DOCUMENTACIÓN TÉCNICA

SERVICIO DE CONSULTORIA Y DESARROLLO ABAP

PY Implementación de Servicios EPP

30 de junio del 2025

Contenido

1 Objetivo del Documento 3
2 Proyecto 3
3 Especificación Funcional 3
4 Documentación Técnica 3
4.1 Objetos Desarrollados 3
4.2 Objeto de Autorización 4
4.3 Estructura del programa 4
4.4 Caso de Prueba Realizado 5
4.5 Listado de OT's 7
4.6 Código Fuente 7
4.6.1 Clase ZEPP_SERVICES 7
4.6.2 Estructura ZEPP_S_MATE 22
4.6.3 Estructura ZEPP_S_MOVI 22
4.6.4 Estructura ZEPP_S_NEWR 23
4.6.5 Estructura ZEPP_S_RESE 23

1 Objetivo del Documento

El presente documento tiene como finalidad describir la solución técnica implementada para la solución del requerimiento.

2 Proyecto

PY Implementación de Servicios Web EPP

3 Especificación Funcional

LAREDO requiere implementar el proceso de EPPs para el personal utilizando un aplicativo Web/Mobile. Este aplicativo requiere consultar información de la web service para consumo de data de Empleados y obreros y también de SAP, ya que tendrá opciones de reportes para lo cual es necesario consultar datos de SAP de materiales EPPs, stock disponible y movimientos de mercancía.

Asimismo, una vez aprobada la reserva que generan en el aplicativo esta sea generada en SAP de manera automática para continuar con el proceso en SAP para el despacho de los materiales.

4 Documentación Técnica

4.1 Objetos Desarrollados

Tipo	Objeto	Descripción
Paquete	ZEPP	
Clase	ZEPP_SERVICES	
Vista DDL	ZEPP_DD_MATERIAL	Servicio de Materiales
Vista DDL	ZEPP_DD_SUM_RESB	Cant. Reserva, Pend. de Despacho
Estructura	ZEPP_S_MATE	Materiales
Estructura	ZEPP_S_MOVI	Movimientos
Estructura	ZEPP_S_NEWR	Nueva Reserva
Estructura	ZEPP_S_RESE	Reserva
Tipo tabla	ZEPP_TT_MATE	Materiales
Tipo tabla	ZEPP_TT_MOVI	Movimientos
Tipo tabla	ZEPP_TT_NEWR	Nueva Reserva
Tipo tabla	ZEPP_TT_RESE	Reserva

Servicio OData	ZWS_EPP_SERVICE	

Objetos desarrollados en SAP Payroll

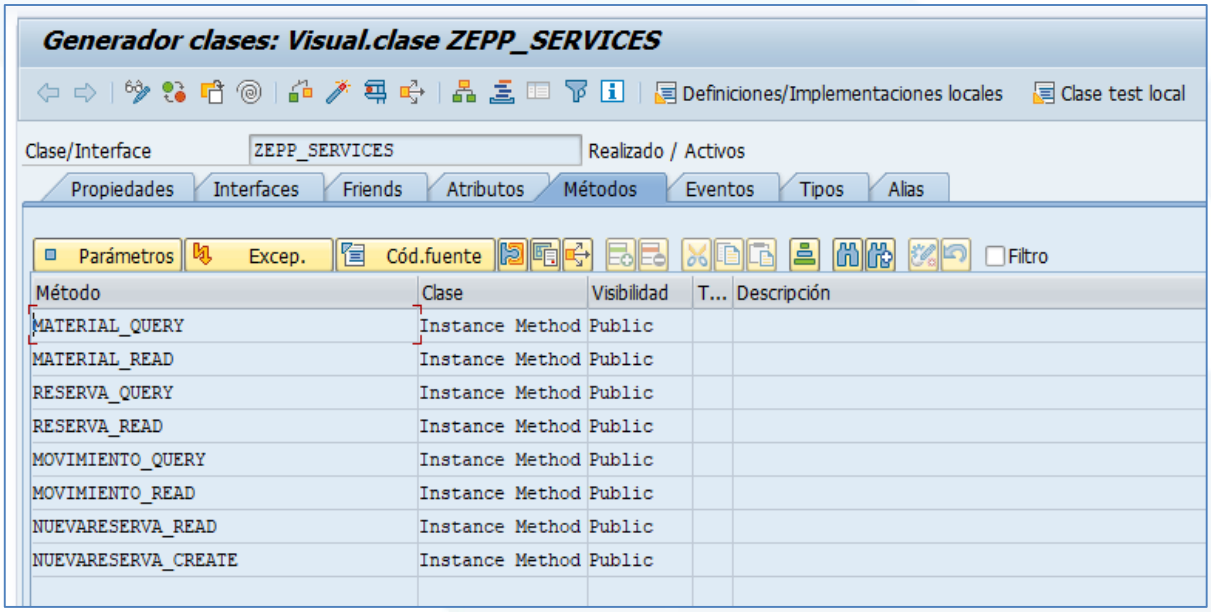
Tipo	Objeto	Descripción
Grupo de funciones	ZGFWS_INF_TRAB2	
Función	ZRFC_INF_TRABAJADORES 2	RFC Envio de información de trabajadores
Clase	ZCL_INF_TRABAJADOR2	REST Información del trabajador
Clase	ZCL_INF_TRABAJADOR_RE SOURCE2	REST Información del trabajador
Estructura	ZES_INF_OBREROS2	Estructura para enviar información de empleados WS
Estructura	ZES_INF_EMPLEADOS2	Estructura para enviar información de empleados WS
Tipo de Tabla	ZTES_OBREROS2	Tipo tabla para enviar información de obreros
Tipo de Tabla	ZTES_EMPLEADOS2	Tipo tabla para enviar información de empleados

4.2 Objeto de Autorización

Al ser un servicio OData, las autorizaciones se van a gestionar mediante el perfil del usuario de comunicación que se utilice para tal fin,

4.3 Estructura del programa

La construcción del servicio OData ZWS_EPP_SERVICE se ha preparado para que gestione desde la clase ZEPP_SERVICES, es ahí donde se ha programado la lógica para ejecución del servicio en tipo READ (un sólo registro) y tipo QUERY (varios registros).

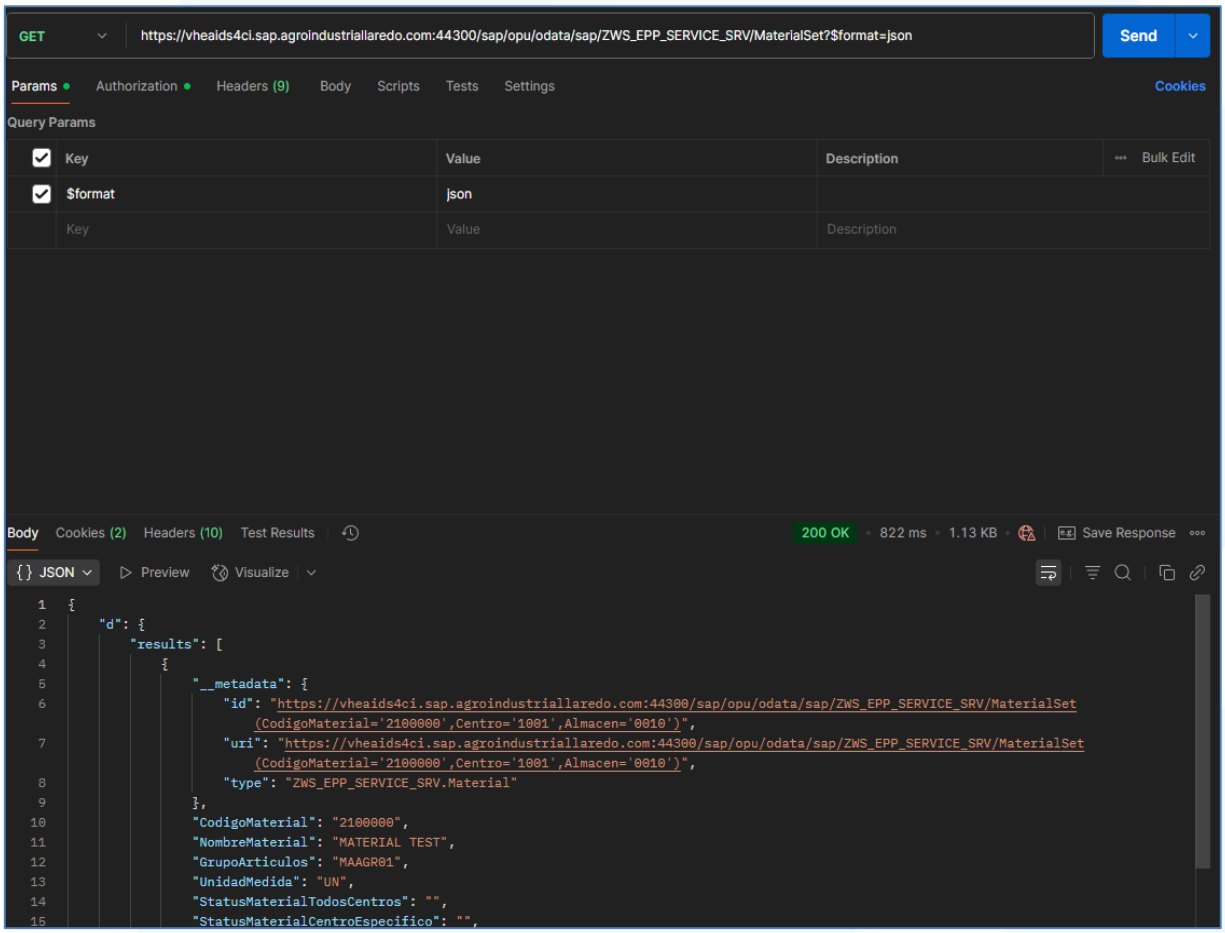


Cada uno de ellos está sujeto al servicio OData creado.

4.4 Caso de Prueba Realizado

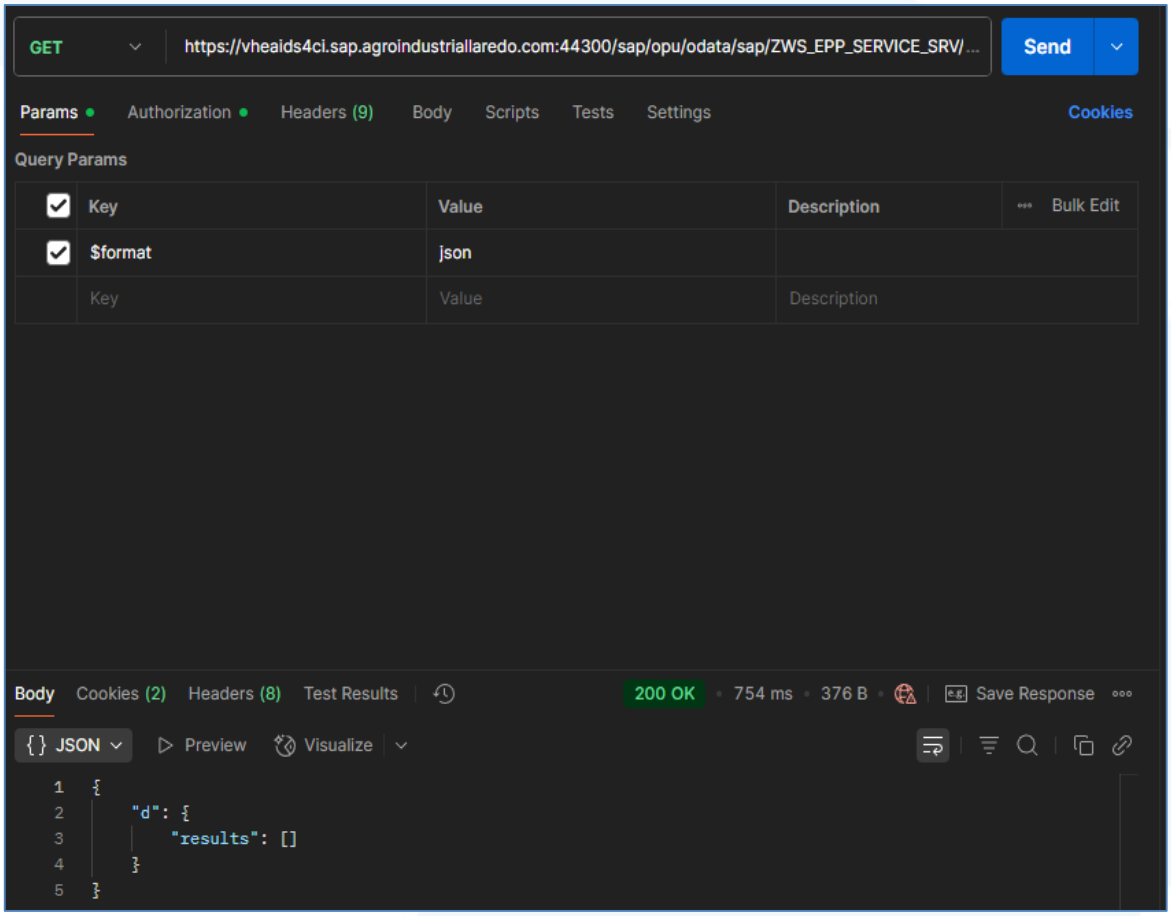
Servicio de Materiales

https://vheaids4ci.sap.agroindustriallaredo.com:44300/sap/opu/odata/sap/ZWS_EPP_SERVICE_SRV/MaterialSet?$format=json



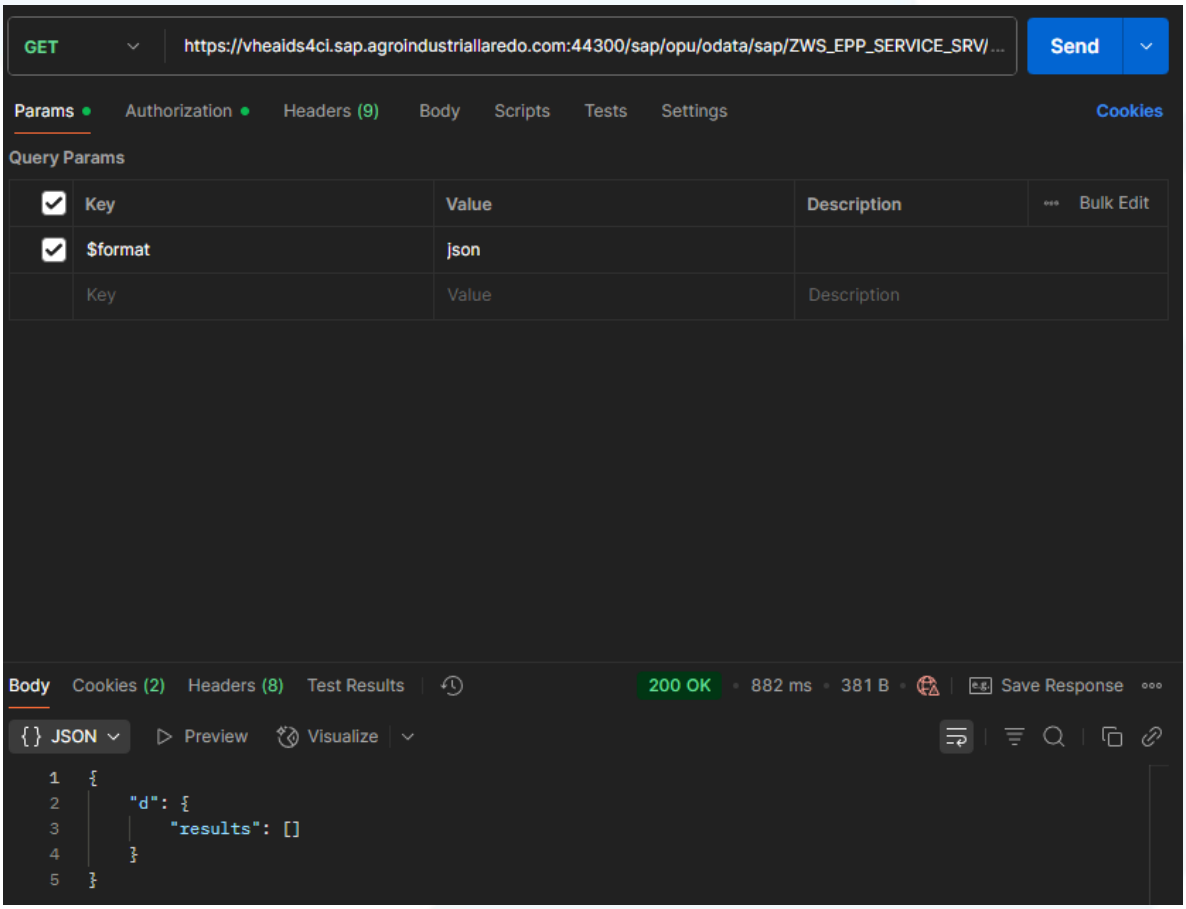
Servicio de Movimientos

https://vheaids4ci.sap.agroindustriallaredo.com:44300/sap/opu/odata/sap/ZWS_EPP_SERVICE_SRV/MovimientoSet?$format=json



Reserva

https://vheaids4ci.sap.agroindustriallaredo.com:44300/sap/opu/odata/sap/ZWS_EPP_SERVICE_SRV/ReservaSet?$format=json



Servicio en SAP Payroll de trabajadores.

