Imagen que contiene exterior, luz

Descripción generada automáticamente

Propuesta Técnico/Económica

A

DISTRITO DE CALI – VALLE DEL CAUCA

DOCUMENTO TÉCNICO DE INTEROPERABILIDAD ENTRE LA SOLUCION TECNOLÓGICA PARA LA GESTIÓN DE CATASTRO Y EL SISTEMA DE GESTIÓN ADMINISTRATIVO FINANCIERO TERRITORIAL (SISTEMA SAP) DEL DEPARTAMENTO ADMINISTRATIVO DE HACIENDA

|  |
| --- |
| **Sistema de Información:** |
| **SAP versión 6.07** |

mayo de 2022

[1. PRESENTACIÓN 2](#_Toc102746853)

[1.1. Cinteli Colombia 3](#_Toc102746854)

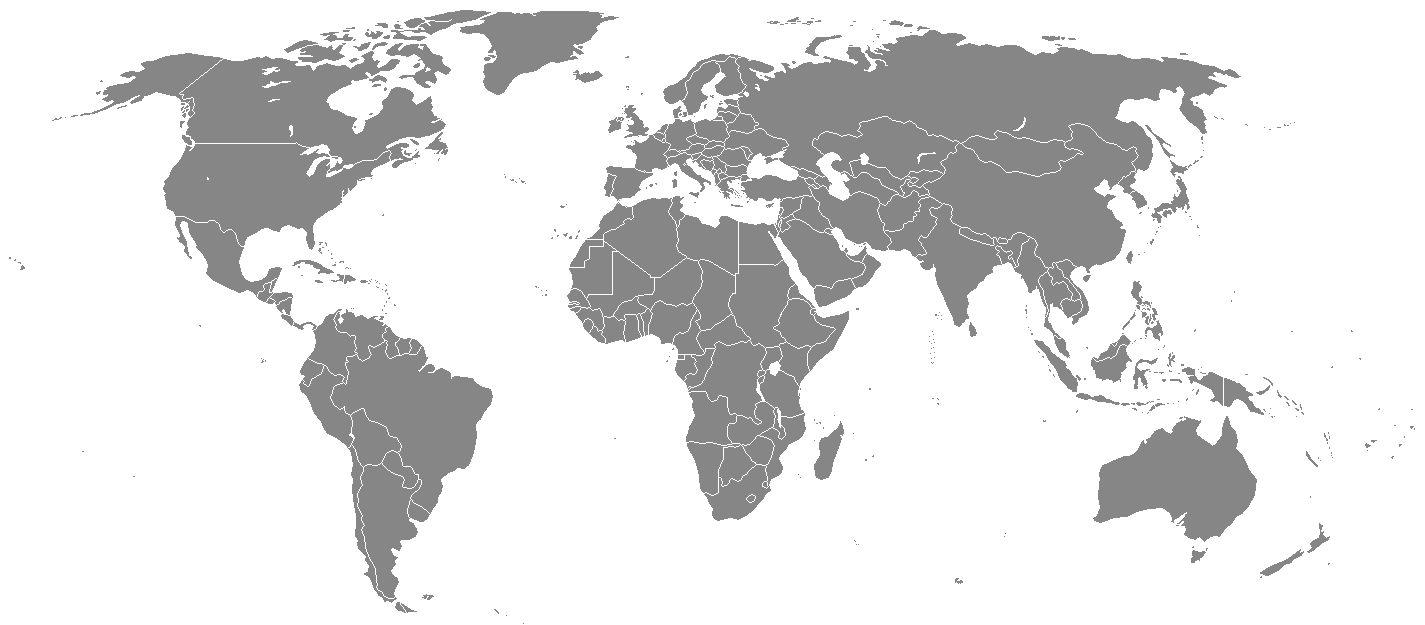
[1.2. Nuestros Servicios 3](#_Toc102746855)

[1.1. Experiencia 4](#_Toc102746856)

[2. DOCUMENTO TÉCNICO 5](#_Toc102746857)

# PRESENTACIÓN

*CINTELI es una empresa colombiana de consultoría estratégica especializada en apoyar a la transformación de las principales ciudades de América en metrópolis eficientes mediante la implantación de productos innovadores y soluciones inteligentes. Para lograr ese objetivo tenemos alianzas con socios tecnológicos para los insumos de alta calidad. Estas empresas son pioneras en los mercados internacionales, con una amplia experiencia en sus sectores, trasladando sus conocimientos a cada uno de nuestros proyectos de Smart City.*



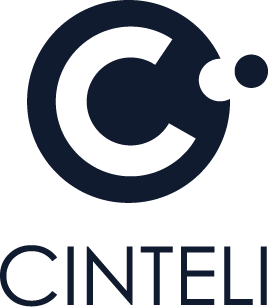
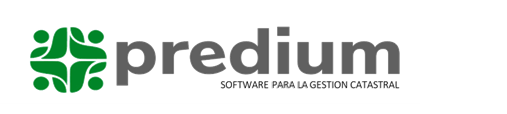


Ilustración 1 - Presencia Cinteli

## Cinteli Colombia

La sociedad CINTELI COLOMBIA S.A.S.es una sociedad anónima por acciones simplificada, cuyo objeto principal es ofrecer *data solutions* en prestación de servicios, desarrollo de tecnología, producción y puesta en marcha de sistemas de información territorial multipropósito para gobiernos y operadores de infraestructuras.

