|  |
| --- |
|  |
| Proyecto: Implementación ERP SAP RE-FX en PROMIGAS  NOMBRE DEL GAP  Diseño Técnico |
| VERSIÓN 1.0 |
|  |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| Escrito por | Francisco Javier Acevedo Guerrero | 23/12/2024 |
| Verificado por: |  | *dd/mm/yyyy* |
| Aprobado por: |  | *dd/mm/yyyy* |

Cambios en el Documento

Los cambios en el documento se detallan a continuación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Día** | **Versión** | **Escrito por** | **Revisado por** | **Descripción del Cambio** |
|  | 1.0 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Tabla 1 Control de Versiones

Index

[2. Introducción 5](#_Toc184724741)

[2.1. Alcance del documento 5](#_Toc184724742)

[2.2. Fuera de este documento 5](#_Toc184724743)

[3. Descripción Técnica de la Solución 6](#_Toc184724744)

[3.1. Resumen Funcional 6](#_Toc184724745)

[3.2. Descripción Técnica de la Solución y Flujos 6](#_Toc184724746)

[3.3. Controles de Acceso y Segregación de responsabilidades 6](#_Toc184724747)

[4. Restricciones 7](#_Toc184724748)

[4.1. Performance y Disponibilidad 7](#_Toc184724749)

[4.2. Restricciones Técnicas 7](#_Toc184724750)

[5. Impacto / Relación con otros sistemas 8](#_Toc184724751)

[6. Anexos Especificación 9](#_Toc184724752)

[7. Detalle Técnico de Construcción 10](#_Toc184724753)

[7.1. Lista de Objetos Workbench 10](#_Toc184724754)

[7.2. Lista Transportes 10](#_Toc184724755)

[7.3. Customizing 10](#_Toc184724756)

[7.4. Anexos Solución 10](#_Toc184724757)

1. **Listado de** **Definiciones de Términos, Acrónimos y Abreviaciones**

|  |  |
| --- | --- |
| **Acronym** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabla 2 – Definiciones de Términos, Acrónimos y Abreviaciones

# Introducción

## Alcance del documento

* + Generar un programa de carga de consumos que se pueda utilizar en las sociedades definidas en el proyecto de Real Estate que postee el cargue de los valores de consumo dependiendo de las reglas especificadas en los tipos de contrato.

## Fuera de este documento

Otro tipo de modificaciones fuera del alcance establecido.

# Descripción Técnica de la Solución

## Resumen Funcional

###### Función(Programa)

Dentro del proceso de promigas. En determinados contratos es necesario cargar el consumo del periodo para generar el valor total a facturar. Para ello dentro del proceso de Real Estate se utiliza la transaccion **“RESRRP”**, la cual ingresa los valores de consumo dentro del contrato.

Se necesita hacer un programa de carga que active la transaccion y haga el llenado de la informacion de acuerdo con unos parametros de entrada.

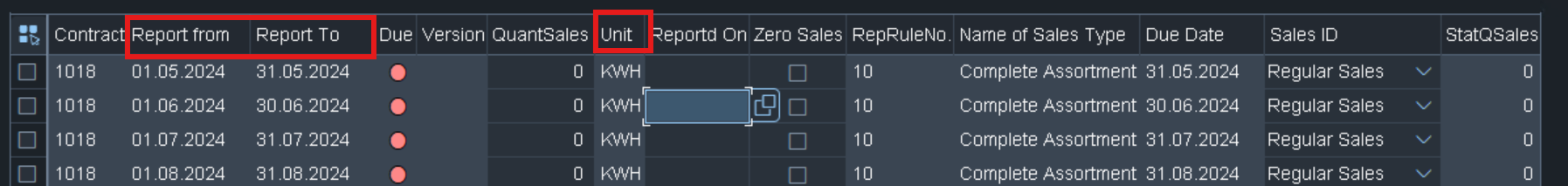
* Contract
* Company Code

|  |  |
| --- | --- |
| llaves de entrada | |
| C-CODE | #-contrato |
| RECN\_CONTRACT - BUKRS | RECN\_CONTRACT - RECNNR |