URL: URL: https://my640147.payroll.ondemand.com/sap/bc/zws_inf_trab2/09092024?sap-client=100&saml2=disabledn

4.5 Listado de OT's

OT	Descripción	Tipo	Orden
DS4K901828	WB: Servicios EPP (SAP ERP)	WorkBench	1
PQMK902716	WS: Trabajadores (SAP Payroll)	WorkBench	2

4.6 Código Fuente

4.6.1 Clase ZEPP_SERVICES

```
class ZEPP_SERVICES definition
  public
  final
  create public .

public section.

  methods MATERIAL_QUERY
    importing
      !IT_FILTER type /IWBEP/T_MGW_SELECT_OPTION optional
      !IV_MATNR type MATNR optional
      !IV_WERKS type WERKS_D optional
      !IV_LGORT type LGORT_D optional
    exporting
      !ET_RETURN type BAPIRET2_TT
      !ET_MATERIAL type ZEPP_TT_MATE .
  methods MATERIAL_READ
    importing
      !IT_KEY_TAB type /IWBEP/T_MGW_NAME_VALUE_PAIR optional
    exporting
      !ES_MATERIAL type ZEPP_S_MATE
      !ET_RETURN type BAPIRET2_TT .
  methods RESERVA_QUERY
    importing
      !IV_MATNR type MATNR optional
      !IV_BUDAT type BUDAT optional
      !IV_WEMPF type WEMPF optional
      !IV_RSNUM type RSNUM optional
      !IV_RSPOS type RSPOS optional
      !IT_FILTER type /IWBEP/T_MGW_SELECT_OPTION optional
    exporting
      !ET_RESERVA type ZEPP_TT_RESE
      !ET_RETURN type BAPIRET2_TT .
  methods RESERVA_READ
    importing
      !IT_KEY_TAB type /IWBEP/T_MGW_NAME_VALUE_PAIR optional
    exporting
      !ES_RESERVA type ZEPP_S_RESE
      !ET_RETURN type BAPIRET2_TT .
  methods MOVIMIENTO_QUERY
    importing
```

```
      !IT_FILTER type /IWBEP/T_MGW_SELECT_OPTION optional
      !IV_RSNUM type RSNUM optional
      !IV_RSPOS type RSPOS optional
      !IV_MBLNR type MBLNR optional
      !IV_MJAHR type MJAHR optional
      !IV_MATNR type MATNR optional
      !IV_ZEILE type MBLPO optional
      !IV_WEMPF type WEMPF optional
      !IV_BUDAT type BUDAT optional
    exporting
      !ET_RETURN type BAPIRET2_TT
      !ET_MOVIMIENTO type ZEPP_TT_MOVI .
  methods MOVIMIENTO_READ
    importing
      !IT_KEY_TAB type /IWBEP/T_MGW_NAME_VALUE_PAIR optional
    exporting
      !ES_MOVIMIENTO type ZEPP_S_MOVI
      !ET_RETURN type BAPIRET2_TT .
  methods NUEVARESERVA_READ
    exporting
      !ES_NUEVARESERVA type ZEPP_S_NEWR
      !ET_RETURN type BAPIRET2_TT .
  methods NUEVARESERVA_CREATE
    exporting
      !EP_RSNUM type RSNUM
      !ET_RETURN type BAPIRET2_TT
    changing
      !IT_NUEVARESERVA type ZEPP_TT_NEWR optional .
  PROTECTED SECTION.
private section.

  data C_MAAGR01 type MATKL value 'MAAGR01' ##NO_TEXT.
ENDCLASS.



CLASS ZEPP_SERVICES IMPLEMENTATION.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->MATERIAL_QUERY
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IT_FILTER                      TYPE
/IWBEP/T_MGW_SELECT_OPTION(optional)
* | [--->] IV_MATNR                       TYPE        MATNR(optional)
* | [--->] IV_WERKS                       TYPE        WERKS_D(optional)
* | [--->] IV_LGORT                       TYPE        LGORT_D(optional)
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* | [<---] ET_MATERIAL                    TYPE        ZEPP_TT_MATE
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD material_query.

    DATA:
      ls_filter TYPE /iwbep/s_mgw_select_option,
```

```
      ls_range  TYPE /iwbep/s_cod_select_option,
      ls_matnr  TYPE zepp_s_mate-matnr,
      ls_werks  TYPE zepp_s_mate-werks,
      ls_lgort  TYPE zepp_s_mate-lgort,
      ls_meins  TYPE mara-meins,
      lr_matnr  TYPE RANGE OF zepp_s_mate-matnr,
      lr_werks  TYPE RANGE OF zepp_s_mate-werks,
      lr_lgort  TYPE RANGE OF zepp_s_mate-lgort.

    FIELD-SYMBOLS:
      <fs_matnr> LIKE LINE OF lr_matnr,
      <fs_werks> LIKE LINE OF lr_werks,
      <fs_lgort> LIKE LINE OF lr_lgort.

    LOOP AT it_filter INTO ls_filter.

      LOOP AT ls_filter-select_options INTO ls_range.

        TRANSLATE ls_filter-property TO UPPER CASE.

        CASE ls_filter-property.
          WHEN 'MATNR'.
            APPEND INITIAL LINE TO lr_matnr ASSIGNING <fs_matnr>.
            <fs_matnr>-sign = ls_range-sign.
            <fs_matnr>-option = ls_range-option.
            ls_matnr = ls_range-low.
            CALL FUNCTION 'CONVERSION_EXIT_MATN1_INPUT'
              EXPORTING
                input  = ls_matnr
              IMPORTING
                output = <fs_matnr>-low.
            ls_matnr = ls_range-high.
            CALL FUNCTION 'CONVERSION_EXIT_MATN1_INPUT'
              EXPORTING
                input  = ls_matnr
              IMPORTING
                output = <fs_matnr>-high.
          WHEN 'WERKS'.
            APPEND INITIAL LINE TO lr_werks ASSIGNING <fs_werks>.
            <fs_werks>-sign = ls_range-sign.
            <fs_werks>-option = ls_range-option.
            ls_werks = ls_range-low.
            <fs_werks>-low = ls_werks.
            ls_werks = ls_range-high.
            <fs_werks>-high = ls_werks.
          WHEN 'LGORT'.
            APPEND INITIAL LINE TO lr_lgort ASSIGNING <fs_lgort>.
            <fs_lgort>-sign = ls_range-sign.
            <fs_lgort>-option = ls_range-option.
            ls_lgort = ls_range-low.
            <fs_lgort>-low = ls_lgort.
            ls_lgort = ls_range-high.
            <fs_lgort>-high = ls_lgort.
        ENDCASE.

      ENDLOOP.
```

```
    ENDLOOP.

    IF iv_matnr IS NOT INITIAL.
      lr_matnr = VALUE #( sign = 'I' option = 'EQ' ( low = iv_matnr ) ).
    ENDIF.

    IF iv_werks IS NOT INITIAL.
      lr_werks = VALUE #( sign = 'I' option = 'EQ' ( low = iv_werks ) ).
    ENDIF.

    IF iv_lgort IS NOT INITIAL.
      lr_lgort = VALUE #( sign = 'I' option = 'EQ' ( low = iv_lgort ) ).
    ENDIF.

    SELECT matnr, maktx, @c_maagr01 AS matkl, meins, mstae, mmsta, werks,
lgort, labst
      INTO TABLE @et_material
      FROM zepp_dd_material( matkl = @c_maagr01 )
      WHERE matnr IN @lr_matnr
        AND werks IN @lr_werks
        AND lgort IN @lr_lgort.
    IF sy-subrc <> 0.
      APPEND VALUE #( type = 'E' id = '63' number = '167' ) TO et_return.
    ENDIF.

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->MATERIAL_READ
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IT_KEY_TAB                     TYPE
/IWBEP/T_MGW_NAME_VALUE_PAIR(optional)
* | [<---] ES_MATERIAL                    TYPE        ZEPP_S_MATE
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD material_read.
  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->MOVIMIENTO_QUERY
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IT_FILTER                      TYPE
/IWBEP/T_MGW_SELECT_OPTION(optional)
* | [--->] IV_RSNUM                       TYPE        RSNUM(optional)
* | [--->] IV_RSPOS                       TYPE        RSPOS(optional)
* | [--->] IV_MBLNR                       TYPE        MBLNR(optional)
* | [--->] IV_MJAHR                       TYPE        MJAHR(optional)
* | [--->] IV_MATNR                       TYPE        MATNR(optional)
* | [--->] IV_ZEILE                       TYPE        MBLPO(optional)
* | [--->] IV_WEMPF                       TYPE        WEMPF(optional)
```

```
* | [--->] IV_BUDAT                        TYPE        BUDAT(optional)
* | [<---] ET_RETURN                       TYPE        BAPIRET2_TT
* | [<---] ET_MOVIMIENTO                   TYPE        ZEPP_TT_MOVI
* +-------------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD movimiento_query.

    DATA:
      lr_rsnum TYPE RANGE OF rsnum,
      lr_rspos TYPE RANGE OF rspos,
      lr_matnr TYPE RANGE OF matnr,
      lr_mblnr TYPE RANGE OF mblnr,
      lr_mjahr TYPE RANGE OF mjahr,
      lr_zeile TYPE RANGE OF mblpo,
      lr_budat TYPE RANGE OF budat,
      lr_wempf TYPE RANGE OF wempf.

    IF iv_rsnum IS NOT INITIAL.
      lr_rsnum = VALUE #( sign = 'I' option = 'EQ' ( low = iv_rsnum ) ).
    ENDIF.

    IF iv_rspos IS NOT INITIAL.
      lr_rspos = VALUE #( sign = 'I' option = 'EQ' ( low = iv_rspos ) ).
    ENDIF.

    IF iv_matnr IS NOT INITIAL.
      lr_matnr = VALUE #( sign = 'I' option = 'EQ' ( low = |{ iv_matnr
ALPHA = IN }| ) ).
    ENDIF.

    IF iv_mblnr IS NOT INITIAL.
      lr_mblnr = VALUE #( sign = 'I' option = 'EQ' ( low = iv_mblnr ) ).
    ENDIF.

    IF iv_mjahr IS NOT INITIAL.
      lr_mjahr = VALUE #( sign = 'I' option = 'EQ' ( low = iv_mjahr ) ).
    ENDIF.

    IF iv_zeile IS NOT INITIAL.
      lr_zeile = VALUE #( sign = 'I' option = 'EQ' ( low = iv_zeile ) ).
    ENDIF.

    IF iv_budat IS NOT INITIAL.
      lr_budat = VALUE #( sign = 'I' option = 'EQ' ( low = iv_budat ) ).
    ENDIF.

    IF iv_wempf IS NOT INITIAL.
      lr_wempf = VALUE #( sign = 'I' option = 'EQ' ( low = iv_wempf ) ).
    ENDIF.

    LOOP AT it_filter
      INTO DATA(lw_filter).

      CASE lw_filter-property.
        WHEN 'RSNUM'. lr_rsnum = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
```

```
        WHEN 'RSPOS'. lr_rspos = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
        WHEN 'MATNR'.
          lr_matnr = VALUE #(
            FOR ls_range IN lw_filter-select_options (
             sign   = ls_range-sign
             option = ls_range-option
             low    = |{ ls_range-low ALPHA = IN }|
             high   = |{ ls_range-high ALPHA = IN }|
            )
          ).
        WHEN 'MBLNR'. lr_mblnr = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
        WHEN 'MJAHR'. lr_mjahr = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
        WHEN 'ZEILE'. lr_zeile = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
        WHEN 'BUDAT'. lr_budat = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
        WHEN 'WEMPF'. lr_wempf = VALUE #( FOR ls_range IN lw_filter-
select_options ( CORRESPONDING #( ls_range ) ) ).
      ENDCASE.

    ENDLOOP.

    IF lines( lr_rsnum ) = 0
      AND lines( lr_rspos ) = 0
      AND lines( lr_matnr ) = 0
      AND lines( lr_mblnr ) = 0
      AND lines( lr_mjahr ) = 0
      AND lines( lr_zeile ) = 0
      AND lines( lr_budat ) = 0
      AND lines( lr_wempf ) = 0.
      lr_budat = VALUE #( sign = 'I' option = 'BT' ( low = sy-datum - 365
high = sy-datum ) ).
    ENDIF.

* Paso 1 - Buscamos los movimientos y nos quedamos con los números de
reservas.
    SELECT rsnum, rspos INTO TABLE @DATA(lt_reservas)
      FROM mseg
        INNER JOIN mara ON mseg~matnr EQ mara~matnr
        INNER JOIN mkpf ON mseg~mblnr EQ mkpf~mblnr AND mseg~mjahr EQ
mkpf~mjahr
      WHERE mseg~mblnr IN @lr_mblnr
        AND mseg~mjahr IN @lr_mjahr
        AND zeile      IN @lr_zeile
        AND budat      IN @lr_budat
        AND mseg~matnr IN @lr_matnr
        AND rsnum      IN @lr_rsnum
        AND wempf      IN @lr_wempf
        AND matkl      EQ @c_maagr01.
    IF sy-subrc = 0 AND lines( lt_reservas ) > 0.

      SORT lt_reservas BY rsnum rspos ASCENDING.
      DELETE ADJACENT DUPLICATES FROM lt_reservas COMPARING rsnum rspos.
```

```
* Paso 2 - Buscar las reservas que tienen movimientos
      SELECT resb~rsnum, resb~rspos, bdmng, resb~matnr, mseg~mblnr,
mseg~mjahr, zeile, mseg~shkzg, menge, budat, resb~kzear, xloek
        INTO TABLE @DATA(lt_resb)
        FROM resb
          INNER JOIN mseg ON resb~rsnum EQ mseg~rsnum AND resb~rspos EQ
mseg~rspos
          INNER JOIN mkpf ON mseg~mblnr EQ mkpf~mblnr AND mseg~mjahr EQ
mkpf~mjahr
        FOR ALL ENTRIES IN @lt_reservas
       WHERE resb~rsnum EQ @lt_reservas-rsnum
         AND resb~rspos EQ @lt_reservas-rspos.
      IF sy-subrc = 0 AND lines( lt_resb ) > 0.

        SORT lt_resb BY mblnr mjahr zeile ASCENDING.
        DELETE ADJACENT DUPLICATES FROM lt_resb COMPARING mblnr mjahr
zeile.

        LOOP AT lt_resb
          INTO DATA(lw_resb).

          APPEND INITIAL LINE TO et_movimiento
            ASSIGNING FIELD-SYMBOL(<fs_movimiento>).
          <fs_movimiento> = CORRESPONDING #( lw_resb ).
          IF lw_resb-shkzg EQ 'H'.
            <fs_movimiento>-labst = lw_resb-menge * ( -1 ).
          ELSE.
            <fs_movimiento>-labst = lw_resb-menge.
          ENDIF.

          SELECT SINGLE firma INTO @<fs_movimiento>-firma
            FROM ztmm_firma_mkpf
            WHERE mblnr EQ @lw_resb-mblnr
              AND mjahr EQ @lw_resb-mjahr.

        ENDLOOP.

      ENDIF.

    ENDIF.

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->MOVIMIENTO_READ
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IT_KEY_TAB                     TYPE
/IWBEP/T_MGW_NAME_VALUE_PAIR(optional)
* | [<---] ES_MOVIMIENTO                  TYPE        ZEPP_S_MOVI
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD movimiento_read.
```

```
    DATA:
      lv_rsnum TYPE rsnum,
      lv_matnr TYPE matnr,
      lv_mjahr TYPE mjahr,
      lv_zeile TYPE mblpo,
      lv_mblnr TYPE mblnr,
      lv_wempf TYPE wempf.

    LOOP AT it_key_tab INTO DATA(ls_key).
      CASE ls_key-name.
        WHEN 'NumeroReserva'.
          lv_rsnum = ls_key-value.
        WHEN 'NumeroDocumentoMaterial'.
          lv_mblnr = ls_key-value.
        WHEN 'EjercicioDocumentoMaterial'.
          lv_mjahr = ls_key-value.
        WHEN 'PosicionDocumentoMaterial'.
          lv_zeile = ls_key-value.
        WHEN 'Destinatario'.
          lv_wempf = ls_key-value.
      ENDCASE.
    ENDLOOP.

    CALL METHOD movimiento_query
      EXPORTING
        iv_rsnum      = lv_rsnum
        iv_mblnr      = lv_mblnr
        iv_mjahr      = lv_mjahr
        iv_zeile      = lv_zeile
        iv_wempf      = lv_wempf
      IMPORTING
        et_return     = et_return
        et_movimiento = DATA(lt_movimiento).

    READ TABLE lt_movimiento INTO es_movimiento INDEX 1.

  ENDMETHOD.


* <SIGNATURE>-----------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->NUEVARESERVA_CREATE
* +-----------------------------------------------------------------------
---------------------------+
* | [<---] EP_RSNUM                       TYPE        RSNUM
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* | [<-->] IT_NUEVARESERVA                TYPE
ZEPP_TT_NEWR(optional)
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD nuevareserva_create.

    DATA:
      lv_rsnum  TYPE resb-rsnum,
      lv_index  TYPE sy-tabix,
      lw_header TYPE bapirkpfc,
      lt_items  TYPE TABLE OF bapiresbc,
```

```
      lt_return TYPE TABLE OF bapireturn.

    CLEAR:
      lw_header,
      lt_items[],
      lt_return[].

    IF lines( it_nuevareserva ) > 0.

      DATA(ls_line1) = it_nuevareserva[ 1 ].

      lw_header-plant        = ls_line1-werks.
      lw_header-move_stloc   = ls_line1-lgort.
      lw_header-cost_ctr     = ls_line1-kostl.
      lw_header-move_type    = '201'.
      lw_header-gr_rcpt      = ls_line1-wempf.
      lw_header-created_by   = sy-uname.
      lw_header-res_date     = sy-datum.

      lt_items = VALUE #(
        FOR ls_item IN it_nuevareserva (
          material   = |{ ls_item-matnr ALPHA = IN }|
          short_text = ls_item-maktx
          plant      = ls_item-werks
          quantity   = ls_item-bdmng
          unit       = ls_item-meins
          store_loc  = ls_item-lgort
          req_date   = ls_item-bdter
          gr_rcpt    = ls_item-wempf
          movement   = abap_true
        )
      ).

      CALL FUNCTION 'BAPI_RESERVATION_CREATE'
        EXPORTING
          reservation_header = lw_header
        IMPORTING
          reservation        = ep_rsnum
        TABLES
          reservation_items  = lt_items
          return             = lt_return.

      IF ep_rsnum IS NOT INITIAL.

        DO 100 TIMES.

          SELECT rsnum, rspos, resb~matnr, maktx, werks, lgort, bdmng,
meins
            INTO TABLE @DATA(lt_resb)
            FROM resb INNER JOIN makt ON resb~matnr EQ makt~matnr
            WHERE rsnum EQ @ep_rsnum
            AND spras EQ @sy-langu.

          IF sy-subrc = 0 AND lines( lt_resb ) = lines( it_nuevareserva
).

            lv_index = 0.
```

```
            DATA(lt_nuevareserva) = it_nuevareserva.

            LOOP AT it_nuevareserva
              ASSIGNING FIELD-SYMBOL(<fs_nuevareserva>).

              ADD 1 TO lv_index.

              READ TABLE lt_nuevareserva INTO DATA(lw_nuevareserva) INDEX
lv_index.
              IF sy-subrc = 0.
                MOVE-CORRESPONDING lw_nuevareserva TO <fs_nuevareserva>.
              ENDIF.

              READ TABLE lt_resb INTO DATA(lw_resb) INDEX lv_index.
              IF sy-subrc = 0.
                MOVE-CORRESPONDING lw_resb TO <fs_nuevareserva>.
              ENDIF.

            ENDLOOP.

            EXIT.

          ENDIF.

          WAIT UP TO 5 SECONDS.

        ENDDO.

      ELSE.

        ep_rsnum = 0.

        et_return = VALUE #(
          FOR ls_return IN lt_return (
            type       = ls_return-type
            id         = ls_return-code
            log_no     = ls_return-log_no
            number     = ls_return-log_no
            message_v1 = ls_return-message_v1
            message_v2 = ls_return-message_v2
            message_v3 = ls_return-message_v3
            message_v4 = ls_return-message_v4
            message    = ls_return-message
          )
        ).

        et_return = VALUE #( ( type = 'E' id = '/ISDFPS/MM' number =
'142' ) ).

      ENDIF.

    ELSE.

      ep_rsnum = 0.

      et_return = VALUE #( ( type = 'E' id = 'W5' number = '136' ) ).
```

```
    ENDIF.

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->NUEVARESERVA_READ
* +-----------------------------------------------------------------------
---------------------------+
* | [<---] ES_NUEVARESERVA                TYPE        ZEPP_S_NEWR
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  method NUEVARESERVA_READ.
  endmethod.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->RESERVA_QUERY
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IV_MATNR                       TYPE        MATNR(optional)
* | [--->] IV_BUDAT                       TYPE        BUDAT(optional)
* | [--->] IV_WEMPF                       TYPE        WEMPF(optional)
* | [--->] IV_RSNUM                       TYPE        RSNUM(optional)
* | [--->] IV_RSPOS                       TYPE        RSPOS(optional)
* | [--->] IT_FILTER                      TYPE
/IWBEP/T_MGW_SELECT_OPTION(optional)
* | [<---] ET_RESERVA                     TYPE        ZEPP_TT_RESE
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD reserva_query.

    DATA:
      lt_filter TYPE       /iwbep/t_mgw_select_option,

      lr_rsnum  TYPE RANGE OF rsnum,
      rw_rsnum  LIKE LINE OF  lr_rsnum,

      lr_rspos  TYPE RANGE OF rspos,
      rw_rspos  LIKE LINE OF  lr_rspos,

      lr_matnr  TYPE RANGE OF matnr,
      rw_matnr  LIKE LINE OF  lr_matnr,

      lr_budat  TYPE RANGE OF budat,
      rw_budat  LIKE LINE OF  lr_budat,

      lr_wempf  TYPE RANGE OF wempf,
      rw_wempf  LIKE LINE OF  lr_wempf.


    lt_filter[] = it_filter[].
```

```
    SORT lt_filter BY property.
    LOOP AT lt_filter ASSIGNING FIELD-SYMBOL(<fs_filters>) WHERE property
EQ 'RSNUM'.
      LOOP AT <fs_filters>-select_options INTO DATA(lw_seloptn).
        MOVE-CORRESPONDING lw_seloptn TO rw_rsnum.
        APPEND rw_rsnum TO lr_rsnum.
        CLEAR: rw_rsnum, lw_seloptn.
      ENDLOOP.
    ENDLOOP.

    LOOP AT it_filter ASSIGNING <fs_filters> WHERE property EQ 'RSPOS'.
      LOOP AT <fs_filters>-select_options INTO lw_seloptn.
        MOVE-CORRESPONDING lw_seloptn TO rw_rspos.
        APPEND rw_rspos TO lr_rspos.
        CLEAR: rw_rspos, lw_seloptn.
      ENDLOOP.
    ENDLOOP.


    LOOP AT it_filter ASSIGNING <fs_filters> WHERE property EQ 'MATNR'.
      LOOP AT <fs_filters>-select_options INTO lw_seloptn.
        MOVE-CORRESPONDING lw_seloptn TO rw_matnr.
        CALL FUNCTION 'CONVERSION_EXIT_MATN1_INPUT'
          EXPORTING
            input  = rw_matnr-low
          IMPORTING
            output = rw_matnr-low.
        CALL FUNCTION 'CONVERSION_EXIT_MATN1_INPUT'
          EXPORTING
            input  = rw_matnr-high
          IMPORTING
            output = rw_matnr-high.
        APPEND rw_matnr TO lr_matnr.
        CLEAR: rw_matnr, lw_seloptn.
      ENDLOOP.
    ENDLOOP.

    LOOP AT it_filter ASSIGNING <fs_filters> WHERE property EQ 'BUDAT'.
      LOOP AT <fs_filters>-select_options INTO lw_seloptn.
        MOVE-CORRESPONDING lw_seloptn TO rw_budat.
        APPEND rw_budat TO lr_budat.
        CLEAR: rw_budat, lw_seloptn.
      ENDLOOP.
    ENDLOOP.

    LOOP AT it_filter ASSIGNING <fs_filters> WHERE property EQ 'WEMPF'.
      LOOP AT <fs_filters>-select_options INTO lw_seloptn.
        MOVE-CORRESPONDING lw_seloptn TO rw_wempf.
        APPEND rw_wempf TO lr_wempf.
        CLEAR: rw_wempf, lw_seloptn.
      ENDLOOP.
    ENDLOOP.


    IF iv_rsnum IS NOT INITIAL.
      lr_rsnum = VALUE #( sign = 'I' option = 'EQ' ( low = iv_rsnum ) ).
    ENDIF.
```

```
    IF iv_rspos IS NOT INITIAL.
      lr_rspos = VALUE #( sign = 'I' option = 'EQ' ( low = iv_rspos ) ).
    ENDIF.

    IF iv_matnr IS NOT INITIAL.
      lr_matnr = VALUE #( sign = 'I' option = 'EQ' ( low = iv_matnr ) ).
    ENDIF.

    IF iv_budat IS NOT INITIAL.
      lr_budat = VALUE #( sign = 'I' option = 'EQ' ( low = iv_budat ) ).
    ENDIF.

    IF iv_wempf IS NOT INITIAL.
      lr_wempf = VALUE #( sign = 'I' option = 'EQ' ( low = iv_wempf ) ).
    ENDIF.

    SELECT rb~rsnum,  "Reserva
           rb~rspos,  "Posición
           rb~bdmng,  "Cantidad necesaria
           rb~meins,  "Unidad de medida base
           rb~matnr,  "Número de material
           ms~mblnr,  "Número de documento de material
           ms~mjahr,  "Ejercicio del documento de material
           ms~zeile,  "Posición en documento de material
           mk~budat,  "Fecha de contabilización en el documento/Fecha
contable del movimiento
           md~labst,  "Cantidad del movimiento(con signo positivo o
negativo según el caso)
           fm~firma,  "Firma
           rb~kzear,  "Salida final de la reserva
           rb~xloek,  "Posición borrada
           rb~wempf   "Destinatario
      INTO TABLE @DATA(lt_resb) UP TO 100 ROWS
      FROM resb AS rb "Detalle reservas
     INNER JOIN rkpf AS rk "Cabecera reservas
        ON rb~rsnum EQ rk~rsnum
      LEFT OUTER JOIN mard AS md "Almacén material
        ON rb~matnr EQ md~matnr
       AND rb~werks EQ md~werks
       AND rb~lgort EQ md~lgort
      LEFT OUTER JOIN mseg AS ms "Seg. documento material
        ON rb~rsnum EQ ms~rsnum
       AND rb~rspos EQ ms~rspos
      LEFT OUTER JOIN mkpf AS mk "Cabecera de doc.material
        ON ms~mblnr EQ mk~mblnr
       AND ms~mjahr EQ mk~mjahr
      LEFT OUTER JOIN ztmm_firma_mkpf AS fm "Firma Conformidad Despacho
        ON ms~mblnr EQ fm~mblnr
       AND ms~mjahr EQ fm~mjahr
     WHERE rb~rsnum IN @lr_rsnum
       AND rb~rspos IN @lr_rspos
       AND rb~wempf IN @lr_wempf
       AND md~matnr IN @lr_matnr
       AND mk~budat IN @lr_budat.
    IF sy-subrc NE 0.
```

```
      APPEND VALUE #( type = 'E' id = '63' number = '167' ) TO et_return.
    ENDIF.

    LOOP AT lt_resb INTO DATA(ls_resb).
      APPEND INITIAL LINE TO et_reserva
        ASSIGNING FIELD-SYMBOL(<fs_reserva>).
      <fs_reserva> = CORRESPONDING #( ls_resb ).
    ENDLOOP.

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Public Method ZEPP_SERVICES->RESERVA_READ
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IT_KEY_TAB                     TYPE
/IWBEP/T_MGW_NAME_VALUE_PAIR(optional)
* | [<---] ES_RESERVA                     TYPE        ZEPP_S_RESE
* | [<---] ET_RETURN                      TYPE        BAPIRET2_TT
* +-----------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD reserva_read.

    DATA:
      lv_rsnum TYPE rsnum,
      lv_matnr TYPE matnr,
      lv_budat TYPE budat,
      lv_wempf TYPE wempf.

    LOOP AT it_key_tab INTO DATA(ls_key).
      CASE ls_key-name.
        WHEN 'NumeroReserva'.
          lv_rsnum = ls_key-value.
        WHEN 'CodigoMaterial'.
          lv_matnr = ls_key-value.
          CALL FUNCTION 'CONVERSION_EXIT_MATN1_INPUT'
            EXPORTING
              input  = lv_matnr
            IMPORTING
              output = lv_matnr.
        WHEN 'FechaContable'.
          lv_budat = ls_key-value.
        WHEN 'Destinatario'.
          lv_wempf = ls_key-value.
      ENDCASE.
    ENDLOOP.

    CALL METHOD reserva_query
      EXPORTING
        iv_rsnum   = lv_rsnum
        iv_matnr   = lv_matnr
        iv_budat   = lv_budat
        iv_wempf   = lv_wempf
      IMPORTING
        et_return  = et_return
```

```
        et_reserva = DATA(lt_reserva).

    READ TABLE lt_reserva INTO es_reserva INDEX 1.

  ENDMETHOD.
ENDCLASS.
```

