamiento del EXIT.
- I\_STRUCTURE\_NAME: nombre de la estructura de la tabla interna de salida.
- IS\_LAYOUT: especificaciones para el layout del listado de salida.
- IT\_FIELDCAT: catálogo de campos y sus descripciones
- IT\_EXCLUDING: tabla para excluir comandos
- IT\_SPECIAL\_GROUPS: campos de agrupamiento.
- IT\_SORT: criterio de ordenamiento para el primer listado de salida desplegado.
- IT\_FILTER: filtros para el primer listado de salida.
- IS\_SEL\_HIDE :
- I\_DEFAULT: variante inicial.
- I\_SAVE: las variantes de visualización pueden ser guardadas por el usuario.
- IS\_VARIANT : información de las variantes
- IT\_EVENTS: tabla de eventos
- IT\_EVENT\_EXIT : tabla para el tratamiento del código de función EXIT

- IS\_PRINT: Imprimir el listado
- IS\_REPREP\_ID: claves de inicialización de interfaces
- I\_SCREEN\_START\_COLUMN: setea las coordenadas para un cuadro de diálogo
- I\_SCREEN\_START\_LINE: setea las coordenadas para un cuadro de diálogo
- I\_SCREEN\_END\_COLUMN: setea las coordenadas para un cuadro de diálogo
- I\_SCREEN\_END\_LINE: setea las coordenadas para un cuadro de diálogo

### Parámetros EXPORT

- E\_EXIT\_CAUSED\_BY\_CALLER: borra la lista  
CALLBACK\_USER\_COMMAND
- ES\_EXIT\_CAUSED\_BY\_USER: cómo el usuario abandona el listado
- T\_OUTTAB: tabla interna de salida con los datos del listado (obligatoria)

### Ejemplo ALV- Sin pantalla de Selección de datos, tipo Lista:

```
*&-----*
*& Report  Z_DEMO_ALV
*&
*&-----*
*&
*&
*&-----*

report z_demo_alv no standard page heading.
*
* Declaración de Datos
data: i_sflight type table of sflight.

*-----*
* Selección
select * from sflight into table i_sflight.

* Call ABAP List Viewer (ALV)
call function 'REUSE_ALV_LIST_DISPLAY'
  exporting
    i_structure_name = 'SFLIGHT'
  tables
    t_outtab         = i_sflight.
```

Al ejecutar el reporte se obtiene el siguiente resultado (F8). Ver que en la parte superior del listado se despliega la barra de tareas estándar de los listados de salida del tipo ALV:



## Diplomatura en Programación Abap - MÓDULO 12: ABAP LIST VIEWER (ALV)

Programa ejemplo ALV												
ID	No.	Date	Airfare	Curr.	Plane Type	Capacity	Occupied	Booking total	Capacity	Occupied	Capacity	Occupie
AA	17	18.05.2006	422,94	USD	747-400	385	370	190.949,24	31	31	21	19
AA	17	07.06.2006	422,94	USD	747-400	385	363	188.428,36	31	29	21	20
AA	17	05.07.2006	422,94	USD	747-400	385	374	194.480,65	31	29	21	21
AA	17	02.08.2006	422,94	USD	747-400	385	362	189.447,81	31	29	21	21
AA	17	30.08.2006	422,94	USD	747-400	385	364	188.711,91	31	29	21	20
AA	17	27.09.2006	422,94	USD	747-400	385	371	192.852,39	31	30	21	20
AA	17	25.10.2006	422,94	USD	747-400	385	373	195.326,65	31	31	21	21
AA	17	22.11.2006	422,94	USD	747-400	385	363	185.852,67	31	29	21	18
AA	17	20.12.2006	422,94	USD	747-400	385	371	193.897,04	31	31	21	21
AA	17	17.01.2007	422,94	USD	747-400	385	142	73.210,99	31	12	21	7
AA	17	14.02.2007	422,94	USD	747-400	385	5	1.924,39	31	0	21	0
AA	17	14.03.2007	422,94	USD	747-400	385	57	29.474,76	31	5	21	3
AA	17	11.04.2007	422,94	USD	747-400	385	34	16.799,19	31	3	21	1
AA	17	09.05.2007	422,94	USD	747-400	385	41	20.639,52	31	3	21	2
AA	17	06.06.2007	422,94	USD	747-400	385	0	0,00	31	0	21	0
AA	64	12.05.2006	422,94	USD	A310-300	280	271	135.505,97	22	22	10	10
AA	64	09.06.2006	422,94	USD	A310-300	280	268	131.382,28	22	20	10	10
AA	64	07.07.2006	422,94	USD	A310-300	280	271	134.359,77	22	22	10	10
AA	64	04.08.2006	422,94	USD	A310-300	280	265	133.014,74	22	22	10	10
AA	64	01.09.2006	422,94	USD	A310-300	280	271	133.945,25	22	22	10	9
AA	64	29.09.2006	422,94	USD	A310-300	280	271	133.484,25	22	22	10	9
AA	64	27.10.2006	422,94	USD	A310-300	280	271	134.042,59	22	21	10	10
AA	64	24.11.2006	422,94	USD	A310-300	280	264	129.999,27	22	21	10	9
AA	64	22.12.2006	422,94	USD	A310-300	280	271	133.572,94	22	21	10	9
AA	64	19.01.2007	422,94	USD	A310-300	280	156	77.351,61	22	12	10	6

### Ejemplo ALV- Sin pantalla de Selección de datos, tipo Grilla:

```

*&-----*
*& Report  Z_DEMO_ALV1
*&
*&-----*
*&
*&
*&-----*

REPORT  Z_DEMO_ALV1.
*
* Data to be displayed
DATA: I_SFLIGHT TYPE TABLE OF SFLIGHT.
*-----*

* Selection
  SELECT * FROM SFLIGHT INTO TABLE I_SFLIGHT.

* Call ABAP List Viewer (ALV)
  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      I_STRUCTURE_NAME  = 'SFLIGHT'
      TABLES
        T_OUTTAB        = I_SFLIGHT.