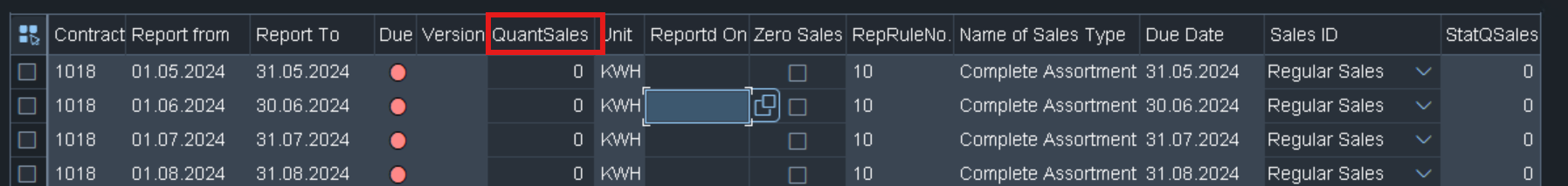
Una vez llenada la informacion de la pantalla de entrada se requiere validar los daots para hacer el posteo de la informacion. Esta validacion se hace a traves de los campos de “fecha desde”, “fecha hasta” y “unidad de medida”

|  |  |  |
| --- | --- | --- |
| campos de verificación | | |
| fecha desde | fecha hasta | unidad de medida |
| VISRREPORT - VALIDFROM | VISRREPORT - VALIDTO | VISRREPORT - SALESUNITS |



Las fechas deben estar ingresadas en formato DD.MM.AAAA. Si los parametros estan correctos se completa la informacion del campo SAP “Cantidades” tomados de la columna de excel “Lectura”

|  |  |
| --- | --- |
| campos de entrada de información | |
| lectura | valor en COP |
| VISRREPORT - QUANTITY | |



Como logica adicional se encuentra el escenario del consumo de Combustible. Para este caso no se ingresa un dato en unidades de valor Ej: 20 KW/H. Sino que se ingresa un valor en moneda Ej: $200.000 COP. Dentro de este escenario el funcionamiento del programa de carga es el mismo. Sin embargo hay que hacer una validacion extra en la entrada de los datos.

Para este caso hay que generar una validacion la cual versa sobre el archivo de carga y la transaccion en SAP. En el archivo de excel la columna a llenar la informacion pasa de ser “Lectura” a ser “Valor en COP”.

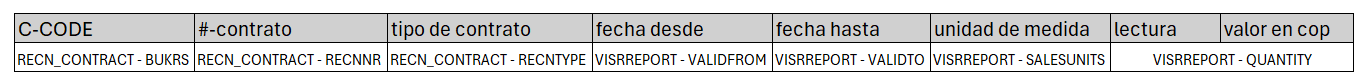
A screenshot of a computer

Description automatically generated

|  |
| --- |
| campo de validación |
| tipo de contrato |
| RECN\_CONTRACT - RECNTYPE |

La validacion que se genera es sobre el campo “Contract Type” en donde si el registro ingresado es igual a **“LI02”** la columna en el excel que se debe llenar es “Valor en COP” de lo contrario se llena la columna de “Lectura”

El layout del archivo de excel deberia tener la siguiente estructura



En ningun caso se puede tener la informacion llena en ambos campos “Lectura” Valor en COP”

A white rectangular box with black text

Description automatically generated

Como nota adicional, Si para determinado periodo no se registra ningun consumo. El registro dentro de la transaccion debe tener activo el flag “Zero Sales”

|  |
| --- |
| Zero Sales |
| No\_Sales |
| RESP\_REPORT\_S |

A screenshot of a computer

Description automatically generated

Para terminar el proceso se debe ejecutar la transaccion y con ello se carga la informacion en los contratos.

Las unidades de medida a ser usadas en el proyeto son:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sociedad** | **Condicion** | **Por** | **Descripcion Unidad** | **unidat** |
| Promigas | venta de energía esperada | fijo | Kilovatios | KW/H |
| venta energía, excedente | variable | Kilovatios (registro del medidor) | KW/H |
| Promisol | venta de energía o | Energía de vapor | Libras unidad libras/hora (Registro de unidad de medida y se calcula el P\*Q) | LB/H |
| Arrendamiento maq/Eq. | Generación energía gas | megavatio/hora | MW/H |
| Todos vehículos | Combustible | variable | valor total en COP | COP |

