\*&---------------------------------------------------------------------\*

\*& Report ZEWM\_COMPARACION\_UBICACIONES

\*&

\*&---------------------------------------------------------------------\*

\*&

\*&

\*&---------------------------------------------------------------------\*

REPORT ZEWM\_COMPARACION\_UBICACIONES.

\*----------------------------------------------------------------------\*

\* CLASS main DEFINITION

\*----------------------------------------------------------------------\*

\*

\*----------------------------------------------------------------------\*

CLASS main DEFINITION.

PUBLIC SECTION.

METHODS: run\_ubi\_report,

run\_hu\_report,

run\_palet\_report,

run\_depot\_report,

run\_diff\_report.

PRIVATE SECTION.

TYPES: BEGIN OF palet\_type,

material TYPE zsaldos\_inv-material, "Material

matkx TYPE maktx, "Nombre del material

cantidad TYPE zsaldos\_inv-cantidad, "Cantidad

palet\_qty TYPE marm-umrez, "Cantidad paletada

ubicacion TYPE zsaldos\_inv-ubicacion,"Ubicacion

hu TYPE zsaldos\_inv-hu, "UMp

unidadm TYPE string, "Unidad de Medida

light TYPE icon-id. "Semaforo

TYPES: END OF palet\_type.

TYPES: BEGIN OF depot\_type,

material TYPE zsaldos\_inv-material, "Material

matkx TYPE maktx, "Nombre del material

ubicacion TYPE zsaldos\_inv-ubicacion,"Ubicacion

tipo\_almacen TYPE zewm\_ubications-tipo\_almacen, "Almacen

hu TYPE zsaldos\_inv-hu, "UMp

lote TYPE zsaldos\_inv-lote, "Lote

ligth1 TYPE icon-id,

ligth2 TYPE icon-id.

TYPES: END OF depot\_type.

TYPES: BEGIN OF saldos\_dif,

material TYPE matnr, "Codigo material de la tabla ZSaldos\_inv

ubicacion TYPE zsaldos\_inv-ubicacion,

lote TYPE zsaldos\_inv-lote,

\* z\_maktx TYPE maktx, "Texto de material

cantidad TYPE zsaldos\_inv-cantidad, "Cantidad en la tabla zsaldos

unidadm TYPE string, "Unidad de Medida en la tabla de zsaldos

hu TYPE zsaldos\_inv-hu,

ernam TYPE ernam,

erdat TYPE erdat,

END OF saldos\_dif.

TYPES: BEGIN OF batch\_diff,

material TYPE matnr, "Codigo material de la tabla ZSaldos\_inv

maktx TYPE maktx,

cantidad TYPE zsaldos\_inv-cantidad, "Cantidad en la tabla zsaldos

qty\_sap TYPE mchb-clabs.

TYPES: END OF batch\_diff.

TYPES: BEGIN OF t\_material,

material TYPE zsaldos\_inv-material.

TYPES: End of t\_material.

Data: gt\_ubicaciones TYPE TABLE OF zewm\_ubications,

gt\_diferencias TYPE TABLE OF zewm\_ubications,

lv\_ubicaciones LIKE LINE OF gt\_ubicaciones,

gt\_saldos TYPE TABLE OF zsaldos\_inv,

gt\_saldos\_d TYPE TABLE OF saldos\_dif,

lv\_saldos LIKE LINE OF gt\_saldos,

lv\_saldos\_d LIKE LINE OF gt\_saldos\_d,

it\_fieldcat TYPE slis\_t\_fieldcat\_alv,

wa\_fieldcat TYPE slis\_fieldcat\_alv,

gt\_palets TYPE TABLE OF palet\_type,

lv\_palets LIKE LINE OF gt\_palets,

gt\_almacen TYPE TABLE OF depot\_type,

lv\_almacen LIKE LINE OF gt\_almacen,

lv\_almacen2 LIKE LINE OF gt\_almacen,

gt\_diffalmacen TYPE TABLE OF batch\_diff,

lv\_diffalmacen LIKE LINE OF gt\_diffalmacen,

gt\_material TYPE TABLE OF t\_material.

