OUM

TA.070-9 DISEÑO TÉCNICO DE EXTENSIONES E INTERFACES

ER01\_AR\_Creación de Transacciones

##### Autor: Pedro Sanchez Martinez

##### Fecha de creación: 13 de marzo de 2024

##### Última actualización: 13 de marzo de 2024

##### Código de referencia OUM: DS.140 Design Specification

##### Version: 1.0

#### Aprobadores:

##### 

|  |
| --- |
| Juan Gonzales |
| Jairo Rojas |

##### 

oracle.gif

# Control de Documento

## Bitácora de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Referencia del cambio** |
| 13 de marzo de 2024 | Pedro Sanchez Martinez | 1.0 | No Previous Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Revisores

**Nombre**

**Posición**

Contenido

[1 Control de Documento 1](#_Toc161247392)

[1.1 Bitácora de Cambios 1](#_Toc161247393)

[1.2 Revisores 1](#_Toc161247394)

[2 Resumen técnico 3](#_Toc161247395)

[2.1 Diagrama de la integración 4](#_Toc161247396)

[3 Lógica de implementación 5](#_Toc161247397)

[3.1 Calendarización y Ejecución 5](#_Toc161247398)

[3.2 InvoicesEnt 6](#_Toc161247399)

[3.3 InvoicesBiz 22](#_Toc161247400)

[4 Diseño de Datos 42](#_Toc161247401)

[4.1 Origen de Datos 42](#_Toc161247402)

[4.2 Lógica de Validación 42](#_Toc161247403)

[5 Reglas de Negocio 43](#_Toc161247404)

[5.1 Diseño del Servicio 43](#_Toc161247405)

[6 Consideraciones de Rendimiento 44](#_Toc161247406)

[6.1 Estrategia de Reinicio 44](#_Toc161247407)

[6.2 Seguridad 44](#_Toc161247408)

[6.3 Personalización 44](#_Toc161247409)

[6.4 Catálogo de Errores 44](#_Toc161247410)

[7 URL de Acceso y seguridad 45](#_Toc161247411)

[8 Temas abiertos y cerrados 45](#_Toc161247412)

[8.1 Temas Abiertos 45](#_Toc161247413)

[8.2 Temas Cerrados 45](#_Toc161247414)

# Resumen técnico

Este documento presenta el complemento para la integración que conforma la importación Transacciones hacia el ERP Cloud. Esta integración es referida por el documento funcional TA020-4 ER01\_AR\_Creación de Transacciones.docx.

El contenido del documento es el detalle para la representan la integración, de esta manera, su objetivo principal es identificar el flujo de la integración que está dividida en 3 capas.

* Enterprise (Ent)
* Business (Biz)
* Technical (Tec)

La capa **Ent** permite exponer aquellos servicios u operaciones de Oracle Service Oriented Architecture (SOA) mediante Oracle Service Bus (OSB), esto con la finalidad de tener una primera línea de seguridad sin exponer la arquitectura de los servidores internos de la empresa.

Por otro lado esta capa se encarga de administra las capas **Biz** utilizadas para esta integración, del mismo modo nos permite calendarizar en el servidor Enterprise Scheduler Service (EES) aquellas operaciones de SOA que se requieran ejecutar cada cierto tiempo.

La capa **Biz** tiene la finalidad de llevar acabo la lógica de negocio correspondiente a dicha integración, es decir, se encarga de orquestar aquellas capas técnicas y manejar aquellas reglas para el negocio.

La capa **Tec** nos permite ofrecer una línea de seguridad en cuestión a servicios de terceros mediante Oracle Service Bus (OSB); por otro lado esta capa nos permite conectar con aquellos orígenes de datos(Base de datos, Servicios Internos, EJB, etc.) permitiendo realizar diferentes operaciones para el guardado o manipulación de información.

Existen dos tipos de integraciones:

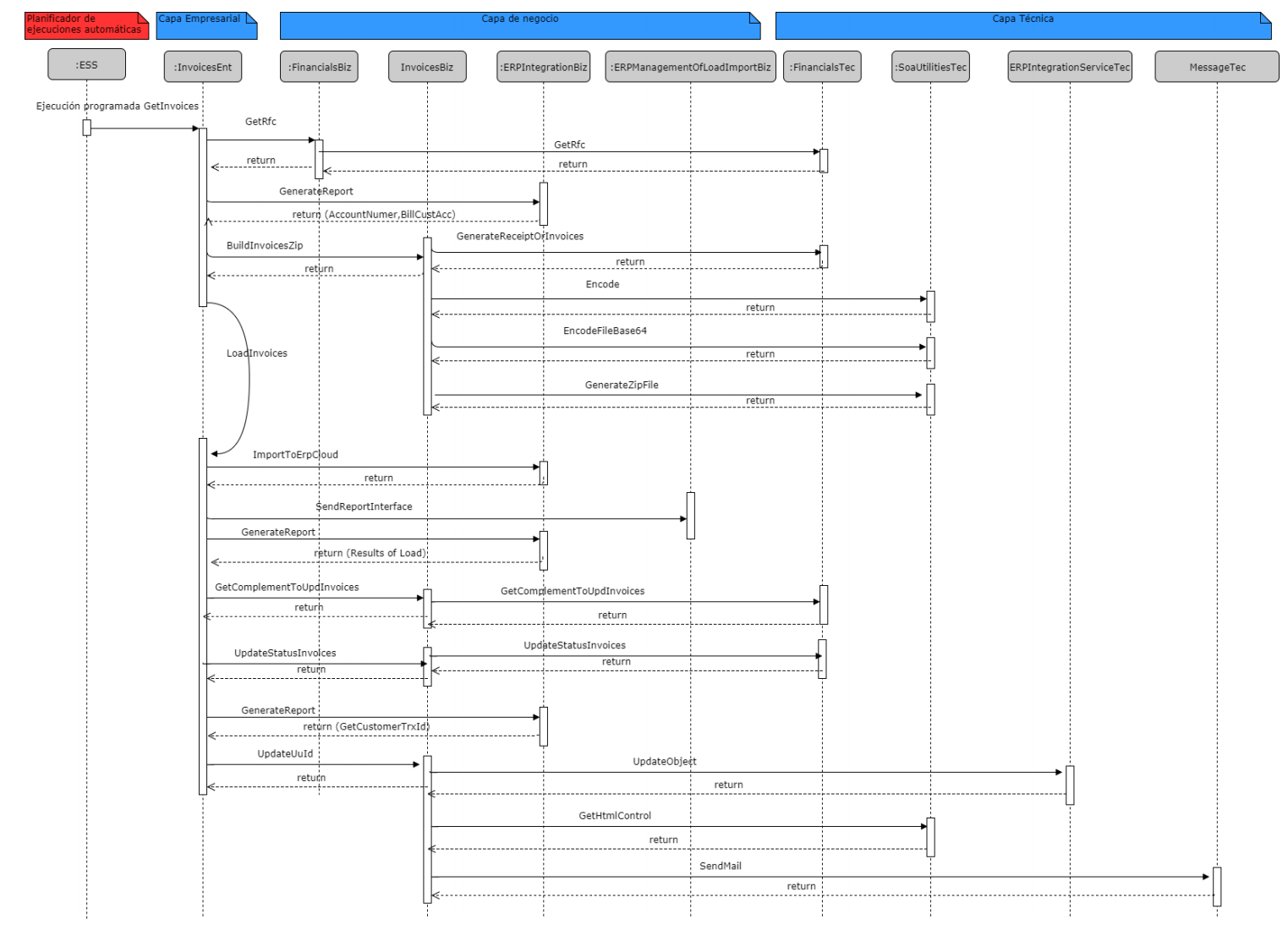
* **Inbound**: En este tipo de integraciones, las aplicaciones legadas llamarán a servicios web SOAP expuestos de lado de los servidores internos de la empresa, enviando la información hacia el ERP Cloud, es decir, son aquellas integraciones que nos permiten gestionar información referente a Clientes, Recibos, Transacciones, etc.
* **Outbound:** Estas integraciones nos permiten extraer información del ERP Cloud por medio de reportes a través de BI Publisher, posteriormente es enviada a diferentes sistemas legados según corresponda, algunas de las integraciones que aplican son: Timbrado de Facturas AR, validación de facturas AP, Validación de Complementos de Pago, etc.

Existen 4 fases importantes para este tipo de integraciones:

* 1. Consiste en la extracción de la información teniendo como fuente el ERP Cloud, donde se encuentran ciertas condiciones que permiten identificar qué información es apta para procesar.
  2. En esta fase consiste en el enriquecimiento del mensaje, es decir, implica realizar el complemento de información del lado de la capa de negocios en SOA para poder enviar al sistema externo (Capa Técnica), controlando cualquier tipo de error durante el flujo.
  3. En este punto se recupera la información devuelta por los servicios externos y se consolida la información de acuerdo a los resultados obtenidos.
  4. Por último, se envía una notificación con la información de todos los registros procesados, número de éxitos y errores con su detalle.

Estas fases están distribuidas en las 3 capas mencionadas anteriormente (Ent, Biz, Tec) las cuales se explicarán a detalle en los siguientes puntos.

## Diagrama de la integración

 Diagrama de la integración

# Lógica de implementación

En esta sección se describe cada uno de las fases que son necesarias para hacer la carga de facturas y notas de crédito, en donde se incluye la recuperación de información desde la base de datos, la creación de archivos CSV con las facturas o notas de crédito, la carga al ERP-Cloud, actualización y notificaciones de resultados. El proceso inicia con la aplicación compuesta InvoicesEnt, donde esta aplicación, es invocada por el ESS.

## Calendarización y Ejecución

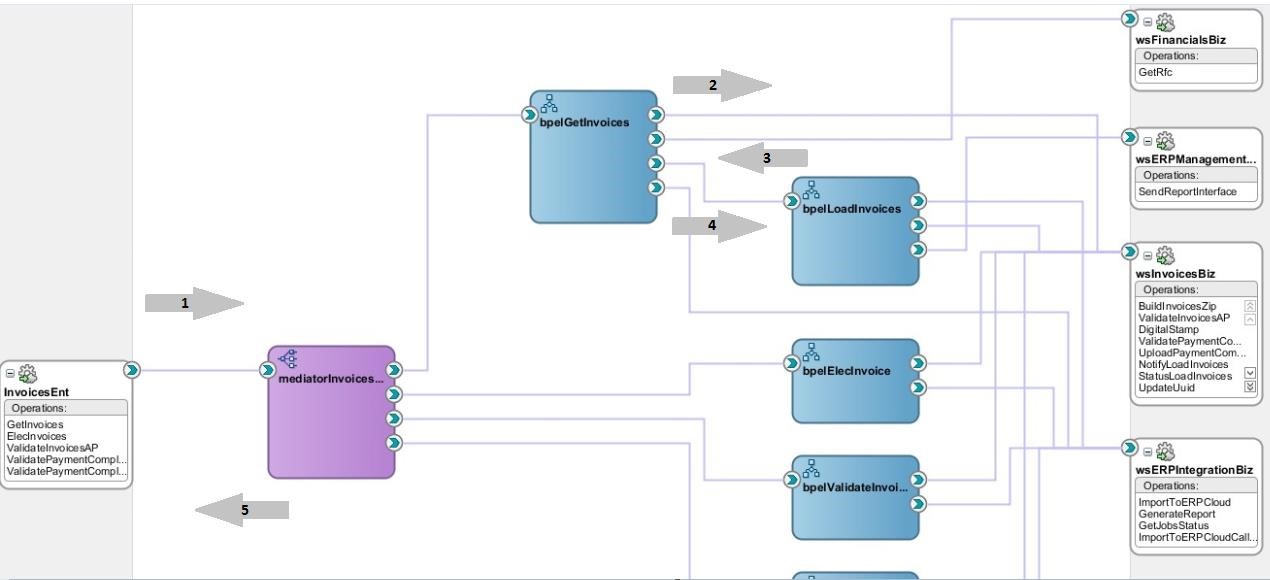
En la capa **ENT** se encuentra la operación **GetInvoices,** estaoperación se encarga de iniciar el proceso de carga de las facturas y notas de crédito por lo tanto es la operación que se manda a ejecutar con el ESS, a continuación, se muestra información de la configuración que se hizo para la ejecución periódica de la carga.

|  |  |  |
| --- | --- | --- |
| **Componente** | **Propiedad** | **Valor** |
| **Trabajo** | Nombre | InvoiceGetFacSch |
| Nombre Mostrado | InvoiceGetFacSch |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Carga Facturas AR |
| Tipo de trabajo | SyncWebserviceJobType |
| WSDL | http://<host>:<port>/soa-  infra/services/PROD/InvoicesEnt/InvoicesEnt?WSDL |
| Tipo de puerto | InvoicesEntPort |
| Operación | GetInvoices |
| **Planificación** | Nombre | InvoiceGetFacSch |
| Nombre Mostrado | InvoiceGetFacSch |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Ejecución diaria a las 02:45:00 p. m. |
| Frecuencia | Diariamente |
| Zona horaria | (UTC-06:00) Ciudad de México |

Desde las pantallas del ESS se pueden buscar las ejecuciones actuales con lo cual se puede validar el estatus de las de acuerdo a la planificación con la que se ejecutó.