## Descripción Técnica de la Solución y Flujos

Se crea la clase ZCX\_EXCEL\_HANDLER Exception class for Excel handler

Se crea la clase ZCL\_EXCEL\_HANDLER General Excel upload/download handler class

Para realizer cargas de archivos de excel

Se crea Programa e includes

REPORT ZRE\_CARGA\_CONSUMO\_CONTRATO.   
INCLUDE ZRE\_CARGA\_CONSUMO\_CONTRATO\_TOP.Variables

INCLUDE ZRE\_CARGA\_CONSUMO\_CONTRATO\_SCR.Pantalla de selección

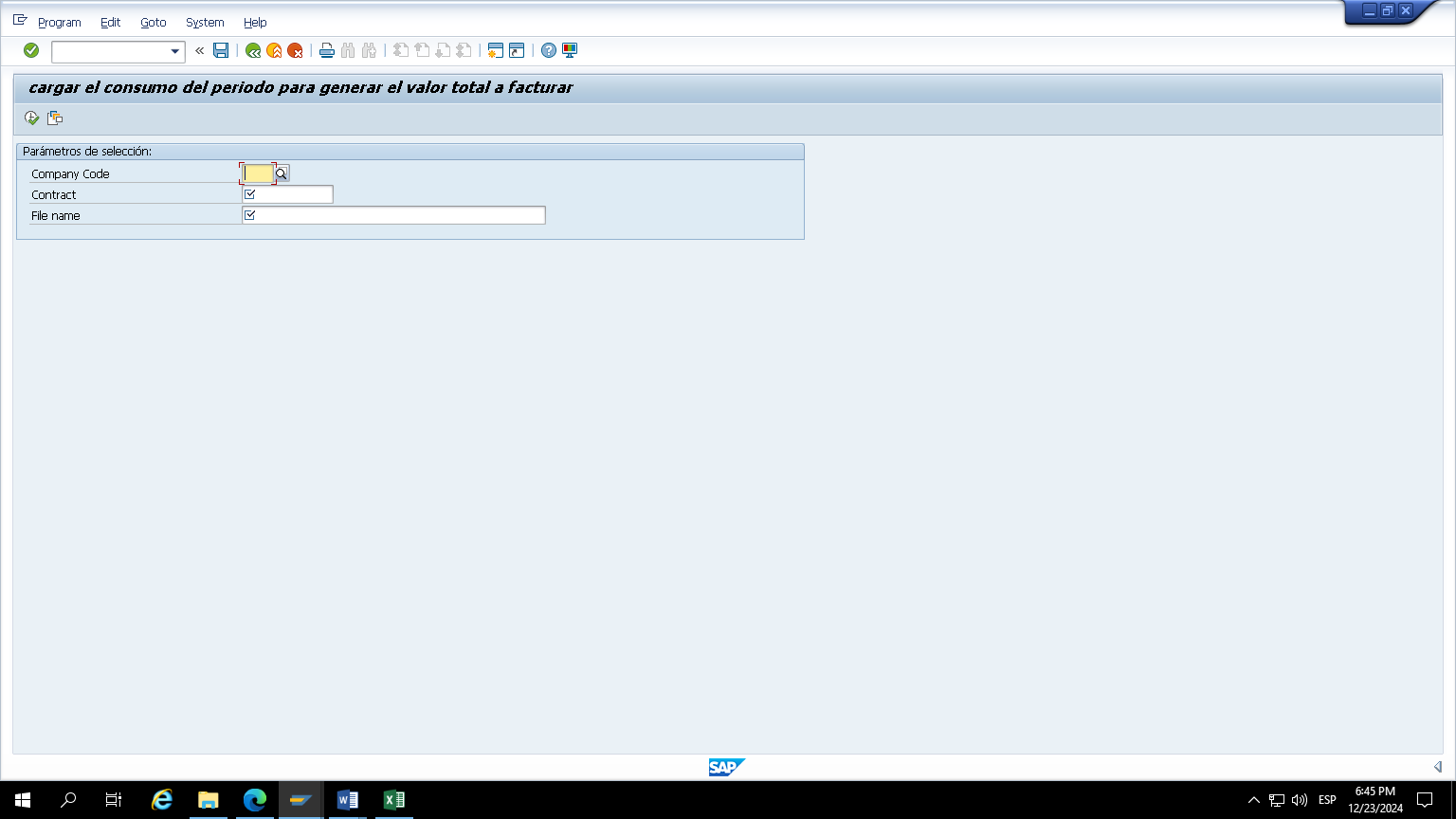
*\*&---------------------------------------------------------------------\**  
*\*& Include          ZRE\_CARGA\_CONSUMO\_CONTRATO\_SCR*  
*\*&---------------------------------------------------------------------\**  
SELECTION-SCREEN: BEGIN OF BLOCK b001 WITH FRAME TITLE TEXT-001.  
PARAMETERS: p\_bukrs TYPE RECN\_CONTRACT-BUKRS     OBLIGATORY .  
PARAMETERS: p\_RCNNR    type RECN\_CONTRACT-RECNNR   OBLIGATORY.  
PARAMETERS: p\_file  TYPE localfile OBLIGATORY.  
PARAMETERS: p\_batch TYPE sy-batch  NO-DISPLAY .  
SELECTION-SCREEN: END OF BLOCK b001.