DATA: lv\_conteo TYPE i,

lv\_conteo\_sap TYPE i.

CONSTANTS: c\_green TYPE icon-id VALUE '@08@', "For green light

c\_yellow TYPE icon-id VALUE '@09@', "For Yellow light

c\_red TYPE icon-id VALUE '@0A@', "For red light

c\_lgort TYPE mchb-lgort VALUE '3000', "Centro

c\_werks TYPE mchb-werks VALUE '3000', "Almacen

c\_langu TYPE c VALUE 'S', "Idioma Español

c\_charg\_unico TYPE mchb-charg VALUE 'UNICO'. "Lote unico.

METHODS: get\_ubi\_data,

get\_hu\_data,

get\_palet\_data,

get\_depot\_data,

get\_diff\_data,

call\_diff\_alv IMPORTING p\_cantidad TYPE i p\_cantidad\_sap TYPE i

CHANGING p\_table TYPE STANDARD TABLE,

call\_depot\_alv CHANGING p\_table TYPE STANDARD TABLE,

call\_palet\_alv CHANGING p\_table TYPE STANDARD TABLE,

call\_hu\_alv CHANGING p\_table TYPE STANDARD TABLE,

call\_ubi\_alv IMPORTING p\_fieldcat TYPE slis\_t\_fieldcat\_alv

p\_table TYPE STANDARD TABLE.

ENDCLASS. "main DEFINITION

\*----------------------------------------------------------------------\*

\* CLASS main IMPLEMENTATION

\*----------------------------------------------------------------------\*

\*

\*----------------------------------------------------------------------\*

CLASS main IMPLEMENTATION.

METHOD get\_ubi\_data.

data: range TYPE RANGE OF zewm\_ubications-ubicacion,

lv\_range LIKE LINE OF range.

SELECT \* INTO CORRESPONDING FIELDS OF TABLE gt\_ubicaciones

FROM zewm\_ubications.

LOOP AT gt\_ubicaciones INTO lv\_ubicaciones.

SELECT COUNT(\*) FROM zsaldos\_inv

WHERE ubicacion = lv\_ubicaciones-ubicacion.

CHECK sy-subrc ne 0.

append lv\_ubicaciones to gt\_diferencias.

ENDLOOP.

EndMethod. "get\_data

METHOD get\_hu\_data.

SELECT hu INTO CORRESPONDING FIELDS OF TABLE gt\_saldos

FROM zsaldos\_inv

GROUP BY hu

HAVING COUNT( \* ) > 1.

DELETE gt\_saldos WHERE hu IS INITIAL.

IF gt\_saldos[] IS NOT INITIAL.

SELECT \* INTO CORRESPONDING FIELDS OF TABLE gt\_saldos\_d

FROM zsaldos\_inv

FOR ALL ENTRIES IN gt\_saldos

WHERE hu = gt\_saldos-hu.

ENDIF.

ENDMETHOD. "get\_hu\_data

METHOD get\_palet\_data.

DATA: meinh TYPE marm-meinh VALUE 'PAL'. "Tipo de unidad de medida para paletas

SELECT \* INTO CORRESPONDING FIELDS OF TABLE gt\_saldos

FROM zsaldos\_inv.

LOOP AT gt\_saldos INTO lv\_saldos.

SELECT SINGLE umrez INTO lv\_palets-palet\_qty

FROM marm

WHERE matnr = lv\_saldos-material

AND meinh = meinh.

CHECK sy-subrc eq 0.

MOVE-CORRESPONDING lv\_saldos to lv\_palets.

\* lv\_palets-material = lv\_saldos-material.

\* lv\_palets-cantidad = lv\_saldos-cantidad.

SELECT SINGLE maktx INTO lv\_palets-matkx

FROM makt

WHERE matnr = lv\_palets-material

AND spras = 'S'.

IF lv\_palets-cantidad = lv\_palets-palet\_qty.

lv\_palets-light = c\_green.

Else.

lv\_palets-light = c\_red.