```

Al ejecutar el reporte se obtiene el siguiente resultado (F8).

## Diplomatura en Programación Abap - MÓDULO 12: ABAP LIST VIEWER (ALV)

Programa ejemplo ALV II

List Edit Goto Views Settings System Help

ID	No.	Flight Date	Airfare	Curr.	Plane Type	Capacity	Occupied	Booking total	Capacity	Occupied bus.	Capacity	Occupied
AA	17	10.05.2006	422,94	USD	747-400	385	370	190.949,24	31	31	21	19
AA	17	07.06.2006	422,94	USD	747-400	385	363	188.428,36	31	29	21	20
AA	17	05.07.2006	422,94	USD	747-400	385	374	194.480,65	31	29	21	21
AA	17	02.08.2006	422,94	USD	747-400	385	362	189.447,81	31	29	21	21
AA	17	30.08.2006	422,94	USD	747-400	385	364	188.711,91	31	29	21	20
AA	17	27.09.2006	422,94	USD	747-400	385	371	192.852,39	31	30	21	20
AA	17	25.10.2006	422,94	USD	747-400	385	373	195.326,65	31	31	21	21
AA	17	22.11.2006	422,94	USD	747-400	385	363	185.852,67	31	29	21	18
AA	17	20.12.2006	422,94	USD	747-400	385	371	193.897,04	31	31	21	21
AA	17	17.01.2007	422,94	USD	747-400	385	142	73.210,99	31	12	21	7
AA	17	14.02.2007	422,94	USD	747-400	385	5	1.924,39	31	0	21	0
AA	17	14.03.2007	422,94	USD	747-400	385	57	29.474,76	31	5	21	3
AA	17	11.04.2007	422,94	USD	747-400	385	34	16.799,19	31	3	21	1
AA	17	09.05.2007	422,94	USD	747-400	385	41	20.639,52	31	3	21	2
AA	17	06.06.2007	422,94	USD	747-400	385	0	0,00	31	0	21	0
AA	64	12.05.2006	422,94	USD	A310-300	280	271	135.505,97	22	22	10	10
AA	64	09.06.2006	422,94	USD	A310-300	280	268	131.382,28	22	20	10	10
AA	64	07.07.2006	422,94	USD	A310-300	280	271	134.359,77	22	22	10	10
AA	64	04.08.2006	422,94	USD	A310-300	280	265	133.014,74	22	22	10	10
AA	64	01.09.2006	422,94	USD	A310-300	280	271	133.945,25	22	22	10	9
AA	64	29.09.2006	422,94	USD	A310-300	280	271	133.484,25	22	22	10	9
AA	64	27.10.2006	422,94	USD	A310-300	280	271	134.042,59	22	21	10	10
AA	64	24.11.2006	422,94	USD	A310-300	280	264	129.999,27	22	21	10	9
AA	64	22.12.2006	422,94	USD	A310-300	280	271	133.572,94	22	21	10	9
AA	64	19.01.2007	422,94	USD	A310-300	280	156	77.351,61	22	12	10	6
AA	64	16.02.2007	422,94	USD	A310-300	280	6	2.326,18	22	0	10	0
AA	64	16.03.2007	422,94	USD	A310-300	280	96	47.843,10	22	7	10	4
AA	64	13.04.2007	422,94	USD	A310-300	280	32	14.879,03	22	3	10	0
AA	64	11.05.2007	422,94	USD	A310-300	280	44	22.669,61	22	4	10	2

### Ejemplo ALV- Sin pantalla de Selección de datos, tipo Grilla:

```
*&-----*
*& Report  Z_DEMO_ALV2
*&
*&-----*
*&
*&
*&-----*
```

REPORT Z\_DEMO\_ALV2.

TYPE-POOLS: slis.

```
DATA: BEGIN OF i_outtab OCCURS 0.
      INCLUDE STRUCTURE sflight.
DATA: w_chk TYPE c.
DATA: END OF i_outtab.
" For multiple selection
```

```
*      I_OUTTAB TYPE SFLIGHT OCCURS 0,
```

```
DATA: i_private TYPE slis_data_caller_exit,
      i_selfield TYPE slis_selfield,
      W_exit(1) TYPE c.
```

```

PARAMETERS: p_title TYPE sy-title.
*
START-OF-SELECTION.

    SELECT * FROM sflight INTO TABLE i_outtab.

    CALL FUNCTION 'REUSE_ALV_POPUP_TO_SELECT'
      EXPORTING
        i_title           = p_title
        i_selection       = 'X'
        i_zebra           = 'X'
        I_SCREEN_START_COLUMN = 0
        I_SCREEN_START_LINE  = 0
        I_SCREEN_END_COLUMN  = 0
        I_SCREEN_END_LINE    = 0
        i_checkbox_fieldname = 'W_CHK'
        I_LINEMARK_FIELDNAME  = 
        I_SCROLL_TO_SEL_LINE  = 'X'
        i_tabname           = 'I_OUTTAB'
        i_structure_name     = 'SFLIGHT'
        IT_FIELDCAT          = 
        IT_EXCLUDING         = 
        I_CALLBACK_PROGRAM   = 
        I_CALLBACK_USER_COMMAND = 
        IS_PRIVATE           = I_PRIVATE
      IMPORTING
        es_selfield        = i_selfield
        e_exit              = w_exit
      TABLES
        t_outtab           = i_outtab
      EXCEPTIONS
        program_error      = 1
        OTHERS              = 2.

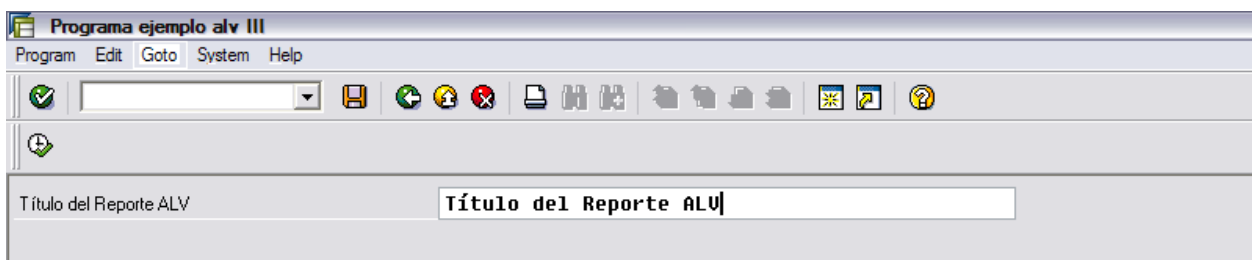
    IF sy-subrc <> 0.
*      MESSAGE i000(0k) WITH sy-subrc.
    ENDIF.

    *****the internal table is modified with a cross sign for marking the
    *****rows selected

    LOOP AT i_outtab WHERE w_chk = 'X'.
      WRITE: / i_outtab-carriid, i_outtab-price.
    ENDLOOP.