4.6.2 Función ZRFC_INF_TRABAJADORES2

```
FUNCTION zrfc_inf_trabajadores2.

*"----------------------------------------------------------------------

*"*"Interfase local

*"  IMPORTING

*"     VALUE(LV_TIPO_TRAB) TYPE  ZE_TIPOE

*"     VALUE(LV_SOCIEDAD) TYPE  BUKRS

*"     VALUE(LV_FECHA) TYPE  DATUM

*"  TABLES

*"      IT_EMPLEADOS TYPE  ZTES_EMPLEADOS2

*"      IT_OBREROS TYPE  ZTES_OBREROS2

*"      IT_PERNR TYPE  ZTT_TRAB OPTIONAL

*"----------------------------------------------------------------------


  TYPES: BEGIN OF t_personal,

           pernr     LIKE   pa0000-pernr,

           endda     LIKE   pa0000-endda,

           begda     LIKE   pa0000-begda,

           massn     LIKE   pa0000-massn,

           bukrs     TYPE   pa0001-bukrs,

           persk     LIKE   pa0001-persk,

           abkrs     LIKE   pa0001-abkrs,
```

```
            ansvh      LIKE   pa0001-ansvh,
            plans      LIKE   pa0001-plans,
            orgeh      LIKE   pa0001-orgeh,
            uni_neg    LIKE pa0001-uni_neg,
            posicion   LIKE pa0001-posicion,
            clasifica LIKE pa0001-clasifica,
            division   LIKE pa0001-division,
*{
            kostl      TYPE pa0001-kostl,
            departam   TYPE pa0001-departam,
*}
          END OF t_personal,
          BEGIN OF t_sup,
            pernr LIKE pa0000-pernr,
            sup   LIKE pa0000-pernr,
          END OF t_sup,

          BEGIN OF t_telef,
            pernr        LIKE pa0000-pernr,
            telefono(80) TYPE c,
          END OF  t_telef,
          BEGIN OF t_correos,
            pernr        LIKE pa0000-pernr,
            correop(200) TYPE c,
```

```
          correol(200) TYPE c,
        END OF  t_correos.

  TYPES: BEGIN OF t_zona.
          INCLUDE STRUCTURE zthcm_tipo_zona.
        TYPES: END OF t_zona.

  TYPES: BEGIN OF t_t5evp.
          INCLUDE STRUCTURE t5evp.
        TYPES: END OF t_t5evp.

  TYPES: BEGIN OF t_region.
          INCLUDE STRUCTURE zthcm_region.
        TYPES: END OF t_region.

  TYPES: BEGIN OF t_prov.
          INCLUDE STRUCTURE zthcm_provincia.
        TYPES: END OF t_prov.

  DATA: g_1(80)      TYPE c,
        g_2(50)      TYPE c,
        g_3(50)      TYPE c,
        g_4(50)      TYPE c,
        g_5(50)      TYPE c,
```

```
        g_6(50)      TYPE c,
        g_7(50)      TYPE c,
        g_8(50)      TYPE c,
        g_9(50)      TYPE c,
        g_10(200)    TYPE c,
        g_11(200)    TYPE c,
        g_12(200)    TYPE c,
        g_13(200)    TYPE c,
        g_14(200)    TYPE c,
        lv_url(200) TYPE c,
        lv_cont      TYPE i.

DATA: s_per      TYPE t_personal,
      s_per_c    TYPE t_personal,
      s_supo     TYPE t_sup,
      s_telef    TYPE t_telef,
      s_correo   TYPE t_correos,
      st_trab    TYPE zes_inf_empleados2,
      st_trab_o TYPE zes_inf_obreros2,
      s_ab_hr    TYPE ztab_const,
      s_const    TYPE ztab_const,
      st_p0002   TYPE pa0002,
      st_p0006   TYPE pa0006,
      st_p0185   TYPE pa0185,
```

```
        st_p9003  TYPE pa9003,
        st_p0016  TYPE pa0016,
        st_p9015  TYPE pa9015,
        st_p2001  TYPE pa2001,
        st_p0041  TYPE pa0041,
        st_t005t  TYPE t005t,
        st_civil  TYPE ztab_const,
        st_t5r06  TYPE t5r06,
        s_t5evp   TYPE t_t5evp,
        s_zona    TYPE t_zona,
        w_region  TYPE t_region,
        w_prov    TYPE t_prov,
        st_t547s  TYPE t547s,
        st_afp    TYPE zthcm_cabpens,
        st_nivel  TYPE ztecp_niveles_j,
        st_dir    TYPE t880,
        w_unidad  TYPE hrp1000,
        st_nom    TYPE t549t,
        st_cese   TYPE ztecp_cesados,
        st_divi   TYPE ztecp_divisiones,
        st_tipo   TYPE ztab_const.


  DATA: it_per      TYPE STANDARD TABLE OF t_personal,
        it_per_c    TYPE STANDARD TABLE OF t_personal,
```

```
        it_supo      TYPE STANDARD TABLE OF t_sup,
        it_telef     TYPE STANDARD TABLE OF t_telef,
        it_correo    TYPE STANDARD TABLE OF t_correos,
        it_trab      TYPE STANDARD TABLE OF zes_inf_empleados2,
        it_p0002     TYPE TABLE OF pa0002 WITH HEADER LINE,
        it_p0006     TYPE TABLE OF pa0006 WITH HEADER LINE,
        it_p0041     TYPE TABLE OF pa0041 WITH HEADER LINE,
        it_p2001     TYPE TABLE OF pa2001 WITH HEADER LINE,
        it_p0185     TYPE TABLE OF pa0185 WITH HEADER LINE,
        it_p0016     TYPE TABLE OF pa0016 WITH HEADER LINE,
        it_p9015     TYPE TABLE OF pa9015 WITH HEADER LINE,
        it_p9015_e   TYPE TABLE OF pa9015 WITH HEADER LINE,
        it_p9003     TYPE TABLE OF pa9003 WITH HEADER LINE,
        it_ab_hr     TYPE STANDARD TABLE OF ztab_const,
        it_const     TYPE STANDARD TABLE OF ztab_const,
        it_civil     TYPE STANDARD TABLE OF ztab_const,
        it_tipo      TYPE STANDARD TABLE OF ztab_const,
        it_t005t     TYPE STANDARD TABLE OF t005t,
        it_t5r06     TYPE STANDARD TABLE OF t5r06,
        it_zona      LIKE STANDARD TABLE OF s_zona,
        it_t5evp     LIKE STANDARD TABLE OF s_t5evp,
        it_region    LIKE STANDARD TABLE OF w_region,
        it_prov      LIKE STANDARD TABLE OF w_prov,
        it_t547s     TYPE STANDARD TABLE OF t547s,
```

```
        it_afp      TYPE STANDARD TABLE OF zthcm_cabpens,
        it_nivel    TYPE STANDARD TABLE OF ztecp_niveles_j,
        it_unidades TYPE STANDARD TABLE OF hrp1000,
        it_divi     TYPE STANDARD TABLE OF ztecp_divisiones,
        it_nom      TYPE STANDARD TABLE OF t549t.
  " it_sup      TYPE STANDARD TABLE OF ztecp_sup_sub.

  DATA: lv_date      TYPE sy-datum,          " Fecha (AAAAMMDD)
        lv_time      TYPE sy-uzeit,          " Hora (HHMMSS)
        lv_timestamp TYPE string,             " Marca de tiempo en formato
ISO 8601
        lv_year      TYPE string,
        lv_month     TYPE string,
        lv_day       TYPE string,
        lv_hour      TYPE string,
        lv_minute    TYPE string,
        lv_second    TYPE string.

  DATA: v_desc TYPE char40.
  DATA: lo_http_client TYPE REF TO if_http_client.
  DATA: response TYPE string.
  DATA handler TYPE REF TO  if_hrpa_message_handler.
  DATA i_edad TYPE num2.
  DATA: lg_comilla(5)  TYPE c,
        lv_pos         TYPE i,
```

```
        lv_correo(150) TYPE c.


DATA: url_ta       TYPE string,
      url_ta_p     TYPE string,
      url_tad      TYPE string,
      url_tad_p    TYPE string,
      url_tad_get TYPE string,
      aut          TYPE string.


DATA: lr_data  TYPE REF TO data,
      lr_data2 TYPE REF TO data.


DATA: w_xstring TYPE xstring,
      w_string  TYPE string,
      w_lenght  TYPE i,
      w_strlen  TYPE string.


DATA: ti_smum   TYPE STANDARD TABLE OF smum_xmltb,
      ti_return TYPE bapiret2 OCCURS 0.
DATA: lv_perso(6)  TYPE c,
      lv_cargo(20) TYPE c,
      g_stext      TYPE hrp1000-stext,
      v_var1(2)    TYPE c,
      v_var2       TYPE c,
```

```
        lv_soc(6)     TYPE c,
        lv_fe_ini     TYPE datum,
        lv_line       TYPE i.
  DATA: lv_fec_cese TYPE sy-datum,
        lv_ingreso  TYPE datum,
        lv_tipo     TYPE c,
        lv_per1(6)  TYPE c,
        lv_per2(6)  TYPE c,
        lv_per3(6)  TYPE c,
        lv_per4(6)  TYPE c,
        lv_pernr    TYPE persno.


  FIELD-SYMBOLS:
    <data>        TYPE data,
    <data2>       TYPE data,
    <results>     TYPE any,
    <d>           TYPE any,
    <results2>    TYPE any,
    <structure>   TYPE any,
    <structure2>  TYPE any,
    <table>       TYPE ANY TABLE,
    <field>       TYPE any,
    <field_value> TYPE data,
```

```
    <fs_const>    LIKE LINE OF it_const.


  DATA: s_perso   TYPE zes_traba.

  RANGES:  r_sub     FOR  p2001-awart,

           r_tipo    FOR  p0001-persk,

           r_perso   FOR  p0000-pernr.




  SORT it_pernr ASCENDING BY pernr.

  IF it_pernr IS NOT INITIAL.

    REFRESH r_perso.


    LOOP AT it_pernr INTO s_perso.

      r_perso-sign   = 'I'.

      r_perso-option = 'EQ'.

      r_perso-low    = s_perso-pernr.

      r_perso-high   = ''.

      APPEND r_perso.

    ENDLOOP.


  ENDIF.


  CLEAR lv_tipo.

  REFRESH: r_tipo, it_tipo[].
```

```
  IF lv_tipo_trab EQ 'Q'.
    SELECT * INTO TABLE it_tipo
      FROM ztab_const
      WHERE modulo   EQ 'HCM'
      AND const01   EQ 'AREAP'
      AND const02   EQ 'EMP'
      AND obj_abap EQ 'ZRFC_INF_TRABAJADORES'.
    lv_tipo = 'X'.
  ELSEIF lv_tipo_trab EQ 'O'.
    SELECT * INTO TABLE it_tipo
    FROM ztab_const
    WHERE modulo   EQ 'HCM'
    AND const01   EQ 'AREAP'
    AND const02   EQ 'OBR'
    AND obj_abap EQ 'ZRFC_INF_TRABAJADORES'.
    lv_tipo = 'X'.
  ENDIF.


  LOOP AT it_tipo INTO st_tipo.
    r_tipo-sign   = 'I'.
    r_tipo-option = 'EQ'.
    r_tipo-low    = st_tipo-zlow.
    r_tipo-high   = ''.
    APPEND r_tipo.
```

```
  ENDLOOP.


  IF lv_tipo EQ 'X'.


    REFRESH: it_ab_hr[].

    SELECT * INTO TABLE it_ab_hr

      FROM ztab_const

      WHERE modulo   EQ 'HCM'

      AND const01   EQ 'HORAS'

      AND obj_abap EQ 'ZRFC_INT_PIMS_022C'.


    REFRESH r_sub.


    LOOP AT it_ab_hr INTO s_ab_hr.

      r_sub-sign   = 'I'.

      r_sub-option = 'EQ'.

      r_sub-low    = s_ab_hr-zlow.

      r_sub-high   = ''.

      APPEND r_sub.

    ENDLOOP.


    SELECT * INTO CORRESPONDING FIELDS OF TABLE it_const

     FROM ztab_const

     WHERE modulo   EQ 'HCM'
```

```
      AND const01  EQ sy-sysid
      AND obj_abap EQ 'ZREHCM_NEW_BALLOT'.


    IF sy-subrc EQ 0.
      LOOP AT it_const ASSIGNING <fs_const>.
        CASE <fs_const>-const02.
          WHEN 'AUTHORIZATION'.
            CONCATENATE <fs_const>-zlow <fs_const>-zhigh INTO aut.
          WHEN 'URL'.
            lv_url = <fs_const>-zlow.
        ENDCASE.
      ENDLOOP.
    ENDIF.


    SELECT * INTO TABLE it_t005t
      FROM t005t
      WHERE spras EQ 'S'.


    SELECT * INTO TABLE it_civil
     FROM ztab_const
   WHERE modulo   EQ 'HCM'
     AND const01  EQ 'ESTADO_CIVIL'
     AND obj_abap EQ 'ZRFC_INF_TRABAJADORES'.
```

```
    SELECT * INTO TABLE it_t5r06
      FROM   t5r06
      WHERE sprsl EQ 'S'
      AND molga EQ '99'.

    "* TIPO DE VIA
    REFRESH it_t5evp[].
    SELECT * INTO TABLE it_t5evp
      FROM t5evp.

* TIPO DE ZONA
    REFRESH it_zona[].
    SELECT * INTO TABLE it_zona
      FROM zthcm_tipo_zona.

    SELECT * INTO TABLE it_t547s
      FROM t547s
      WHERE sprsl EQ 'S'.

    SELECT * INTO TABLE it_afp
      FROM zthcm_cabpens.

    SELECT * INTO TABLE it_nivel
```

```
      FROM ztecp_niveles_j.

    CONCATENATE '00' lv_sociedad INTO lv_soc.

    SELECT SINGLE * INTO st_dir
      FROM t880
      WHERE rcomp EQ lv_soc.

    SELECT * INTO TABLE it_unidades  FROM hrp1000 WHERE
    plvar = '01' AND otype = 'O' AND langu = 'S' AND
     begda LE lv_fecha AND endda GE lv_fecha.

    SELECT * INTO TABLE it_nom
      FROM t549t
      WHERE sprsl EQ 'S'.

    REFRESH it_divi[].
    SELECT * INTO TABLE it_divi
      FROM ztecp_divisiones.

    REFRESH: it_empleados[], it_obreros[].
    CONCATENATE lv_fecha+0(4) '0101' INTO lv_fe_ini.
```

```
    "--------Lectura de trabajadores activos a la fecha de consulta------
--


    SELECT DISTINCT a~pernr a~endda a~begda a~massn b~bukrs b~persk b~abkrs
b~ansvh b~plans b~orgeh b~uni_neg b~posicion
                    b~clasifica b~division b~kostl b~departam
        INTO CORRESPONDING FIELDS OF TABLE it_per
        FROM pa0000 AS a INNER JOIN pa0001 AS b ON a~pernr = b~pernr
        WHERE a~massn NE 'Z3'
        AND a~begda LE lv_fecha
        AND a~endda GE lv_fecha
        AND b~persk  IN r_tipo
        AND a~pernr IN r_perso
        AND b~uni_neg NE space.


    SORT it_per DESCENDING BY  pernr endda.
    DELETE ADJACENT DUPLICATES FROM it_per COMPARING pernr endda.




*{

    SELECT kostl, ltext INTO TABLE @DATA(lt_cskt)
      FROM cskt FOR ALL ENTRIES IN @it_per
      WHERE kostl EQ @it_per-kostl.


    SELECT departam, descrip INTO TABLE @DATA(lt_departam)
```

```
      FROM ztecp_departam FOR ALL ENTRIES IN @it_per

      WHERE departam EQ @it_per-departam.

*}


    "-------------lectura de cesados desde el 01.01 año actual hasta la
fecha de consulta------------excluyendo los que ya se han enviado

    SELECT DISTINCT a~pernr a~endda a~begda a~massn b~bukrs b~persk b~abkrs
b~ansvh b~plans b~orgeh b~uni_neg b~posicion

                    b~clasifica b~division

     INTO CORRESPONDING FIELDS OF TABLE it_per_c

     FROM pa0000 AS a INNER JOIN pa0001 AS b ON a~pernr = b~pernr

     WHERE NOT EXISTS (

      SELECT * FROM ztecp_cesados AS ex

      WHERE ex~pernr = a~pernr

        AND ex~fe_cese = a~begda )

     AND a~massn EQ 'Z3'

     AND a~begda LE lv_fecha

     AND a~endda GE lv_fe_ini

      AND b~persk  IN r_tipo

      AND a~pernr IN r_perso

       AND b~uni_neg NE space.

    IF sy-subrc EQ 0.

      SORT it_per_c DESCENDING BY  pernr endda.

      DELETE ADJACENT DUPLICATES FROM it_per_c COMPARING pernr endda.
```

```
      LOOP AT it_per_c INTO s_per_c.

        READ TABLE it_per INTO s_per WITH KEY pernr = s_per_c-pernr.
        IF sy-subrc NE 0.
          INSERT s_per_c INTO TABLE it_per.
        ENDIF.
      ENDLOOP.
    ENDIF.

    DESCRIBE TABLE it_per LINES lv_line.
    IF  lv_line GT 0.

      REFRESH it_p0002[].
      SELECT * INTO TABLE it_p0002
        FROM pa0002
         FOR ALL ENTRIES IN it_per
       WHERE pernr EQ it_per-pernr.

      SORT it_p0002 DESCENDING BY pernr endda.
      DELETE ADJACENT DUPLICATES FROM it_p0002 COMPARING pernr.

      REFRESH it_p0041[].
      SELECT * INTO TABLE it_p0041
```

```
        FROM pa0041
         FOR ALL ENTRIES IN it_per
       WHERE pernr EQ it_per-pernr.

      SORT it_p0041 DESCENDING BY pernr endda.
      DELETE ADJACENT DUPLICATES FROM it_p0041 COMPARING pernr.

      REFRESH it_p0006[].
      SELECT * INTO TABLE it_p0006
        FROM pa0006
         FOR ALL ENTRIES IN it_per
       WHERE pernr EQ it_per-pernr.

      SORT it_p0006 DESCENDING BY pernr endda.
      DELETE ADJACENT DUPLICATES FROM it_p0006 COMPARING pernr.

      REFRESH it_p0185[].
      SELECT * INTO TABLE it_p0185
        FROM pa0185
         FOR ALL ENTRIES IN it_per
       WHERE pernr EQ it_per-pernr.

      SORT it_p0185 DESCENDING BY pernr endda.
      DELETE ADJACENT DUPLICATES FROM it_p0185 COMPARING pernr.
```

```
REFRESH it_p0016[].
SELECT * INTO TABLE it_p0016
  FROM pa0016
   FOR ALL ENTRIES IN it_per
 WHERE pernr EQ it_per-pernr.

SORT it_p0016 DESCENDING BY pernr endda.
DELETE ADJACENT DUPLICATES FROM it_p0016 COMPARING pernr.


REFRESH it_p9003[].
SELECT * INTO TABLE it_p9003
  FROM pa9003
   FOR ALL ENTRIES IN it_per
 WHERE pernr EQ it_per-pernr.

SORT it_p9003 DESCENDING BY pernr endda.
DELETE ADJACENT DUPLICATES FROM it_p9003 COMPARING pernr.

REFRESH it_p9015[].
SELECT * INTO TABLE it_p9015
  FROM pa9015
   FOR ALL ENTRIES IN it_per
```

```
      WHERE pernr EQ it_per-pernr
       AND zvsart EQ '9002'
       AND zzcompa NE space.

    SORT it_p9015 DESCENDING BY pernr endda.
    DELETE ADJACENT DUPLICATES FROM it_p9015 COMPARING pernr.

    REFRESH it_p9015_e[].
    SELECT * INTO TABLE it_p9015_e
      FROM pa9015
       FOR ALL ENTRIES IN it_per
     WHERE pernr EQ it_per-pernr
      AND zvsart EQ '9001'.

    SORT it_p9015_e DESCENDING BY pernr endda.
    DELETE ADJACENT DUPLICATES FROM it_p9015_e COMPARING pernr.

    REFRESH it_p2001[].
    SELECT * INTO TABLE it_p2001
      FROM pa2001
       FOR ALL ENTRIES IN it_per
      WHERE pernr EQ it_per-pernr
       AND begda LE lv_fecha
       AND endda GE lv_fecha
```

```
        AND awart NE r_sub.



      lv_per1 = '000001'.

      lv_per2 = '999999'.

      SORT  it_per ASCENDING BY pernr.

      "-------------------------Consulta ODATA ----------------------

      "----------------ODATA SUPERVISORES --------------------------

      CLEAR lv_perso.

      lv_perso = lv_per1.

      lg_comilla = ''''.

      lg_comilla = lg_comilla(1).

      CLEAR: response, url_ta_p.


      CONCATENATE lv_url '/odata/v2/EmpJob' INTO url_ta.


      url_ta_p = url_ta.

      CONCATENATE url_ta_p '?$filter= userId ge' INTO url_ta_p.

      CONCATENATE url_ta_p lg_comilla INTO url_ta_p SEPARATED BY space.

      CONCATENATE url_ta_p lv_per1 lg_comilla INTO url_ta_p.

      CONCATENATE  url_ta_p  'and  userId  le'  INTO  url_ta_p  SEPARATED  BY
space.

      CONCATENATE url_ta_p lg_comilla INTO url_ta_p SEPARATED BY space.

      CONCATENATE url_ta_p lv_per2 lg_comilla INTO url_ta_p.
```

```
      CONCATENATE                                                    url_ta_p
'&$select=userId,managerId&$format=JSON&$orderby=userId     asc&$top=1000'
INTO url_ta_p.




**"create HTTP client by url

*  "API endpoint for API sandbox

      CALL METHOD cl_http_client=>create_by_url

        EXPORTING

          url                 = url_ta_p

        IMPORTING

          client              = lo_http_client

        EXCEPTIONS

          argument_not_found = 1

          plugin_not_active  = 2

          internal_error     = 3

          OTHERS              = 4.


      IF sy-subrc <> 0.

        "error handling

      ENDIF.




      "setting request method

      lo_http_client->request->set_method('GET').
```

```
      "adding headers
      lo_http_client->request->set_header_field(  name  =  'Content-Type'
value = 'application/json' ).
      lo_http_client->request->set_header_field( name = 'Accept' value =
'application/json' ).


      "Basic  Auth  :  provide  username:password  in  Base64  encoded  in
Authorization header
      lo_http_client->request->set_header_field(  name  =  'Authorization'
value = aut ).


      CALL METHOD lo_http_client->send
        EXCEPTIONS
          http_communication_failure = 1
          http_invalid_state         = 2
          http_processing_failed     = 3
          http_invalid_timeout       = 4
          OTHERS                     = 5.


      IF sy-subrc = 0.
        CALL METHOD lo_http_client->receive
          EXCEPTIONS
            http_communication_failure = 1
            http_invalid_state         = 2
            http_processing_failed     = 3
```

```
            OTHERS                      = 5.
      ENDIF.

      IF sy-subrc <> 0.
        "error handling
      ENDIF.

      response = lo_http_client->response->get_cdata( ).

      CLEAR: lr_data.

      CALL METHOD /ui2/cl_json=>deserialize
        EXPORTING
          json         = response
          pretty_name  = /ui2/cl_json=>pretty_mode-user
          assoc_arrays = abap_true
        CHANGING
          data         = lr_data.

      IF lr_data IS BOUND.
        UNASSIGN: <data>, <d>, <data2>, <results>, <table>, <structure>.
        ASSIGN lr_data->* TO <data>.
        ASSIGN COMPONENT 'D' OF STRUCTURE <data> TO <d>.
        MOVE <d> TO lr_data2.
```

```
      ASSIGN lr_data2->* TO <data2>.
      ASSIGN COMPONENT 'RESULTS' OF STRUCTURE <data2> TO <results>.
      ASSIGN <results>->* TO <table>.
    ENDIF.

    LOOP AT <table> ASSIGNING <structure>.
      ASSIGN <structure>->* TO <data>.



      ASSIGN COMPONENT 'USERID' OF STRUCTURE <data> TO <field>.
      IF <field> IS ASSIGNED.
        lr_data = <field>.
        ASSIGN lr_data->* TO <field_value>.
        s_supo-pernr = <field_value>.

      ENDIF.

      ASSIGN COMPONENT 'MANAGERID' OF STRUCTURE <data> TO <field>.
      IF <field> IS ASSIGNED.
        lr_data = <field>.
        ASSIGN lr_data->* TO <field_value>.
        IF <field_value> NE 'NO_MANAGER'.
          s_supo-sup = <field_value>.
          APPEND s_supo TO it_supo.
```

```
            CLEAR s_supo.
          ENDIF.
        ENDIF.
        UNASSIGN: <field>, <field_value>.
      ENDLOOP.

      "-----------------ODATA-----TELEFONO----------------------

      lg_comilla = ''''.
      lg_comilla = lg_comilla(1).
      CLEAR: response, url_ta_p.

      CONCATENATE lv_url '/odata/v2/PerPhone' INTO url_ta.

      url_ta_p = url_ta.
      CONCATENATE url_ta_p '?$filter= personIdExternal ge' INTO url_ta_p.
      CONCATENATE url_ta_p lg_comilla INTO url_ta_p SEPARATED BY space.
      CONCATENATE url_ta_p lv_per1 lg_comilla INTO url_ta_p.
      CONCATENATE  url_ta_p  'and  personIdExternal  le'  INTO  url_ta_p
SEPARATED BY space.
      CONCATENATE url_ta_p lg_comilla INTO url_ta_p SEPARATED BY space.
      CONCATENATE url_ta_p lv_per2 lg_comilla INTO url_ta_p.
      CONCATENATE                                                  url_ta_p
'&$select=personIdExternal,phoneNumber&$format=JSON&$orderby=personIdExte
rnal asc&$top=1000' INTO url_ta_p.
```

```
**"create HTTP client by url
*  "API endpoint for API sandbox
      CALL METHOD cl_http_client=>create_by_url
        EXPORTING
          url                 = url_ta_p
        IMPORTING
          client              = lo_http_client
        EXCEPTIONS
          argument_not_found = 1
          plugin_not_active  = 2
          internal_error     = 3
          OTHERS              = 4.

      IF sy-subrc <> 0.
        "error handling
      ENDIF.


      "setting request method