Actualmente CINTELI cuenta con oficinas en Barcelona (España), Panamá y Barranquilla (Colombia) y un recurso humano interdisciplinario idóneo, con la experticia necesaria para la prestación del servicio objeto de la presente propuesta, lograda a través de la puesta en marcha, ejecución y *delivery* de procesos masivos de actualización catastral con especificaciones IGAC[[1]](#footnote-1) y tecnología de punta que asegura los más altos estándares de calidad en cada uno de los procesos que desarrolla.

## Nuestros Servicios

CINTELI actualiza y gestiona diversos de modelos de datos y procesos con carácter multifinalitarios, tales como:

* Catastros convencionales, multipropósito y multifinalitarios
* Generación, procesamiento y análisis de cartografías temáticas
* Reconocimiento predial y padrones
* Transcripciones parcelarias
* Estratificación de ciudades
* Estudio, diseño y puesta en marcha de aplicaciones para el fortalecimiento del recaudo público
* Geocoding masivo
* Catálogos e inventarios urbanos (comerciales, inquilinatos, equipamientos, servicios, subsuelo, mobiliario urbano, señalética y lumínica)
* Toponimia
* Generalizaciones multiescala
* Correlación y analítica de datos (Big Data / Data Science)
* Linaje de los datos
* Diseño de aplicaciones espaciales para geolocalización de elementos de acuerdo a las necesidades territoriales o necesidades específicas del cliente



Ilustración 2 - Catálogo Cinteli

## 

## Experiencia

La sociedad CINTELI COLOMBIA S.A.S. es una sociedad anónima por acciones simplificada, cuyo objeto principal es ofrecer *data solutions* en prestación de servicios, desarrollo de tecnología, producción y puesta en marcha de sistemas de información territorial multipropósito para gobiernos y operadores de infraestructuras.

Logotipo

Descripción generada automáticamenteLogotipo

Descripción generada automáticamenteDespués de más de 80 años de exitosa actividad empresarial, desde 2012 nuestra matriz Infraplan-Group traslada todo su *know-how* y *expertise* a Cinteli Colombia, para el desarrollo de geo-soluciones Smart City en Colombia. En efecto, dentro de la experiencia que podemos destacar se encuentra la suscripción, ejecución y terminación de dos contratos de prestación de servicios suscritos con el Distrito de Barranquilla de apoyo a la gestión de la Secretaría de Hacienda en la identificación, clasificación, organización y georreferenciación de los sujetos pasivos del impuesto de industria y comercio (012018006169 y 012019003299), uno con el Área Metropolitana de Barranquilla de prestación de servicios de apoyo a la gestión para brindar acompañamiento en la labor técnica de identificación, clasificación, organización y georreferenciación de los sujetos pasivos del impuesto de industria y comercio en los municipios que conforman el Área Metropolitana de Barranquilla (116-2019), los cuales se encuentran debidamente publicados en la página del SECOP, así como con otras ciudades (Barcelona y Santa Marta).





# Imagen que contiene exterior, luz Descripción generada automáticamenteDOCUMENTO TÉCNICO DE INTEROPERABILIDAD ENTRE LA SOLUCION TECNOLÓGICA PARA LA GESTIÓN DE CATASTRO Y EL SISTEMA DE GESTIÓN ADMINISTRATIVO FINANCIERO TERRITORIAL (SISTEMA SAP) DEL DEPARTAMENTO ADMINISTRATIVO DE HACIENDA.

|  |
| --- |
| **Sistema de Información:** |
| **SAP versión 6.07** |

Se hace entrega de código de la BAPI con las funcionalidades requerida según requerimiento.

* Crear y Transferir al Sistema de Gestión Administrativo Financiero Territorial – SAP de la Alcaldía del Distrito de Santiago de Cali – Vertical Tax and Revenue Management de SAP de forma natural los datos maestros del objeto contrato “Predio” a la cuenta contrato “Impuesto Predial”, al Business Partner “Contribuyente”.

Para la creación de Business Partner “Contribuyente” se utiliza la bapi “BAPI\_BUPA\_CREATE\_FROM\_DATA”.

Para la creación de la cuenta contrato “Impuesto Predial”, se utiliza la bapi “BAPI\_CTRACCOUNT\_EASYCREATE”.

Para la creación de objeto contrato “Predio” se utiliza la bapi “BAPI\_CTRACPSOBJECT\_EASYCREATE”.

Validar que tipo de identificación tiene el contribuyente.

Si es NIT, se completa el tipo de IC como “2” Organización, se debe validar el digito de verificación.

partnercategory = '2'.

Completar los datos de organización.

centraldataorgan-name1 = partner-firstname.  
centraldataorgan-name2 = partner-lastname.   
centraldata-searchterm1 = partner-firstname.  
centraldata-partnertype = ‘PJPR’.  
  
Validar el dígito de verificación

stcd1 = is\_partner-id\_number.  
 CALL FUNCTION 'TAX\_NUMBER\_CHECK'  
 EXPORTING  
 country = 'CO'  
 natural\_person\_flag = ' '  
 tax\_code\_1 = stcd1  
 EXCEPTIONS  
 not\_valid = 1  
 different\_fprcd = 2  
 OTHERS = 3.

Si no es NIT entonces completar el tipo de IC como “1” Persona.

partnercategory = '1'.

centraldataperson-firstname = partner-firstname.  
centraldataperson-lastname = partner-lastname.

CONCATENATE firstname lastname  
 INTO centraldata-searchterm1 SEPARATED BY space.

Completar como personal natural.

centraldata-partnertype = ‘PNAT’.