```

Al ejecutar el reporte se obtiene el siguiente resultado (F8).



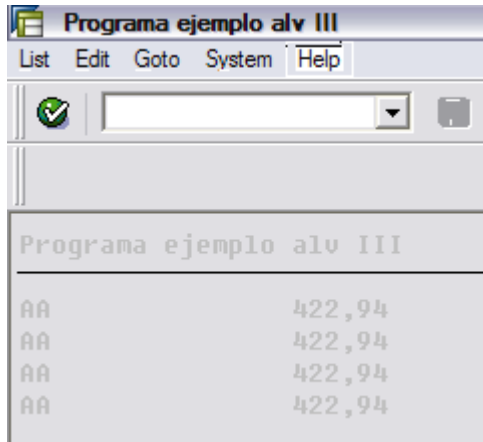


Si ejecutamos nuevamente, obtendremos el resultado del reporte en una ventana emergente (F8).

ID	No.	Date	Airfare	Curr.	Plane Type	Capacity	Occupied
AA	17	10.05.2006	422,94	USD	747-400	385	370
AA	17	07.06.2006	422,94	USD	747-400	385	363
AA	17	05.07.2006	422,94	USD	747-400	385	374
AA	17	02.08.2006	422,94	USD	747-400	385	362
AA	17	30.08.2006	422,94	USD	747-400	385	364
AA	17	27.09.2006	422,94	USD	747-400	385	371
AA	17	25.10.2006	422,94	USD	747-400	385	373
AA	17	22.11.2006	422,94	USD	747-400	385	363
AA	17	20.12.2006	422,94	USD	747-400	385	371
AA	17	17.01.2007	422,94	USD	747-400	385	142
AA	17	14.02.2007	422,94	USD	747-400	385	5
AA	17	14.03.2007	422,94	USD	747-400	385	57
AA	17	11.04.2007	422,94	USD	747-400	385	34
AA	17	09.05.2007	422,94	USD	747-400	385	41
AA	17	06.06.2007	422,94	USD	747-400	385	0
AA	64	12.05.2006	422,94	USD	A310-300	280	271
AA	64	09.06.2006	422,94	USD	A310-300	280	268
AA	64	07.07.2006	422,94	USD	A310-300	280	271
AA	64	04.08.2006	422,94	USD	A310-300	280	265
AA	64	01.09.2006	422,94	USD	A310-300	280	271
AA	64	29.09.2006	422,94	USD	A310-300	280	271

Podremos seleccionar los registros de los cuales deseamos ver un detalle, y activar el pushbutton :

ID	No.	Date	Airfare	Curr.	Plane Type	Capacity	Occupied
<input checked="" type="checkbox"/> AA	17	10.05.2006	422,94	USD	747-400	385	370
<input checked="" type="checkbox"/> AA	17	07.06.2006	422,94	USD	747-400	385	363
<input checked="" type="checkbox"/> AA	17	05.07.2006	422,94	USD	747-400	385	374
<input checked="" type="checkbox"/> AA	17	02.08.2006	422,94	USD	747-400	385	362
<input checked="" type="checkbox"/> AA	17	30.08.2006	422,94	USD	747-400	385	364



**Ejemplo ALV- Con pantalla de Selección de datos, tipo interactivo:**

```

*&-----*
*& Report  Z_DEMO_ALV3
*&
*&-----*
*&
*&-----*

REPORT Z_DEMO_ALV3 NO STANDARD PAGE HEADING.
* ALV related data declaration
TYPE-POOLS: slis.
* DB-Table
TABLES sflight.
* Includes
INCLUDE <icon>.
INCLUDE <symbol>.
*
CONSTANTS:
c_formname_top_of_page TYPE slis_formname VALUE 'TOP_OF_PAGE'.

DATA: i_fieldcat TYPE slis_t_fieldcat_alv,
      i_layout   TYPE slis_layout_alv,
      i_sp_group TYPE slis_t_sp_group_alv,
      i_events   TYPE slis_t_event,
      i_print    TYPE slis_print_alv,
      i_sort     TYPE slis_t_sortinfo_alv.

*internal table for data to be displayed
DATA: BEGIN OF i_sflight OCCURS 0.
      INCLUDE STRUCTURE sflight.
DATA: box,
      lights.
DATA: END OF i_sflight.
*
DATA: w_repid LIKE sy-repid.
DATA: i_list_top_of_page TYPE slis_t_listheader.
* Report Selections
SELECT-OPTIONS s_carriid FOR sflight-carriid.
SELECT-OPTIONS s_connid FOR sflight-connid.