      lo_http_client->request->set_method('GET').

      "adding headers
```

```
      lo_http_client->request->set_header_field(  name  =  'Content-Type'
value = 'application/json' ).

      lo_http_client->request->set_header_field( name = 'Accept' value =
'application/json' ).


      "Basic  Auth  :  provide  username:password  in  Base64  encoded  in
Authorization header

      lo_http_client->request->set_header_field(  name  =  'Authorization'
value = aut ).


      CALL METHOD lo_http_client->send

        EXCEPTIONS

          http_communication_failure = 1

          http_invalid_state         = 2

          http_processing_failed     = 3

          http_invalid_timeout       = 4

          OTHERS                     = 5.


      IF sy-subrc = 0.

        CALL METHOD lo_http_client->receive

          EXCEPTIONS

            http_communication_failure = 1

            http_invalid_state         = 2

            http_processing_failed     = 3

            OTHERS                     = 5.

      ENDIF.
```

```
IF sy-subrc <> 0.
  "error handling
ENDIF.

response = lo_http_client->response->get_cdata( ).

CLEAR: lr_data.

CALL METHOD /ui2/cl_json=>deserialize
  EXPORTING
    json         = response
    pretty_name  = /ui2/cl_json=>pretty_mode-user
    assoc_arrays = abap_true
  CHANGING
    data         = lr_data.

IF lr_data IS BOUND.
  UNASSIGN: <data>, <d>, <data2>, <results>, <table>, <structure>.
  ASSIGN lr_data->* TO <data>.
  ASSIGN COMPONENT 'D' OF STRUCTURE <data> TO <d>.
  MOVE <d> TO lr_data2.
  ASSIGN lr_data2->* TO <data2>.
  ASSIGN COMPONENT 'RESULTS' OF STRUCTURE <data2> TO <results>.
```

```
        ASSIGN <results>->* TO <table>.
      ENDIF.

      LOOP AT <table> ASSIGNING <structure>.
        ASSIGN <structure>->* TO <data>.

        ASSIGN COMPONENT 'PERSONIDEXTERNAL' OF STRUCTURE <data> TO <field>.
        IF <field> IS ASSIGNED.
          lr_data = <field>.
          ASSIGN lr_data->* TO <field_value>.

          s_telef-pernr = <field_value>.

        ENDIF.

        ASSIGN COMPONENT 'PHONENUMBER' OF STRUCTURE <data> TO <field>.
        IF <field> IS ASSIGNED.
          lr_data = <field>.
          ASSIGN lr_data->* TO <field_value>.

          s_telef-telefono = <field_value>.
          APPEND s_telef TO it_telef.
          CLEAR s_telef.
        ENDIF.
```

```
        UNASSIGN: <field>, <field_value>.

      ENDLOOP.

      "-------------------------------------------------------

      "-----------------ODATA-----CORREOS  TRABAJADOR--------------------
--



      lg_comilla = ''''.

      lg_comilla = lg_comilla(1).

      CLEAR: response, url_ta_p.


      CONCATENATE lv_url '/odata/v2/PerEmail' INTO url_ta.


      url_ta_p = url_ta.

      CONCATENATE url_ta_p '?$filter= personIdExternal ge' INTO url_ta_p.

      CONCATENATE url_ta_p lg_comilla INTO url_ta_p SEPARATED BY space.

      CONCATENATE url_ta_p lv_per1 lg_comilla INTO url_ta_p.

      CONCATENATE  url_ta_p  'and  personIdExternal  le'  INTO  url_ta_p
SEPARATED BY space.

      CONCATENATE url_ta_p lg_comilla INTO url_ta_p SEPARATED BY space.

      CONCATENATE url_ta_p lv_per2 lg_comilla INTO url_ta_p.

      CONCATENATE                                                  url_ta_p
'&$select=personIdExternal,emailAddress&$format=JSON&$orderby=personIdExt
ernal asc&$top=1000' INTO url_ta_p.



**"create HTTP client by url
```

```
*  "API endpoint for API sandbox

      CALL METHOD cl_http_client=>create_by_url

        EXPORTING

          url                 = url_ta_p

        IMPORTING

          client              = lo_http_client

        EXCEPTIONS

          argument_not_found = 1

          plugin_not_active  = 2

          internal_error     = 3

          OTHERS              = 4.


      IF sy-subrc <> 0.

        "error handling

      ENDIF.




      "setting request method

      lo_http_client->request->set_method('GET').


      "adding headers

      lo_http_client->request->set_header_field(  name  =  'Content-Type'
value = 'application/json' ).

      lo_http_client->request->set_header_field( name = 'Accept' value =
'application/json' ).
```

```
      "Basic  Auth  :  provide  username:password  in  Base64  encoded  in
Authorization header
      lo_http_client->request->set_header_field(  name  =  'Authorization'
value = aut ).


      CALL METHOD lo_http_client->send
        EXCEPTIONS
          http_communication_failure = 1
          http_invalid_state         = 2
          http_processing_failed     = 3
          http_invalid_timeout       = 4
          OTHERS                     = 5.


      IF sy-subrc = 0.
        CALL METHOD lo_http_client->receive
          EXCEPTIONS
            http_communication_failure = 1
            http_invalid_state         = 2
            http_processing_failed     = 3
            OTHERS                     = 5.
      ENDIF.


      IF sy-subrc <> 0.
        "error handling
```

```
      ENDIF.


      response = lo_http_client->response->get_cdata( ).


      CLEAR: lr_data.


      CALL METHOD /ui2/cl_json=>deserialize
        EXPORTING
          json         = response
          pretty_name  = /ui2/cl_json=>pretty_mode-user
          assoc_arrays = abap_true
        CHANGING
          data         = lr_data.


      IF lr_data IS BOUND.
        UNASSIGN: <data>, <d>, <data2>, <results>, <table>, <structure>.
        ASSIGN lr_data->* TO <data>.
        ASSIGN COMPONENT 'D' OF STRUCTURE <data> TO <d>.
        MOVE <d> TO lr_data2.
        ASSIGN lr_data2->* TO <data2>.
        ASSIGN COMPONENT 'RESULTS' OF STRUCTURE <data2> TO <results>.
        ASSIGN <results>->* TO <table>.
      ENDIF.
```

```
      LOOP AT <table> ASSIGNING <structure>.
        ASSIGN <structure>->* TO <data>.

        ASSIGN COMPONENT 'PERSONIDEXTERNAL' OF STRUCTURE <data> TO <field>.
        IF <field> IS ASSIGNED.
          lr_data = <field>.
          ASSIGN lr_data->* TO <field_value>.

          s_correo-pernr = <field_value>.

        ENDIF.

        ASSIGN COMPONENT 'EMAILADDRESS' OF STRUCTURE <data> TO <field>.
        IF <field> IS ASSIGNED.
          lr_data = <field>.
          ASSIGN lr_data->* TO <field_value>.

          CLEAR: lv_pos, lv_correo.
          lv_correo =  <field_value>.
          IF lv_correo IS NOT INITIAL.
            FIND 'agroindustriallaredo' IN lv_correo MATCH COUNT lv_pos.
            IF sy-subrc = 0.

              s_correo-correol = <field_value>.
```

```
            ELSE.
              s_correo-correop = <field_value>.
            ENDIF.


            APPEND s_correo TO it_correo.
            CLEAR s_correo.
          ENDIF.
        ENDIF.
        UNASSIGN: <field>, <field_value>.
      ENDLOOP.
      "--------------------------------------------------------
```

```
      "-------------------------------------------------------------------
-------------------------
      IF it_supo[] IS NOT INITIAL.
        CLEAR lv_cont.
        lv_per3 = lv_per1.
        WHILE lv_cont LE 10.
          SORT it_supo DESCENDING BY pernr.
          READ TABLE  it_supo INTO s_supo INDEX 1.
          IF lv_per3 NE s_supo-pernr+2(6).
            lv_per3 = s_supo-pernr+2(6).
            lv_per1 = s_supo-pernr+2(6) + 1.
            CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
```

```
              EXPORTING
                input  = lv_per1
              IMPORTING
                output = lv_per1.


            lv_cont = lv_cont + 1.
            lv_per2 = '999999'.
            CLEAR lv_perso.
            lv_perso = lv_per1.
            lg_comilla = ''''.
            lg_comilla = lg_comilla(1).
            CLEAR: response, url_ta_p.


            CONCATENATE lv_url '/odata/v2/EmpJob' INTO url_ta.


            url_ta_p = url_ta.
            CONCATENATE url_ta_p '?$filter= userId ge' INTO url_ta_p.
            CONCATENATE  url_ta_p  lg_comilla  INTO  url_ta_p  SEPARATED  BY
space.
            CONCATENATE url_ta_p lv_per1 lg_comilla INTO url_ta_p.
            CONCATENATE url_ta_p 'and userId le' INTO url_ta_p SEPARATED
BY space.
            CONCATENATE  url_ta_p  lg_comilla  INTO  url_ta_p  SEPARATED  BY
space.
            CONCATENATE url_ta_p lv_per2 lg_comilla INTO url_ta_p.
```

```
            CONCATENATE                                                    url_ta_p
'&$select=userId,managerId&$format=JSON&$orderby=userId     asc&$top=1000'
INTO url_ta_p.




**"create HTTP client by url

*  "API endpoint for API sandbox

            CALL METHOD cl_http_client=>create_by_url

              EXPORTING

                url                 = url_ta_p

              IMPORTING

                client              = lo_http_client

              EXCEPTIONS

                argument_not_found = 1

                plugin_not_active  = 2

                internal_error     = 3

                OTHERS             = 4.


            IF sy-subrc <> 0.

              "error handling

            ENDIF.




            "setting request method

            lo_http_client->request->set_method('GET').
```

```
            "adding headers
            lo_http_client->request->set_header_field(  name  =  'Content-
Type' value = 'application/json' ).
            lo_http_client->request->set_header_field(  name  =  'Accept'
value = 'application/json' ).


            "Basic Auth : provide username:password in Base64 encoded in
Authorization header
            lo_http_client->request->set_header_field(        name        =
'Authorization' value = aut ).


            CALL METHOD lo_http_client->send
              EXCEPTIONS
                http_communication_failure = 1
                http_invalid_state         = 2
                http_processing_failed     = 3
                http_invalid_timeout       = 4
                OTHERS                     = 5.


            IF sy-subrc = 0.
              CALL METHOD lo_http_client->receive
                EXCEPTIONS
                  http_communication_failure = 1
                  http_invalid_state         = 2
                  http_processing_failed     = 3
```

```
                OTHERS                      = 5.
            ENDIF.

            IF sy-subrc <> 0.
              "error handling
            ENDIF.

            response = lo_http_client->response->get_cdata( ).

            CLEAR: lr_data.

            CALL METHOD /ui2/cl_json=>deserialize
              EXPORTING
                json          = response
                pretty_name   = /ui2/cl_json=>pretty_mode-user
                assoc_arrays = abap_true
              CHANGING
                data          = lr_data.

            IF lr_data IS BOUND.
              UNASSIGN:  <data>,  <d>,  <data2>,  <results>,  <table>,
<structure>.
              ASSIGN lr_data->* TO <data>.
              ASSIGN COMPONENT 'D' OF STRUCTURE <data> TO <d>.
              MOVE <d> TO lr_data2.
```

```
            ASSIGN lr_data2->* TO <data2>.
            ASSIGN COMPONENT 'RESULTS' OF STRUCTURE <data2> TO <results>.
            ASSIGN <results>->* TO <table>.
          ENDIF.

          LOOP AT <table> ASSIGNING <structure>.
            ASSIGN <structure>->* TO <data>.



            ASSIGN COMPONENT 'USERID' OF STRUCTURE <data> TO <field>.
            IF <field> IS ASSIGNED.
              lr_data = <field>.
              ASSIGN lr_data->* TO <field_value>.
              s_supo-pernr = <field_value>.

            ENDIF.

            ASSIGN COMPONENT 'MANAGERID' OF STRUCTURE <data> TO <field>.
            IF <field> IS ASSIGNED.
              lr_data = <field>.
              ASSIGN lr_data->* TO <field_value>.
              IF <field_value> NE 'NO_MANAGER'.
                s_supo-sup = <field_value>.
                APPEND s_supo TO it_supo.
```

```
                CLEAR s_supo.
              ENDIF.
            ENDIF.
            UNASSIGN: <field>, <field_value>.
          ENDLOOP.

          "-----------------ODATA-----TELEFONO----------------------

          lg_comilla = ''''.
          lg_comilla = lg_comilla(1).
          CLEAR: response, url_ta_p.

          CONCATENATE lv_url '/odata/v2/PerPhone' INTO url_ta.

          url_ta_p = url_ta.
          CONCATENATE  url_ta_p  '?$filter=  personIdExternal  ge'  INTO
url_ta_p.
          CONCATENATE  url_ta_p  lg_comilla  INTO  url_ta_p  SEPARATED  BY
space.
          CONCATENATE url_ta_p lv_per1 lg_comilla INTO url_ta_p.
          CONCATENATE url_ta_p 'and personIdExternal le' INTO url_ta_p
SEPARATED BY space.
          CONCATENATE  url_ta_p  lg_comilla  INTO  url_ta_p  SEPARATED  BY
space.
          CONCATENATE url_ta_p lv_per2 lg_comilla INTO url_ta_p.
```

```
            CONCATENATE                                                 url_ta_p
'&$select=personIdExternal,phoneNumber&$format=JSON&$orderby=personIdExte
rnal asc&$top=1000' INTO url_ta_p.



**"create HTTP client by url

*  "API endpoint for API sandbox

            CALL METHOD cl_http_client=>create_by_url

              EXPORTING

                url                 = url_ta_p

              IMPORTING

                client              = lo_http_client

              EXCEPTIONS

                argument_not_found = 1

                plugin_not_active  = 2

                internal_error     = 3

                OTHERS             = 4.


            IF sy-subrc <> 0.

              "error handling

            ENDIF.



            "setting request method

            lo_http_client->request->set_method('GET').
```

```
            "adding headers
            lo_http_client->request->set_header_field(  name  =  'Content-
Type' value = 'application/json' ).
            lo_http_client->request->set_header_field(  name  =  'Accept'
value = 'application/json' ).


            "Basic Auth : provide username:password in Base64 encoded in
Authorization header
            lo_http_client->request->set_header_field(       name       =
'Authorization' value = aut ).


            CALL METHOD lo_http_client->send
              EXCEPTIONS
                http_communication_failure = 1
                http_invalid_state         = 2
                http_processing_failed     = 3
                http_invalid_timeout       = 4
                OTHERS                     = 5.


            IF sy-subrc = 0.
              CALL METHOD lo_http_client->receive
                EXCEPTIONS
                  http_communication_failure = 1
                  http_invalid_state         = 2
                  http_processing_failed     = 3
```

```
                OTHERS                      = 5.
            ENDIF.

            IF sy-subrc <> 0.
              "error handling
            ENDIF.

            response = lo_http_client->response->get_cdata( ).

            CLEAR: lr_data.

            CALL METHOD /ui2/cl_json=>deserialize
              EXPORTING
                json          = response
                pretty_name   = /ui2/cl_json=>pretty_mode-user
                assoc_arrays = abap_true
              CHANGING
                data          = lr_data.

            IF lr_data IS BOUND.
              UNASSIGN:  <data>,  <d>,  <data2>,  <results>,  <table>,
<structure>.
              ASSIGN lr_data->* TO <data>.
              ASSIGN COMPONENT 'D' OF STRUCTURE <data> TO <d>.
              MOVE <d> TO lr_data2.
```

```
              ASSIGN lr_data2->* TO <data2>.
              ASSIGN COMPONENT 'RESULTS' OF STRUCTURE <data2> TO <results>.
              ASSIGN <results>->* TO <table>.
            ENDIF.

            LOOP AT <table> ASSIGNING <structure>.
              ASSIGN <structure>->* TO <data>.

              ASSIGN COMPONENT 'PERSONIDEXTERNAL' OF STRUCTURE <data> TO
<field>.
              IF <field> IS ASSIGNED.
                lr_data = <field>.
                ASSIGN lr_data->* TO <field_value>.

                s_telef-pernr = <field_value>.

              ENDIF.

              ASSIGN COMPONENT 'PHONENUMBER' OF STRUCTURE <data> TO
<field>.
              IF <field> IS ASSIGNED.
                lr_data = <field>.
                ASSIGN lr_data->* TO <field_value>.

                s_telef-telefono = <field_value>.
```

```
                APPEND s_telef TO it_telef.
                CLEAR s_telef.
              ENDIF.
              UNASSIGN: <field>, <field_value>.
            ENDLOOP.
            "-------------------------------------------------------
            "-----------------ODATA-----CORREOS  TRABAJADOR---------------
--------


            lg_comilla = ''''.
            lg_comilla = lg_comilla(1).
            CLEAR: response, url_ta_p.


            CONCATENATE lv_url '/odata/v2/PerEmail' INTO url_ta.


            url_ta_p = url_ta.
            CONCATENATE  url_ta_p  '?$filter=  personIdExternal  ge'  INTO
url_ta_p.
            CONCATENATE  url_ta_p  lg_comilla  INTO  url_ta_p  SEPARATED  BY
space.
            CONCATENATE url_ta_p lv_per1 lg_comilla INTO url_ta_p.
            CONCATENATE url_ta_p 'and personIdExternal le' INTO url_ta_p
SEPARATED BY space.
            CONCATENATE  url_ta_p  lg_comilla  INTO  url_ta_p  SEPARATED  BY
space.
            CONCATENATE url_ta_p lv_per2 lg_comilla INTO url_ta_p.
```

```
            CONCATENATE                                                    url_ta_p
'&$select=personIdExternal,emailAddress&$format=JSON&$orderby=personIdExt
ernal asc&$top=1000' INTO url_ta_p.



**"create HTTP client by url

*  "API endpoint for API sandbox

            CALL METHOD cl_http_client=>create_by_url

              EXPORTING

                url                 = url_ta_p

              IMPORTING

                client              = lo_http_client

              EXCEPTIONS

                argument_not_found = 1

                plugin_not_active  = 2

                internal_error     = 3

                OTHERS             = 4.


            IF sy-subrc <> 0.

              "error handling

            ENDIF.



            "setting request method

            lo_http_client->request->set_method('GET').
```

```
            "adding headers
            lo_http_client->request->set_header_field(  name  =  'Content-
Type' value = 'application/json' ).
            lo_http_client->request->set_header_field(  name  =  'Accept'
value = 'application/json' ).


            "Basic Auth : provide username:password in Base64 encoded in
Authorization header
            lo_http_client->request->set_header_field(        name        =
'Authorization' value = aut ).


            CALL METHOD lo_http_client->send
              EXCEPTIONS
                http_communication_failure = 1
                http_invalid_state         = 2
                http_processing_failed     = 3
                http_invalid_timeout       = 4
                OTHERS                     = 5.


            IF sy-subrc = 0.
              CALL METHOD lo_http_client->receive
                EXCEPTIONS
                  http_communication_failure = 1
                  http_invalid_state         = 2
                  http_processing_failed     = 3
```

```
                OTHERS                      = 5.
            ENDIF.

            IF sy-subrc <> 0.
              "error handling
            ENDIF.

            response = lo_http_client->response->get_cdata( ).

            CLEAR: lr_data.

            CALL METHOD /ui2/cl_json=>deserialize
              EXPORTING
                json          = response
                pretty_name   = /ui2/cl_json=>pretty_mode-user
                assoc_arrays  = abap_true
              CHANGING
                data          = lr_data.

            IF lr_data IS BOUND.
              UNASSIGN:  <data>,  <d>,  <data2>,  <results>,  <table>,
<structure>.
              ASSIGN lr_data->* TO <data>.
              ASSIGN COMPONENT 'D' OF STRUCTURE <data> TO <d>.
              MOVE <d> TO lr_data2.
```

```
            ASSIGN lr_data2->* TO <data2>.
            ASSIGN COMPONENT 'RESULTS' OF STRUCTURE <data2> TO <results>.
            ASSIGN <results>->* TO <table>.
          ENDIF.

          LOOP AT <table> ASSIGNING <structure>.
            ASSIGN <structure>->* TO <data>.

            ASSIGN COMPONENT 'PERSONIDEXTERNAL' OF STRUCTURE <data> TO
<field>.
            IF <field> IS ASSIGNED.
              lr_data = <field>.
              ASSIGN lr_data->* TO <field_value>.

              s_correo-pernr = <field_value>.

            ENDIF.

            ASSIGN COMPONENT 'EMAILADDRESS' OF STRUCTURE <data> TO
<field>.
            IF <field> IS ASSIGNED.
              lr_data = <field>.
              ASSIGN lr_data->* TO <field_value>.

              CLEAR: lv_pos, lv_correo.
```

```
                lv_correo =  <field_value>.
                IF lv_correo IS NOT INITIAL.
                  FIND  'agroindustriallaredo'  IN  lv_correo  MATCH  COUNT
lv_pos.
                  IF sy-subrc = 0.


                    s_correo-correol = <field_value>.
                  ELSE.
                    s_correo-correop = <field_value>.