Completar la estructura centraldata.

Completar partnergroup con “0001” por defecto Contribuyente.

Completar por defecto “0001” centraldata-title\_key .

Completar estructura de dirección.  
addressdata-street = partner-street.  
addressdata-country = partner-country.  
addressdata-region = partner-region.  
addressdata-city\_no = partner-city.

Completar las tablas de teléfono y correo  
lt\_telefondata.  
lt\_maildata

Para predial utilizar el tipo de dirección “ZESTABLECI”

addressusage-addresstype = 'ZESTABLECI'.  
addressusage-usagevalidto = '99991231235959'.  
addressusage\_x-usagevalidto = abap\_true.  
CONCATENATE sy-datlo '000000' INTO addressusage-usagevalidfrom.  
addressusage\_x-usagevalidfrom = abap\_true.  
addressusage-standardaddressusage = abap\_true.  
addressusage\_x-standardaddressusage = abap\_true.  
  
IF addressusage-addresstype IS NOT INITIAL.  
 addressusage\_x-addresstype = abap\_true.  
  
 CLEAR addressusage-validtoreadforchange.  
 addressusage\_x-updateflag = 'I'.  
  
 APPEND addressusage TO lt\_addressusage.  
 APPEND addressusage\_x TO lt\_addressusage\_x.  
ENDIF.

CALL FUNCTION 'BAPI\_BUPA\_CREATE\_FROM\_DATA'  
 EXPORTING  
 partnercategory = partnercategory  
 partnergroup = partnergroup  
 centraldata = centraldata  
 centraldataperson = centraldataperson  
 centraldataorganization = centraldataorganization  
 addressdata = addressdata  
 IMPORTING  
 businesspartner = businesspartner  
 TABLES  
 telefondata = lt\_telefondata  
 e\_maildata = lt\_maildata  
 faxdata = lt\_faxdata  
 return = lt\_return.

Si se crea correctamente el IC se realiza el commit para que se cree los datos en las tablas de SAP, luego se procede a actualizar la dirección

CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
 EXPORTING  
 wait = 'X'.  
  
 REFRESH lt\_return.  
 CALL FUNCTION 'BAPI\_BUPA\_ADDRESS\_CHANGE'  
 EXPORTING  
 businesspartner = partner  
 TABLES  
 addressusage = lt\_addressusage  
 addressusage\_x = lt\_addressusage\_x  
 return = lt\_return*.*

* Actualizar al Sistema de Gestión Administrativo Financiero Territorial – SAP de la Alcaldía del Distrito de Santiago de Cali – Vertical Tax and Revenue Management de SAP de forma natural los datos maestros del objeto contrato “Predio” a la cuenta contrato “Impuesto Predial”, al Business Partner “Contribuyente” de forma natural los datos maestros, contribuyentes, cuenta contrato y el objeto contrato.

Para modificar Business Partner “Contribuyente” se utiliza la bapi “BAPI\_BUPA\_CENTRAL\_CHANGE”.

Para modificar la cuenta contrato “Impuesto Predial”, se utiliza la bapi “BAPI\_CTRACCOUNT\_EASYCREATE”.

Para modificar el objeto contrato “Predio” se utiliza la bapi “BAPI\_CTRACPSOBJECT\_CHANGE”.

CASE is\_partner-id\_type.  
 WHEN 'NIT'.  
 lv\_tbp = abap\_false.  
 WHEN OTHERS.  
 lv\_tbp = abap\_true.  
 ENDCASE.  
  
 CALL FUNCTION 'BAPI\_BUPA\_CENTRAL\_GETDETAIL'  
 EXPORTING  
 businesspartner = iv\_partner  
 IMPORTING  
 centraldata = ls\_centraldata  
 centraldataperson = ls\_centraldataperson  
 centraldataorganization = ls\_centraldataorgan.  
  
 CALL FUNCTION 'BAPI\_BUPA\_ADDRESS\_GETDETAIL'  
 EXPORTING  
 businesspartner = iv\_partner  
 IMPORTING  
 addressdata = ls\_addressdata  
 TABLES  
 bapiadtel = lt\_telefondata  
 bapiadfax = lt\_faxdata  
 bapiadsmtp = lt\_maildata  
 addressusage = lt\_addressusage.  
  
  
 CLEAR ls\_centraldataperson\_x.  
  
 *"Determina si es persona natural o jurídica*  
 CASE lv\_tbp.  
 WHEN abap\_true.  
 IF ls\_centraldataperson-firstname NE is\_partner-firstname.  
 ls\_centraldataperson-firstname = is\_partner-firstname.  
 ls\_centraldataperson\_x-firstname = abap\_true.  
 ENDIF.  
 IF ls\_centraldataperson-lastname NE is\_partner-lastname.  
 ls\_centraldataperson-lastname = is\_partner-lastname.  
 ls\_centraldataperson\_x-lastname = abap\_true.  
 ENDIF.  
  
 IF ls\_centraldataperson\_x-firstname IS NOT INITIAL OR  
 ls\_centraldataperson\_x-lastname IS NOT INITIAL.  
 CONCATENATE is\_partner-firstname is\_partner-lastname  
 INTO ls\_centraldata-searchterm1 SEPARATED BY space.  
 ls\_centraldata\_x-searchterm1 = abap\_true.  
 ENDIF.  
  