```

```
SELECT-OPTIONS s_fldate FOR sflight-fldate.
*SELECTION-SCREEN SKIP 1.
* Parameters
PARAMETERS: p_maxrow TYPE i DEFAULT 30."to limit the selection
SELECTION-SCREEN SKIP 1.

* Variant for ALV display
SELECTION-SCREEN BEGIN OF BLOCK 0 WITH FRAME TITLE text-000.
PARAMETERS: p_varnt LIKE disvariant-variant.
SELECTION-SCREEN END OF BLOCK 0.
* Layout of the report display
SELECTION-SCREEN BEGIN OF BLOCK a WITH FRAME TITLE text-001.
PARAMETERS: p_zebra AS CHECKBOX DEFAULT ' ', "Striped pattern
             p_nocolh AS CHECKBOX DEFAULT ' ', "No column heading
             p_novlin AS CHECKBOX DEFAULT ' ', "No vertical lines
             p_colopt AS CHECKBOX DEFAULT ' ', "Optimizes col. wd
             p_keyhot AS CHECKBOX DEFAULT ' ', "Key fields hot
             p_noinpt AS CHECKBOX DEFAULT ' '. "No field for input
SELECTION-SCREEN END OF BLOCK a.

SELECTION-SCREEN BEGIN OF BLOCK b WITH FRAME TITLE text-002.
PARAMETERS: p_lights AS CHECKBOX DEFAULT 'X',
             p_lightc AS CHECKBOX DEFAULT 'X'.
SELECTION-SCREEN END OF BLOCK b.

SELECTION-SCREEN BEGIN OF BLOCK c WITH FRAME TITLE text-003.
PARAMETERS: p_totonl AS CHECKBOX DEFAULT ' ',
             p_totext(60),
             p_sttext(60).
SELECTION-SCREEN END OF BLOCK c.

SELECTION-SCREEN BEGIN OF BLOCK d WITH FRAME TITLE text-004.
PARAMETERS: p_chkbox AS CHECKBOX DEFAULT 'X',
             p_detpop AS CHECKBOX DEFAULT 'X',
             p_groups AS CHECKBOX DEFAULT ' '.
SELECTION-SCREEN END OF BLOCK d.

SELECTION-SCREEN BEGIN OF BLOCK e WITH FRAME TITLE text-005.
PARAMETERS: p_print AS CHECKBOX DEFAULT ' ',
             p_nosinf AS CHECKBOX DEFAULT ' ',
             p_nocove AS CHECKBOX DEFAULT ' ',
             p_nonewp AS CHECKBOX DEFAULT ' ',
             p_nolinf AS CHECKBOX DEFAULT ' ',
             p_reserv TYPE i.
SELECTION-SCREEN END OF BLOCK e.

DATA: w_boxnam TYPE slis_fieldname VALUE 'BOX',
      w_f2code LIKE sy-ucomm VALUE '&ETA',
      w_lignam TYPE slis_fieldname VALUE 'LIGHTS',
      w_save(1) TYPE c,
      w_default(1) TYPE c,
      w_exit(1) TYPE c,
      i_variant LIKE disvariant,
      i_variant1 LIKE disvariant.

*-----*
INITIALIZATION.
```

```
w_repid = sy-repid.

PERFORM fieldcat_init USING i_fieldcat.
PERFORM eventtab_build USING i_events.
PERFORM comment_build USING i_list_top_of_page.
PERFORM sp_group_build USING i_sp_group.
PERFORM t_sort_build USING i_sort.

* Set Options: save variant user specific or general
**** 'A' or 'U' are for user-specific variants list
**** 'X' or 'space' for general
w_save = 'A'.

PERFORM variant_init.
* Get default variant
i_variant1 = i_variant.

CALL FUNCTION 'REUSE_ALV_VARIANT_DEFAULT_GET'
  EXPORTING
    i_save      = w_save
  CHANGING
    cs_variant = i_variant1
  EXCEPTIONS
    not_found  = 2.

IF sy-subrc = 0.
  p_varnt = i_variant1-variant.
ENDIF.

* Process on value request (list of possible variants)
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_varnt.
  PERFORM f4_for_variant.

* PAI
AT SELECTION-SCREEN.
  PERFORM pai_of_selection_screen.

START-OF-SELECTION.

  PERFORM selection.

END-OF-SELECTION.

PERFORM layout_build USING i_layout. "wg. Parameters
PERFORM print_build USING i_print.  "wg. Parameters

CALL FUNCTION 'REUSE_ALV_FIELDCATALOG_MERGE'
  EXPORTING
    i_program_name      = w_repid
    i_internal_tabname  = 'I_SFLIGHT'
    i_structure_name    = 'SFLIGHT'
    i_client_never_display = 'X'
    i_inclname          = w_repid
  CHANGING
    ct_fieldcat         = i_fieldcat[]
  EXCEPTIONS
    inconsistent_interface = 1
```

```

        program_error          = 2
        OTHERS                  = 3.

IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE 'S' NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

* Call ABAP/4 List Viewer
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
*       I_INTERFACE_CHECK          = ' '
*       i_callback_program         = w_repid
*       I_CALLBACK_PF_STATUS_SET   = ' '
*       I_CALLBACK_USER_COMMAND    = ' '
*       I_CALLBACK_TOP_OF_PAGE     = ' '
*       I_CALLBACK_HTML_TOP_OF_PAGE = ' '
*       I_CALLBACK_HTML_END_OF_LIST = ' '
*       i_structure_name           = 'SFLIGHT'
*       i_background_id            = 'ALV_BACKGROUND'
*       I_GRID_TITLE               =
*       I_GRID_SETTINGS            =
*       is_layout                  = i_layout
*       it_fieldcat                = i_fieldcat[]
*       IT_EXCLUDING               =
*       it_special_groups          = i_sp_group[]
*       it_sort                    = i_sort[]
*       IT_FILTER                  =
*       IS_SEL_HIDE                =
*       I_DEFAULT                  = 'X'
*       i_save                     = w_save
*       is_variant                 = i_variant
*       it_events                  = i_events[]
*       IT_EVENT_EXIT              =
*       is_print                   = i_print
*       IS_REPREP_ID               =
*       I_SCREEN_START_COLUMN      = 0
*       I_SCREEN_START_LINE        = 0
*       I_SCREEN_END_COLUMN        = 0
*       I_SCREEN_END_LINE          = 0
*       IMPORTING
*       E_EXIT_CAUSED_BY_CALLER    =
*       ES_EXIT_CAUSED_BY_USER     =
    TABLES
*       t_outtab                   = i_sflight
    EXCEPTIONS
        program_error              = 1
        OTHERS                     = 2.

IF sy-subrc <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*       WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

* CALL FUNCTION 'REUSE_ALV_LIST_DISPLAY'
*       EXPORTING
*           i_callback_program      = w_repid
```

```

*          i_structure_name      = 'SFLIGHT'
*          is_layout             = i_layout
*          it_fieldcat           = i_fieldcat[]
**          IT_EXCLUDING         =
*          it_special_groups     = i_sp_group[]
*          it_sort               = i_sort[]
**          IT_FILTER            =
**          IS_SEL_HIDE          =
**          i_default            = W_DEFAULT
*          i_save                = w_save
*          is_variant            = i_variant
*          it_events             = i_events[]
**          IT_EVENT_EXIT       =
*          is_print              = i_print
**          I_SCREEN_START_COLUMN = 0
**          I_SCREEN_START_LINE  = 0
**          I_SCREEN_END_COLUMN  = 0
**          I_SCREEN_END_LINE    = 0
**          IMPORTING
**          E_EXIT_CAUSED_BY_CALLER =
*          TABLES
*          t_outtab              = i_sflight.

*-----*
*          FORM FIELDCAT_INIT                                         *
*-----*
*  -->  L_FIELDCAT                                                  *
*-----*
FORM fieldcat_init USING l_fieldcat TYPE slis_t_fieldcat_alv.

    DATA: ls_fieldcat TYPE slis_fieldcat_alv.
*
    CLEAR ls_fieldcat.
    ls_fieldcat-fieldname = 'SEATSOCC'.
*The field is not displayed in the initial output, can be interactively
* chosen for display
    ls_fieldcat-no_out    = 'X'.
*This field is assigned to a special group with tech. key 'A' and can be
*displayed using the special group buttons
    ls_fieldcat-sp_group  = 'A'.
*The field cannot be summed irrespective of its data type
    ls_fieldcat-no_sum    = 'X'.
    APPEND ls_fieldcat TO l_fieldcat.
*
    CLEAR ls_fieldcat.
    ls_fieldcat-fieldname = 'SEATSMAX'.
    ls_fieldcat-no_out    = 'X'.
    ls_fieldcat-sp_group  = 'A'.
    APPEND ls_fieldcat TO l_fieldcat.
*
    CLEAR ls_fieldcat.
    ls_fieldcat-fieldname = 'PRICE'.
    ls_fieldcat-no_out    = 'X'.
    ls_fieldcat-sp_group  = 'B'.
    APPEND ls_fieldcat TO l_fieldcat.
*
    CLEAR ls_fieldcat.