## InvoicesEnt

Dentro de la capa Ent se expone el servicio **InvoicesEnt** el cual expone la operación GetInvoices la cual consume la operación LoadInvoices y otros servicios comunes para completar el proceso de inicio del proceso de transacciones.



La implementación se basa en la aplicación compuesta mostrada en la imagen anterior, se hace uso de los componentes:

|  |  |
| --- | --- |
| Componente | Icono del componente |
| Servicio Web |  |
| Mediador |  |
| BPEL |  |

El módulo principal se denomina mediador, este componente es el encargado de enrutar las peticiones desde el servicio de entrada hacia los demás componentes que se encargan de diferente funcionalidad y viceversa, es decir, las respuestas de cada componente son enrutadas y así entregadas a la petición que se generó.

### Lista de Objetos

Los siguientes objetos fueron usados/creados para implementar la funcionalidad del servicio Web

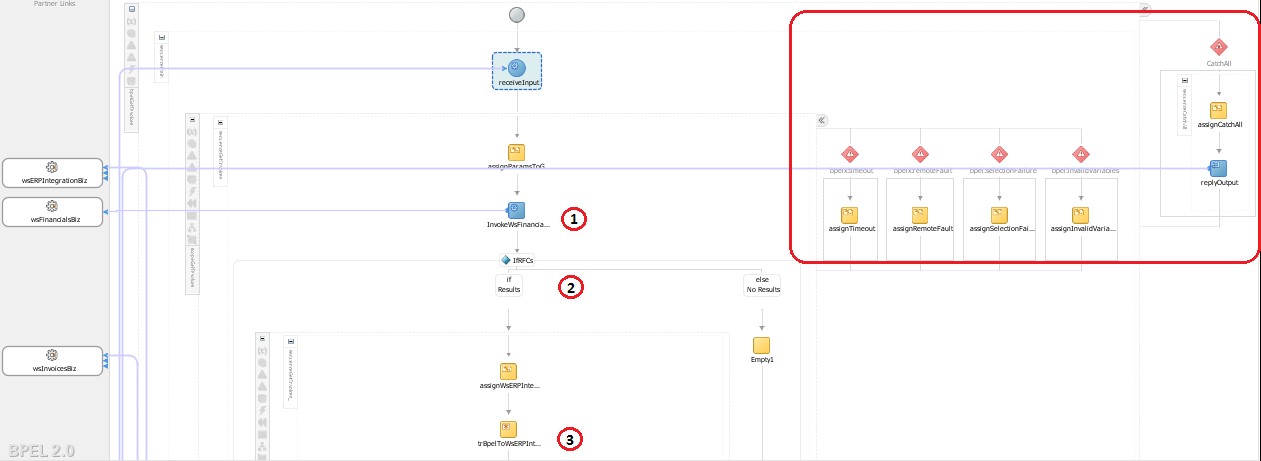
|  |  |  |
| --- | --- | --- |
| **Elemento** | **Tipo** | **Descripción** |
| InvoiceEnt | Web  Service | Servicio Expuesto |
| mediatorInvoiceEnt | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| bpelGetInvoices | Bpel | Componente de Orquestación |
| bpelLoadInvoices | Bpel | Componente de Orquestación |
| wsInvoicesBiz | Web  Service | Servicio externo que tiene la para generar el template de carga de facturas |
| wsERPIntegrationBiz | Web  Service | Servicio externo que Realiza la importación de los Datos al ERP |
| wsFinancialsBiz | Web  Service | Servicio externo que consulta los RFC a procesar. |
| wsERPManagementOfLoadImpo rtBiz | Web  Service | Servicio externo que Realiza la importación de los template al ERP para la carga masiva de facturas. |

### GetInvoices

**GetInvoices**: Esta es la primera operación en ser invocada para el proceso de carga, el BPEL que se usa es de tipo síncrono a continuación se detallan los componentes que integran esta operación:

1. Se invoca la operación “FinancialsTec”-“GetCustomerRfc” para recuperar los RFC de los clientes que estarán disponibles para la carga de facturas y notas de crédito.
2. Se valida que la invocación para recuperar los RFC, haya traído resultados para poder continuar con el flujo.
3. Se hace uso de una transformación para establecer valores a la variable de entrada para la invocación de la siguiente operación que es consumir un reporte.

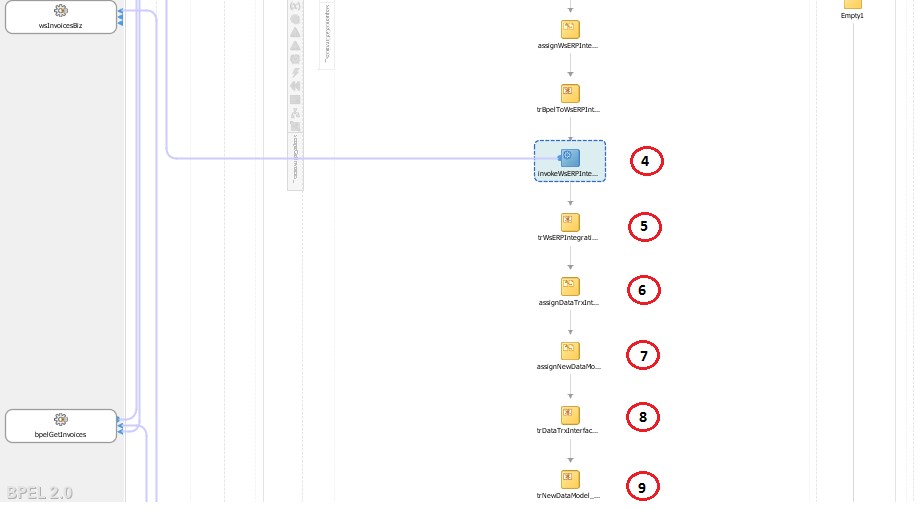
a. Se integra el control de excepciones para capturar aquellas excepciones más comunes que lleguen a pasar en el flujo completo de la operación GetInvoices.



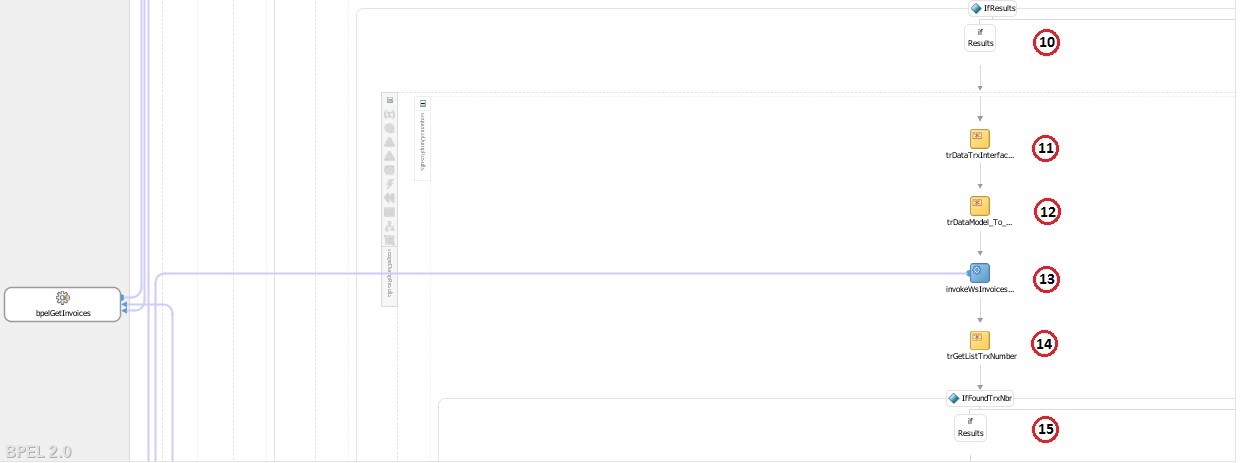
1. Se hace invocación al servicio **“ERPIntegrationBiz”-“GenerateReport”** para consumir el Data Model “XXER\_AR\_TRX\_INTERFACE” que devolverá información complementaria para poder importar las Facturas o notas de crédito.
2. En este paso se hace uso de una transformación

(trWsERPIntegrationBizGenerateReportRs\_To\_WsERPIntegrationBizGenerateReportRs) para poder tomar la información resultante de la llamada al servicio **“ERPIntegrationBiz”-“GenerateReport”** ytransfórmala a información en donde a los nodos se les pueda establecer determinado Namespace.

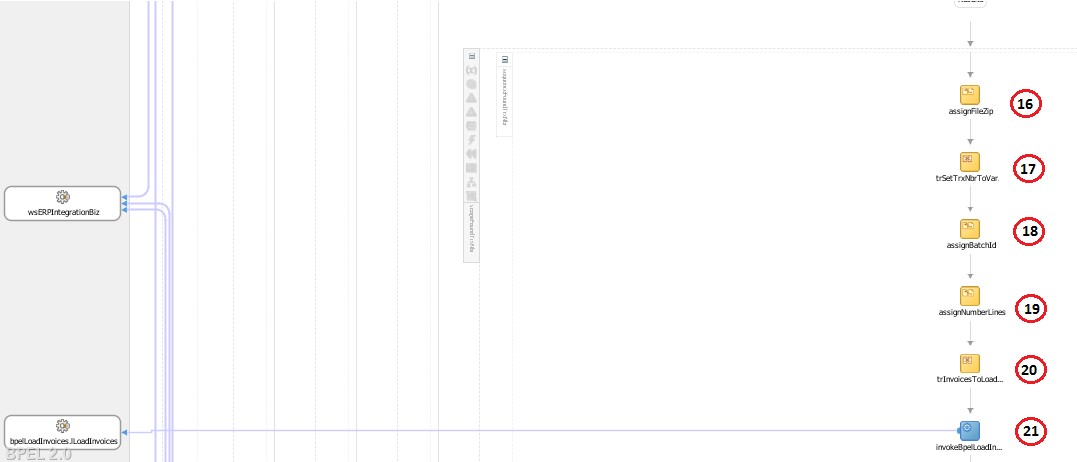
1. En este paso se agrega una asignación (assignDataTrxInterfaces) en la que se toma la información del paso anterior y se convierte a información de tipo XML.
2. Este proceso es similar al anterior solo que la información es asignada a otra variable.
3. En este paso se toma la información del paso 6, y ya que es información de tipo XML, se utiliza una transformación para asignar los datos a una variable con la estructura del reporte OTBI que se consultó en la invocación de servicio anterior y así poder usar la información para complementar los datos de carga.
4. En este paso se hace el mismo procedimiento que el paso anterior, solo que la información se establece a otra variable.



1. En este paso se usa el componente IF para evaluar si hubo datos devueltos del reporte OTBI anterior, si hubo datos se continua con el flujo, si no es así se entra al ELSE del IF y se termina el flujo.
2. Se agrega transformación para tomar datos de la variable que contiene la información del reporte y pasarlos a variable de entrada para la ejecución de Invoke.
3. En este paso se hace el mismo procedimiento que el paso anterior, solo que el origen es de distinta variable.
4. Se hace la invocación del servicio “InvoicesEnt”-“BuildInvoicesZip” que es la operación que construirá el CSV que contendrá las facturas o facturas a importar.
5. Se agrega transformación para obtener la lista de TRX\_NUMBER de las facturas que se tienen en el archivo CSV también se recupera el BATCH\_ID.
6. Se evalúa si la lista de TRX\_NUMBER no es nula para poder continuar con el flujo de importación de facturas.