                  ENDIF.


                  APPEND s_correo TO it_correo.
                  CLEAR s_correo.
                ENDIF.
              ENDIF.
              UNASSIGN: <field>, <field_value>.
            ENDLOOP.
            "-------------------------------------------------------
          ELSE.
            lv_cont = 11.
          ENDIF.
        ENDWHILE.
      ENDIF.


      SELECT * INTO TABLE @DATA(it_cargo_sup)
```

```
        FROM pa0001
        FOR ALL ENTRIES IN @it_supo
        WHERE pernr EQ @it_supo-sup
        AND begda LE @lv_fecha
       AND endda GE @lv_fecha
       AND uni_neg NE @space.
      SORT it_cargo_sup DESCENDING BY pernr endda.
      DELETE ADJACENT DUPLICATES FROM it_cargo_sup COMPARING pernr.

      LOOP AT it_per INTO s_per.
        "-----------------ODATA-----SUPERVISOR----------------------
        CLEAR lv_pernr.

        READ TABLE it_supo INTO s_supo WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.
          st_trab-emp_jefe_codigo = s_supo-sup+2(6).
          lv_pernr = s_supo-sup.
        ENDIF.

        "-----------------ODATA-----CORREO  SUPERVISOR-------------------
---
        IF st_trab-emp_jefe_codigo IS NOT INITIAL.

          LOOP AT it_correo INTO s_correo WHERE pernr+2(6) = st_trab-
emp_jefe_codigo.
```

```
            IF s_correo-correop IS NOT INITIAL.
            ELSEIF s_correo-correol IS NOT INITIAL.
              st_trab-emp_email_laboral_jefe =  s_correo-correol.
            ENDIF.
          ENDLOOP.



          READ TABLE it_cargo_sup INTO DATA(st_cargos) WITH KEY pernr =
lv_pernr.
          IF sy-subrc EQ 0.
            st_trab-emp_cargo_jefe = st_cargos-posicion.


            CLEAR: g_stext.
            SELECT SINGLE stext INTO g_stext
              FROM hrp1000
             WHERE plvar EQ '01'
               AND otype EQ 'S'
               AND objid EQ st_cargos-plans
               AND istat EQ '1'
               AND langu EQ 'S'
               AND seqnr EQ '000'.
            IF sy-subrc EQ 0.
              st_trab-desc_cargo_jefe =  g_stext.
              TRANSLATE st_trab-desc_cargo_jefe TO UPPER CASE.
            ENDIF.
```

```
          ENDIF.


        ENDIF.
        "--------------------------------------
*        "-----------------ODATA-----TELEFONO-----------------------
        READ TABLE it_telef INTO s_telef WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.
          st_trab-emp_telefono = s_telef-telefono.
        ENDIF.


*        "-------------------------------------------------------
        st_trab-emp_codigo = s_per-pernr+2(6).
        st_trab-emp_sociedad = s_per-bukrs.
        st_trab-emp_gerencia = s_per-uni_neg.
        st_trab-sitio_pago = s_per-division.
        st_trab-emp_tipo_nomina = s_per-abkrs.
        st_trab-liquidacion_nombre = 'MENSUAL'.
        st_trab-emp_cargo = s_per-posicion.


        READ TABLE it_divi INTO st_divi WITH KEY division = s_per-division.
        IF sy-subrc EQ 0.
          st_trab-nombre_sitio =  st_divi-descrip.
          TRANSLATE st_trab-nombre_sitio TO UPPER CASE.
        ENDIF.
```

```
st_trab-nombre_sociedad = st_dir-name1.

LOOP AT it_correo INTO s_correo WHERE pernr = s_per-pernr.
  IF s_correo-correop IS NOT INITIAL.
    st_trab-emp_email_personal =  s_correo-correop.
  ELSEIF s_correo-correol IS NOT INITIAL.
    st_trab-emp_email_laboral = s_correo-correol.
  ENDIF.
ENDLOOP.

CLEAR st_t005t.
READ TABLE it_t005t INTO st_t005t WITH KEY land1 = st_dir-cntry.
IF sy-subrc EQ 0.
  st_trab-emp_pais_sociedad  = st_t005t-landx.
  TRANSLATE  st_trab-emp_pais_sociedad TO UPPER CASE.
ENDIF.

st_trab-emp_ciudad_sociedad = st_dir-city.
TRANSLATE st_trab-emp_ciudad_sociedad TO UPPER CASE.
st_trab-nombre_localidad    = st_dir-stret.
TRANSLATE st_trab-nombre_localidad TO UPPER CASE.

CLEAR: g_stext.
```

```
        SELECT SINGLE stext INTO g_stext
          FROM hrp1000
         WHERE plvar EQ '01'
           AND otype EQ 'S'
           AND objid EQ s_per-plans
           AND istat EQ '1'
           AND langu EQ 'S'
           AND seqnr EQ '000'.
        IF sy-subrc EQ 0.
          st_trab-cargo_nombre =  g_stext.
          TRANSLATE st_trab-cargo_nombre TO UPPER CASE.
        ENDIF.


        CLEAR g_stext.
        READ TABLE it_unidades INTO w_unidad WITH KEY objid = s_per-orgeh.
        IF sy-subrc EQ 0.
          st_trab-nombre_gerencia = w_unidad-stext.
          TRANSLATE st_trab-nombre_gerencia TO UPPER CASE.
        ENDIF.


        READ TABLE it_nivel INTO st_nivel WITH KEY zcod_clas = s_per-
clasifica.
        IF sy-subrc EQ 0.
          st_trab-nivel_jerarquico = st_nivel-znivel.
        ENDIF.
```

```
        READ TABLE it_nom INTO st_nom WITH KEY abkrs = s_per-abkrs.
        IF sy-subrc EQ 0.
          st_trab-tipo_nomina_nombre = st_nom-atext.
          TRANSLATE st_trab-tipo_nomina_nombre TO UPPER CASE.
        ENDIF.


        CLEAR st_p0002.
        READ TABLE it_p0002 INTO st_p0002 WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.
          CONCATENATE  st_p0002-nachn  st_p0002-name2  st_p0002-vorna  INTO
st_trab-emp_nombre_completo SEPARATED BY space.
          TRANSLATE st_trab-emp_nombre_completo TO UPPER CASE.


          st_trab-emp_apellido1 = st_p0002-nachn.
          TRANSLATE  st_trab-emp_apellido1 TO UPPER CASE.
          st_trab-emp_apellido2 = st_p0002-name2.
          TRANSLATE  st_trab-emp_apellido2 TO UPPER CASE.
          st_trab-nombre = st_p0002-vorna.
          TRANSLATE  st_trab-nombre TO UPPER CASE.



          IF st_p0002-gesch EQ '1'.
            st_trab-emp_sexo =  'M'.
```

```
          ELSE.
            st_trab-emp_sexo =  'F'.
          ENDIF.

          " FECHA DE NACIMIENTO convertir a UTC--------------
          CLEAR:   lv_date,      lv_time,   lv_year,   lv_month,   lv_day,
lv_hour,lv_minute, lv_second, lv_timestamp.
          lv_date = st_p0002-gbdat." FECHA DE NACIMIENTO
          lv_time = '000000'.    " Hora predeterminada: medianoche
          " Extraer componentes de la fecha y hora
          lv_year = lv_date+0(4).
          lv_month = lv_date+4(2).
          lv_day = lv_date+6(2).

          lv_hour = lv_time+0(2).
          lv_minute = lv_time+2(2).
          lv_second = lv_time+4(2).

          " Construir el formato ISO 8601
          lv_timestamp = |{ lv_year }-{ lv_month }-{ lv_day }T{ lv_hour }:{
lv_minute }:{ lv_second }Z|.

          st_trab-emp_fecha_naci = lv_timestamp.
          "-------------------------------------------------------------
          CALL FUNCTION 'HR_ECM_GET_PERIOD_BETW_DATES'
```

```
            EXPORTING
              begda            = st_p0002-gbdat
              endda            = sy-datum
              message_handler = handler
            IMPORTING
              num_years        = i_edad.

          st_trab-emp_edad =  i_edad.
          st_trab-ugn1_codigo_naci = st_p0002-gblnd.

          CLEAR st_t005t.
          READ TABLE it_t005t INTO st_t005t WITH KEY land1 = st_p0002-
gblnd.
          IF sy-subrc EQ 0.
            st_trab-ugn1_desc_naci = st_t005t-landx.
            TRANSLATE st_trab-ugn1_desc_naci TO UPPER CASE.
          ENDIF.

          st_trab-ugn2_desc_naci = st_p0002-gbort.
          st_trab-ugn3_codigo_naci = st_p0002-nati2.

          CLEAR st_t005t.
          READ TABLE it_t005t INTO st_t005t WITH KEY land1 = st_p0002-
nati2.
          IF sy-subrc EQ 0.
```

```
            st_trab-ugn3_desc_naci = st_t005t-landx.
            TRANSLATE st_trab-ugn3_desc_naci TO UPPER CASE.
          ENDIF.

          st_trab-emp_estado_civil = st_p0002-famst.

          CLEAR st_civil.
          READ TABLE it_civil INTO st_civil WITH KEY zlow =  st_p0002-
famst.
          IF sy-subrc EQ 0.
            st_trab-desc_est_civil = st_civil-zhigh.
          ENDIF.

          IF st_p0002-zztipo_sang IS NOT INITIAL.
            CLEAR: v_var1, v_var2.
            FIND '+' IN st_p0002-zztipo_sang MATCH COUNT lv_pos.
            IF sy-subrc = 0.
              st_trab-emp_sangre_rh = st_p0002-zztipo_sang+lv_pos(1).

              SPLIT st_p0002-zztipo_sang AT '+' INTO v_var1 v_var2.
              st_trab-emp_tipo_sangre =  v_var1.
            ELSE.
              st_trab-emp_sangre_rh = '-'.
              SPLIT st_p0002-zztipo_sang AT '-' INTO v_var1 v_var2.
              st_trab-emp_tipo_sangre =  v_var1.
```

```
            ENDIF.


          ENDIF.


        ENDIF.


        IF s_per-massn NE 'Z3'."Cesado


          CLEAR st_p2001.

          READ TABLE it_p2001 INTO st_p2001 WITH KEY pernr = s_per-pernr.

          IF sy-subrc EQ 0.

            IF st_p2001-subty EQ '9120' OR st_p2001-subty EQ '9137'.

              st_trab-emp_estado = 'V'.

              st_trab-estado = 'VACACIONES'.

            ELSE.

              st_trab-emp_estado = 'A'.

              st_trab-estado = 'AUSENTE'.

            ENDIF.

          ELSE.

            st_trab-emp_estado = 'N'.

            st_trab-estado = 'ACTIVO'.

          ENDIF.


        ELSE.
```

```
          st_trab-emp_estado = 'C'.
          st_trab-estado = 'CESADO'.
        ENDIF.

        CLEAR: st_p0041, lv_ingreso.
        READ TABLE it_p0041 INTO st_p0041 WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.
          lv_ingreso = st_p0041-dat01.
          "convertir UTC
          CLEAR:   lv_date,      lv_time,   lv_year,   lv_month,   lv_day,
lv_hour,lv_minute, lv_second, lv_timestamp.
          lv_date = st_p0041-dat01." FECHA DE ingreso
          lv_time = '000000'.    " Hora predeterminada: medianoche
          " Extraer componentes de la fecha y hora
          lv_year = lv_date+0(4).
          lv_month = lv_date+4(2).
          lv_day = lv_date+6(2).

          lv_hour = lv_time+0(2).
          lv_minute = lv_time+2(2).
          lv_second = lv_time+4(2).

          " Construir el formato ISO 8601
          lv_timestamp = |{ lv_year }-{ lv_month }-{ lv_day }T{ lv_hour }:{
lv_minute }:{ lv_second }Z|.
```

```
          st_trab-emp_fecha_ingreso = lv_timestamp."fecha de ingreso UTC
        ENDIF.

        IF s_per-massn EQ 'Z3'."Cesado

          CALL FUNCTION 'RP_CALC_DATE_IN_INTERVAL'
            EXPORTING
              date      = s_per-begda
              days      = 01
              months    = 00
              signum    = '-'
              years     = 00
            IMPORTING
              calc_date = lv_fec_cese.
          "convertir UTC
          CLEAR:   lv_date,      lv_time,   lv_year,   lv_month,   lv_day,
lv_hour,lv_minute, lv_second, lv_timestamp.
          lv_date = lv_fec_cese." FECHA DE CESE
          lv_time = '000000'.    " Hora predeterminada: medianoche
          " Extraer componentes de la fecha y hora
          lv_year = lv_date+0(4).
          lv_month = lv_date+4(2).
          lv_day = lv_date+6(2).

          lv_hour = lv_time+0(2).
```

```
          lv_minute = lv_time+2(2).
          lv_second = lv_time+4(2).

          " Construir el formato ISO 8601
          lv_timestamp = |{ lv_year }-{ lv_month }-{ lv_day }T{ lv_hour }:{
lv_minute }:{ lv_second }Z|.
          st_trab-emp_fecha_cese = lv_timestamp."fecha de CESE UTC

        ENDIF.

        CLEAR st_p0185.
        READ TABLE it_p0185 INTO st_p0185 WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.
          st_trab-emp_cedula = st_p0185-icnum.
          st_trab-emp_tipo_identif = st_p0185-ictyp.
          st_trab-ugn1_codigo_ident =  st_p0185-idcot.
          CLEAR st_t005t.
          READ TABLE it_t005t INTO st_t005t WITH KEY land1 = st_p0185-
idcot.
          IF sy-subrc EQ 0.
            st_trab-ugn1_desc_ident = st_t005t-landx.
            TRANSLATE st_trab-ugn1_desc_ident TO UPPER CASE.
          ENDIF.
```

```
          READ TABLE it_t5r06 INTO st_t5r06 WITH KEY ictyp = st_p0185-
ictyp.
          IF sy-subrc EQ 0.
            st_trab-nombre_tip_doc = st_t5r06-ictxt.
            TRANSLATE st_trab-nombre_tip_doc TO UPPER CASE.
          ENDIF.
        ENDIF.

        CLEAR st_p0006.
        READ TABLE it_p0006 INTO st_p0006 WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.

          CLEAR: s_t5evp, v_desc.
          READ TABLE it_t5evp INTO s_t5evp WITH KEY strds =  st_p0006-
zztipvia.
          IF sy-subrc EQ 0.
            v_desc   =  s_t5evp-viapt.          " TIPO DE VIA
          ENDIF.

          CLEAR: g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_10, g_11,
g_12, g_13, g_14.

          CONCATENATE v_desc   st_p0006-zznombvi   INTO g_1 SEPARATED BY
space. CONDENSE g_1.
          IF st_p0006-zznrovia IS NOT INITIAL AND st_p0006-zznrovia NE '0'.
```

```
            CONCATENATE 'Nro. de vía' st_p0006-zznrovia INTO g_2 SEPARATED
BY space. CONDENSE g_2.

          ENDIF.

          IF st_p0006-zzdepto1 IS NOT INITIAL.

            CONCATENATE 'Dpto.'st_p0006-zzdepto1    INTO g_3 SEPARATED BY
space. CONDENSE g_3.

          ENDIF.

          IF st_p0006-zzinteri1 IS NOT INITIAL.

            CONCATENATE 'Int.' st_p0006-zzinteri1    INTO g_4 SEPARATED BY
space. CONDENSE g_4.

          ENDIF.

          IF st_p0006-zzmanzan1 IS NOT INITIAL.

            CONCATENATE 'Mz.' st_p0006-zzmanzan1     INTO g_5 SEPARATED BY
space. CONDENSE g_5.

          ENDIF.

          IF st_p0006-zzlote1 IS NOT INITIAL.

            CONCATENATE 'Lt.'st_p0006-zzlote1        INTO g_6 SEPARATED BY
space. CONDENSE g_6.

          ENDIF.

          IF st_p0006-zzkilome1 IS NOT INITIAL.

            CONCATENATE 'Km.' st_p0006-zzkilome1     INTO g_7 SEPARATED BY
space. CONDENSE g_7.

          ENDIF.

          IF st_p0006-zzblock1 IS NOT INITIAL.

            CONCATENATE 'Block.' st_p0006-zzblock1   INTO g_8 SEPARATED BY
space. CONDENSE g_8.

          ENDIF.
```

```
          IF st_p0006-zzetapa1 IS NOT INITIAL.
            CONCATENATE 'Etapa.' st_p0006-zzetapa1  INTO g_9 SEPARATED BY
space. CONDENSE g_9.
          ENDIF.


          CONCATENATE g_1 g_2 g_3 g_4 g_5 g_6 g_7 g_8 g_9 st_p0006-zzrefere1
INTO g_10 SEPARATED BY space. CONDENSE g_10.


          CLEAR: s_zona, v_desc.
          IF st_p0006-zztipozo1 NE '99' AND st_p0006-zztipozo1 NE '13'.
            READ TABLE it_zona INTO s_zona WITH KEY tipo_zona = st_p0006-
zztipozo1.
            IF sy-subrc EQ 0.
              v_desc  = s_zona-desc_tipo.    " DESCRIPCION DE TIPO DE ZONA
              TRANSLATE v_desc TO  UPPER CASE.
            ENDIF.
          ENDIF.


          IF s_per-persk EQ 'Z0' OR s_per-persk EQ 'Z1' OR s_per-persk EQ
'Z2' OR s_per-persk EQ 'Z3'
           OR s_per-persk EQ 'Z6'.
            IF v_desc IS NOT INITIAL.
              CONCATENATE v_desc st_p0006-zznombzo1 INTO g_11 SEPARATED BY
space. CONDENSE g_11.


            ELSE.
```

```
              g_11 = st_p0006-zznombzo1. CONDENSE g_11.
            ENDIF.

            st_trab-emp_barrio = g_11.

            st_trab-emp_direccion = g_10.

            CONDENSE st_trab-emp_direccion.
            TRANSLATE st_trab-emp_direccion TO UPPER CASE.

          ELSE.
            IF v_desc IS NOT INITIAL.
              CONCATENATE v_desc st_p0006-zznombzo1 INTO g_11 SEPARATED BY
space. CONDENSE g_11.

            ELSE.
              g_11 = st_p0006-zznombzo1. CONDENSE g_11.
            ENDIF.

            CONCATENATE g_10 g_11 INTO st_trab-emp_direccion SEPARATED BY
space.

            CONDENSE st_trab-emp_direccion.
            TRANSLATE st_trab-emp_direccion TO UPPER CASE.
          ENDIF.
```

```
          st_trab-ugn1_codigo_resid =  st_p0006-land1.
          CLEAR st_t005t.
          READ TABLE it_t005t INTO st_t005t WITH KEY land1 = st_p0006-
land1.
          IF sy-subrc EQ 0.
            st_trab-ugn1_desc_resid = st_t005t-landx.
            TRANSLATE st_trab-ugn1_desc_resid TO UPPER CASE.
          ENDIF.

          st_trab-ugn2_codigo_resid = st_p0006-zzregion1.

          CLEAR: w_region.
          READ TABLE it_region INTO w_region WITH KEY region = st_p0006-
zzregion1.
          IF sy-subrc EQ 0.
            st_trab-ugn2_desc_resid  = w_region-descregi.    " DESCRIPCION
DE REGIO O DEPARTAMENTO
          ENDIF.

          st_trab-ugn3_codigo_resid = st_p0006-zzprovin1.      " PROVINCIA
          CLEAR: w_prov.
          READ TABLE it_prov INTO w_prov WITH KEY region = st_p0006-
zzregion1
                                                 provin = st_p0006-zzprovin1.
          IF sy-subrc EQ 0.
```

```
            st_trab-ugn3_desc_resid  = w_prov-descprov.      " DESCRIPCION
DE PROVINCIA
          ENDIF.

        ENDIF.

        CLEAR st_p0016.
        READ TABLE it_p0016 INTO st_p0016 WITH KEY pernr = s_per-pernr.
        IF sy-subrc EQ 0.

          "convertir UTC Fecha de inicio de contrato
          CLEAR:   lv_date,      lv_time,   lv_year,   lv_month,   lv_day,
lv_hour,lv_minute, lv_second, lv_timestamp.
          lv_date =  st_p0016-begda.
          lv_time = '000000'.    " Hora predeterminada: medianoche
          " Extraer componentes de la fecha y hora
          lv_year = lv_date+0(4).
          lv_month = lv_date+4(2).
          lv_day = lv_date+6(2).

          lv_hour = lv_time+0(2).
          lv_minute = lv_time+2(2).
          lv_second = lv_time+4(2).

          " Construir el formato ISO 8601
```

```
          lv_timestamp = |{ lv_year }-{ lv_month }-{ lv_day }T{ lv_hour }:{
lv_minute }:{ lv_second }Z|.

          st_trab-emp_fecha_ini_contrato  = lv_timestamp." UTC


          IF st_p0016-cttyp NE 'Z1'. "Indeterminado


            "convertir UTC Fecha fin de contrato

            CLEAR:  lv_date,     lv_time,  lv_year,  lv_month,  lv_day,
lv_hour,lv_minute, lv_second, lv_timestamp.

            lv_date =  st_p0016-ctedt.

            lv_time = '000000'.    " Hora predeterminada: medianoche

            " Extraer componentes de la fecha y hora

            lv_year = lv_date+0(4).

            lv_month = lv_date+4(2).

            lv_day = lv_date+6(2).


            lv_hour = lv_time+0(2).

            lv_minute = lv_time+2(2).

            lv_second = lv_time+4(2).


            " Construir el formato ISO 8601

            lv_timestamp = |{ lv_year }-{ lv_month }-{ lv_day }T{ lv_hour
}:{ lv_minute }:{ lv_second }Z|.

            st_trab-emp_fecha_fin_contrato  = lv_timestamp." UTC
```

```
          ENDIF.
          READ TABLE it_t547s INTO st_t547s WITH KEY cttyp = st_p0016-
cttyp.
          IF sy-subrc EQ 0.
            st_trab-tipo_contrato = st_t547s-cttxt.
            TRANSLATE st_trab-tipo_contrato TO UPPER CASE.
          ENDIF.
        ENDIF.


        CLEAR st_p9015.
        READ TABLE it_p9015_e INTO st_p9015 WITH KEY pernr = s_per-pernr
zvsart = '9001'.
        IF sy-subrc EQ 0.
          st_trab-eps = 'EPS'.
        ENDIF.


        CLEAR st_p9015.
        READ TABLE it_p9015 INTO st_p9015 WITH KEY pernr = s_per-pernr
zvsart = '9002'.
        IF sy-subrc EQ 0.
          st_trab-arl =  st_p9015-zzcompa.
          TRANSLATE st_trab-arl TO UPPER CASE.
        ENDIF.


        CLEAR st_p9003.
```

```
        READ TABLE it_p9003 INTO st_p9003 WITH KEY pernr = s_per-pernr .
        IF sy-subrc EQ 0.
          IF st_p9003-zzinstitu NE '02' AND st_p9003-zzinstitu NE '99'.
            READ TABLE it_afp INTO st_afp WITH KEY zinstitu = st_p9003-
zzinstitu.
            IF sy-subrc EQ 0.
              st_trab-afp = st_afp-zdescins.
            ENDIF.
          ENDIF.
        ENDIF.


*{
        CLEAR: st_trab-desc_centro_coste, st_trab-desc_departamento.
        st_trab-departamento = s_per-departam.
        st_trab-centro_coste = s_per-kostl.
        READ TABLE lt_cskt INTO DATA(ls_cskt)
          WITH KEY kostl = s_per-kostl.
        IF sy-subrc = 0.
          st_trab-desc_centro_coste = ls_cskt-ltext.
        ENDIF.
        READ TABLE lt_departam INTO DATA(ls_departam)
          WITH KEY departam = s_per-departam.
        IF sy-subrc = 0.
          st_trab-desc_departamento = ls_departam-descrip.
        ENDIF.
```

```
*}

        IF s_per-persk EQ 'Z0' OR s_per-persk EQ 'Z1' OR s_per-persk EQ
'Z2' OR s_per-persk EQ 'Z3'
          OR s_per-persk EQ 'Z6'.
          st_trab-emp_tipoliq = 'Q'.
          APPEND st_trab TO it_empleados.

        ELSE.
          st_trab-emp_tipoliq = 'O'.
          MOVE-CORRESPONDING st_trab TO st_trab_o.
          APPEND st_trab_o TO it_obreros.
        ENDIF.

        IF s_per-massn EQ 'Z3'."Cesado
          st_cese-mandt = sy-mandt.
          st_cese-pernr = s_per-pernr.
          st_cese-fe_cese = s_per-begda."fecha de medida se cese
          st_cese-fe_ing = lv_ingreso.
          INSERT INTO ztecp_cesados VALUES st_cese.
          IF sy-subrc = 0.
            COMMIT WORK.
          ELSE.
            ROLLBACK WORK.
          ENDIF.
```

```
        ENDIF.


        CLEAR st_trab.

      ENDLOOP.


    ENDIF.

    "-----------------------------------------------------------------
--------

  ENDIF.


  IF lv_tipo_trab EQ 'Q'.

    REFRESH: it_obreros[].

  ELSEIF lv_tipo_trab EQ 'O'.

    REFRESH: it_empleados[].

  ELSE.

    REFRESH: it_empleados[], it_obreros[].

  ENDIF.


ENDFUNCTION.
```