```

```
ls_fieldcat-fieldname    = 'CARRID'.
ls_fieldcat-outputlen    = 7.
APPEND ls_fieldcat TO l_fieldcat.

ENDFORM.

*-----*
*      FORM DATA_ADD                      *
*-----*
* --> L_SFLIGHT
*-----*
FORM data_add TABLES l_sflight STRUCTURE i_sflight.

  LOOP AT l_sflight.

    IF sy-tabix > 10.
      l_sflight-box    = 'X'.
      l_sflight-lights = '3'.
    ELSE.
      IF sy-tabix = 1.
        l_sflight-lights = '2'.
      ELSE.
        l_sflight-lights = '1'.
      ENDIF.
    ENDIF.

    MODIFY l_sflight.

  ENDLOOP.

ENDFORM.

*-----*
*      FORM EVENTTAB_BUILD                  *
*-----*
* -->  L_EVENTS
*-----*
FORM eventtab_build USING l_events TYPE slis_t_event.

  DATA: ls_event TYPE slis_alv_event.
*
  CALL FUNCTION 'REUSE_ALV_EVENTS_GET'
    EXPORTING
      i_list_type = 0
    IMPORTING
      et_events   = l_events.

  READ TABLE l_events WITH KEY name = slis_ev_top_of_page INTO ls_event.

  IF sy-subrc = 0.
    MOVE c_formname_top_of_page TO ls_event-form.
    APPEND ls_event TO l_events.
  ENDIF.

ENDFORM.

*-----*
```

```
*          FORM COMMENT_BUILD                                     *
*-----*
* -->  L_TOP_OF_PAGE                                           *
*-----*
FORM comment_build USING l_top_of_page TYPE slis_t_listheader.

    DATA: ls_line TYPE slis_listheader.

***Header
    CLEAR ls_line.
    ls_line-typ = 'H'.
* LS_LINE-KEY: not used for this type
    ls_line-info = 'Heading list'.
    APPEND ls_line TO l_top_of_page.

***Selection
    CLEAR ls_line.
    ls_line-typ = 'S'.
    ls_line-key = 'Key 1'.
    ls_line-info = 'Information'.
    APPEND ls_line TO l_top_of_page.
    ls_line-key = 'Key 2'.
    APPEND ls_line TO l_top_of_page.

***Action
    CLEAR ls_line.
    ls_line-typ = 'A'.
* LS_LINE-KEY: not used for this type
    ls_line-info = 'Status list'.
    APPEND ls_line TO l_top_of_page.

ENDFORM.

*-----*
*          FORM LAYOUT_BUILD                                     *
*-----*
* <->  LS_LAYOUT                                           *
*-----*
FORM layout_build USING ls_layout TYPE slis_layout_alv.

    ls_layout-f2code          = w_f2code.
    ls_layout-zebra           = p_zebra.
    ls_layout-colwidth_optimize = p_colopt.
    IF p_chkbox = 'X'.
*Fieldname for check box on the report output
        ls_layout-box_fieldname = w_boxnam.
    ELSE.
        ls_layout-box_fieldname = space.
    ENDIF.
    ls_layout-no_input        = p_noinput.
    ls_layout-no_vline        = p_novlin.
    ls_layout-no_colhead      = p_nocolh.
    IF p_lights = 'X' OR p_lightc = 'X'.
*Fieldname for lights on the report output
        ls_layout-lights_fieldname = w_lignam.
    ELSE.
        CLEAR ls_layout-lights_fieldname.
    ENDIF.
```



```

ENDIF.

ls_layout-lights_condense    = p_lightc.
ls_layout-totals_text        = p_totext.
ls_layout-subtotals_text      = p_sttext.
ls_layout-totals_only        = p_totonl.
ls_layout-key_hotspot        = p_keyhot.
ls_layout-detail_popup       = p_detpop.
ls_layout-group_change_edit  = p_groups.
* E05_LS_LAYOUT-GROUP_BUTTONS = P_GROUPB.
* ls_layout-group_buttons    = 'X'.

ENDFORM.

*-----*
*      FORM SP_GROUP_BUILD                                     *
*-----*
* -->  L_SP_GROUP                                           *
*-----*
FORM sp_group_build USING l_sp_group TYPE slis_t_sp_group_alv.

  DATA: ls_sp_group TYPE slis_sp_group_alv.
  *Fields are assigned to the special group
  CLEAR ls_sp_group.
  ls_sp_group-sp_group = 'A'.
  ls_sp_group-text     = 'Reservation status'.
  APPEND ls_sp_group TO l_sp_group.

  CLEAR ls_sp_group.
  ls_sp_group-sp_group = 'B'.
  ls_sp_group-text     = 'Flight charges'.
  APPEND ls_sp_group TO l_sp_group.

ENDFORM.

*-----*
*      FORM SELECTION                                       *
*-----*
FORM selection.

  SELECT * FROM sflight INTO CORRESPONDING FIELDS OF TABLE i_sflight
    UP TO p_maxrow ROWS WHERE carrid IN s_carrid
    AND connid IN s_connid AND fldate IN s_fldate.

  PERFORM data_add TABLES i_sflight.
ENDFORM.

*-----*
*      FORM TOP_OF_PAGE                                     *
*-----*
*      .....
*-----*
FORM top_of_page.
*
  CALL FUNCTION 'REUSE_ALV_COMMENTARY_WRITE'
    EXPORTING
      i_logo          = 'ENJOYSAP_LOGO'