1. En este paso se agregan valores a los parámetros de la variable de entrada para invocar el bpelLoadInvoices.
2. Se Utiliza transformación para pasar la lista de TRX\_NUMBER y otras variables a una variable que incluye los datos de entrada para la ejecución del servicio **“ERPIntegrationBiz”-“GenerateReport”.**
3. Se agrega Asignación para recuperar el BATCH\_ID.
4. Se agrega asignación para calcular el número de líneas que incluye el CSV, que se ha de cargar.
5. Se agrega transformación (trInvoicesToLoadToBpelLoadInvoices) para establecer valores a variable de entrada de invocación de bpel asíncrono (bpelLoadInvoices)



1. Se invoca al bpel “bpelLoadInvoices” para poder llamar al bpel asíncrono que se encargará de ejecutar las operaciones para importar las facturas o notas de crédito al ERP-Cloud.

#### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad de obtener información para cargar las facturas y notas de crédito al ERP-Cloud.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Nombre | Tipo | Descripción |
| 1 | receiveInput | Entrada | Recibe los parámetros de entrada |
| 2 | bpelGetInvoices | Alcance | Define un nuevo alcance |
| 3 | sequenceMain | Secuencia | Secuencia de asignaciones |
| 4 | assignParamsToGetRFCs | Asignación | Asignación de datos |
| 5 | InvokeWsFinancialsBizGetRfc | Invocación | Invocación de servicio |
| 6 | scopeGetInvoices | Alcance | Define un nuevo alcane |
| 7 | sequenceGetInvoices | Secuencia | Secuencia de asignaciones |
| 8 | IfRFCs | Condición | Condicionante de datos |
| 9 | assignWsERPIntegrationBizReferenceOperation Rq | Asignación | Asignación de datos |
| 10 | scopeGetInvoices\_ | Alcance | Define un nuevo alcance |
| 11 | sequenceGetInvoices\_ | Secuencia | Secuencia de asignaciones |
| 12 | trBpelToWsERPIntegrationBizGenerateReportRq | Transformación | Transformación de esquemas |
| 13 | invokeWsERPIntegrationBizReferenceOperation | Invocación | Invocación de servicio |
| 14 | trWsERPIntegrationBizGenerateReportRs\_To\_W sERPIntegrationBizGenerateReportRs | Transformación | Transformación de esquemas |
| 15 | assignDataTrxInterfaces | Asignación | Asignación de datos |
| 16 | assignNewDataModelTest | Asignación | Asignación de datos |
| 17 | trDataTrxInterface\_To\_DataTrxInterfaceResult | Transformación | Transformación de esquemas |
| 18 | trNewDataModel\_To\_DataModelResult | Transformación | Transformación de esquemas |
| 19 | IfResults | Condición | Condicionante de datos |
| 20 | trDataTrxInterface\_To\_BuildInvoicesZipRq | Transformación | Transformación de esquemas |
| 21 | scopeDumpResults | Alcance | Define un nuevo alcance |
| 22 | sequenceDumpResults | Secuencia | Secuencia de asignaciones |
| 23 | trDataModel\_To\_BuildInvoicesZipRq | Transformación | Transformación de esquemas |
| 24 | invokeWsInvoicesBizToBpel | invocación | Invocación de servicio |
| 25 | trGetListTrxNumber | Transformación | Transformación de esquemas |
| 26 | IfFoundTrxNbr | Condición | Condicionante de datos |
| 27 | assignFileZip | Asignación | Asignación de datos |
| 28 | scopeFoundTrxNbr | Alcance | Define un nuevo alcance |
| 29 | sequenceFoundTrxNbr | Secuencia | Secuencia de asignaciones |
| 30 | trSetTrxNbrToVarGenerateReport | Transformación | Transformación de esquemas |
| 31 | assignBatchId | Asignación | Asignación de datos |
| 32 | assignNumberLines | Asignación | Asignación de datos |
| 33 | trInvoicesToLoadToBpelLoadInvoices | Transformación | Transformación de esquemas |
| 34 | invokeBpelLoadInvoices | Invocación | Invocación de proceso |
| 35 | replyOutput | Salida | Contiene la respuesta de la operación |

#### Pruebas

Al hacer una prueba del proceso en general para la carga, actualización y notificación de facturas y notas de crédito es necesario llamar primero al servicio InvoicesEnt.GetInvoices.

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección del siguiente WSDL: http://<host>:<puerto>/soa[-infra/services/PROD/InvoicesEnt/InvoicesEnt?WSDL](http://ersoacsprod-wls-1:9073/soa-infra/services/default/InvoicesEnt/InvoicesEnt?WSDL)

A continuación, se muestran los parámetros de entrada y la respuesta que se presentan en esta operación:

**Request:** No se necesitan parámetros de entrada para la ejecución de esta operación:



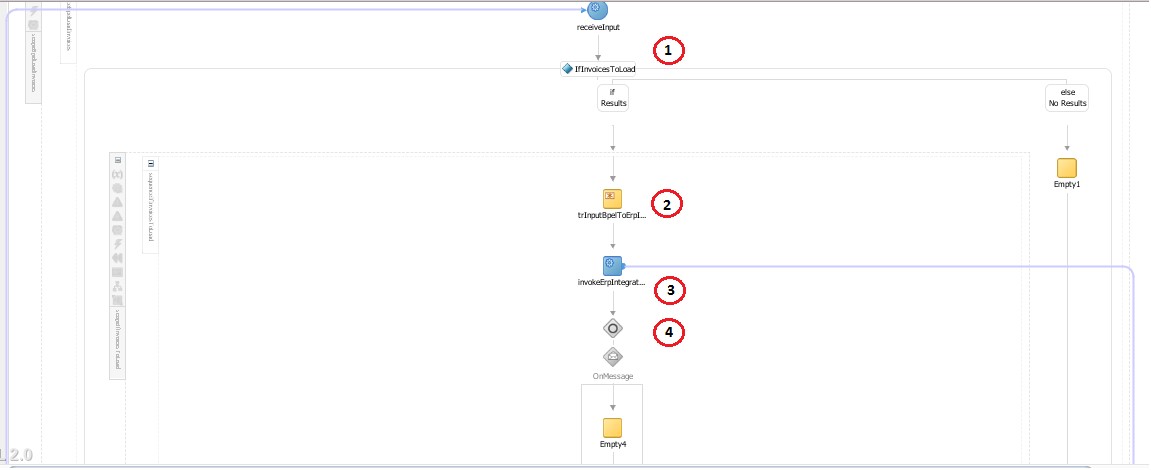
**Response:** El servicio no requiere algún dato de respuesta por lo que no se recibe nada:



### LoadInvoices

**LoadInvoices**: Es la operación que conforma el bpel asíncrono bpelLoadInvoices, este bpel es encargado de hacer la carga de facturas o notas de crédito al ERP-Cloud, se eligió este tipo de servicio ya que el proceso de importación demora el tiempo suficiente para que un bpel síncrono presente problemas de Timeout, problema que no se presenta en este tipo de bpel, los componentes que integran este bpel se detallan a continuación.

1. Se agrega IF para evaluar que los parámetros no sean nulos para poder avanzar con el flujo
2. Se agrega transformación (trInputBpelToErpIntegrationBizImportErpCloud) para asignar los valores de los parámetros a la variable request de la invocación del servicio para importar las facturas o notas de crédito al ERP-Cloud.
3. Se invoca el servicio **“ERPIntegrationBiz”-“ImportToErpCloud”**, para poder cargar las facturas y notas de crédito al ERP-Cloud.
4. Se agrega Pick para poder poner en pausa el flujo hasta la operación de cargar al ERP-Cloud finalice.



1. Se asigna valores a la variable request para la invocación del servicio

**“ERPManagementOfLoadImportBiz”-“SendReportInterface”**

1. Se invoca el servicio **“ERPManagementOfLoadImportBiz”-“SendReportInterface”** para poder recuperar los resultados de carga y notificarlos vía correo electrónico.
2. “IF” para evaluar si se activara el flujo de actualización del “UUID” de las facturas y continuar con el flujo.
3. Asignación (assignVarParamsRepIntCtrl) para establecer parte de los valores a la variable request para la invocación del servicio para consumir un reporte OTBI
4. Transformación para establecer los valores restantes a la variable de request para la invocación del servicio para consumir reporte.
5. Se invoca el servicio **“ERPIntegrationBiz”-“GenerateReport”,** para consumir el reporte OTBI “XXER\_FND\_INTERFACE\_CONTROL” con el que se recuperara los resultados de carga junto con información adicional que ayudara actualizar el UUID de las facturas o notas de crédito.



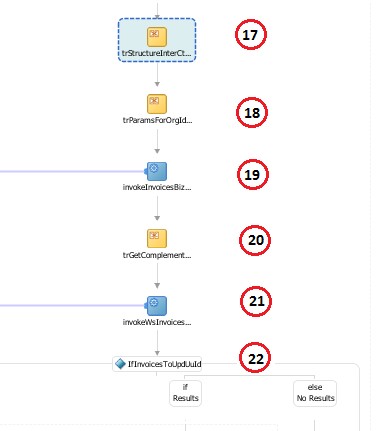
1. Se agrega transformación

(trvarInvokeWsErpIntegrationBizGenerateReportRsTovarInvokeWsErpIntegrationBizGenerateReportRs) para tomar la información resultante de carga de facturas que se hizo en el paso anterior y poner a los nodos el namespace correspondiente.

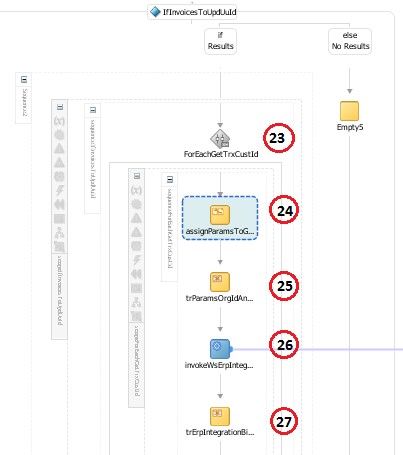
1. Se agrega asignación (assignDataInterfaceCtrl) para convertir la información resultante del paso anterior a información en formato XML.
2. Se agrega transformación (trStructureInterfaceCtrlToStructureInterfaceCtrlResult) para recuperar la información en formato XML y con estructura del reporte “XXER\_FND\_INTERFACE\_CONTROL” para poder hacer uso de la información recuperada, además de agregar funciones a los valores para filtrar información recuperada.
3. Se agrega IF para evaluar si hubo registros devueltos por el reporte OTBI consultado anteriormente, si hay resultados se puede continuar con el flujo.
4. Se agrega la transformación (trResultLoadInvoicesToparamsToUpdStatusInvoices) para recuperar parte de los valores a establecer para ejecutar el servicio que actualizará el estado de las variables que pudieron importarse al ERP-Cloud y aquella que marcaron error.
5. Se agrega transformación (trParamsToUpdStatusInvToWsInvoicesBizUpdStatusInvRq) para asignar valores a la variable request para invocar el servicio que actualizará en BD aquellas facturas o notas de crédito que intentaron cargarse al ERP-Cloud.