INCLUDE ZRE\_CARGA\_CONSUMO\_CONTRATO\_CLD.Definición de clases locales  
*\*&---------------------------------------------------------------------\**  
*\*& Include          ZRE\_CARGA\_CONSUMO\_CONTRATO\_CLD*  
*\*&---------------------------------------------------------------------\**  
class lcl\_carga\_consumo definition.  
  
  PUBLIC SECTION.  
  data gt\_list type STANDARD TABLE OF RESR\_REPORT\_S.  
  data gt\_listx type re\_t\_report\_x.  
  methods constructor  
    importing  
      !i\_bukrs  type recn\_contract-bukrs  
      !i\_recnnr type recn\_contract-recnnr  
      !i\_file   type localfile  
      !i\_batch  type sy-batch .  
  class-methods f4\_file  
    returning  
      value(r\_localfile) type localfile .  
  
  METHODS start\_of\_selection.  
  
  methods set\_file  
    importing  
      !i\_file type localfile .  
    PRIVATE SECTION.  
  TYPES: BEGIN OF ltp\_data\_excel,  
        bukrs  type RESR\_REPORT\_S-bukrs,  
        recnnr type RESR\_REPORT\_S-recnnr,  
        RECNTYPE type RECNCONTRACTTYPE,  
        VALIDFROM Type  RESRRPFROM,  
        VALIDTO Type  RESRRPTO,  
        SALESUNIT Type  VVUMSEINH,  
        SALESCURR Type  WAERS,  
        QUANTITY  Type  RESRQUANTITY,  
        GROSS\_SALES Type  RESRSALESGROSS,  
        END  OF ltp\_data\_excel.  
  METHODs call\_bapi  
  IMPORTING  
    !I\_BUKRS  type BUKRS  
    !I\_RECNNR  type RECNNUMBER.  
    methods upload\_excel .  
  METHODs get\_data\_RESRRP.  
  METHODs display\_data.  
  METHODS validate\_excel.  
  METHODS show\_log  
    IMPORTING  
      !i\_t\_return type BAPIRET2\_T.  
  DATA et\_VICNCN type VICNCN.  
  data GV\_BUKRS type recn\_contract-bukrs.  
  data GV\_LOCALFILE type LOCALFILE .  
  data GV\_recnnr type recn\_contract-recnnr.  
  data GT\_RETURN type BAPIRET2\_T .  
  data: sales\_rep type standard table of bapi\_re\_sales\_rep\_dat,  
        return    type standard table of  bapiret2.  
  DATA gt\_excel type STANDARD TABLE OF ltp\_data\_excel.  
  data GV\_BATCH type SY-BATCH .  
endclass.

INCLUDE ZRE\_CARGA\_CONSUMO\_CONTRATO\_CLI.Implementación de clases locales  
*\*&---------------------------------------------------------------------\**  
*\*& Include          ZRE\_CARGA\_CONSUMO\_CONTRATO\_CLI*  
*\*&---------------------------------------------------------------------\**  
class lcl\_carga\_consumo implementation.  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Instance Public Method lcl\_carga\_consumo->CONSTRUCTOR*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_BUKRS                        TYPE        recn\_contract-bukrs*  
*\* | [--->] I\_RECNNR                        TYPE        recn\_contract-recnnr*  
*\* | [--->] I\_FILE                         TYPE        LOCALFILE*  
*\* | [--->] I\_BATCH                        TYPE        SY-BATCH*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  method constructor.  
    me->gv\_bukrs = i\_bukrs.  
    me->gv\_recnnr = i\_recnnr.  
*\*    me->create\_ref\_cont( ).*  
    me->gv\_batch = i\_batch.  
  endmethod.  
  
  method f4\_file.  
    data: lv\_desktop  type string,  
          lv\_wind\_tit type string,  
          lt\_files    type filetable,  
          lv\_rcode    type int4.  
  
    call method cl\_gui\_frontend\_services=>get\_desktop\_directory  
      changing  
        desktop\_directory    = lv\_desktop  
      exceptions  
        cntl\_error           = 1  
        error\_no\_gui         = 2  
        not\_supported\_by\_gui = 3  
        others               = 4.  
    if sy-subrc <> 0.  
      message e001(00) with text-002.  
    endif.  
  