```

```
        it_list_commentary = i_list_top_of_page.

ENDFORM.

*-----*
*      FORM F4_FOR_VARIANT                                *
*-----*
FORM f4_for_variant.
*
    CALL FUNCTION 'REUSE_ALV_VARIANT_F4'
      EXPORTING
        is_variant      = i_variant
        i_save           = w_save
*      it_default_fieldcat =
      IMPORTING
        e_exit           = w_exit
        es_variant       = i_variant1
      EXCEPTIONS
        not_found = 2.
    IF sy-subrc = 2.
      MESSAGE ID sy-msgid TYPE 'S' NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ELSE.
      IF w_exit = space.
        p_varnt = i_variant1-variant.
      ENDIF.
    ENDIF.
ENDFORM.

*&-----*
*&      Form  PAI_OF_SELECTION_SCREEN                    *
*&-----*
*      to check whether right variant is entered on the selection scr
*-----*
FORM pai_of_selection_screen.
*
    IF NOT p_varnt IS INITIAL.
      MOVE i_variant TO i_variant1.
      MOVE p_varnt TO i_variant1-variant.

      CALL FUNCTION 'REUSE_ALV_VARIANT_EXISTENCE'
        EXPORTING
          i_save      = w_save
        CHANGING
          cs_variant = i_variant1.

      i_variant = i_variant1.
    ELSE.
      PERFORM variant_init.
    ENDIF.

ENDFORM.                                " PAI_OF_SELECTION_SCREEN

*&-----*
*&      Form  VARIANT_INIT                                *
*&-----*
FORM variant_init.
```

```
*
  CLEAR i_variant.
  i_variant-report = w_repid.

ENDFORM.                                " VARIANT_INIT

*-----*
*      FORM PRINT_BUILD                      *
*-----*
*      .....                               *
*-----*
FORM print_build USING l_print TYPE slis_print_alv.
*
  l_print-print          = p_print.
  l_print-no_print_selinfos = p_nosinf.
  l_print-no_coverpage    = p_nocove.
  l_print-no_new_page     = p_nonewp.
  l_print-no_print_listinfos = p_nolinf.
  l_print-reserve_lines   = p_reserv.
  l_print-print          = p_print.

ENDFORM.

*-----*
*      FORM T_SORT_BUILD                      *
*-----*
FORM t_sort_build USING l_sort TYPE slis_t_sortinfo_alv.

  DATA: ls_sort TYPE slis_sortinfo_alv.

  ls_sort-fieldname = 'CARRID'.
  ls_sort-spos      = 1.
  ls_sort-up        = 'X'.
  ls_sort-subtot    = 'X'.
  APPEND ls_sort TO l_sort.

ENDFORM.
```

Al ejecutar el reporte se obtiene la siguiente pantalla de Selección (F8):

Con los siguientes parámetros de Selección:

P_CHKBOX	Checkbox
P_COLOPT	Colores Resaltados
P_DETPOP	Detalle emergente
P_GROUPS	Nuevos Grupos de Usuario
P_KEYHOT	Claves como hotspot
P_LIGHTC	Semáforo
P_LIGHTS	Subtotales con Semáforo
P_MAXROW	Cantidad de Registros Máx.
P_NOCOLH	Sin encabezados en Columnas
P_NOCOVE	Sin Cobertura
P_NOINPT	Campos sólo de Visualización
P_NOLINF	Sin detalle de Líneas