ENDIF.

Append lv\_palets to gt\_palets.

ENDLOOP.

SORT gt\_palets by material ASCENDING.

EndMETHOD. "get\_palet\_data

METHOD get\_depot\_data.

\* data: almacenes\_dobles TYPE string VALUE ''.

SELECT a~material a~hu a~lote a~ubicacion b~tipo\_almacen

INTO CORRESPONDING FIELDS OF TABLE gt\_almacen

FROM zsaldos\_inv as a

INNER JOIN zewm\_ubications as b on a~ubicacion = b~ubicacion

\* LEFT JOIN makt as c ON c~matnr = a~material AND spras = 'S'

WHERE b~tipo\_almacen in ('2001', '2003', '2005' )

ORDER BY a~ubicacion ASCENDING.

LOOP AT gt\_almacen INTO lv\_almacen.

SELECT SINGLE maktx INTO lv\_almacen-matkx

FROM makt

WHERE matnr = lv\_almacen-material

AND Spras = c\_langu.

IF lv\_almacen2 IS INITIAL.

lv\_almacen2 = lv\_almacen.

Modify gt\_almacen FROM lv\_almacen INDEX sy-tabix.

CONTINUE.

ENDIF.

IF lv\_almacen-ubicacion = lv\_almacen2-ubicacion.

IF lv\_almacen-material = lv\_almacen2-material.

lv\_almacen-ligth1 = c\_green.

ELSE.

lv\_almacen-ligth1 = c\_red.

ENDIF.

IF lv\_almacen-lote = lv\_almacen2-lote.

lv\_almacen-ligth2 = c\_green.

ELSE.

lv\_almacen-ligth2 = c\_red.

ENDIF.

ENDIF.

Modify gt\_almacen FROM lv\_almacen INDEX sy-tabix.

lv\_almacen2 = lv\_almacen.

ENDLOOP.

ENDMETHOD. "get\_depot\_data

METHOD get\_diff\_data.

DATA clabs TYPE mchb-clabs.

DATA: r\_material TYPE RANGE OF zsaldos\_inv-material,

lv\_material TYPE zsaldos\_inv-material.

SELECT \* INTO CORRESPONDING FIELDS OF TABLE gt\_saldos

FROM zsaldos\_inv.

SELECT COUNT( DISTINCT material ) INTO lv\_conteo

FROM zsaldos\_inv.

SELECT COUNT( DISTINCT matnr ) INTO lv\_conteo\_sap

FROM mchb

WHERE lgort = c\_lgort

AND werks = c\_werks

AND charg = c\_charg\_unico.

LOOP AT gt\_saldos INTO lv\_saldos.

MOVE-CORRESPONDING lv\_saldos to lv\_diffalmacen.

SELECT SINGLE clabs INTO (clabs)

FROM mchb

WHERE matnr = lv\_saldos-material

AND werks = c\_lgort

AND lgort = c\_werks

AND charg = lv\_saldos-lote.

CHECK clabs = 0.

SELECT SINGLE maktx INTO lv\_diffalmacen-maktx

FROM makt

WHERE matnr = lv\_diffalmacen-material

AND spras = c\_langu.

lv\_diffalmacen-qty\_sap = clabs.

APPEND lv\_diffalmacen to gt\_diffalmacen.

ENDLOOP.

SORT gt\_diffalmacen by material ASCENDING.

DELETE ADJACENT DUPLICATES FROM gt\_diffalmacen COMPARING material.

ENDMETHOD. "get\_diff\_data

METHOD call\_depot\_alv.

DATA: I\_START\_COLUMN TYPE i VALUE 15,

I\_START\_LINE TYPE i VALUE 2,

I\_END\_COLUMN TYPE i VALUE 60,

I\_END\_LINE TYPE i VALUE 30,

i\_title TYPE string VALUE 'Revision ubicacion dobles',

i\_popup TYPE flag VALUE '',

it\_alv TYPE TABLE OF zewm\_ubications.

data: lr\_grid type ref to cl\_salv\_form\_layout\_grid,

lr\_grid\_1 TYPE REF TO cl\_salv\_form\_layout\_grid,

lr\_label TYPE REF TO cl\_salv\_form\_label,

lr\_text TYPE REF TO cl\_salv\_form\_text,

message TYPE string,

lr\_header type ref to cl\_salv\_form\_header\_info,

lr\_content type ref to cl\_salv\_form\_element.