    call method cl\_gui\_cfw=>update\_view  
      exceptions  
        cntl\_system\_error = 1  
        cntl\_error        = 2  
        others            = 3.  
  
    lv\_wind\_tit = text-003.  
  
    call method cl\_gui\_frontend\_services=>file\_open\_dialog  
      exporting  
        window\_title            = lv\_wind\_tit  
        default\_extension       = '.xls'  
        file\_filter             = '.xls'  
        initial\_directory       = lv\_desktop  
      changing  
        file\_table              = lt\_files  
        rc                      = lv\_rcode  
      exceptions  
        file\_open\_dialog\_failed = 1  
        cntl\_error              = 2  
        error\_no\_gui            = 3  
        not\_supported\_by\_gui    = 4  
        others                  = 5.  
    if sy-subrc <> 0.  
      message e001(00) with text-004.  
    endif.  
  
    read table lt\_files index 1 into data(ls\_files).  
    if sy-subrc eq 0.  
*\*      me->gv\_localfile = ls\_files-filename.*  
      r\_localfile      = ls\_files-filename.  
    else.  
      message i001(00) with text-005.  
    endif.  
  
  endmethod.  
  
  method set\_file.  
    me->gv\_localfile = i\_file.  
  endmethod.  
  
  method upload\_excel.  
    data(lo\_excel\_handler) = new zcl\_excel\_handler( ).  
    lo\_excel\_handler->upload\_xlsx(  
    exporting  
     iv\_file\_path = conv #( me->gv\_localfile )  
     iv\_server    = abap\_false  
     ir\_table     = ref #( gt\_excel ) ).  
    validate\_excel( ).  
  endmethod.  
  method start\_of\_selection.  
    me->upload\_excel( ).  
    if me->gt\_return is initial.  
      me->get\_data\_resrrp( ).  
      if me->gt\_list is not initial.  
        me->call\_bapi(  
          exporting  
            i\_bukrs  = et\_vicncn-bukrs  
            i\_recnnr = et\_vicncn-recnnr  
        ).  
        refresh: me->gt\_list,me->gt\_listx.  
        me->get\_data\_resrrp( ).  
        me->display\_data( ).  
      else.  
        message i001(00) with text-006.  
      endif.  
  
    else.  
      me->show\_log( i\_t\_return = me->gt\_return ).  
    endif.  
  endmethod.  
  method get\_data\_resrrp.  
    data:         *"grid data (buffered for all pbo/pai-processing)*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
      go\_cn       type ref to if\_recn\_contract,  
      gto\_cn      type re\_t\_if\_recn\_contract,  
      gd\_activity type reca1\_activity,  
      gf\_enqueue  type abap\_bool.  
    gd\_activity = '02'.  
  
    select single \*  
              into @et\_vicncn  
              from vicncn  
              where bukrs       eq @gv\_bukrs  
                 and recnnr     eq @gv\_recnnr.  
  
    call method cf\_resr\_contract=>find\_by\_intreno  
      exporting  
        id\_intreno    = et\_vicncn-intreno  
        id\_activity   = gd\_activity  
        if\_auth\_check = abap\_true  
*\*       IF\_ENQUEUE    = ABAP\_FALSE*  
      receiving  
        ro\_instance   = go\_cn  
      exceptions  
        error         = 1  
        others        = 2.  
  
  
    insert go\_cn into table gto\_cn.  
  
    data:  
      lo\_report\_mngr  type ref to  if\_resr\_report\_multiple\_mngr,  
      lt\_rsosalestype type re\_t\_rsosalestype,  
      ls\_rsosalestype like line of lt\_rsosalestype,  
      lt\_rhythm       type re\_t\_rhythmtype\_x,  
      lo\_msglist      type ref to if\_reca\_message\_list.  
  