 IF ls\_centraldata-partnertype NE gc\_ot-pnat.  
 ls\_centraldata\_x-partnertype = abap\_true.  
 ls\_centraldata-partnertype = gc\_ot-pnat.  
 ENDIF.  
 WHEN OTHERS.  
 IF ls\_centraldataorgan-name1 NE is\_partner-firstname  
 AND is\_partner-firstname IS NOT INITIAL.  
 ls\_centraldataorgan-name1 = is\_partner-firstname.  
 ls\_centraldataorgan\_x-name1 = abap\_true.  
 ls\_centraldata-searchterm1 = is\_partner-firstname.  
 ls\_centraldata\_x-searchterm1 = abap\_true.  
 ENDIF.  
 IF ls\_centraldataorgan-name2 NE is\_partner-lastname  
 AND is\_partner-lastname IS NOT INITIAL.  
 ls\_centraldataorgan-name2 = is\_partner-lastname.  
 ls\_centraldataorgan\_x-name2 = abap\_true.  
 ENDIF.  
  
 IF ls\_centraldata-partnertype NE gc\_ot-pjpr.  
 ls\_centraldata\_x-partnertype = abap\_true.  
 ls\_centraldata-partnertype = gc\_ot-pjpr.  
 ENDIF.  
 ENDCASE.  
  
 lv\_partnergroup = gc\_ot-group.  
 IF ls\_centraldata-title\_key NE gc\_ot-title\_key.  
 ls\_centraldata-title\_key = gc\_ot-title\_key.  
 ls\_centraldata\_x-title\_key = abap\_true.  
 ENDIF.  
  
 *"Dirección*  
 IF ls\_addressdata-street NE is\_partner-street  
 AND is\_partner-street IS NOT INITIAL.  
 ls\_addressdata-street = is\_partner-street.  
 ls\_addressdata\_x-street = abap\_true.  
 ENDIF.  
  
 IF ls\_addressdata-country NE is\_partner-country  
 AND is\_partner-country IS NOT INITIAL.  
 ls\_addressdata-country = is\_partner-country.  
 ls\_addressdata\_x-country = abap\_true.  
 ENDIF.  
  
 IF ls\_addressdata-region NE is\_partner-region  
 AND is\_partner-region IS NOT INITIAL.  
 ls\_addressdata-region = is\_partner-region.  
 ls\_addressdata\_x-region = abap\_true.  
 ENDIF.  
  
 CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
 EXPORTING  
 input = is\_partner-city  
 IMPORTING  
 output = lv\_city\_code.  
  
 IF ls\_addressdata-city\_no NE lv\_city\_code  
 AND is\_partner-city IS NOT INITIAL.  
 ls\_addressdata-city\_no = is\_partner-city.  
 ls\_addressdata\_x-city\_no = abap\_true.  
  
 SELECT SINGLE city\_name INTO ls\_addressdata-city FROM v\_adrcity  
 WHERE langu = gc\_ot-spanish  
 AND country = ls\_addressdata-country  
 AND city\_code = lv\_city\_code  
 AND region = ls\_addressdata-region.  
  
 ls\_addressdata\_x-city = abap\_true.  
 ENDIF.  
  
  
 *"Teléfonos*  
 REFRESH lt\_telefondata\_new.  
 DESCRIBE TABLE lt\_telefondata LINES lv\_lines.  
 *"Valida fijo*  
 IF is\_partner-telf1 IS NOT INITIAL.  
 READ TABLE lt\_telefondata INTO ls\_telefondata WITH KEY r\_3\_user = space.  
 IF sy-subrc NE 0.  
 READ TABLE lt\_telefondata INTO ls\_telefondata WITH KEY r\_3\_user = '1'.  
 ENDIF.  
 IF sy-subrc NE 0.  
 IF is\_partner-telf1 IS NOT INITIAL.  
 ADD 1 TO lv\_lines.  
 CLEAR ls\_telefondata.  
 CLEAR ls\_telefondata\_x.  
  
 ls\_telefondata-country = is\_partner-country.  
 ls\_telefondata-std\_no = abap\_true.  
 ls\_telefondata-telephone = is\_partner-telf1.  
 ls\_telefondata-extension = space.  
 ls\_telefondata-tel\_no = is\_partner-telf1.  
 ls\_telefondata-home\_flag = abap\_true.  
 ls\_telefondata-consnumber = lv\_lines.  
 APPEND ls\_telefondata TO lt\_telefondata\_new.  
 ls\_telefondata\_x-country = abap\_true.  
 ls\_telefondata\_x-std\_no = abap\_true.  
 ls\_telefondata\_x-telephone = abap\_true.  
 ls\_telefondata\_x-extension = abap\_true.  
 ls\_telefondata\_x-tel\_no = abap\_true.  
 ls\_telefondata\_x-home\_flag = abap\_true.  
 ls\_telefondata\_x-consnumber = abap\_true.  
 ls\_telefondata\_x-updateflag = 'I'.  
 APPEND ls\_telefondata\_x TO lt\_telefondata\_x.  
 ENDIF.  
  
 ELSEIF ls\_telefondata-telephone NE is\_partner-telf1.*"Si lo encuentra valida si se actualiza*  
  
 ls\_telefondata-telephone = is\_partner-telf1.  
 ls\_telefondata-tel\_no = is\_partner-telf1.  
 APPEND ls\_telefondata TO lt\_telefondata\_new.  
 CLEAR ls\_telefondata\_x.  
 ls\_telefondata\_x-telephone = abap\_true.  
 ls\_telefondata\_x-tel\_no = abap\_true.  
 ls\_telefondata\_x-updateflag = 'U'.  
 APPEND ls\_telefondata\_x TO lt\_telefondata\_x.  
 ENDIF.  
 ENDIF.  
  