4.6.3 Clase ZCL_INF_TRABAJADOR2

```
class ZCL_INF_TRABAJADOR2 definition

  public

  inheriting from CL_REST_HTTP_HANDLER

  final
```

```
  create public .


public section.


  methods IF_REST_APPLICATION~GET_ROOT_HANDLER
    redefinition .
protected section.


  methods HANDLE_CSRF_TOKEN
    redefinition .
private section.
ENDCLASS.




CLASS ZCL_INF_TRABAJADOR2 IMPLEMENTATION.



* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Protected Method ZCL_INF_TRABAJADOR2->HANDLE_CSRF_TOKEN
* +-----------------------------------------------------------------------
---------------------------+
* | [--->] IO_CSRF_HANDLER                TYPE REF TO IF_REST_CSRF_HANDLER
* | [--->] IO_REQUEST                     TYPE REF TO IF_REST_REQUEST
```

```
* | [--->] IO_RESPONSE                    TYPE REF TO IF_REST_RESPONSE

*  +-----------------------------------------------------------------------
----------------</SIGNATURE>

  METHOD HANDLE_CSRF_TOKEN.

    DATA:

      lv_method           TYPE string,

      lv_csrf_token       TYPE string,

      lv_csrf_token_valid TYPE abap_bool VALUE abap_false.



    lv_method = io_request->get_method( ).

    lv_csrf_token              =              io_request->get_header_field(
if_rest_request=>gc_header_csrf_token ).



    CASE lv_method.

      WHEN if_rest_request=>gc_method_get OR

           if_rest_request=>gc_method_head OR

           if_rest_request=>gc_method_options.

        lv_csrf_token = to_lower( lv_csrf_token ).

        IF lv_csrf_token = 'fetch' ##no_text.

          io_response->set_header_field(

            iv_name  = if_rest_request=>gc_header_csrf_token

            iv_value = io_csrf_handler->get_csrf_token( ) ).

        ENDIF.

      WHEN OTHERS.
```

```
        lv_csrf_token_valid     =     io_csrf_handler->validate_csrf_token(
lv_csrf_token ).

        IF lv_csrf_token_valid <> abap_true.

          io_response->set_header_field(

            iv_name  = if_rest_request=>gc_header_csrf_token

            iv_value = 'Required' ) ##no_text.

          RAISE EXCEPTION TYPE cx_rest_resource_exception

            EXPORTING

              status_code = cl_rest_status_code=>gc_client_error_forbidden

              textid                                                    =
cx_rest_resource_exception=>csrf_token_validation_failed.

        ENDIF.

    ENDCASE.

*     .

  ENDMETHOD.




*  <SIGNATURE>-----------------------------------------------------------------
---------------------------+

*       |       Instance       Public       Method       ZCL_INF_TRABAJADOR2-
>IF_REST_APPLICATION~GET_ROOT_HANDLER

*  +----------------------------------------------------------------------
---------------------------+

* | [<-()] RO_ROOT_HANDLER                TYPE REF TO IF_REST_HANDLER

*  +----------------------------------------------------------------------
----------------</SIGNATURE>
```

```
  METHOD IF_REST_APPLICATION~GET_ROOT_HANDLER.


    DATA lo_handler TYPE REF TO cl_rest_router.

    CREATE OBJECT lo_handler.

*        lo_handler->attach( iv_template = '/test' iv_handler_class =
'ZCL_INF_TRABAJADOR_RESOURCE' ).

    lo_handler->attach( iv_template = '/{PAR1}' iv_handler_class =
'ZCL_INF_TRABAJADOR_RESOURCE2' ).

    ro_root_handler = lo_handler.


  ENDMETHOD.

ENDCLASS.
```