\* it\_alv = p\_table.

DATA go\_alv TYPE REF TO cl\_salv\_table.

TRY.

cl\_salv\_table=>factory(

IMPORTING

r\_salv\_table = go\_alv

CHANGING

t\_table = p\_table ).

CATCH cx\_salv\_msg.

ENDTRY.

message = 'Reporte de Ubicaciones Dobles '.

CREATE OBJECT lr\_grid.

lr\_grid->create\_header\_information(

row = 1

column = 1

text = message

tooltip = message ).

lr\_content = lr\_grid.

go\_alv->set\_top\_of\_list( lr\_content ).

DATA: lr\_functions TYPE REF TO cl\_salv\_functions\_list,

lr\_columns TYPE REF to CL\_SALV\_COLUMNS\_TABLE,

lr\_column TYPE REF TO cl\_salv\_column,

lr\_layout TYPE REF TO CL\_SALV\_LAYOUT,

lr\_layoutkey TYPE salv\_s\_layout\_key.

lr\_layout = go\_alv->get\_layout( ).

lr\_layoutkey-REPORT = sy-repid.

lr\_layout->set\_key( lr\_layoutkey ).

lr\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).

lr\_functions = go\_alv->get\_functions( ).

lr\_functions->set\_all( 'X' ).

lr\_columns = go\_alv->get\_columns( ).

lr\_columns->set\_optimize( 'X' ).

lr\_column = lr\_columns->get\_column( 'LIGTH1' ).

lr\_column->set\_short\_text( 'Material' ).

lr\_column = lr\_columns->get\_column( 'LIGTH2' ).

lr\_column->set\_long\_text( 'Lote' ).

lr\_column = lr\_columns->get\_column( 'TIPO\_ALMACEN' ).

lr\_column->set\_long\_text( 'Almacen' ).

lr\_column->set\_medium\_text( 'Almacen' ).

lr\_column->set\_short\_text( 'Almacen' ).

IF go\_alv IS BOUND.

IF i\_popup = 'X'.

go\_alv->set\_screen\_popup(

start\_column = i\_start\_column

end\_column = i\_end\_column

start\_line = i\_start\_line

end\_line = i\_end\_line ).

ENDIF.

go\_alv->display( ).

ENDIF.

EndMETHOD. "call\_depot\_alv

METHOD call\_palet\_alv.

DATA: I\_START\_COLUMN TYPE i VALUE 15,

I\_START\_LINE TYPE i VALUE 2,

I\_END\_COLUMN TYPE i VALUE 60,

I\_END\_LINE TYPE i VALUE 30,

i\_title TYPE string VALUE 'UMp duplicadas',

i\_popup TYPE flag VALUE '',

it\_alv TYPE TABLE OF zewm\_ubications.

data: lr\_grid type ref to cl\_salv\_form\_layout\_grid,

lr\_grid\_1 TYPE REF TO cl\_salv\_form\_layout\_grid,

lr\_label TYPE REF TO cl\_salv\_form\_label,

lr\_text TYPE REF TO cl\_salv\_form\_text,

message TYPE string,

lr\_header type ref to cl\_salv\_form\_header\_info,

lr\_content type ref to cl\_salv\_form\_element.

\* it\_alv = p\_table.

DATA go\_alv TYPE REF TO cl\_salv\_table.

TRY.

cl\_salv\_table=>factory(

IMPORTING

r\_salv\_table = go\_alv

CHANGING

t\_table = p\_table ).

CATCH cx\_salv\_msg.

ENDTRY.

message = 'Reporte de Comparación Palets de Materiales'.