 *"Valida Movil*  
 IF is\_partner-mobile IS NOT INITIAL.  
 READ TABLE lt\_telefondata INTO ls\_telefondata WITH KEY r\_3\_user = '3'.  
 IF sy-subrc NE 0.  
 IF is\_partner-telf1 IS NOT INITIAL.  
 ADD 1 TO lv\_lines.  
 CLEAR ls\_telefondata.  
 CLEAR ls\_telefondata\_x.  
  
 ls\_telefondata-country = is\_partner-country.  
 ls\_telefondata-std\_no = abap\_true.  
 ls\_telefondata-telephone = is\_partner-mobile.  
 ls\_telefondata-extension = space.  
 ls\_telefondata-tel\_no = is\_partner-mobile.  
 ls\_telefondata-r\_3\_user = '3'.  
 ls\_telefondata-home\_flag = space.  
 ls\_telefondata-consnumber = lv\_lines.  
 APPEND ls\_telefondata TO lt\_telefondata\_new.  
 ls\_telefondata\_x-country = abap\_true.  
 ls\_telefondata\_x-std\_no = abap\_true.  
 ls\_telefondata\_x-telephone = abap\_true.  
 ls\_telefondata\_x-extension = abap\_true.  
 ls\_telefondata\_x-tel\_no = abap\_true.  
 ls\_telefondata\_x-r\_3\_user = abap\_true.  
 ls\_telefondata\_x-home\_flag = abap\_true.  
 ls\_telefondata\_x-consnumber = abap\_true.  
 ls\_telefondata\_x-updateflag = 'I'.  
 APPEND ls\_telefondata\_x TO lt\_telefondata\_x.  
 ENDIF.  
  
 ELSEIF ls\_telefondata-telephone NE is\_partner-mobile.*"Si lo encuentra valida si se actualiza*  
  
 ls\_telefondata-telephone = is\_partner-mobile.  
 ls\_telefondata-tel\_no = is\_partner-mobile.  
 APPEND ls\_telefondata TO lt\_telefondata\_new.  
 CLEAR ls\_telefondata\_x.  
 ls\_telefondata\_x-telephone = abap\_true.  
 ls\_telefondata\_x-tel\_no = abap\_true.  
 ls\_telefondata\_x-updateflag = 'U'.  
 APPEND ls\_telefondata\_x TO lt\_telefondata\_x.  
 ENDIF.  
 ENDIF.  
  
  
 *"Correo electrónico*  
 IF is\_partner-email IS NOT INITIAL.  
 DESCRIBE TABLE lt\_maildata LINES lv\_lines.  
 READ TABLE lt\_maildata INTO ls\_maildata WITH KEY consnumber = '001'.  
 IF sy-subrc NE 0.  
 ADD 1 TO lv\_lines.  
 CLEAR ls\_maildata.  
 ls\_maildata-std\_no = abap\_true.  
 ls\_maildata-e\_mail = is\_partner-email.  
 ls\_maildata-std\_recip = abap\_true.  
 ls\_maildata-home\_flag = abap\_true.  
 ls\_maildata-consnumber = lv\_lines.  
 APPEND ls\_maildata TO lt\_maildata\_new.  
  
 ls\_maildata\_x-std\_no = abap\_true.  
 ls\_maildata\_x-e\_mail = abap\_true.  
 ls\_maildata\_x-std\_recip = abap\_true.  
 ls\_maildata\_x-home\_flag = abap\_true.  
 ls\_maildata\_x-consnumber = abap\_true.  
 ls\_maildata\_x-updateflag = 'I'.  
  
 APPEND ls\_maildata\_x TO lt\_maildata\_x.  
  
  
 ELSEIF is\_partner-email NE ls\_maildata-e\_mail.  
 ls\_maildata-e\_mail = is\_partner-email.  
 APPEND ls\_maildata TO lt\_maildata\_new.  
 ls\_maildata\_x-e\_mail = abap\_true.  
 ls\_maildata\_x-updateflag = 'U'.  
 APPEND ls\_maildata\_x TO lt\_maildata\_x.  
  
 ENDIF.  
 ENDIF.  
  
 *"Fax*  
 IF is\_partner-telfx IS NOT INITIAL.  
 DESCRIBE TABLE lt\_faxdata LINES lv\_lines.  
 READ TABLE lt\_faxdata INTO ls\_faxdata WITH KEY consnumber = '001'.  
 IF sy-subrc NE 0.  
 ADD 1 TO lv\_lines.  
  
 ls\_faxdata-country = is\_partner-country.  
 ls\_faxdata-std\_no = abap\_true.  
 ls\_faxdata-fax = is\_partner-telfx.  
 ls\_faxdata-extension = space.  
 ls\_faxdata-fax\_no = is\_partner-telfx.  
 ls\_faxdata-home\_flag = abap\_true.  
 ls\_faxdata-consnumber = lv\_lines.  
  
 APPEND ls\_faxdata TO lt\_faxdata\_new.  
  
 ls\_faxdata\_x-country = abap\_true.  
 ls\_faxdata\_x-std\_no = abap\_true.  
 ls\_faxdata\_x-fax = abap\_true.  
 ls\_faxdata\_x-extension = abap\_true.  
 ls\_faxdata\_x-fax\_no = abap\_true.  
 ls\_faxdata\_x-home\_flag = abap\_true.  
 ls\_faxdata\_x-consnumber = abap\_true.  
 ls\_faxdata\_x-updateflag = 'I'.  
 APPEND ls\_faxdata\_x TO lt\_faxdata\_x.  
  