    call method cf\_reca\_message\_list=>create  
      receiving  
        ro\_instance = lo\_msglist.  
    call method cf\_resr\_report\_multiple\_mngr=>find\_by\_list  
      exporting  
        ito\_contract    = gto\_cn  
*\*       IT\_RSOTERMNO    =*  
        id\_rhythmtype   = '0001' *"<ls\_rhythm>-rhythmtype*  
        it\_rsosalestype = lt\_rsosalestype  
*\*       ID\_RPFROM       = RECA0\_DATE-MIN*  
        id\_rpto         = sy-datum *"r\_viewto*  
        id\_typeofreport = abap\_false *"r\_type*  
        id\_selecttype   = '02' *" 04 r\_select*  
        if\_enqueue      = abap\_false *"gf\_enqueue*  
        io\_msglist      = lo\_msglist  
      receiving  
        ro\_instance     = lo\_report\_mngr.  
    constants:  
      lc\_classname\_data type char30    value 'CL\_RESR\_DATA\_REPORT',  
      lc\_classname\_view type char30    value 'CL\_REIS\_VIEW\_DEFAULT',  
      lc\_classname\_edit type char30    value 'CL\_REIS\_VIEW\_EDIT',  
      lc\_reportid       type reisrepid value 'SR\_REPORT'.  
    data:  
      lo\_data\_specific type ref to cl\_resr\_data\_report,  
*\*   model and view container*  
      lo\_data          type ref to cl\_reis\_data.  
    create object lo\_data  
     type  
     (lc\_classname\_data)  
     exporting  
       id\_reportid         = lc\_reportid.  
  
  
    try.  
        lo\_data\_specific ?= lo\_data.  
        lo\_data\_specific->set\_data\_mngr( lo\_report\_mngr ).  
        call method lo\_data\_specific->get\_data  
          exceptions  
            error  = 1  
            others = 2.  
        if sy-subrc <> 0.  
*\*        mac\_symsg\_send\_as\_type 'S'.*  
          return.  
        endif.  
        field-symbols:  
          <fs\_lisx>  type re\_t\_report\_x,  
          <lt\_listx> type standard table,  
          <lt\_list>  type standard table,  
          <fs\_lis>   type resr\_report\_s.  
        assign lo\_data->mr\_list\_display->\* to <lt\_listx>.  
        move-corresponding <lt\_listx> to gt\_listx.  
        assign lo\_data->mr\_list->\* to <lt\_list>.  
        data bapi\_re type bapi\_re\_sales\_rep\_int.  
        data bapi\_re\_dat  type  bapi\_re\_sales\_rep\_dat.  
        sort gt\_excel by bukrs recnnr validfrom validto.  
        clear sales\_rep.  
        loop at <lt\_list> assigning <fs\_lis>.  
          read table gt\_excel into data(ls\_excel) with key validfrom = <fs\_lis>-validfrom validto = <fs\_lis>-validto binary search.  
          if sy-subrc eq 0.  
            <fs\_lis>-quantity = ls\_excel-quantity.  
            <fs\_lis>-salesunit = ls\_excel-salesunit.  
            <fs\_lis>-salescurr = ls\_excel-salescurr.  
            <fs\_lis>-net\_sales = ls\_excel-gross\_sales.  
  
          else.  
            <fs\_lis>-no\_sales = abap\_true.  
          endif.  
*\*          <fs\_lis>-quantity = 10.*  
*\*          <fs\_lis>-net\_sales = 256.*  
          move-corresponding <fs\_lis> to bapi\_re.  
          call function 'CONV\_TO\_RE\_SALES\_REP\_DAT'  
            exporting  
              bapi\_re\_sales\_rep\_int = bapi\_re  
            changing  
              bapi\_re\_sales\_rep\_dat = bapi\_re\_dat.  
          append bapi\_re\_dat to sales\_rep.  
          append <fs\_lis> to gt\_list.  
        endloop.  
      catch cx\_sy\_move\_cast\_error..  
    endtry.  
  endmethod.  
  
  method call\_bapi.  
    clear return.  
    call function 'BAPI\_RE\_SALES\_REP\_REPORT'  
      exporting  
        compcode       = et\_vicncn-bukrs  
        contractnumber = et\_vicncn-recnnr  
*\*       TEST\_RUN       = ' '*  
      tables  
        sales\_rep      = sales\_rep  
        return         = return.  
  
    call function 'BAPI\_TRANSACTION\_COMMIT'  
      exporting  
        wait = abap\_true  
*\*     IMPORTING*  
*\*       RETURN        =*  
      .  
me->show\_log( i\_t\_return = return ).  
  endmethod.  
  method display\_data.  
    data: lo\_gr\_alv       type ref to cl\_salv\_table, *" Variables for ALV properties*  
          lo\_gr\_functions type ref to cl\_salv\_functions\_list.  
  