4.6.4 Clase ZCL_INF_TRABAJADOR2

```
class ZCL_INF_TRABAJADOR2 definition

  public

  inheriting from CL_REST_HTTP_HANDLER

  final

  create public .


public section.


  methods IF_REST_APPLICATION~GET_ROOT_HANDLER

    redefinition .

protected section.
```

```
  methods HANDLE_CSRF_TOKEN
    redefinition .
private section.
ENDCLASS.




CLASS ZCL_INF_TRABAJADOR2 IMPLEMENTATION.



* <SIGNATURE>-------------------------------------------------------------
---------------------------+
* | Instance Protected Method ZCL_INF_TRABAJADOR2->HANDLE_CSRF_TOKEN
* +------------------------------------------------------------------------
---------------------------+
* | [--->] IO_CSRF_HANDLER                TYPE REF TO IF_REST_CSRF_HANDLER
* | [--->] IO_REQUEST                     TYPE REF TO IF_REST_REQUEST
* | [--->] IO_RESPONSE                    TYPE REF TO IF_REST_RESPONSE
* +------------------------------------------------------------------------
----------------</SIGNATURE>
  METHOD HANDLE_CSRF_TOKEN.
    DATA:
      lv_method           TYPE string,
      lv_csrf_token       TYPE string,
      lv_csrf_token_valid TYPE abap_bool VALUE abap_false.
```

```
    lv_method = io_request->get_method( ).
    lv_csrf_token              =              io_request->get_header_field(
if_rest_request=>gc_header_csrf_token ).


    CASE lv_method.
      WHEN if_rest_request=>gc_method_get OR
           if_rest_request=>gc_method_head OR
           if_rest_request=>gc_method_options.
        lv_csrf_token = to_lower( lv_csrf_token ).
        IF lv_csrf_token = 'fetch' ##no_text.
          io_response->set_header_field(
            iv_name  = if_rest_request=>gc_header_csrf_token
            iv_value = io_csrf_handler->get_csrf_token( ) ).
        ENDIF.
      WHEN OTHERS.
        lv_csrf_token_valid     =     io_csrf_handler->validate_csrf_token(
lv_csrf_token ).
        IF lv_csrf_token_valid <> abap_true.
          io_response->set_header_field(
            iv_name  = if_rest_request=>gc_header_csrf_token
            iv_value = 'Required' ) ##no_text.
          RAISE EXCEPTION TYPE cx_rest_resource_exception
            EXPORTING
              status_code = cl_rest_status_code=>gc_client_error_forbidden
```

```
              textid                                                    =
cx_rest_resource_exception=>csrf_token_validation_failed.

        ENDIF.

    ENDCASE.

*     .

  ENDMETHOD.




*  <SIGNATURE>-------------------------------------------------------------
---------------------------+

*       |       Instance       Public       Method       ZCL_INF_TRABAJADOR2-
>IF_REST_APPLICATION~GET_ROOT_HANDLER

*  +------------------------------------------------------------------------
---------------------------+

* | [<-()] RO_ROOT_HANDLER                TYPE REF TO IF_REST_HANDLER

*  +-----------------------------------------------------------------------
----------------</SIGNATURE>

  METHOD IF_REST_APPLICATION~GET_ROOT_HANDLER.


    DATA lo_handler TYPE REF TO cl_rest_router.

    CREATE OBJECT lo_handler.

*        lo_handler->attach(  iv_template  =  '/test'  iv_handler_class  =
'ZCL_INF_TRABAJADOR_RESOURCE' ).

    lo_handler->attach(  iv_template  =  '/{PAR1}'  iv_handler_class  =
'ZCL_INF_TRABAJADOR_RESOURCE2' ).

    ro_root_handler = lo_handler.
```

```
  ENDMETHOD.

ENDCLASS.
```