 ELSEIF is\_partner-telfx NE ls\_faxdata-fax.  
 ls\_faxdata-fax = is\_partner-telfx.  
 ls\_faxdata-fax\_no = is\_partner-telfx.  
 ls\_faxdata-home\_flag = abap\_true.  
 APPEND ls\_faxdata TO lt\_faxdata\_new.  
 ls\_faxdata\_x-country = abap\_true.  
 ls\_faxdata\_x-std\_no = abap\_true.  
 ls\_faxdata\_x-fax = abap\_true.  
 ls\_faxdata\_x-extension = abap\_true.  
 ls\_faxdata\_x-fax\_no = abap\_true.  
 ls\_faxdata\_x-home\_flag = abap\_true.  
 ls\_faxdata\_x-consnumber = abap\_true.  
 ls\_faxdata\_x-updateflag = 'U'.  
 APPEND ls\_faxdata\_x TO lt\_faxdata\_x.  
  
 ENDIF.  
 ENDIF.  
  
 *"Consulta la información de utilización de dirección*  
 REFRESH lt\_addressusage.  
 CLEAR ls\_addressusage.  
 CASE is\_partner-cont\_acct1.  
 WHEN '51' OR '52'.*"Predial o Valorización*  
 ls\_addressusage-addresstype = 'ZESTABLECI'.  
 WHEN '55'.*"Circulación y tránsito*  
 ls\_addressusage-addresstype = 'ZNOTIFIVEH'.  
 WHEN '58'.*"Código de Policia*  
 ls\_addressusage-addresstype = 'ZNOTIFICCP'.  
 WHEN '53' OR '54'.*"ICA o RETEICA*  
 ls\_addressusage-addresstype = 'ZNOTIFICAC'.  
 WHEN '59'.*"Procultura*  
 ls\_addressusage-addresstype = 'ZNOTIFICPC'.  
 WHEN '60'.*"Espectáculos públicos y procultura*  
 ls\_addressusage-addresstype = 'ZNOTIFICEP'.  
 WHEN '61'.*"Obras públicas*  
 ls\_addressusage-addresstype = 'ZNOTIFICCO'.  
 ENDCASE.  
 ls\_addressusage-usagevalidto = '99991231235959'.  
 ls\_addressusage\_x-usagevalidto = abap\_true.  
  
 CONCATENATE sy-datlo '000000' INTO ls\_addressusage-usagevalidfrom.  
 ls\_addressusage\_x-usagevalidfrom = abap\_true.  
  
 ls\_addressusage-standardaddressusage = abap\_true.  
 ls\_addressusage\_x-standardaddressusage = abap\_true.  
  
 IF ls\_addressusage-addresstype IS NOT INITIAL.  
 SELECT SINGLE \* INTO @DATA(ls\_but021\_fs)  
 FROM but021\_fs  
 WHERE partner = @iv\_partner  
 AND adr\_kind = @ls\_addressusage-addresstype.  
 IF sy-subrc NE 0.  
 ls\_addressusage\_x-addresstype = abap\_true.  
  
 CLEAR ls\_addressusage-validtoreadforchange.  
 ls\_addressusage\_x-updateflag = 'I'.  
  
 APPEND ls\_addressusage TO lt\_addressusage.  
 APPEND ls\_addressusage\_x TO lt\_addressusage\_x.  
 ENDIF.  
 ENDIF.  
  
 CALL FUNCTION 'BAPI\_BUPA\_CENTRAL\_CHANGE'  
 EXPORTING  
 businesspartner = iv\_partner  
 centraldata = ls\_centraldata  
 centraldataperson = ls\_centraldataperson  
 centraldataorganization = ls\_centraldataorgan  
 centraldata\_x = ls\_centraldata\_x  
 centraldataperson\_x = ls\_centraldataperson\_x  
 centraldataorganization\_x = ls\_centraldataorgan\_x  
  
 TABLES  
 return = lt\_return.  
  
 CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
 EXPORTING  
 wait = 'X'.  
  
 CALL FUNCTION 'BAPI\_BUPA\_ADDRESS\_CHANGE'  
 EXPORTING  
 businesspartner = iv\_partner  
*\* ADDRESSGUID =*  
 addressdata = ls\_addressdata  
 addressdata\_x = ls\_addressdata\_x  
*\* DUPLICATE\_MESSAGE\_TYPE =*  
*\* ACCEPT\_ERROR = ' '*  
 TABLES  
 bapiadtel = lt\_telefondata\_new  
 bapiadfax = lt\_faxdata\_new  
 bapiadsmtp = lt\_maildata\_new  
 addressusage = lt\_addressusage  
*\* bapiaduse =*  
 bapiadtel\_x = lt\_telefondata\_x  
 bapiadfax\_x = lt\_faxdata\_x  
 bapiadsmt\_x = lt\_maildata\_x  
 addressusage\_x = lt\_addressusage\_x  
*\* bapiaduse\_x =*  
 return = lt\_return  
*\* ADDRESSDUPLICATES =*  
 .  
  
 CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
 EXPORTING  
 wait = 'X'.