CREATE OBJECT lr\_grid.

lr\_grid->create\_header\_information(

row = 1

column = 1

text = message

tooltip = message ).

lr\_content = lr\_grid.

go\_alv->set\_top\_of\_list( lr\_content ).

DATA: lr\_functions TYPE REF TO cl\_salv\_functions\_list,

lr\_columns TYPE REF to CL\_SALV\_COLUMNS\_TABLE,

lr\_column TYPE REF TO cl\_salv\_column,

lr\_layout TYPE REF TO CL\_SALV\_LAYOUT,

lr\_layoutkey TYPE salv\_s\_layout\_key.

lr\_layout = go\_alv->get\_layout( ).

lr\_layoutkey-REPORT = sy-repid.

lr\_layout->set\_key( lr\_layoutkey ).

lr\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).

lr\_functions = go\_alv->get\_functions( ).

lr\_functions->set\_all( 'X' ).

lr\_columns = go\_alv->get\_columns( ).

lr\_columns->set\_optimize( 'X' ).

lr\_column = lr\_columns->get\_column( 'PALET\_QTY' ).

lr\_column->set\_optimized( ).

lr\_column->set\_long\_text( 'Cantidad por Palets' ).

lr\_column = lr\_columns->get\_column( 'CANTIDAD' ).

lr\_column->set\_long\_text( 'Cantidad contada' ).

lr\_column = lr\_columns->get\_column( 'UNIDADM').

lr\_column->set\_long\_text( 'Unidad de Medida').

IF go\_alv IS BOUND.

IF i\_popup = 'X'.

go\_alv->set\_screen\_popup(

start\_column = i\_start\_column

end\_column = i\_end\_column

start\_line = i\_start\_line

end\_line = i\_end\_line ).

ENDIF.

go\_alv->display( ).

ENDIF.

ENDMETHOD. "call\_palet\_alv

METHOD call\_diff\_alv.

data: lr\_grid type ref to cl\_salv\_form\_layout\_grid,

lr\_grid\_1 TYPE REF TO cl\_salv\_form\_layout\_grid,

lr\_label TYPE REF TO cl\_salv\_form\_label,

lr\_text TYPE REF TO cl\_salv\_form\_text,

message TYPE string,

lr\_header type ref to cl\_salv\_form\_header\_info,

lr\_content type ref to cl\_salv\_form\_element.

DATA: I\_START\_COLUMN TYPE i VALUE 15,

I\_START\_LINE TYPE i VALUE 2,

I\_END\_COLUMN TYPE i VALUE 60,

I\_END\_LINE TYPE i VALUE 30,

i\_title TYPE string VALUE 'UMp duplicadas',

i\_popup TYPE flag VALUE ''.

DATA go\_alv TYPE REF TO cl\_salv\_table.

TRY.

cl\_salv\_table=>factory(

IMPORTING

r\_salv\_table = go\_alv

CHANGING

t\_table = p\_table ).

CATCH cx\_salv\_msg.

ENDTRY.

message = 'Reporte de Inventariado'.

CREATE OBJECT lr\_grid.

lr\_grid->create\_header\_information(

row = 1

column = 1

text = message

tooltip = message ).

lr\_grid->add\_row( ).

lr\_grid\_1 = lr\_grid->Create\_grid(

row = 3

column = 1 ).

lr\_label = lr\_grid\_1->create\_label(

row = 1

column = 1

text = 'Cantidad contada'

tooltip = 'Cantidad contada' ).

lr\_text = lr\_grid\_1->create\_text(

row = 1

column = 2

text = p\_cantidad

tooltip = p\_cantidad ).

lr\_label = lr\_grid\_1->create\_label(

row = 2

column = 1

text = 'Cantidad en SAP'

tooltip = 'Cantidad en SAP' ).

lr\_text = lr\_grid\_1->create\_text(

row = 2

column = 2

text = p\_cantidad\_sap

tooltip = p\_cantidad\_sap ).

lr\_label->set\_label\_for( lr\_text ).

lr\_content = lr\_grid.

go\_alv->set\_top\_of\_list( lr\_content ).