*\*  data: lo\_event\_handler type ref to cl\_handler, " Variables for events*  
*\*        lo\_events        type ref to cl\_salv\_events\_table.*  
  
    data: lo\_grid        type ref to cl\_salv\_form\_layout\_grid, *" Variables for header*  
          lo\_layout\_logo type ref to cl\_salv\_form\_layout\_logo,  
          lo\_content     type ref to cl\_salv\_form\_element,  
          lv\_title       type string,  
          lv\_rows        type string.  
  
    data: lo\_layout type ref to cl\_salv\_layout, *" Variables for enabling Save button*  
          lv\_key    type salv\_s\_layout\_key.  
  
    data: lo\_display type ref to cl\_salv\_display\_settings. *" Variable for layout settings*  
  
    data: lo\_selections type ref to cl\_salv\_selections, *" Variables for selection mode and column properties*  
          lo\_columns    type ref to cl\_salv\_columns,  
          lo\_column     type ref to cl\_salv\_column\_table.  
  
*\* Create the ALV object*  
    try.  
        call method cl\_salv\_table=>factory  
          importing  
            r\_salv\_table = lo\_gr\_alv  
          changing  
            t\_table      = gt\_listx.  
      catch cx\_salv\_msg.  
    endtry.  
  
*\* Let's show all default buttons of ALV*  
    lo\_gr\_functions = lo\_gr\_alv->get\_functions( ).  
    lo\_gr\_functions->set\_all( abap\_true ).  
  
*\* Fit the columns*  
    lo\_columns = lo\_gr\_alv->get\_columns( ).  
    lo\_columns->set\_optimize( 'X' ).  
  
*\* Create header*  
    describe table gt\_listx lines lv\_rows.  
    concatenate 'Número de registros: ' lv\_rows into lv\_title separated by space.  
  
*\*  concatenate 'Number of lv\_rows: ' lv\_rows into lv\_title separated by space.*  
  
    create object lo\_grid.  
    create object lo\_layout\_logo.  
    lo\_grid->create\_label( row = 1 column = 1 text = lv\_title tooltip = lv\_title ).  
    lo\_layout\_logo->set\_left\_content( lo\_grid ).  
    lo\_content = lo\_layout\_logo.  
    lo\_gr\_alv->set\_top\_of\_list( lo\_content ).  
  
*\* Apply zebra style to lv\_rows*  
    lo\_display = lo\_gr\_alv->get\_display\_settings( ).  
    lo\_display->set\_striped\_pattern( cl\_salv\_display\_settings=>true ).  
  
*\* Enable the save layout buttons*  
    lv\_key-report = sy-repid.  
    lo\_layout = lo\_gr\_alv->get\_layout( ).  
    lo\_layout->set\_key( lv\_key ).  
    lo\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).  
    lo\_layout->set\_default( abap\_true ).  
  
*\*\* Register events*  
*\*  lo\_events = lo\_gr\_alv->get\_event( ).*  
*\*  create object lo\_event\_handler.*  
*\*  set handler lo\_event\_handler->on\_double\_click for lo\_events.*  
  
*\* Enable cell selection mode*  
    lo\_selections = lo\_gr\_alv->get\_selections( ).  
    lo\_selections->set\_selection\_mode( if\_salv\_c\_selection\_mode=>row\_column ).  
  
*\*  try.*  
*\*      lo\_column ?= lo\_columns->get\_column( 'BUKRS' ). "*  
*\*      lo\_column->set\_visible( if\_salv\_c\_bool\_sap=>true ).*  
*\*      lo\_column->set\_long\_text( 'MyTitle' ).*  
*\*      lo\_column->set\_medium\_text( 'MyTitle' ).*  
*\*      lo\_column->set\_short\_text( 'MyTitle' ).*  
*\*    catch cx\_salv\_not\_found.*  
*\*    catch cx\_salv\_existing.*  
*\*    catch cx\_salv\_data\_error.*  
*\*  endtry.*  
  
    lo\_gr\_alv->display( ).  
  endmethod.  
  method validate\_excel.  
    data ls\_return type bapiret2.  
    loop at gt\_excel assigning field-symbol(<fs\_excel>).  
      clear ls\_return.  
      ls\_return-message\_v2 = sy-tabix.  
      <fs\_excel>-recnnr = |{ <fs\_excel>-recnnr alpha = in }|.  
      if <fs\_excel>-quantity is not initial and <fs\_excel>-gross\_sales is not initial.  
*\*        ls\_return-message\_v2 = sy-tabix.*  
        concatenate 'Fila' ls\_return-message\_v2 text-007 into ls\_return-message.  
        ls\_return-id         = '0'.  
        ls\_return-number     = '001'.  
        ls\_return-type       = 'E'.  
        ls\_return-message\_v1 = text-007.  
*\*      ls\_return-message\_v2 = TEXT-007.*  
*\*      ls\_return-message\_v3 = lv\_bukrs.*  
*\*      ls\_return-message\_v4 = TEXT-008.*  
  