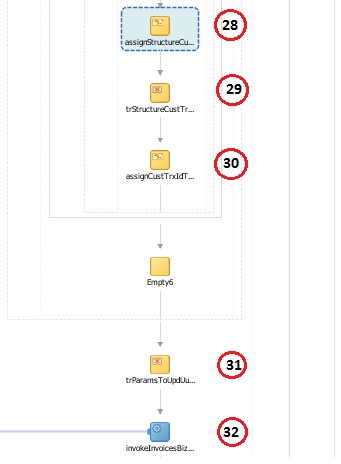
1. Se agrega transformación (trStructureInterCtrlToParamsToGetOrgId) para recuperar valores que usaran como parámetros para invocar el servicio que obtendrá complementos para recuperar el CUSOMER\_TRX\_ID.
2. Se agrega transformación (trParamsForOrgIdToInvoicesBizGetComplementToUpdateInvoices) para asignar valores a la variable request para invocar el servicio que recuperará la información complementaria (ORGANIZATION\_ID) para poder ir por el CUSTOMER\_TRX\_ID posteriormente.
3. Se invoca al servicio “InvoicesBiz”-“GetComplementToUpdInvoices” para recuperar el ORGANIZATION\_ID y UUID que se usarán para actualizar los registros de facturas o notas de crédito ya cargados en el ERPCloud.
4. Se agrega transformación (trGetComplementToUpdInvToParamsTrxNumAndOrgId) para recuperar la información resultante de la invocación del servicio anterior, información que servirá para hacer la actualización del UUID de las facturas cargadas en el ERP.
5. Se hace la invocación del servicio “InvoicesBiz”-“UpdateStatusInvoices”, este servicio actualizará en BD el status que se tubo de intentar cargar las facturas o notas de crédito al ERP-Cloud estableciendo “S” si la carga fue correcta o “E” si es que la carga tubo errores.
6. Se agrega IF para evaluar si la invocación del servicio que recupera el complemento (ORGANIZATION\_ID y UUID) para actualizar el UuId, recupero algún dato, si es así se continua con el flujo.



1. FOR EACH para iterar la lista de valores que contiene las facturas y sus complementos para recuperar el CUSTOMER\_TRX\_ID necesario para la actualización del UuId.
2. Se agrega asignación (assignParamsToGenerateReportCustTrxId) para pasar valores de datos complementarios a variable con estructura similar a la variable que se utilizara como request para la invocación del servicio que consulta el reporte que recupera el CUSTOMER\_TRX\_ID.
3. Se agrega transformación (trParamsOrgIdAndTrxNumToGenerateReportCustTrxId) para asignar valores a la variable request que se usará para invocar el servicio que consume reportes del Data Model.
4. Se invoca el servicio “ERPIntegrationBiz”-“GenerateReport” para ejecutar el reporte OTBI” XXER\_AR\_TRX\_INTRF\_CUST\_TRX\_ID.” El cual recuperar el CUSTOMER\_TRX\_ID de cada factura, este valor que se recupera será necesario para poder actualizar el UuId de las facturas o notas de crédito ya cargadas en el ERP-Cloud.
5. Se agrega transformación para tomar los datos recuperados de la invocación del servicio del paso anterior y asignar el namespace correspondiente a los nodos devueltos.



1. Se utiliza asignación (assignStructureCustTrxId) para tomar la información resultante del paso anterior y convertirla a formato XML.
2. Se usa transformación (trStructureCustTrxIdToStructureCustTrxIdResult) para pasar la información resultante del paso anterior a una variable con la estructura similar a la del reporte XXER\_AR\_TRX\_INTRF\_CUST\_TRX\_ID, esto para que la información pueda ser accesible y usada para los siguientes pasos del flujo.
3. Se agrega asignación (assignCustTrxIdToParmsUpdUuId) para pasar la información recuperada en los pasos anteriores a una variable similar a la variable request que se usará para invocar el servicio que actualizará el UuId.
4. Se agrega transformación (trParamsToUpdUuIdToInvoicesBizUpdateUuIdRq) para pasar información a variable request para invocar el servicio para actualizar el UuId.
5. Se invoca el servicio “InvoicesBiz”-“UpdateUuId” para mandar a actualizar el UuId de aquellas facturas o notas de crédito que pudieron cargarse correctamente en el ERP-Cloud.



#### Lista de objetos

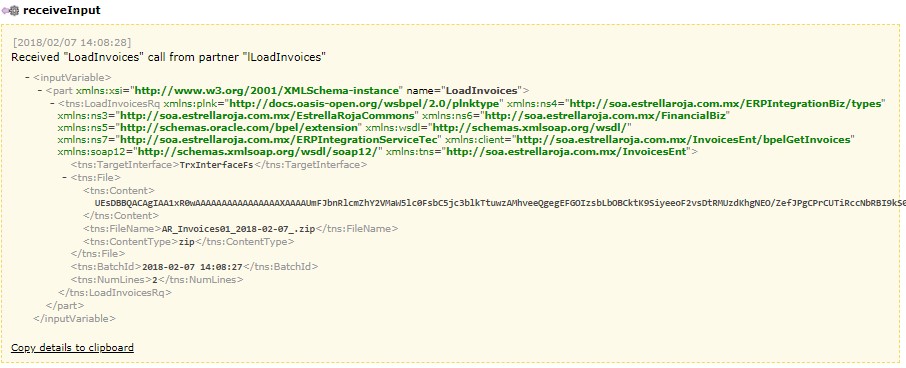
La siguiente tabla contiene los objetos que se utilizaron para la funcionalidad de mandar a cargar las facturas y notas de crédito al ERP-Cloud.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Nombre | Tipo | Descripción |
| 1 | receiveInput | Entrada | Recibe los parámetros del bpel |
| 2 | scopeBpelLoadInvoices | Alcance | Define un nuevo alcance |
| 3 | sequenceBpelLoadInvoices | Secuencia | Secuencia de asignaciones |
| 4 | IfInvoicesToLoad | Condición | Condicionante de datos |
| 5 | scopeIfInvoicesToLoad | Alcance | Define un nuevo alcance |
| 6 | sequenceIfInvoicesToLoad | Secuencia | Secuencia de asignaciones |
| 7 | trInputBpelToErpIntegrationBizImportErpCloud | Transformación | Transformación de esquemas |
| 8 | invokeErpIntegrationBizImportToErpCloud | Invocación | Invocación de servicio |
| 9 | PickWaitLoad | Espera de finalización de proceso | Pausa el proceso hasta que termine el proceso anterior |
| 10 | assignParamsNotifyLoad | Asignación | Asignación de datos |
| 11 | invokeWsERPManagementOfLoadImportBiz | Invocación | Invocación de servicio |
| 12 | IfActivateFlowUpdUuid | Condición | Condicionante de datos |
| 13 | assignVarParamsRepIntCtrl | Asignación | Asignación de datos |
| 14 | scopeNotifyLoadInvoices | Alcance | Define un nuevo alcance |
| 15 | sequenceNotifyLoadInvoices | Secuencia | Secuencia de asignaciones |
| 16 | trParamsToGenerateReport | Transformación | Transformación de esquemas |
| 17 | invokeWsErpIntegrationBizGenerateReport | Invocación | Invocación de servicio |
| 18 | trVarInvokeWsErpIntegrationBizGenerateReportR  sToVarInvokeWsErpIntegrationBizGenerateRepo  rtRs | Transformación | Transformación de esquemas |
| 19 | assignDataInterfaceCtrl | Asignación | Asignación de datos |
| 20 | trStructureInterfaceCtrlToStructureInterfaceCtrlRe sult | Transformación | Transformación de esquemas |
| 21 | IfResultsDm | Condición | Condición de datos |
| 22 | trResultLoadInvoicesToparamsToUpdStatusInvoi ces | Transformación | Transformación de esquemas |
| 23 | scopePickWaitForLoadInvoices | Alcance | Define un nuevo alcance |
| 24 | sequencePickWaitForLoadInvoices | Secuencia | Secuencia de asignaciones |
| 25 | trParamsToUpdStatusInvToWsInvoicesBizUpdSt atusInvRq | Transformación | Transformación de esquemas |
| 26 | trStructureInterCtrlToParamsToGetOrgId | Transformación | Transformación de esquemas |
| 27 | trParamsForOrgIdToInvoicesBizGetComplement ToUpdateInvoices | Transformación | Transformación de esquemas |
| 28 | invokeInvoicesBizGetComplementToUpdInvoices | Invocación | Invocación de servicio |
| 29 | trGetComplementToUpdInvToParamsTrxNumAn dOrgId | Transformación | Transformación de esquemas |
| 30 | invokeWsInvoicesBizUpdateStatusInvoices | Invocación | Invocación de servicio |
| 31 | IfInvoicesToUpdUuId | Condición | Condicionante de datos |
| 32 | scopeIfResultsDm | Alcance | Define un nuevo alcance |
| 33 | sequenceIfResultsDm | Secuencia | Secuencia de asignaciones |
| 34 | ForEachGetTrxCustId | Ciclo | Ciclo de registros |
| 35 | scopeIfInvoicesToUpdUuId | Alcance | Define un nuevo alcance |
| 36 | sequenceIfInvoicesToUpdUuId | Secuencia | Secuencia de asignaciones |
| 37 | assignParamsToGenerateReportCustTrxId | Asignación | Asignación de datos |
| 38 | trParamsOrgIdAndTrxNumToGenerateReportCus tTrxId | Transformación | Transformación de esquemas |
| 39 | invokeWsErpIntegrationBizGenerateReportCustT rxId | Invocación | Invocación de servicios |
| 40 | trErpIntegrationBizGenRepCustTrxIdRsToErpInte grationBizGenRepCustTrxIdRs | Transformación | Transformación de esquemas |
| 41 | assignStructureCustTrxId | Asignación | Asignación de datos |
| 42 | scopeForEachGetTrxCustId | Alcance | Define un nuevo alcance |
| 43 | sequenceForEachGetTrxCustId | Secuencia | Secuencia de asignaciones |
| 44 | trStructureCustTrxIdToStructureCustTrxIdResult | Transformación | Transformación de esquemas |
| 45 | assignCustTrxIdToParmsUpdUuId | Asignación | Asignación de datos |
| 46 | trParamsToUpdUuIdToInvoicesBizUpdateUuIdRq | Transformación | Transformación de esquemas |
| 47 | invokeInvoicesBizUpdateUuid | Invocación | Invocación de servicio |
| 48 | callbackClient | Respuesta | Respuesta Asíncrona |

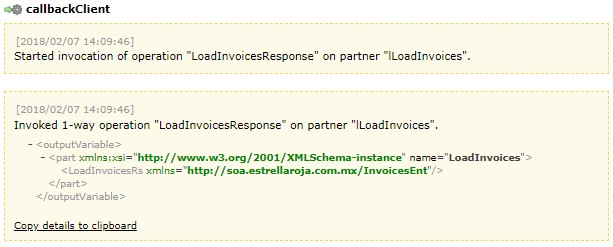
#### Pruebas

Conforme a la prueba general que se hizo, se mostrará a continuación los parámetros de entrada y la respuesta que se presentan en esta operación:

**Request:**

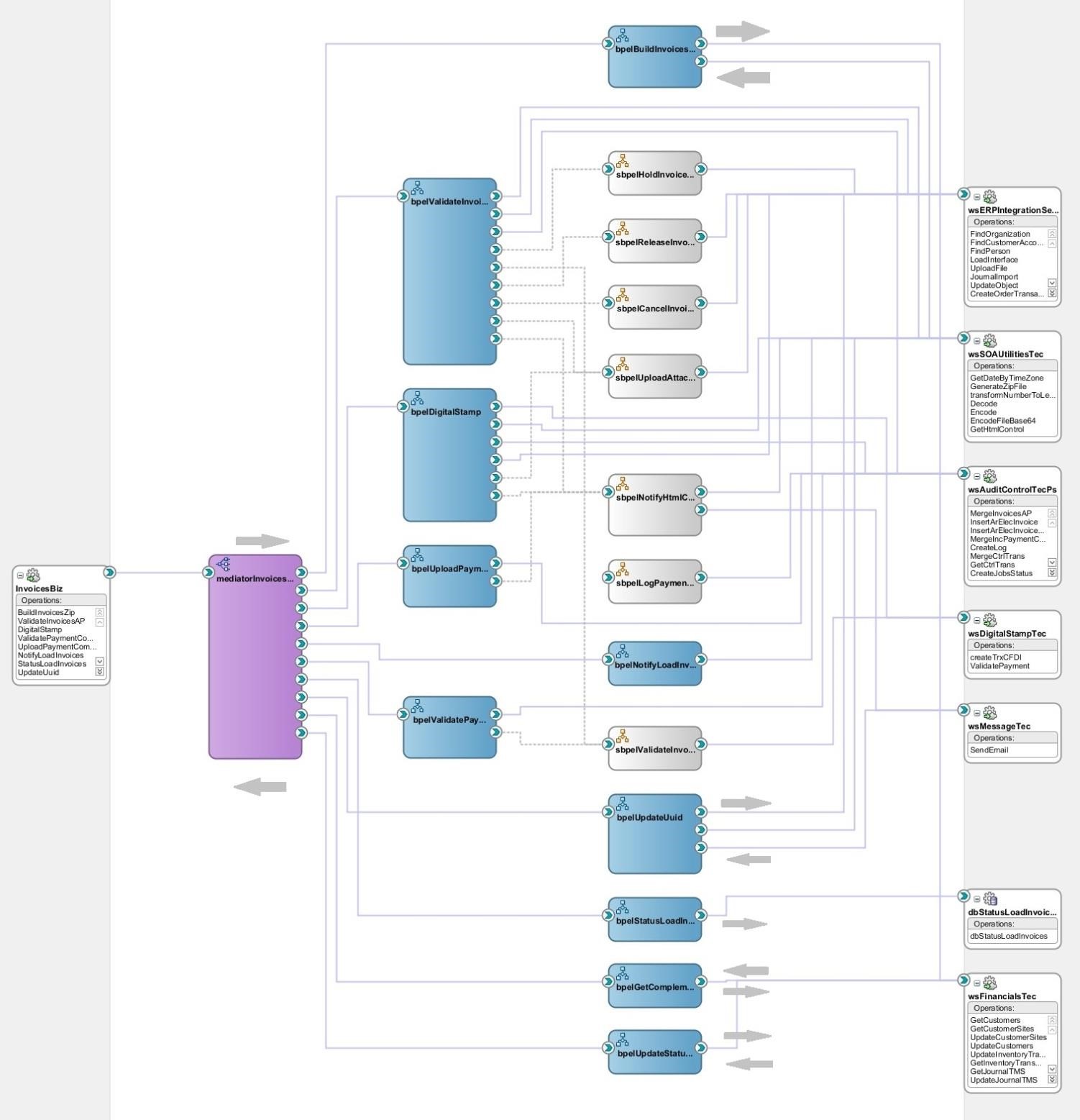


**Response:**



## InvoicesBiz

InvoicesBiz es el servicio que contiene la operación de la capa de negocio, esta capa es consumida por la capa Enterprise que en este caso es InvoicesEnt y la capa Biz hace uso de recursos de la capa Tec. En esta capa Biz se encuentran los procesos para ayudar a obtener información de la base de datos con la ayuda de la capa Tec, también en esta capa se usa para ayudar a hacer notificaciones de las cargas y actualizaciones de Facturas y notas de crédito. El siguiente es el compuesto del bpel InvoicesBiz, para el flujo de carga, actualización y notificación de transacciones serán los BPEL que estén incluidos en las flechas.



La implementación se basa en la aplicación compuesta mostrada en la imagen anterior, se hace uso de los componentes:

|  |  |
| --- | --- |
| Componente | Icono del componente |
| Servicio Web |  |
| Mediador |  |
| BPEL |  |

El módulo principal se denomina mediador, este componente es el encargado de enrutar las peticiones desde el servicio de entrada hacia los demás componentes que se encargan de diferente funcionalidad y viceversa, es decir, las respuestas de cada componente son enrutadas y así entregadas a la petición que se generó.

### Lista de Objetos

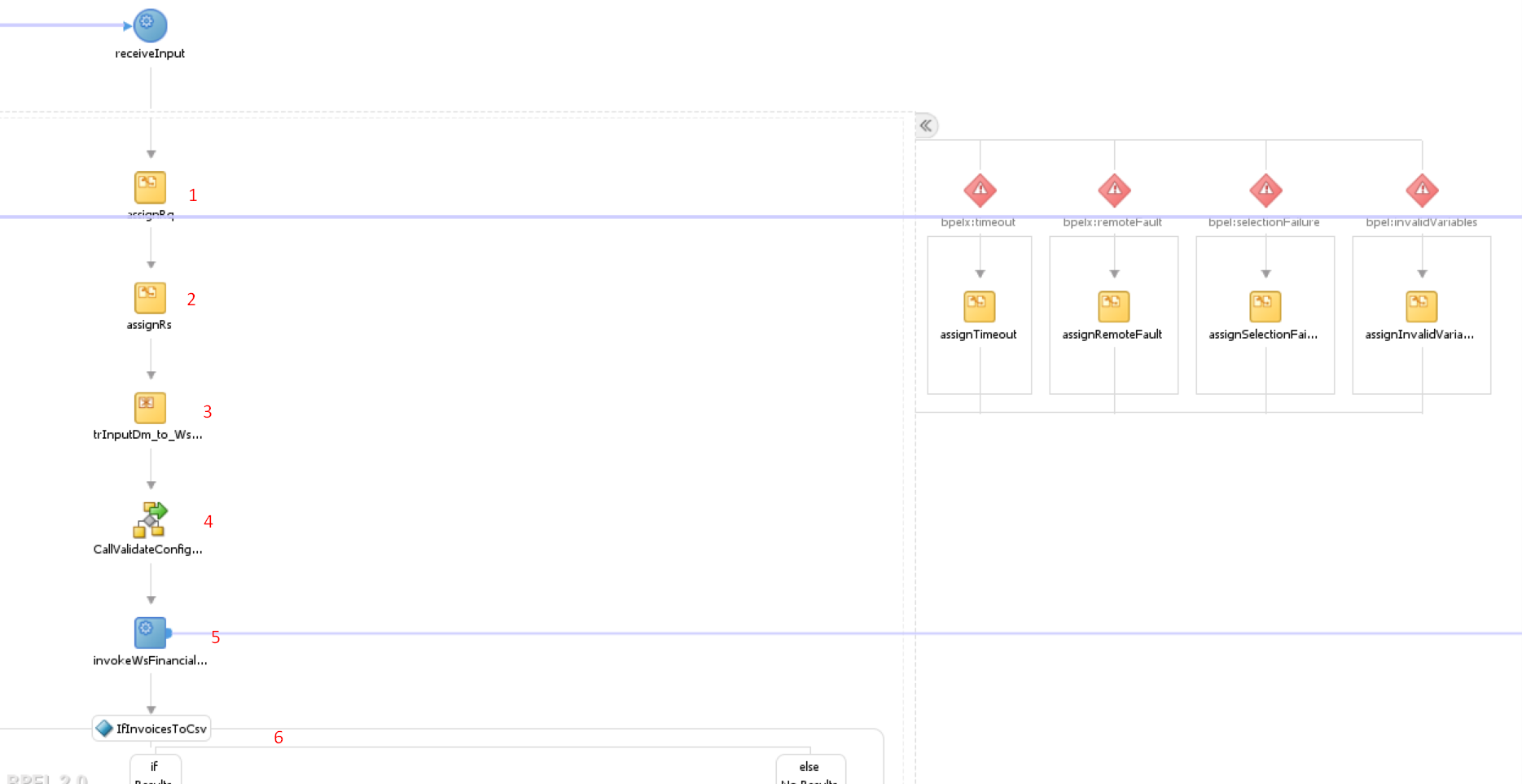
Los siguientes objetos fueron usados/creados para implementar la funcionalidad del servicio Web

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Tipo** | **Descripción** |
| JournalEnt | Web  Service | Servicio Expuesto |
| mediatorJournalEnt | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| bpelSendFileJournal | Bpel | Componente de Orquestación |
| wsJournalBiz | Web  Service | Servicio externo que tiene la lógica de extracion de las polizas |
| wsERPIntegrationBiz | Web    Service | Servicio externo que Reliza la importación de los Datos al ERP |

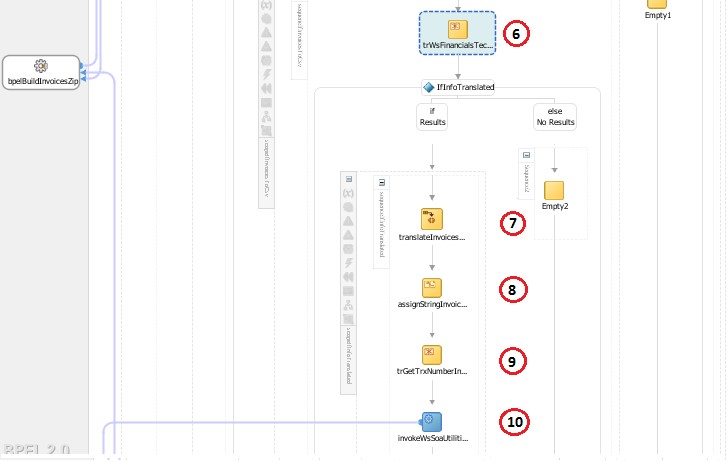
### BuildInvoices

Este es la operación que ayuda a construir el archivo Zip que contiene al archivo CSV donde se encuentran los registros de facturas o notas de crédito, para sus procesos se hace uso de la capa Tec. A continuación, se detallan los componentes que incluye este bpel.

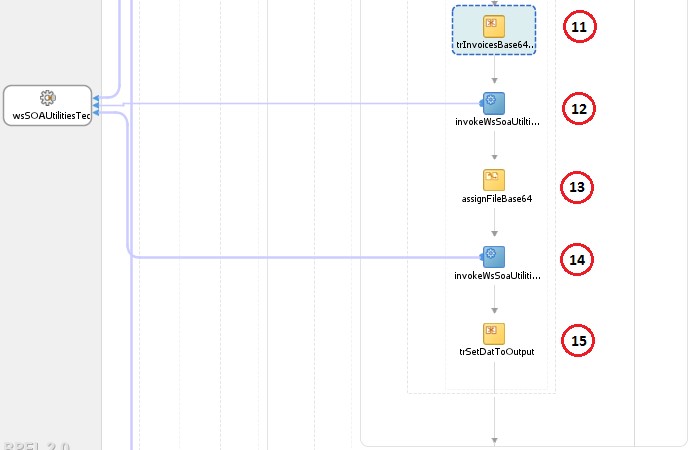
1. Se agrega asignación para establecer valor de algunas literales comunes para el flujo.
2. Se agrega asignación para establecer valor de algunas literales comunes para el flujo.
3. Se agrega la transformación (trInputDm\_to\_WsGetInvoicesReq) para asignar los parámetros a una variable y también establecer un valor como parámetro para la invocación del siguiente servicio que ayudará a generar el archivo CSV.
4. Se invoca el subBpel que “sbpelValidateConfigLines” se encarga de validar que las facturas vengan con todas sus líneas de caso contrario las cambia a estatus de error y indica cuales no tienen configuración y notifica por correo electrónico.
5. Se invoca el servicio “FinancialsTec”-“ GetReciptOrInvoices” para obtener la colección de datos de la base de datos para construir el archivo CSV.
6. Se agrega IF para validar que la variable que almacena los datos consultados a la base de datos, no venga nula, sí es así se podrá continuar con el flujo. **En la imagen se puede apreciar también el uso de control de excepciones.**



1. Se usa la transformación (trWsFinancialsTecGetReceiptOrInvoiceToBpel) para pasar los valores de la variable de salida de la ejecución del servicio anterior.
2. Se usa Translate (translateInvoicesToCsvToString) para convertir los datos recuperados en el paso anterior a contenido tipo cadena.
3. Se usa asignación (assignStringInvoicesToWsSoaUtils) para establecer valores a la variable request para invocar el siguiente servicio que es para convertir la cadena a contenido base 64.
4. Se usa Transformación (trGetTrxNumberInProcess) para recuperar la lista de TRX\_NUMBER que incluirá el archivo CSV.
5. Se invoca el servicio “SOAUtilitiesTec”-“Encode” para tomar la cadena que se obtuvo en el paso 7 y convertirlo a contenido Base 64 bits.