P_NONEWP	Sin nueva página x cambio
P_NOSINF	Sin información de campos
P_NOVLIN	Sin Líneas Verticales
P_PRINT	Imprimir
P_RESERV	Líneas p. END of Page
P_STTEXT	Texto de Subtotal
P_TOTEXT	Texto de Total
P_TOTONL	Sólo Totales
P_VARNT	Variante
P_ZEBRA	Zebra
S_CARRID	Aerolínea
S_CONNID	Vuelo
S_FLDATE	Fecha del vuelo

De acuerdo a los criterios de selección obtendremos diferentes resultados de disposición y agrupamiento del ALV:

Programa Ejemplo ALV IV

List Edit Goto Views Settings System Help

Heading list

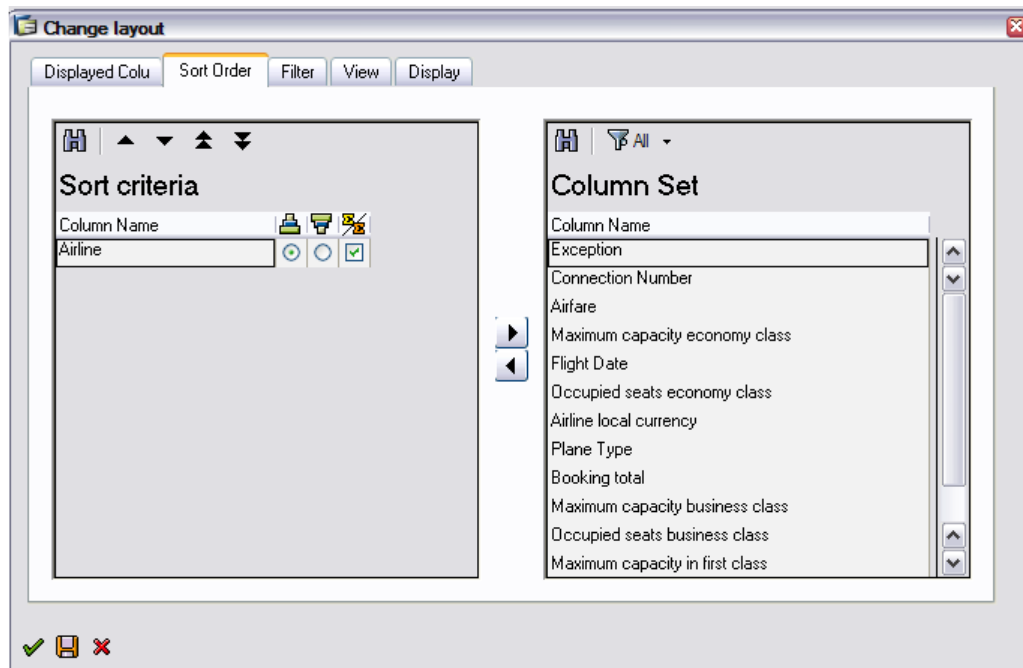
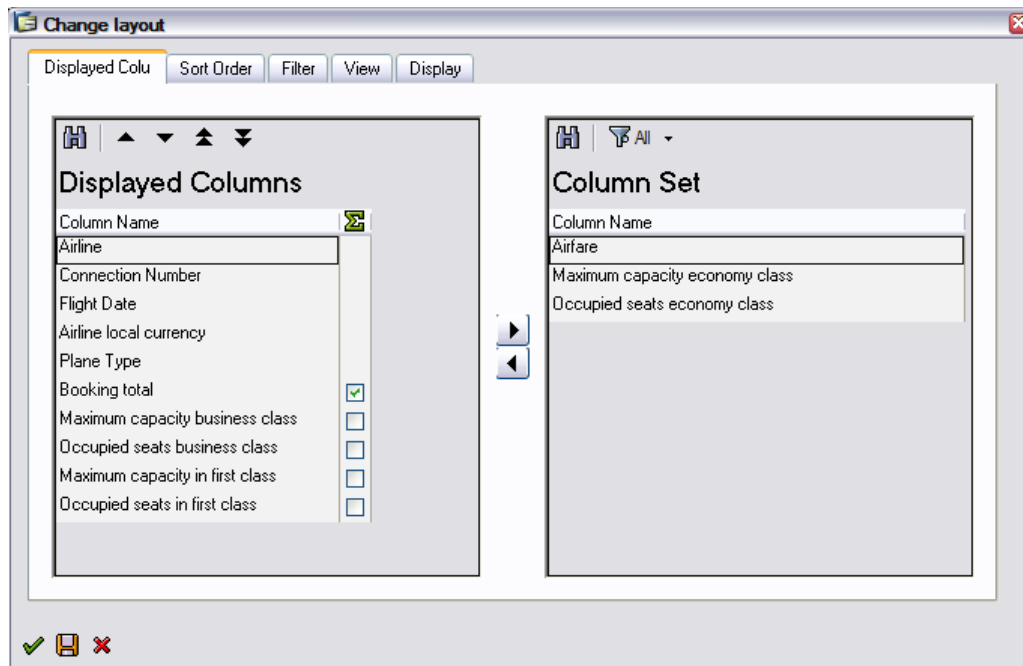
**Key 1** Information  
**Key 2** Information  
*Status list*