        append ls\_return to me->gt\_return.  
      endif.  
      IF <fs\_excel>-recntype EQ 'LI01' AND <fs\_excel>-quantity is INITIAL.  
        concatenate 'Fila' ls\_return-message\_v2 text-008 into ls\_return-message.  
        ls\_return-id         = '0'.  
        ls\_return-number     = '001'.  
        ls\_return-type       = 'E'.  
        ls\_return-message\_v1 = text-008.  
*\*      ls\_return-message\_v2 = TEXT-007.*  
*\*      ls\_return-message\_v3 = lv\_bukrs.*  
*\*      ls\_return-message\_v4 = TEXT-008.*  
  
        append ls\_return to me->gt\_return.  
  
      ENDIF.  
            IF <fs\_excel>-recntype EQ 'LI02' AND <fs\_excel>-gross\_sales is INITIAL.  
        concatenate 'Fila' ls\_return-message\_v2 text-009 into ls\_return-message.  
        ls\_return-id         = '0'.  
        ls\_return-number     = '001'.  
        ls\_return-type       = 'E'.  
        ls\_return-message\_v1 = text-009.  
*\*      ls\_return-message\_v2 = TEXT-007.*  
*\*      ls\_return-message\_v3 = lv\_bukrs.*  
*\*      ls\_return-message\_v4 = TEXT-008.*  
  
        append ls\_return to me->gt\_return.  
  
      ENDIF.  
    endloop.  
    delete gt\_excel where bukrs ne gv\_bukrs.  
    delete gt\_excel where recnnr ne gv\_recnnr.  
  
  endmethod.  
  METHOD show\_log.  
*\*---Variables locales*  
    data: lv\_object    type balobj\_d,  
          lv\_subobject type balsubobj,  
          lv\_id        type balnrext.  
  
    data(lo\_log) = new zcl\_bclog(  
        i\_object    = lv\_object  
        i\_subobject = lv\_subobject  
        i\_prog      = sy-cprog  
        i\_id        = lv\_id ).  
  
*\*---Add Table with messages*  
    lo\_log->add\_t\_message\_bapiret( i\_t\_return ).  
*\*---Show Log*  
    lo\_log->show\_log( ).  
  endmethod.  
endclass.

INCLUDE ZRE\_CARGA\_CONSUMO\_CONTRATO\_EVT.Eventos

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZRE\_CARGA\_CONSUMO\_CONTRATO\_EVT*  
*\*&---------------------------------------------------------------------\**  
at selection-screen on value-request for p\_file.  
  p\_file = lcl\_carga\_consumo=>f4\_file( ).  
  
start-of-selection.  
  data(lo\_cc) = new lcl\_carga\_consumo(  
    i\_bukrs  =  p\_bukrs  
    i\_recnnr =  p\_rcnnr  
    i\_file   =  p\_file  
    i\_batch  =  p\_batch  
  ).  
  
  if p\_file is not initial.  
    lo\_cc->set\_file( p\_file ).  
  endif.  
  lo\_cc->start\_of\_selection( ).



## Controles de Acceso y Segregación de responsabilidades

# Restricciones

## Performance y Disponibilidad

## Restricciones Técnicas

# Impacto / Relación con otros sistemas

# Anexos Especificación

# Detalle Técnico de Construcción

## Lista de Objetos Workbench

Class (ABAP Objects) R3TR CLAS ZCX\_EXCEL\_HANDLER

Class (ABAP Objects) R3TR CLAS ZCL\_EXCEL\_HANDLER

Program R3TR PROG ZRE\_CARGA\_CONSUMO\_CONTRATO

Program R3TR PROG ZRE\_CARGA\_CONSUMO\_CONTRATO\_TOP

Program R3TR PROG ZRE\_CARGA\_CONSUMO\_CONTRATO\_SCR

Program R3TR PROG ZRE\_CARGA\_CONSUMO\_CONTRATO\_EVT

Program R3TR PROG ZRE\_CARGA\_CONSUMO\_CONTRATO\_CLD

Program R3TR PROG ZRE\_CARGA\_CONSUMO\_CONTRATO\_CLI

Transaction R3TR TRAN ZRE\_CARGA\_CONSUMO

## Lista Transportes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Secuencia | Orden | Descripción breve | Titular | C/W |
| 1 | DEVK909273 | GAP02 cargar el consumo del periodo #1 | DAHE00C |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

## Customizing

## Anexos Solución