4.6.5 Estructura ZEPP_S_MATE

Estructura: ZEPP_S_MATE — activo

Descripción breve: Materiales

Atributos | Componentes | Ayuda p./Verif.entr. | Campos de moneda/cantidad

Tipo instalado — 1 / 9

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Sist.coord.	Descripción breve
MATNR	Type	MATNR	CHAR	40	0	0	Número de material
MAKTX	Type	MAKTX	CHAR	40	0	0	Texto breve de material
MATKL	Type	MATKL	CHAR	9	0	0	Grupo de artículos
MEINS	Type	MEINS	UNIT	3	0	0	Unidad de medida base
MSTAE	Type	MSTAE	CHAR	2	0	0	Status material para todos los centros
MMSTA	Type	MMSTA	CHAR	2	0	0	Status material específico centro
WERKS	Type	WERKS_D	CHAR	4	0	0	Centro
LGORT	Type	LGORT_D	CHAR	4	0	0	Almacén
LABST	Type	LABST	QUAN	13	3	0	Stock valorado de libre utilización

4.6.6 Estructura ZEPP_S_MOVI

Estructura: ZEPP_S_MOVI — activo

Descripción breve: Movimientos

Atributos | Componentes | Ayuda p./Verif.entr. | Campos de moneda/cantidad

Tipo instalado — 1 / 14

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Sist.coord.	Descripción breve
RSNUM	Type	RSNUM	NUMC	10	0	0	Número de la reserva/las necesidades secundarias
RSPOS	Type	RSPOS	NUMC	4	0	0	Número de posición de reserva/necesidades secundarias
BDMNG	Type	BDMNG	QUAN	13	3	0	Cantidad necesaria
MATNR	Type	MATNR	CHAR	40	0	0	Número de material
MBLNR	Type	MBLNR	CHAR	10	0	0	Número de documento de material
MJAHR	Type	MJAHR	NUMC	4	0	0	Ejercicio del documento de material
ZEILE	Type	MBLPO	NUMC	4	0	0	Posición en documento de material
BUDAT	Type	DATE	CHAR	8	0	0	Fecha con formato CHAR
LABST	Type	LABST	QUAN	13	3	0	Stock valorado de libre utilización
MEINS	Type	MEINS	UNIT	3	0	0	Unidad de medida base
FIRMA	Type	ALPFSTRING	STRING	0	0	0	String de longitud variable
KZEAR	Type	KZEAR	CHAR	1	0	0	Salida final de la reserva
XLOEK	Type	XLOEK	CHAR	1	0	0	Posición borrada
WEMPF	Type	WEMPF	CHAR	12	0	0	Destinatario de mercancías

4.6.7 Estructura ZEPP_S_NEWR

Estructura: ZEPP_S_NEWR — activo

Descripción breve: Nueva Reserva

Atributos | Componentes | Ayuda p./Verif.entr. | Campos de moneda/cantidad

Tipo instalado — 1 / 15

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Sist.coord.	Descripción breve
RSNUM	Type	RSNUM	NUMC	10	0	0	Número de la reserva/las necesidades secundarias
RSPOS	Type	RSPOS	NUMC	4	0	0	Número de posición de reserva/necesidades secundarias
MATNR	Type	MATNR	CHAR	40	0	0	Número de material
MAKTX	Type	MAKTX	CHAR	40	0	0	Texto breve de material
MEINS	Type	MEINS	UNIT	3	0	0	Unidad de medida base
WERKS	Type	WERKS_D	CHAR	4	0	0	Centro
LGORT	Type	LGORT_D	CHAR	4	0	0	Almacén
BDMNG	Type	BDMNG	QUAN	13	3	0	Cantidad necesaria
BDTER	Type	DATE	CHAR	8	0	0	Fecha con formato CHAR
KOSTL	Type	KOSTL	CHAR	10	0	0	Centro de coste
WEMPF	Type	WEMPF	CHAR	12	0	0	Destinatario de mercancías
MSG_TYPE	Type	BAPI_MTYPE	CHAR	1	0	0	Tipo mensaje: S Success, E Error, W Warning, I Info A Abort
MSG_ID	Type	SYMSGID	CHAR	20	0	0	Clase de mensajes
MSG_NUMBER	Type	SYMSGNO	NUMC	3	0	0	Número de mensaje
MSG_TEXT	Type	BAPI_MSG	CHAR	220	0	0	Texto de mensaje

4.6.8 Estructura ZEPP_S_RESE

Estructura: ZEPP_S_RESE — activo

Descripción breve: Reserva

Atributos | Componentes | Ayuda p./Verif.entr. | Campos de moneda/cantidad

Tipo instalado — 1 / 14

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Sist.coord.	Descripción breve
RSNUM	Type	RSNUM	NUMC	10	0	0	Número de la reserva/las necesidades secundarias
RSPOS	Type	RSPOS	NUMC	4	0	0	Número de posición de reserva/necesidades secundarias
BDMNG	Type	BDMNG	QUAN	13	3	0	Cantidad necesaria
MEINS	Type	MEINS	UNIT	3	0	0	Unidad de medida base
MATNR	Type	MATNR	CHAR	40	0	0	Número de material
MBLNR	Type	MBLNR	CHAR	10	0	0	Número de documento de material
MJAHR	Type	MJAHR	NUMC	4	0	0	Ejercicio del documento de material
ZEILE	Type	MBLPO	NUMC	4	0	0	Posición en documento de material
BUDAT	Type	DATE	CHAR	8	0	0	Fecha con formato CHAR
LABST	Type	LABST	QUAN	13	3	0	Stock valorado de libre utilización
FIRMA	Type	ALPFSTRING	STRING	0	0	0	String de longitud variable
KZEAR	Type	KZEAR	CHAR	1	0	0	Salida final de la reserva
XLOEK	Type	XLOEK	CHAR	1	0	0	Posición borrada
WEMPF	Type	WEMPF	CHAR	12	0	0	Destinatario de mercancías

4.6.9 Estructura: ZES_INF_OBREROS2

Estructura: ZES_INF_EMPLEADOS2 activo

Descripción breve: Estructura para enviar información de empleados WS

Atributos | Componentes | Ayuda p./Verif.entr. | Campos de moneda/cantidad

Tipo instalado 1 / 77

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve	Gr
EMP_CODIGO	Type		CHAR	6	0	Código de empleado	
EMP_APELLIDO1	Type		CHAR	40	0	Primer Apellido	
EMP_APELLIDO2	Type		CHAR	40	0	Segundo Apellido	
NOMBRE	Type		CHAR	40	0	Nombre	
EMP_NOMBRE_COMP...	Type		CHAR	150	0	Nombre Completo	
EMP_SOCIEDAD	Type		CHAR	4	0	Sociedad	
NOMBRE_SOCIEDAD	Type		CHAR	40	0	Nombre sociedad	
EMP_PAIS_SOCIED...	Type		CHAR	40	0	Nombre de país de la empresa	
EMP_CIUDAD_SOCI...	Type		CHAR	40	0	Nombre de la ciudad de la sociedad	
EMP_LOCALIDAD	Type		CHAR	2	0	Código de la localidad	
NOMBRE_LOCALIDAD	Type		CHAR	40	0	Nombre de la localidad	
EMP_GERENCIA	Type		CHAR	40	0	Código de la gerencia(Unidad de negocio)	

4.6.10 Estructura ZES_INF_EMPLEADOS2

Estructura: ZES_INF_OBREROS2 activo

Descripción breve: Estructura para enviar información de Obreros WS

Atributos | Componentes | Ayuda p./Verif.entr. | Campos de moneda/cantidad

Tipo instalado 1 / 73

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve	Grupo
EMP_CODIGO	Type		CHAR	6	0	Código de empleado	
EMP_APELLIDO1	Type		CHAR	40	0	Primer Apellido	
EMP_APELLIDO2	Type		CHAR	40	0	Segundo Apellido	
NOMBRE	Type		CHAR	40	0	Nombre	
EMP_NOMBRE_COMP...	Type		CHAR	150	0	Nombre Completo	
EMP_SOCIEDAD	Type		CHAR	4	0	Sociedad	
NOMBRE_SOCIEDAD	Type		CHAR	40	0	Nombre sociedad	
EMP_PAIS_SOCIED...	Type		CHAR	40	0	Nombre de país de la empresa	
EMP_CIUDAD_SOCI...	Type		CHAR	40	0	Nombre de la ciudad de la sociedad	
EMP_LOCALIDAD	Type		CHAR	2	0	Código de la localidad	
NOMBRE_LOCALIDAD	Type		CHAR	40	0	Nombre de la localidad	