REFRESH dfacts\_t.  
 SELECT \*  
 INTO TABLE dfacts\_t  
 FROM dfacts  
 WHERE objid = <lfs\_nov\_det>-objeto\_contrato.  
 REFRESH lti\_return.  
 PERFORM obtener\_facts TABLES psobjectfacts dfacts\_b USING <lfs\_nov\_det>.  
  
 CLEAR: psobjectdetailout, psobjectaccountout.  
 REFRESH: lti\_return.  
 CALL FUNCTION 'BAPI\_CTRACPSOBJECT\_CHANGE'  
 EXPORTING  
 psobjectkey = <lfs\_nov\_det>-objeto\_contrato  
 xupdatetask = 'X'  
 TABLES  
 psobjectfacts = psobjectfacts  
 return = lti\_return.

* Actualizar las novedades del predio “objeto contrato” de manera natural en el Sistema de Gestión Administrativo Financiero Territorial – SAP de la Alcaldía del Distrito de Santiago de Cali – Vertical Tax and Revenue Management de SAP de forma natural relacionados con el objeto contrato “Predio”, la cuenta contrato “Impuesto Predial” y el Business Partner “Contribuyente”. Se deben considerar por lo menos las siguientes novedades:
  + Primera
  + Segunda
  + Desenglobes
  + Englobes
  + Revisiones de avaluó
  + Auto estimaciones
  + Terceras
  + Quintas nuevas
  + Quintas omitidas
  + Rectificación datos del predio
  + Rectificación datos del terreno
  + Rectificación datos de los propietarios
  + Rectificación datos de las construcciones
  + Complementación datos del predio
  + Complementación datos de los propietarios
  + Cancelaciones
  + Resolución conjunta Supernotariado y Registro SNR

Primero obtener el código de Interlocutor comercial.

SELECT SINGLE partner INTO partner  
 FROM but0id  
 WHERE idnumber = identificacion AND  
 type = clase\_identi.

Obtener los datos del contribuyente

SELECT SINGLE name\_first name\_last name\_org1  
 INTO (fullname, last, orgn)  
 FROM but000  
 WHERE partner = partner.  
 IF sy-subrc = 0.  
 CONCATENATE fullname last INTO fullname SEPARATED BY space.  
 ENDIF.

Obtener el objeto contrato y datos del propietario.

Se van acumulando todos los propietarios por cada registro que viene del detalle de novedades.

Se adicionan los nuevos valores a los predios.

predio-psobkey = objeto\_contrato.  
predio-vigencia = vigencia.  
predio-tip\_operacion = operacion.  
APPEND predio TO lti\_predio.

Validamos que existan los predios.

SELECT SINGLE psobkey INTO predio  
 FROM dpsob  
 WHERE psobkey = p\_predio.

Se procede a actualizar los datos de la BDT y también los datos históricos.

Consultar la última vigencia liquidada

SELECT \* INTO TABLE it\_imp  
 FROM zpr\_d\_imp WHERE  
 psobkey = i\_psobkey AND  
 zstatus = “C”   
 ORDER BY persl DESCENDING.  
  
 READ TABLE it\_imp INTO l\_wa\_imp INDEX 1.  
 l\_vig\_ant = l\_wa\_imp-persl.   
  
 CONCATENATE i\_persl '1231' INTO l\_fecfin\_vig.  
  
Valida que el predio tenga datos BDT

SELECT SINGLE persl INTO l\_vig\_actual  
 FROM zpr\_d\_bdt  
 WHERE psobkey = i\_psobkey.  
  
Consultar los datos históricos

SELECT \* INTO TABLE l\_ti\_historicos  
 FROM zpr\_d\_hist  
 WHERE vktyp = g\_c\_vktyp AND  
 obj\_id = i\_psobkey AND  
 obj\_name = g\_c\_tab\_bdt.

Validar si el predio existe, en caso de no existir se debe crear y actualizarle los valores históricos de la BDT.

Primero crear la cuenta contrato.

ctraccreateinfo-appl\_area = 'P'.   
bpartner = partner.   
ctraccreateinfo-buspartner = bpartner.

ctracpartnerdetail-line\_number = '0001'.  
ctracpartnerdetail-buspartner = bpartner.  
  
APPEND ctracpartnerdetail.

Crear la cuenta contrato desde la 'BAPI\_CTRACCOUNT\_EASYCREATE'.  
  
 CALL FUNCTION 'BAPI\_CTRACCOUNT\_EASYCREATE'  
 EXPORTING  
 ctraccreateinfo = lwa\_ctraccreateinfo  
 ctracdetail = lwa\_ctracdetail  
 IMPORTING  
 contractaccount = l\_contractaccount  
 TABLES  
 ctracpartnerdetail = lti\_ctracpartnerdetail  
 return = lti\_return.

Inicializar los valores de la BDT predial, con los valores históricos.

g\_read\_bdt = i\_read\_bdt.  
 g\_read\_distrib = i\_read\_distrib.  
 g\_read\_participa = i\_read\_participa.  
 g\_read\_exonera = i\_read\_exonera.  
  
 g\_num\_acto = i\_num\_acto.  
 g\_fch\_acto = i\_fch\_acto.  
  
Datos Cabecera Predio Ampliacion BDT

IF g\_read\_bdt IS INITIAL.  
 IF i\_bdt IS SUPPLIED.  
 wa\_datab = i\_bdt.  
 ENDIF.  
  