DATA: lr\_functions TYPE REF TO cl\_salv\_functions\_list,

lr\_columns TYPE REF to CL\_SALV\_COLUMNS\_TABLE,

lr\_column TYPE REF TO cl\_salv\_column,

lr\_layout TYPE REF TO CL\_SALV\_LAYOUT,

lr\_layoutkey TYPE salv\_s\_layout\_key.

lr\_layout = go\_alv->get\_layout( ).

lr\_layoutkey-REPORT = sy-repid.

lr\_layout->set\_key( lr\_layoutkey ).

lr\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).

lr\_functions = go\_alv->get\_functions( ).

lr\_functions->set\_all( 'X' ).

lr\_columns = go\_alv->get\_columns( ).

lr\_columns->set\_optimize( 'X' ).

IF go\_alv IS BOUND.

IF i\_popup = 'X'.

go\_alv->set\_screen\_popup(

start\_column = i\_start\_column

end\_column = i\_end\_column

start\_line = i\_start\_line

end\_line = i\_end\_line ).

ENDIF.

go\_alv->display( ).

ENDIF.

ENDMETHOD. "call\_diff\_alv

METHOD run\_ubi\_report.

get\_ubi\_data( ).

call\_ubi\_alv( p\_fieldcat = it\_fieldcat p\_table = gt\_diferencias ).

ENDMETHOD. "run

METHOD run\_hu\_report.

get\_hu\_data( ).

call\_hu\_alv( CHANGING p\_table = gt\_saldos\_d ).

ENDMETHOD. "run\_hu\_report

METHOD run\_palet\_report.

get\_palet\_data( ).

call\_palet\_alv( CHANGING p\_table = gt\_palets ).

Endmethod. "run\_palet\_report

METHOD run\_depot\_report.

get\_depot\_data( ).

call\_depot\_alv( CHANGING p\_table = gt\_almacen ).

EndMETHOD. "run\_depot\_report

METHOD run\_diff\_report.

get\_diff\_data( ).

call\_diff\_alv( EXPORTING p\_cantidad = lv\_conteo p\_cantidad\_sap = lv\_conteo\_sap CHANGING p\_table = gt\_diffalmacen ).

ENDMETHOD. "run\_diff\_report

METHOD call\_ubi\_alv.

DATA: I\_START\_COLUMN TYPE i VALUE 15,

I\_START\_LINE TYPE i VALUE 2,

I\_END\_COLUMN TYPE i VALUE 60,

I\_END\_LINE TYPE i VALUE 30,

i\_title TYPE string VALUE 'Log',

i\_popup TYPE flag VALUE '',

it\_alv TYPE TABLE OF zewm\_ubications.

it\_alv = p\_table.

DATA go\_alv TYPE REF TO cl\_salv\_table.

TRY.

cl\_salv\_table=>factory(

IMPORTING

r\_salv\_table = go\_alv

CHANGING

t\_table = it\_alv[] ).

CATCH cx\_salv\_msg.

ENDTRY.

data: lr\_grid type ref to cl\_salv\_form\_layout\_grid,

lr\_grid\_1 TYPE REF TO cl\_salv\_form\_layout\_grid,

lr\_label TYPE REF TO cl\_salv\_form\_label,

lr\_text TYPE REF TO cl\_salv\_form\_text,

message TYPE string,

lr\_header type ref to cl\_salv\_form\_header\_info,

lr\_content type ref to cl\_salv\_form\_element.

message = 'Reporte de Ubicaciones NO utilizadas'.

CREATE OBJECT lr\_grid.

lr\_grid->create\_header\_information(

row = 1

column = 1

text = message

tooltip = message ).

lr\_content = lr\_grid.

go\_alv->set\_top\_of\_list( lr\_content ).