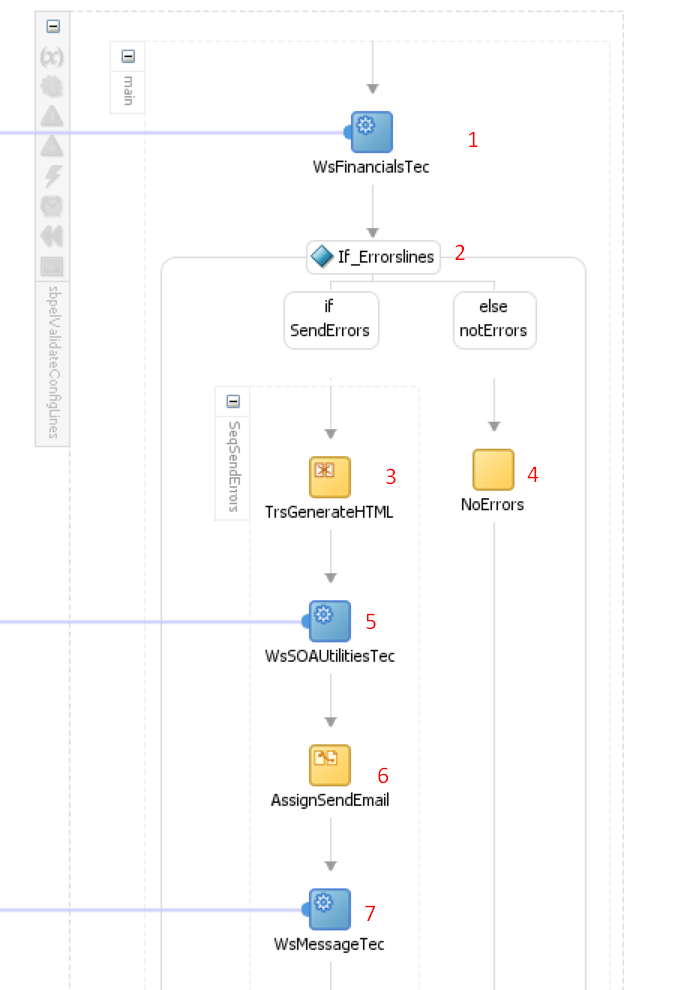


1. Se agrega transformación (trInvoicesBase64ToEncode64Zip) para tomar el contenido devuelto de la invocación del paso anterior y establecerlo a la variable request para la invocación del próximo servicio.
2. Se invoca el servicio “SoaUtiltiesTe”-“EncodeFileBase64” para crear el archivo CSV y codificarlo en base 64.
3. Se usa asignación (assignFileBase64) para pasar el contenido recuperado en la invocación de servicio de paso anterior y pasarlo a la variable request para la invocación del siguiente servicio.
4. Se invoca el servicio “SoaUtilitiesGenerateFileZip”-“GenerateZipFile” para genera el archivo Zip que contiene al archivo CSV, y también el Zip es convertido en Base 64 bits.
5. Se usa asignación (trSetDatToOutput) para pasar los resultados del paso anterior a la variable de respuesta del bpel.



**sbpelValidateConfigLines**

1. Se invoca el servicio “FinancialsTec”-“ ValidateLinesInvoiceConfig” operación que actualiza las facturas en la base de datos que no tienen todas las líneas configuración para que no sean procesadas en este caso.
2. Si el proceso es correcto notificamos si no no se hace nada
3. Transformación de generación de html de respuesta listando los que no tienen configuración
4. Invoca servicio SOAUtilitiesTec-GetHTML .
5. Asignamos el cuerpo del correo y a quien enviamos.
6. Invocamos servicio de notificación correo MessageTec.



#### Lista de objetos BuildInvoices

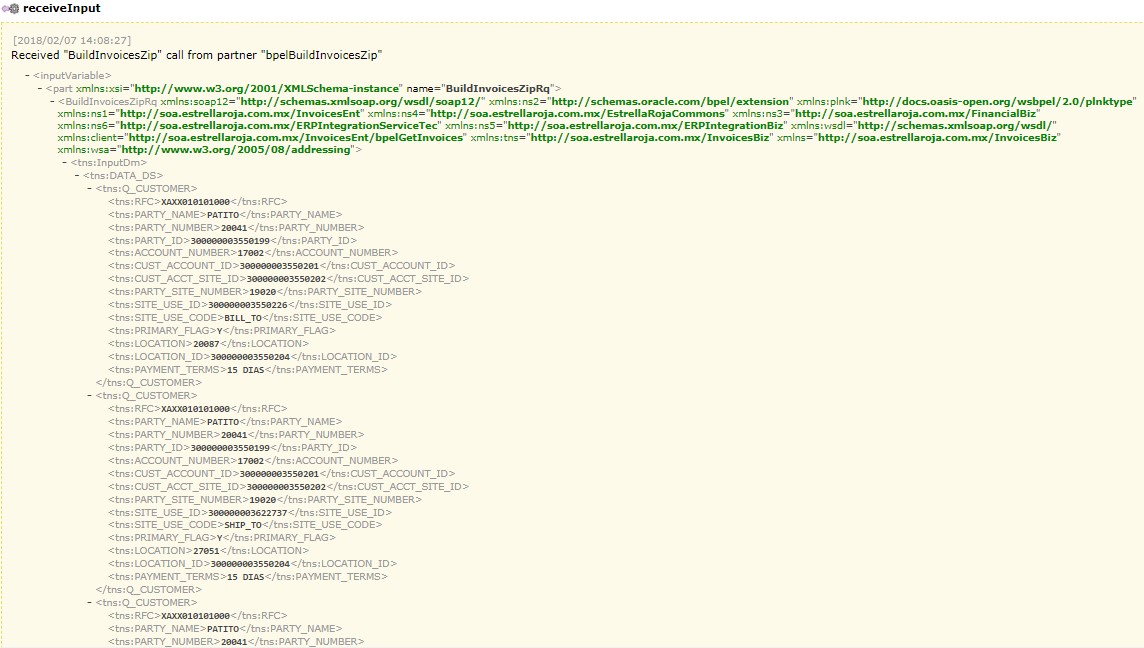
La siguiente tabla contiene la lista de objetos que se utilizaron para la funcionalidad de la construcción de del archivo CSV que contiene las facturas y notas de crédito y construye también el Zip que comprime al CSV.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Nombre | Tipo | Descripción |
| 1 | receiveInput | Entrada | Recibe los parámetros del bpel |
| 2 | sequenceMain | Secuencia | Secuencia de asignaciones |
| 3 | scopeBuildInvoicesZip | Alcance | Define un nuevo alcance |
| 4 | sequenceBuildInvoicesZip | Secuencia | Secuencia de asignaciones |
| 5 | assignRq | Asignación | Asignación de datos |
| 6 | AssignRs | Asignación | Asignación de datos |
| 7 | trInputDm\_to\_WsGetInvoicesReq | Transformación | Transformación de esquemas |
| 8 | InvokeWsFinancialsTecGetReciptOrInv oices | Invocación | Invocación de servicio |
| 9 | IfInvoicesToCsv | Condición | Condición de datos |
| 10 | scopeIfInvoicesToCsv | Alcance | Define un nuevo alcance |
| 11 | sequenceIfInvoicesToCsv | Secuencia | Secuencia de asignaciones |
| 12 | trWsFinancialsTecGetReceiptOrInvoice ToBpel | Transformación | Transformación de esquemas |
| 13 | IfInfoTranslated | Condición | Condicionante de datos |
| 14 | scopeIfInfoTranslated | Alcance | Define un nuevo alcance |
| 15 | sequenceIfInfoTranslated | Secuencia | Secuencia de asignaciones |
| 16 | translateInvoicesToCsvToString | Traductor | Traduce el contenido a cadena (en este caso) |
| 18 | assignStringInvoicesToWsSoaUtils | Asignación | Asignación de datos |
| 19 | trGetTrxNumberInProcess | Transformación | Transformación de esquemas |
| 20 | invokeWsSoaUtilitiesTecToBpel | Invocación | Invocación de servicio |
| 21 | trInvoicesBase64ToEncode64Zip | Transformación | Transformación de esquemas |
| 22 | invokeWsSoaUtiltiesTecEncodeFileBas e64Rq | Invocación | Invocación de servicio |
| 23 | assignFileBase64 | Asignación | Asignación de datos |
| 24 | InvokeWsSoaUtilitiesGenerateFileZip | Invocación | Invocación de servicio |
| 25 | trSetDatToOutput | Transformación | Transformación de esquemas |
| 26 | replyOutput | Respuesta | Respuesta de bpel síncrono |

#### Pruebas BuildInvoices

Con base en la prueba general que se hizo, se presentara a continuación los parámetros de entrada y respuesta de la operación:

**Request**:

 **Response:**



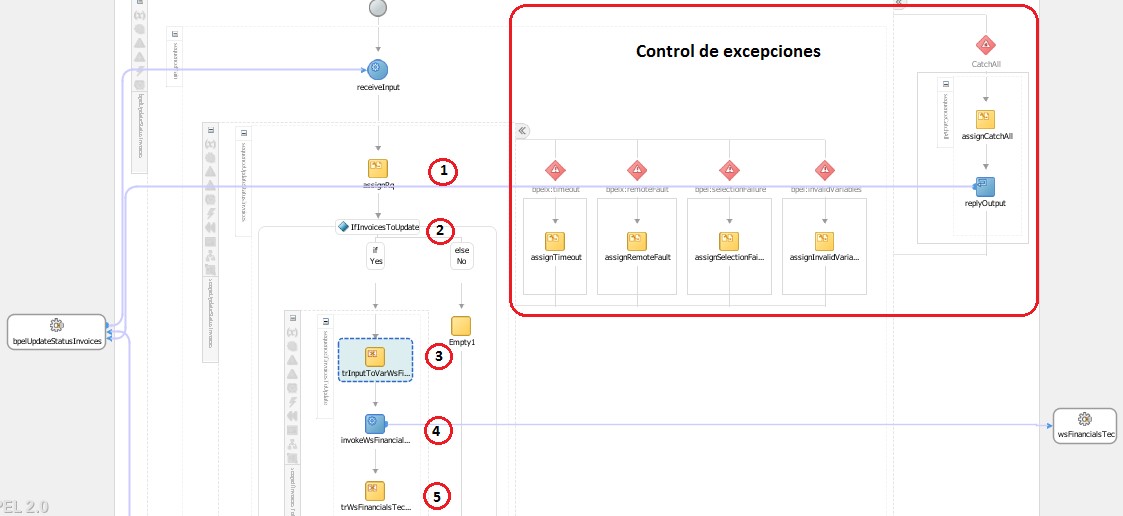
### UpdateStatusInvoices

UpdateStatusInvoices es otra operación de la capa Biz, esta operación se encarga de modificar el estado de las facturas y notas de crédito, dependiendo del éxito que haya tenido la importación al ERP-Cloud se actualizará en BD el estado, “S” para indicar que la carga fue exitosa y “E” para indicar que se tuvieron errores en la carga. A continuación, se detallan los componentes del bpel bpelUpdateStatusInvoices.