 IF i\_bdt\_old IS SUPPLIED.  
 gwa\_bdt\_old = i\_bdt\_old.  
 ENDIF.  
 ENDIF.  
Datos Distribución en Uso-Actividad  
 REFRESH: gti\_distribucion.  
 LOOP AT t\_distrib ASSIGNING <lfs\_distrib\_ori>.  
 APPEND INITIAL LINE TO gti\_distribucion ASSIGNING <lfs\_distrib\_des>.  
 MOVE-CORRESPONDING <lfs\_distrib\_ori> TO <lfs\_distrib\_des>.  
 ENDLOOP.  
  
Datos Exoneraciones  
 REFRESH: gti\_exoneracion.  
 LOOP AT t\_exonera ASSIGNING <lfs\_exonera\_ori>.  
 APPEND INITIAL LINE TO gti\_exoneracion ASSIGNING <lfs\_exonera\_des>.  
 MOVE-CORRESPONDING <lfs\_exonera\_ori> TO <lfs\_exonera\_des>.  
 ENDLOOP.  
  
Datos Propietarios  
 REFRESH: gti\_participa.  
 LOOP AT t\_propietarios ASSIGNING <lfs\_propiet\_ori>.  
 APPEND INITIAL LINE TO gti\_participa ASSIGNING <lfs\_propiet\_des>.  
 MOVE-CORRESPONDING <lfs\_propiet\_ori> TO <lfs\_propiet\_des>.  
 ENDLOOP.

Utilizar la BAPI 'BAPI\_CTRACPSOBJECT\_EASYCREATE'  
 psobjecttype = predio\_data-psobtyp.  
 psobjectkeyext = predio\_data-psobkey.  
 psobjectdetail-psobjecttext = predio\_data-psobtxt.  
 psobjectdetail-pslegacyobject = predio\_data-legacyobjk1.  
 psobjectdetail-authoritygroup = 'RE52'.

Se crea el objeto contrato  
 CALL FUNCTION 'BAPI\_CTRACPSOBJECT\_EASYCREATE'  
 EXPORTING  
 psobjecttype = psobjecttype  
 psobjectkeyext = psobjectkeyext  
 psobjectdetail = psobjectdetail  
 IMPORTING  
 psobjectkey = psobjkey  
 TABLES  
 PSOBJECTFACTS = T\_PSOBJECTFACTS  
 return = lti\_return.

Si la creación es correcta se procede a realizar el commit en tablas.

CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
 EXPORTING  
 wait = 'X'.

Actualizar también la tabla de uso distribución zpr\_d\_distrib.

Se procede a actualizar las facts del objeto contrato.

SELECT SINGLE \* INTO dfacts\_t  
 FROM dfacts WHERE  
 objid = objeto\_contrato AND  
 fact\_type\_seq = ‘0004’ AND  
 fact\_cat\_seq = ‘06’ AND  
 fact\_category = ‘0104’.  
  
 LOOP AT psobjectfacts INTO wa\_psobjectfacts WHERE  
 CONTRACT\_OBJECT\_KEY = p\_wa\_nov\_det-objeto\_contrato AND  
 fact\_type\_sequence = 0004 AND  
 fact\_category\_sequence = 06 AND  
 split\_date\_new = 00010101 AND  
 valid\_to\_date\_key = '99991231' AND  
 valid\_to\_time\_key = '99991231'  
 EXIT.  
 ENDLOOP.  
  
 IF dfacts\_t IS NOT INITIAL AND psobjectfacts IS NOT INITIAL.  
  
 LOOP AT gti\_predio\_conex INTO wa\_predio\_conex.  
 lwa\_dfacts\_t-value\_generic = wa\_predio\_conex-id\_predio\_conex.  
 ADD 1 TO lwa\_dfacts\_t-FACT\_TYPE\_REP.  
 wa\_psobjectfacts-value\_generic = wa\_predio\_conex-id\_predio\_conex.  
 ADD 1 TO wa\_psobjectfacts-fact\_type\_repetition.  
  
 APPEND lwa\_dfacts\_t TO dfacts\_t.  
 APPEND wa\_psobjectfacts TO psobjectfacts.  
 ENDLOOP.  
  
 MODIFY dfacts FROM TABLE dfacts\_t.  
  
 REFRESH lti\_return.

Se actualizan las tablas facts del objeto contrato mediante la BAPI BAPI\_CTRACPSOBJECT\_CHANGE'.   
 CALL FUNCTION 'BAPI\_CTRACPSOBJECT\_CHANGE'  
 EXPORTING  
 psobjectkey = objeto\_contrato  
 xupdatetask = 'X'  
 TABLES  
 psobjectfacts = psobjectfacts  
 return = lti\_return.

Si el proceso es correcto se procede a actualizar las tablas standard con el commit.  
 CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
 EXPORTING  
 wait = 'X'.

Actualizar la relación IC y OC.

Consultar si existe la relación.

SELECT SINGLE partner xobsl INTO (partner, xobsl)  
 FROM dpsob\_bp\_acc  
 WHERE psobkey = i\_data-psobkey AND  
 partner = l\_partner AND

Actualizar las tablas BDTs.

Crea Asociacion entre Contribuyente y Predio (Propietario) con la BAPI 'BAPI\_CTRACPSOBJECT\_CREATE'.  
 CALL FUNCTION 'BAPI\_CTRACPSOBJECT\_CREATE'  
 EXPORTING  
 psobjecttype = claseob  
 psobjectkeyext = psobkey  
 partner = partner  
 ctraccounttype = acctyp  
 ctraccount = vkont  
 psobjectaccount = account  
 TABLES  
 return = return.

1. Instituto Geográfico Agustín Codazzi [↑](#footnote-ref-1)