DATA: lr\_functions TYPE REF TO cl\_salv\_functions\_list,

lr\_columns TYPE REF to CL\_SALV\_COLUMNS\_TABLE,

lr\_column TYPE REF TO cl\_salv\_column,

lr\_layout TYPE REF TO CL\_SALV\_LAYOUT,

lr\_layoutkey TYPE salv\_s\_layout\_key.

lr\_layout = go\_alv->get\_layout( ).

lr\_layoutkey-REPORT = sy-repid.

lr\_layout->set\_key( lr\_layoutkey ).

lr\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).

lr\_functions = go\_alv->get\_functions( ).

lr\_functions->set\_all( 'X' ).

lr\_columns = go\_alv->get\_columns( ).

lr\_columns->set\_optimize( 'X' ).

lr\_column = lr\_columns->get\_column( 'UBICACION').

lr\_column->set\_long\_text( 'Ubicación').

lr\_columns = go\_alv->get\_columns( ).

lr\_column = lr\_columns->get\_column( 'TIPO\_ALMACEN').

lr\_column->set\_short\_text( 'Almancen').

lr\_column->set\_medium\_text( 'Almacen').

lr\_column->set\_long\_text( 'Almacen').

lr\_column = lr\_columns->get\_column( 'AREA\_ALMACEN').

lr\_column->set\_short\_text( 'Area').

lr\_column->set\_medium\_text( 'Area').

lr\_column->set\_long\_text( 'Area').

lr\_column = lr\_columns->get\_column( 'GRUPO\_ALMACEN').

lr\_column->set\_short\_text( 'Grupo').

lr\_column->set\_medium\_text( 'Grupo').

lr\_column->set\_long\_text( 'Grupo').

lr\_column = lr\_columns->get\_column( 'PESO\_MAXIMO').

lr\_column->set\_short\_text( 'Peso M').

lr\_column->set\_medium\_text( 'Peso M').

lr\_column->set\_long\_text( 'Peso M').

lr\_column = lr\_columns->get\_column( 'UNIDADM').

lr\_column->set\_short\_text( 'Unidad M').

lr\_column->set\_medium\_text( 'Unidad M').

lr\_column->set\_long\_text( 'Unidad M').

lr\_column = lr\_columns->get\_column( 'VOL\_MAXIMO').

lr\_column->set\_long\_text( 'Volumen').

lr\_column->set\_medium\_text( 'Volumen').

lr\_column->set\_long\_text( 'Volumen').

lr\_column = lr\_columns->get\_column( 'UNIDAD\_VOL').

lr\_column->set\_short\_text( 'Unidad V').

lr\_column->set\_medium\_text( 'Unidad V').

lr\_column->set\_long\_text( 'Unidad V').

lr\_column = lr\_columns->get\_column( 'PASILLO').

lr\_column->set\_short\_text( 'Pasillo').

lr\_column->set\_medium\_text( 'Pasillo').

lr\_column->set\_long\_text( 'Pasillo').

lr\_column = lr\_columns->get\_column( 'COLUMNA').

lr\_column->set\_short\_text( 'Columna').

lr\_column->set\_medium\_text( 'Columna').

lr\_column->set\_long\_text( 'Columna').

lr\_column = lr\_columns->get\_column( 'NIVEL').

lr\_column->set\_short\_text( 'Nivel').

lr\_column->set\_medium\_text( 'Nivel').

lr\_column->set\_long\_text( 'Nivel').

lr\_column = lr\_columns->get\_column( 'SUBDIV\_UBIC').

lr\_column->set\_short\_text( 'Subdivi').

lr\_column->set\_medium\_text( 'Subdivi').

lr\_column->set\_long\_text( 'Subdivi').

lr\_column = lr\_columns->get\_column( 'VERIFICACION').

lr\_column->set\_long\_text( 'Cod. Verificador').

lr\_column->set\_medium\_text( 'Cod. Verificador').

lr\_column->set\_short\_text( 'Verifica').

IF go\_alv IS BOUND.

IF i\_popup = 'X'.

go\_alv->set\_screen\_popup(

start\_column = i\_start\_column

end\_column = i\_end\_column

start\_line = i\_start\_line

end\_line = i\_end\_line ).

ENDIF.

go\_alv->display( ).

ENDIF.

ENDMETHOD. "call\_alv

METHOD call\_hu\_alv.