1. Se agrega asignación para establecer valor a literal común del bpel.
2. Se agrega IF para validar que los parámetros para el bpel no vengan nulos, si no están nulos entonces se puede continuar con el flujo.
3. Se usa la transformación (trInputToVarWsFinancialsTecUpdateStatusInvoice)

Lógica de implementación

1. Se invoca el servicio “FinancialsTecUpdateStatusInvoices”-“UpdateStatusInvoice” para ejecutar el servicio de la capa Tec que interactúa directamente con la BD y actualiza el estado de las facturas o notas de crédito asignando “S” en caso de que se haya hecho correctamente la importación y “E” en caso de que hayan surgido errores.
2. Se usa transformación (trWsFinancialsTecUpdateStatusInvoices) para asignar los resultados de éxito de la actualización a la variable de salida del bpel. **El bpel incluye manejador de excepciones, se puede apreciar en el recuadro rojo de la imagen.**



#### Lista de objetos UpdateStatusInvoices

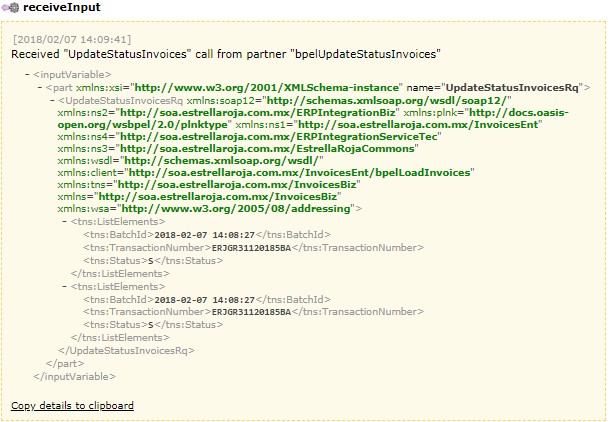
La siguiente tabla contiene la lista de objetos que se utilizaron para la funcionalidad de actualizar el estado de las facturas en base de datos con base en el éxito de carga al ERP-Cloud.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Nombre | Tipo | Descripción |
| 1 | ReceiveInput | Entrada | Recibe los parámetros de entrada |
| 2 | scopeUpdateStatusInvoices | Alcance | Define un nuevo alcance |
| 3 | seqenceUpdateStatusInvoices | Secuencia | Secuencia de asignaciones |
| 4 | AssignRq | Asignación | Asignación de datos |
| 5 | IfInvoicesToUpdate | Condición | Condicionante de datos |
| 6 | scopeIfInvoicesToUpdate | Alcance | Define un nuevo alcance |
| 7 | sequenceIfInvoicesToUpdate | Secuencia | Secuencia de asignaciones |
| 8 | TrInputToVarWsFinancialsTecUpdateStatusInvoic e | Transformación | Transformación de esquemas |
| 9 | invokeWsFinancialsTecUpdateStatusInvoices | Invocación | Invocación de servicio |
| 10 | trWsFinancialsTecUpdateStatusInvoices | Transformación | Transformación de esquemas |
| 11 | assignRs | Asignación | Asignación de datos |
| 12 | replyOutput | Salida | Respuesta de bpel síncrono |

#### Pruebas UpdateStatusInvoices

Con base en la prueba general que se hizo, se presentara a continuación los parámetros de entrada y respuesta de la operación:

**Request:**



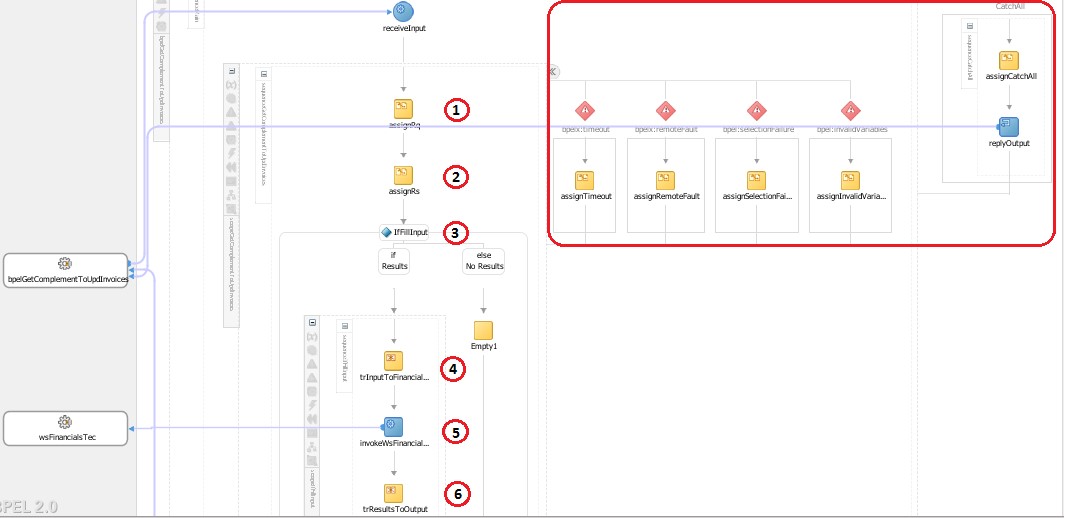
**Response:**



### GetComplementToUpdateInvoices

GetComplementToUpdateInvoices es otra operación del servicio InvoicesBiz, esta operación se encarga de recuperar campos necesarios para actualizar el UuId de las facturas que pudieron cargarse correctamente al ERPCloud, los campos a recuperar es el ORGANIZATION\_ID y el UUID. A continuación, se detallan los componentes del bpel.

1. Se asignan valores a literales comunes del bpel.
2. Se asignan valores a literales comunes del bpel.
3. Se usa IF para validar que los parámetros del bpel no vengan nulos, si no son nulos entonces se puede continuar con el flujo.
4. Se usa transformación (trInputToFinancialsTecGetComplementInvoices) para pasar los parámetros del bpel a variable request para invocar el servicio del siguiente paso.
5. Se invoca el servicio “FinancialsTec”-“GetComplementInvoices”, para interactuar con la BD y obtener la información complementaria (ORGANIZATION\_ID y UUID) para actualizar el UuId de las facturas y notas de crédito que pudieron cargarse al ERP-Cloud.
6. Se usa transformación (trResultsToOutput) para pasar los datos de respuesta de la invocación del servicio del paso anterior a la variable de salida del bpel.



#### Lista de objetos GetComplementInvoice

La siguiente tabla muestra la lista de objetos que fueron utilizados para la funcionalidad de la recuperación de información complementaria para la actualización del UuId de las facturas y notas de crédito en el ERP-Cloud.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Nombre | Tipo | Descripción |
| 1 | receiveInput | Entrada | Recibe los parámetros del bpel |
| 2 | scopeGetComplementToUpdInvoices | Alcance | Define un nuevo alcance |
| 3 | sequenceGetComplementToUpdInvoices | Secuencia | Secuencia de asignaciones |
| 4 | assignRq | Asignación | Asignación de datos |
| 5 | assignRs | Asignación | Asignación de datos |
| 6 | IfFillInput | Condición | Condicionante de datos |
| 7 | trInputToFinancialsTecGetComplementInvoices | Transformación | Transformación de esquemas |
| 8 | scopeIfFillInput | Alcance | Define un nuevo alcance |
| 9 | SequenceIfFillInput | Secuencia | Secuencia de asignaciones |
| 10 | invokeWsFinancialsTecGetComplementInvoices | Invocación | Invocación de servicio |
| 11 | trResultsToOutput | Transformación | Transformación de esquemas |
| 12 | replyOutput | Respuesta | Respuesta de bpel síncrono |

#### Pruebas GetComplementInvoices

Con base en la prueba general que se hizo, se presentara a continuación los parámetros de entrada y respuesta de la operación:

**Request:**



Lógica de implementación

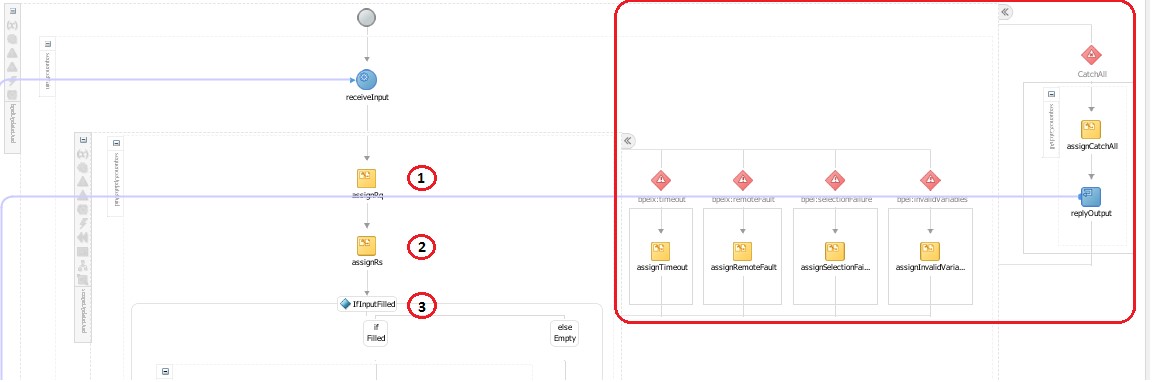
**Response:**



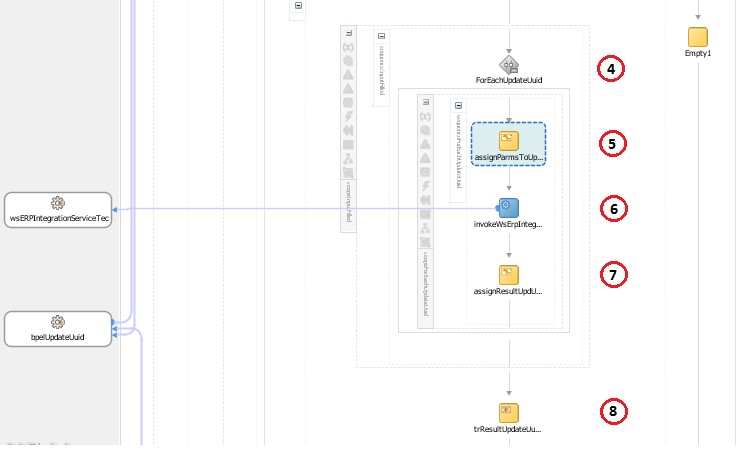
### UpdateUuId

UpdateUuId es otra operación del servicio InvoicesBiz, que se utiliza para el flujo de la solución, esta operación se encarga de actualizar el UuId de las facturas y notas de crédito que pudieron cargarse en el ERP-Cloud y de notificar el éxito de la actualización vía correo electrónico. A continuación, se detallan los componentes del bpel.

1. Se asignan valores a literales comunes del bpel.
2. Se asignan valores a literales comunes del bpel.
3. Se agrega IF para validar que los parámetros del BPEL no vengan nulos, si no están nulos se puede continuar con el flujo. **Se agrega manejador de excepciones al alcance principal.**

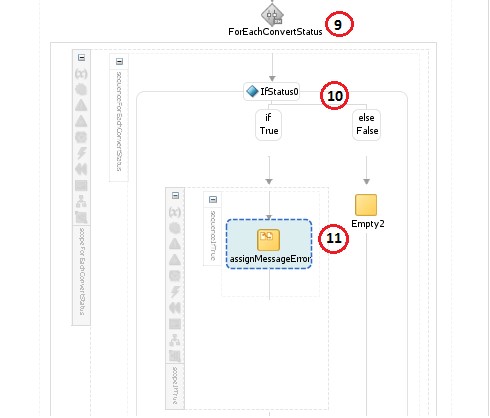


1. FOR EACH para iterar la lista de parámetro de facturas a actualizar.
2. Se usa asignación (assignParmsToUpdateObjectRq) para establecer valores a variable request para invocar servicio del siguiente paso.
3. Se invoca servicio “ErpIntegrationServiceTec”-“UpdateObject”, para mandar a actualizar el UuId la factura o nota de crédito.
4. Asignación (assignResultUpdUuIdToListResults) para recuperar en una variable los resultados de actualización del UuId.
5. Se agrega transformación (trResultUpdateUuIdToResultUpdateUuIdResult) para tomar los datos de paso anterior y pasarlo a una variable de similar estructura y se transforman algunos campos que se utilizaran en los siguientes pasos.



1. FOR EACH para iterar la colección de datos que se recuperó en el paso anterior, para posteriormente agregar funcionalidad de identificar los mensajes de respuesta del servicio anterior y determinar si se trata de un error o de un éxito.
2. IF para evaluar la respuesta del servicio del paso 6, si es “0” entonces se trata de un error si es “1" se trata de un éxito, si es “0” se entra al alcance del IF para hacer conteo de los errores.
3. Asignación (assignMessageError) para agregar mensaje de error a variable de lista de errores y agrega una unidad de error a la variable que hace el conteo de los errores de actualización de UuId.

Lógica de implementación



1. Asignación (assignTotalUpdUuId) para asignar a variable de tipo lista, los mensajes de error que llegaron a presentarse en los pasos anteriores y asigna también el valor de la variable del conteo de errores de actualización de UuId.
2. Se usa transformación (trParamsHtmlToSoaUtilitesTecGetHtmlControl) para asignar los valores recuperados en el paso anterior y establecerlos a la variable request para la invocación del servicio del próximo paso.
3. Se invoca el servicio “SoaUtilitiesTec”-“GetHtmlControl”, para obtener el los resultados de actualización en etiquetas HTML, contenido que se mandará en correo electrónico para notificar sobre la actualización.
4. Asignación (assignHtmlResult) para recuperar los datos devueltos por el servicio del paso anterior y establecerlos a la variable request para la invocación del servicio del siguiente paso.
5. Se invoca el servicio “MessageTec”-“SendMail”, para notificar vía correo electrónico, los resultados de la actualización del UuId
6. Transformación (trResultUpdateUuIdToOutput) para recuperar los mensajes de resultado de la ejecución del servicio del paso anterior y pasarlo a la variable de salida del bpel.