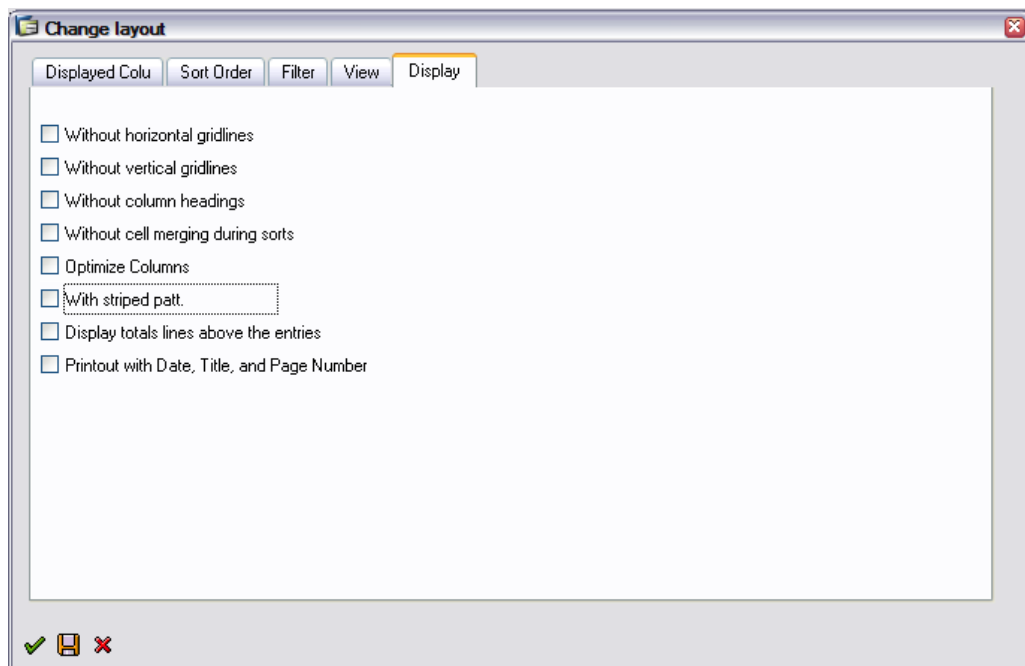
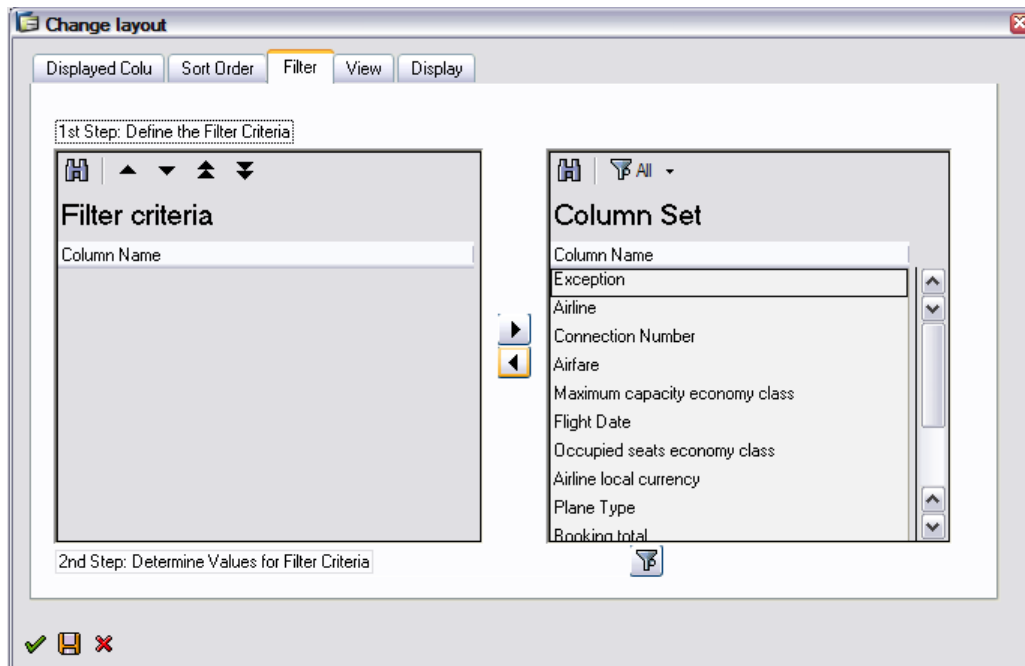
Exce...	ID /	No.	Flight Date	Curr.	Plane Type	Total	Capacity	Occupied	Capacity	Occupied
AA	17	17.01.2007	USD	747-400	73.210,99	31	12	21	7	
	17	14.02.2007	USD	747-400	1.924,39	31	0	21	0	
	17	14.03.2007	USD	747-400	29.474,76	31	5	21	3	
	17	11.04.2007	USD	747-400	16.799,19	31	3	21	1	
	17	09.05.2007	USD	747-400	20.639,52	31	3	21	2	
	17	06.06.2007	USD	747-400	0,00	31	0	21	0	
	64	12.05.2006	USD	A310-300	135.505,97	22	22	10	10	
	64	09.06.2006	USD	A310-300	131.382,28	22	20	10	10	
	64	07.07.2006	USD	A310-300	134.359,77	22	22	10	10	
	64	04.08.2006	USD	A310-300	133.014,74	22	22	10	10	
	64	01.09.2006	USD	A310-300	133.945,25	22	22	10	9	
	64	29.09.2006	USD	A310-300	133.484,25	22	22	10	9	
	64	27.10.2006	USD	A310-300	134.042,59	22	21	10	10	
	64	24.11.2006	USD	A310-300	129.999,27	22	21	10	9	
	64	22.12.2006	USD	A310-300	133.572,94	22	21	10	9	
	64	19.01.2007	USD	A310-300	77.351,61	22	12	10	6	
	64	16.02.2007	USD	A310-300	2.326,18	22	0	10	0	
	64	16.03.2007	USD	A310-300	47.843,10	22	7	10	4	
	64	13.04.2007	USD	A310-300	14.879,03	22	3	10	0	
	64	11.05.2007	USD	A310-300	22.669,61	22	4	10	2	
	64	08.06.2007	USD	A310-300	0,00	22	0	10	0	
AA			USD		3.226...					
			USD		3.22...					

A través del Icono de disposición:



podremos adicionar opciones de visualización:





A continuación, en la próxima Unidad, veremos todos los detalles y funcionalidades del BI (Batch Input) herramienta muy útil para la migración de datos desde un sistema legacy, por ejemplo. En tanto, los invitamos a realizar las actividades de la Unidad actual.