DATA: I\_START\_COLUMN TYPE i VALUE 15,

I\_START\_LINE TYPE i VALUE 2,

I\_END\_COLUMN TYPE i VALUE 60,

I\_END\_LINE TYPE i VALUE 30,

i\_title TYPE string VALUE 'UMp duplicadas',

i\_popup TYPE flag VALUE '',

it\_alv TYPE TABLE OF zewm\_ubications,

c\_layout TYPE disvariant-variant.

data: lr\_grid type ref to cl\_salv\_form\_layout\_grid,

lr\_grid\_1 TYPE REF TO cl\_salv\_form\_layout\_grid,

lr\_label TYPE REF TO cl\_salv\_form\_label,

lr\_text TYPE REF TO cl\_salv\_form\_text,

message TYPE string,

lr\_header type ref to cl\_salv\_form\_header\_info,

lr\_content type ref to cl\_salv\_form\_element.

\* it\_alv = p\_table.

DATA go\_alv TYPE REF TO cl\_salv\_table.

TRY.

cl\_salv\_table=>factory(

IMPORTING

r\_salv\_table = go\_alv

CHANGING

t\_table = p\_table ).

CATCH cx\_salv\_msg.

ENDTRY.

message = 'Reporte de UMp duplicadas'.

CREATE OBJECT lr\_grid.

lr\_grid->create\_header\_information(

row = 1

column = 1

text = message

tooltip = message ).

lr\_content = lr\_grid.

go\_alv->set\_top\_of\_list( lr\_content ).

DATA: lr\_functions TYPE REF TO cl\_salv\_functions\_list,

lr\_columns TYPE REF to CL\_SALV\_COLUMNS\_TABLE,

lr\_column TYPE REF TO cl\_salv\_column,

lr\_layout TYPE REF TO CL\_SALV\_LAYOUT,

lr\_layoutkey TYPE salv\_s\_layout\_key.

lr\_functions = go\_alv->get\_functions( ).

lr\_functions->set\_all( 'X' ).

lr\_columns = go\_alv->get\_columns( ).

lr\_columns->set\_optimize( 'X' ).

lr\_column = lr\_columns->get\_column( 'UNIDADM').

lr\_column->set\_long\_text( 'Unidad de Medida').

\* lr\_columns = go\_alv->get\_columns( ).

lr\_layout = go\_alv->get\_layout( ).

lr\_layoutkey-REPORT = sy-repid.

lr\_layout->set\_key( lr\_layoutkey ).

lr\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).

IF go\_alv IS BOUND.

IF i\_popup = 'X'.

go\_alv->set\_screen\_popup(

start\_column = i\_start\_column

end\_column = i\_end\_column

start\_line = i\_start\_line

end\_line = i\_end\_line ).

ENDIF.

go\_alv->display( ).

ENDIF.

ENDMETHOD. "call\_hu\_alv

ENDCLASS. "main IMPLEMENTATION

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.

PARAMETERS: r\_b1 RADIOBUTTON GROUP grb1,

r\_b2 RADIOBUTTON GROUP grb1,

r\_b3 RADIOBUTTON GROUP grb1,

r\_b4 RADIOBUTTON GROUP grb1,

r\_b5 RADIOBUTTON GROUP grb1.

SELECTION-SCREEN END OF BLOCK b1.

START-OF-SELECTION.

DATA lcl\_main TYPE REF TO main.

CREATE OBJECT lcl\_main.

IF r\_b1 IS NOT INITIAL.

lcl\_main->run\_ubi\_report( ).

ELSEIF r\_b2 IS NOT INITIAL.

lcl\_main->run\_hu\_report( ).

ELSEIF r\_b3 IS NOT INITIAL.

lcl\_main->run\_palet\_report( ).

ELSEIF r\_b4 IS NOT INITIAL.

lcl\_main->run\_depot\_report( ).

ELSEIF r\_b5 IS NOT INITIAL.

lcl\_main->run\_diff\_report( ).

ENDIF.

End-of-SELECTION.