#### Lista de objetos UpdateUuId

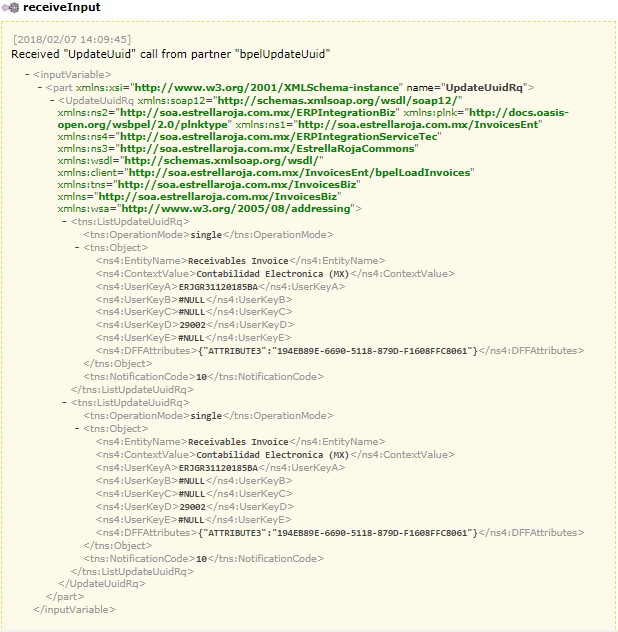
La siguiente tabla muestra la lista de objetos que se ocuparon para la funcionalidad de actualizar el UuId de las facturas y notas de crédito que se cargaron al ERP-Cloud.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Nombre | Tipo | Descripción |
| 1 | ReceiveInput | Entrada | Recibe los parámetros de entrada del bpel |
| 2 | scopeUpdateUuid | Alcance | Define un nuevo alcance |
| 3 | sequenceUpdateUuid | secuencia | Secuencia de asignaciones |
| 4 | assignRq | Asignación | Asignación de datos |
| 5 | assignRs | Asignación | Asignación de datos |
| 6 | IfInputFilled | Condición | Condición de datos |
| 7 | scopeInputFilled | Alcance | Define un nuevo alcance |
| 8 | sequenceInputFilled | Secuencia | Secuencia de asignaciones |
| 9 | ForEachUpdateUuid | Ciclo | Ciclo de registros |
| 10 | scopeForEachUpdateUuid | Alcance | Define un nuevo alcance |
| 11 | sequenceForEachUpdateUuid | Secuencia | Secuencia de asignaciones |
| 12 | assignParmsToUpdateObjectRq | Asignación | Asignación de datos |
| 13 | invokeWsErpIntegrationService TecUpdateObject | Invocación | Invocación de servicio |
| 14 | assignResultUpdUuIdToListRes ults | Asignación | Asignación de datos |
| 15 | trResultUpdateUuIdToResultUp dateUuIdResult | Transformación | Transformación de esquemas |
| 16 | ForEachConvertStatus | Ciclo | Ciclo de registros |
| 17 | scopeForEachConvertStatus | Alcance | Define un nuevo alcance |
| 18 | sequenceForEachConvertStatus | Secuencia | Secuencia de asignaciones |
| 19 | assignMessageError | Asignación | Asignación de datos |
| 20 | assignTotalUpdUuId | Asignación | Asignación de datos |
| 21 | trParamsHtmlToSoaUtilitesTecG etHtmlControl | Transformación | Transformación de esquemas |
| 22 | invokeWsSoaUtilitiesTecGetHtm lControl | Invocación | Invocación de servicio |
| 23 | assignHtmlResult | Asignación | Asignación de datos |
| 24 | invokeWsMessageTecSendMail | Invocación | Invocación de servicio |
| 25 | trResultUpdateUuIdToOutput | Transformación | Transformación de esquemas |
| 26 | replyOut | Respuesta | Respuesta síncrona de bpel |

#### **Pruebas UpdateUuId**

Con base en la prueba general que se hizo, se presentara a continuación los parámetros de entrada y respuesta de la operación:

**Request:**



**Response:**



# Diseño de Datos

A continuación, se listan los componentes SOA utlizados en las capas Ent y Biz.

## Origen de Datos

|  |  |  |
| --- | --- | --- |
| **Tabla** | **Columna** | **Comentarios** |
| XXER\_TBL\_PUENTE\_HEADER | RFC\_DESTINO | RFC del cliente a agregar Transacción |
|  | FECHA\_MOVIMIENTO | Fecha de transacción |
|  | NUMERO\_DOCUMENTO | Numero de transacción |
|  | ADICIONAL2 | Campo utilizado para almacenar el BATCH\_ID |
|  | TABLA\_PUENTE\_ID | Clave Primaria de la tabla |
|  |  |  |
| XXER\_CNF\_INTERFASE\_INGRESOS\_A | UNIDAD\_NEGOCIO\_ORIGEN | Unidad de negocio |
| R | TIPO\_TRANSACCION\_AR | Tipo de transacción |
|  | MEMO\_LINE | Cadena MEMOLINE |
|  | ORGANIZATION\_ID | Numero de organización |
|  |  |  |
| XXER\_TBL\_PUENTE\_LINES | LINEA\_ARTICULO | Línea |
|  | SUBTOTAL | Subtotal de monto de transacción |
|  | TABLA\_PUENTE\_LINEA\_ID | Clave primaria de la tabla |
|  |  |  |
| XXER\_TBL\_PUENTE\_AUX | UUID | CDFI de la transacción |

## Lógica de Validación

Las principales validaciones de la integración de transacciones son las siguientes

* El proceso solo toma en cuenta a aquellas facturas que no tengan el status “S” (Success) que significa éxito de carga, pudiendo tomar a aquellas facturas o notas de crédito que tengan el status “E” (Error) para reintentar cargarlas o bien status en nulo para procesar por primera vez la carga.
* Ya que se utilizan reportes del OTBI de Cloud, aquellos RFC a los que no se les encuentre cliente cargado en la nube, no se considerará para la carga de facturas o notas de crédito
* La notificación de carga de facturas y notas de crédito se hace por línea cargada.
* Aquellas facturas o notas de crédito que hayan logrado cargarse al ERP-Cloud pero no tenga ORGANIZATION\_ID válido, no se les podrá actualizar su UuId.
* No se tomará en cuenta la actualización de UuId de aquellas facturas o notas de crédito que hayan presentado algún error de carga.

La actualización de status de facturas y notas de crédito se hará ya sea por factura o por Nota de crédito, no por línea.

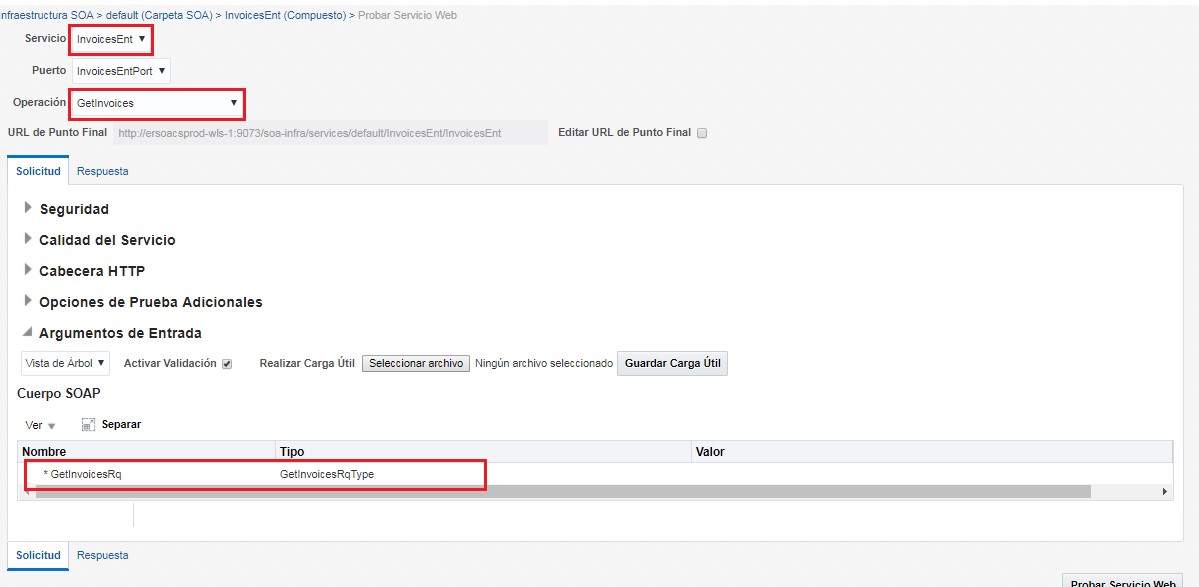
# Reglas de Negocio

Las reglas de negocio para esta integración son las siguientes:

* Validar el estado de las facturas por cargar, ya que solo se cargarán aquellas que no se hayan cargado correctamente antes.
* Validar que el cliente con el que se tiene referencia en BD On Premise, exista en la base de Cloud, ya que si no se encuentra no se podrán derivar datos y no podrá cargarse la información.
* No se podrán cargar Facturas o notas de crédito repetidas.
* Es necesario importar primero la factura o nota de crédito para poder asignar el UuId o CDFI, ya que si se intenta cargar desde la primera importación esto no será posible

## Diseño del Servicio

Para poder iniciar la integración es necesario ejecutar el compuesto **InvoicesEnt – GetInvoices**; para esta ejecución no se requieren parámetros.



# Consideraciones de Rendimiento

Este requerimiento ha sido probado con un conjunto de facturas y notas de crédito pequeñas, tener contemplado que el compuesto puede presentar demora con una cantidad de registros muy grande a procesar.

## Estrategia de Reinicio

* Se pude reiniciar la integración, siempre y cuando no existan instancias de SOA corriendo, pues de lo contrario serán abortadas.
* En caso de existir instancias en ejecución, esperar hasta su terminación.

## Seguridad

N/A

## Personalización

* En caso de requerir modificar el proyecto, tomar la versión más reciente del controlador de versiones con el que se cuente.
* Revisar que los servicio comunes estén desplegados.
* La versión del IDE de desarrollo de JDeveloper con la que se implementó dicha solución es JDEVADF\_12.2.1.2.0\_GENERIC\_161008.1648.S

## Catálogo de Errores

El código de error que se maneja para esta integración se especifica en el documento

**“CatalogosErroresEstrellaRoja.xlsx”**.

# URL de Acceso y seguridad

En la siguiente tabla se define las URL que se utilizan en esta integración:

|  |  |
| --- | --- |
| Servicio | URL |
| InvoicesEnt | http://<host>:<puerto>[/soa-infra/services/PROD/InvoicesEnt/InvoicesEnt?WSDL](http://ersoacsprod-wls-1:9073/soa-infra/services/default/InvoicesEnt/InvoicesEnt?WSDL) |
| InvoicesBiz | http://<host>:<puerto>[/soa-infra/services/PROD/InvoicesBiz/InvoicesBiz?WSDL](http://ersoacsprod-wls-1:9073/soa-infra/services/default/InvoicesBiz/InvoicesBiz?WSDL) |
| ERPManagementOfLoadImpo rtBiz | http://<host>:<puerto>[/soa-](http://ersoacsprod-wls-1:9073/soa-infra/services/default/ERPManagementOfLoadImportBiz/ERPManagementOfLoadImportBiz?WSDL)  [infra/services/PROD/ERPManagementOfLoadImportBiz/ERPManagementOfLoadImp ortBiz?WSDL](http://ersoacsprod-wls-1:9073/soa-infra/services/default/ERPManagementOfLoadImportBiz/ERPManagementOfLoadImportBiz?WSDL) |

URL de Acceso y seguridad

# Temas abiertos y cerrados

## Temas Abiertos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Tema** | **Solución** | **Responsabilidad** | **Fecha Objetivo** | **Fecha impacto** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Temas Cerrados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Tema** | **Solución** | **Responsabilidad** | **Fecha Objetivo** | **Fecha impacto** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |