|  |
| --- |
|  |

OUM

TA.070-8 DISEÑO TÉCNICO DE EXTENSIONES E

INTERFACES

Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

ER25 AR Conciliacion Cash TP TMS

|  |  |
| --- | --- |
| Autor: | Forte Innovation |
| Fecha de creación: | 21 de octubre de 2021 |
| Última actualización: | 21 de octubre de 2021 |
| Código de referencia OUM: |  |
| Versión:  **Aprobadores:** | 3.1 |
| Juan Gonzalez |  |

Gregorio Flores

|  |  |
| --- | --- |
| Wendi Holguin |  |

Miryan Lavin



**Contenido**

1 Control de Documento 2

1.1 Bitácora de Cambios 2

1.2 Revisores 2

2 Resumen técnico 3

3 Diagrama de la integración 4

4 Lógica de implementación 5

4.1 Calendarización y Ejecución 5

4.2 ConciliationEnt - SftpReadBancoER 7

4.3 ConciliationEnt - SftpReadBancoAmex 13

4.4 ConciliationEnt - Conciliation 18

4.5 ConciliationBiz– ConciliationCashTP 25

4.6 ConciliationTec 39

5 Diseño de Datos 54

5.1 Origen de Datos 54

5.2 Lógica de Validación 55

6 Diseño SQL 56

6.1 Sentencias SQL 56

7 Reglas de Negocio 57

7.1 Diseño del Servicio 58

8 Consideraciones de Rendimiento 59

8.1 Estrategia de Reinicio 59

8.2 Seguridad 59

8.3 Personalización 59

8.4 Catálogo de Errores 60

9 Consideraciones de Instalación 61

10 URL de Acceso y seguridad 62

11 Temas abiertos y cerrados 63

11.1 Temas Abiertos 63

11.2 Temas Cerrados 63

# Control de Documento

## Bitácora de Cambios

3

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Referencia del cambio** |
| 16/12/21 | Pedro Sanchez | 3.1 | Cambio en reportes APEX para agregar campos diferencia y fecha de pago |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Revisores

|  |  |
| --- | --- |
| **Nombre** | **Posición** |
|  |  |
|  |  |
|  |  |
|  |  |

# Resumen técnico

Esta especificación documenta el diseño para conciliación de pagos de Cash contra las transacciones de tabla puente de cloud.

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



# Lógica de implementación

Como mencionamos anteriormente, se cuenta con tres capas en la integración de Conciliación , en esta sección describiremos la lógica que se utilizó en cada una de ellas.

## Calendarización y Ejecución

A continuación, se describe la programación de esta capa en el ESS, consta de 1 elemento, un job que se ejecuta cada ? minutos:

|  |  |  |
| --- | --- | --- |
| Componente | Propiedad | Valor |
| Trabajo | Nombre | JobCocilicaicon |
| Nombre mostrado | JobCocilicaicon |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | |  |  | | --- | --- | |  | Ejecucion de servicio conciliacion cash TMS TP | |
| Tipo de trabajo | SyncWebserviceJobType |
| Operación | Conciliation |
| Solicitud | <ns1:ConciliationEntRq xmlns:ns1="http://soa.estrellaroja.com.mx/ConciliationEnt"> <ns1:BussinessUnit></ns1:BussinessUnit> <ns1:FromDate></ns1:FromDate> <ns1:PaymentReference></ns1:PaymentReference> <ns1:ReceiptMethod></ns1:ReceiptMethod> </ns1:ConciliationEntRq> |
| Planificación | Nombre | SchedulerConciliacion |
| Nombre mostrado | SchedulerConciliacion |
| Paquete | |  |  | | --- | --- | |  | /oracle/apps/ess/custom/soa | |
| Descripción | |  | | --- | | Scheduler de ejecucion de conciliacion cash tms TP | |
| Frecuencia | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Diariamente | Cada | 1 |  | Día(s) | |
|  | Zona horaria | |  |  | | --- | --- | |  | Hora estándar Central | |

A continuación, se describe la pantalla de ejecución manual así como valores no obligatorios de ejecución, la cual estará disponible en la siguiente URL:

<https://129.159.106.1/ords/ERSEPDB1/f?p=133>

Tabla

Descripción generada automáticamente

Ejecución de Job de base de datos de obtención de información de TMS es diariamente.

|  |  |
| --- | --- |
| Nombre | Valor |
| OWNER | PXERINTUSER |
| JOB\_NAME | XXER\_EJECUTAR\_CONCILIACION\_TMS |
| JOB\_CLASS | DEFAULT\_JOB\_CLASS |
| COMMENTS | Job que ejeucta paquete que concilia la informacion de TMS con TP AFIL y AMEX |
| ENABLED | TRUE |
| CREDENTIAL\_NAME |  |
| DESTINATION |  |
| PROGRAM\_NAME |  |
| JOB\_TYPE | PLSQL\_BLOCK |
| JOB\_ACTION | DECLARE  p\_businessUnit VARCHAR2(200);  p\_fromDate VARCHAR2(200);  p\_paymentReference VARCHAR2(200);  p\_metodo\_recibo VARCHAR2(200) ;  p\_error NUMBER;  p\_error\_desc VARCHAR2(300);  BEGIN  XXER\_CONCILIATION\_PKG.GET\_CONCILIATION\_TMS(p\_error\_desc => p\_error\_desc,p\_error => p\_error,p\_metodo\_recibo=> p\_metodo\_recibo,p\_paymentReference=> p\_paymentReference,p\_fromDate=>p\_fromDate,p\_businessUnit=>p\_businessUnit); commit; EXCEPTION when OTHERS then rollback; RAISE; END; |
| START\_DATE | 14/11/21 11:33:55.000000000 AMERICA/MEXICO\_CITY |
| REPEAT\_INTERVAL | FREQ=DAILY;BYTIME=060000;BYDAY=MON,TUE,WED,THU,FRI,SAT,SUN |

## ConciliationEnt - SftpReadBancoER

En la capa ConciliationEnt – SftpReadBancoER se realiza una serie de pasos concentrados en un bpel como se muestra en la siguiente imagen: Diagrama, Esquemático

Descripción generada automáticamente

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 |  |
| SFTP |  |

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** |
| SftpReadBancoER | FTP Adapter | Adaptador a escucha de un directorio |
| MediatorSftpReadBancoER | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| BpelConciliationBankER | Bpel | Componente de Orquestación |
| SftpTec | OSB  Service | Servicio externo que tiene la lógica de operaciones a SFTP |
| ConciliationTec | OSB    Service | Servicio externo que Reliza operaciones Relacionadas a la conciliación |
| SOAUtilitiesTec | Web    Service | Servicio de Utilidades Custom |
| MessageTec | Web    Service | Servicio de envió de emails |

### Diagrama de integración



### BpelConciliationBankER

Como servicio expuesto de lado izquierdo tenemos la operación **SftpReadBancoER** que es la operación que se invoca cuando es depositado un archivo en el buzón SFTP del banco, posterior utilizamos un mediator para dirigir a **BpelConciliationBankER** donde se concentra la lógica para realizar la sincronización de la información del buzón del banco con la BD de CLOUD.

El bpel realiza las siguientes acciones:

1. Se recibe una entrada del archivo en B64 se decodifica para el manejó de la información .
2. Se utiliza un NXSD que nos ayuda a darle formato a la información decodificada sea manipulable como XML.
3. Realizamos el mapeo para poblar la variable que le enviaremos al servicio ConciliationTec- SendConciliationBankRq.
4. Invocamos el servicio ConciliationTec- SendConciliationBank
5. Validamos si el Resultado es incorrecto.
6. Ejecutamos un error para que termine el proceso.
7. Asignamos las variables para enviar a la carpeta de errores el archivo que no se importó de forma correcta.
8. Invocamos servicio SftpTec- PutFileBancoER el cual coloca un archivo en el SFTP
9. Asignamos el error a request que genera el msj de notifiacion.
10. Invocamos el servicio que genera msj de notificación SOAUtilitiesTec- GetHtmlControl.
11. Asignamos el msj de error al requets del servicio de envio de Email.
12. Invocamos servicio de notifiaciones de email MessageTec- SendEmail.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Asignamos variable para ejecución de servicio que coloca archivo en carpetas de correctos.
2. Invocamos servicio que envia archivo procesado a carpeta en SFTP SftpTec- PutFileBancoER.

Gráfico, Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del BpelConciliationBankER

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | Java\_decode | Java Enbedding | Ejecuta código java |
| 2 | Trs\_StringFileToXMLPayments | Traslate | Parsea información a XML |
| 3 | Trs\_sendCociliationBank | Transformation | Transformación de datos. |
| 4 | WsConciliationTec | invoke | Invocación de Servicio. |
| 5 | If\_Error | if | Condicional |
| 6 | Throw | Throw | Excepción-genera error |
| 7 | AssignPutError | assign | Asignación de valores. |
| 8 | WsSftpTec | invoke | Invocación de Servicio. |
| 9 | AssignErrorMsg | assign | Asignación de valores. |
| 10 | WsSOAUtilities | invoke | Invocación de Servicio. |
| 11 | assignHtmlResult | assign | Asignación de valores. |
| 12 | WsMessage | invoke | Invocación de Servicio. |
| 13 | AssignPutSuccess | assign | Asignación de valores. |
| 14 | WsSftpTec | invoke | Invocación de Servicio. |

### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo colocando los adjuntos en la ruta del SFTP del banco.

**sftp.mitec.com.mx** **/Outbox**

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



Response: El servicio no retorna nada.

## ConciliationEnt - SftpReadBancoAmex

En la capa ConciliationEnt – SftpReadBancoAmex se realiza una serie de pasos concentrados en un bpel como se muestra en la siguiente imagen: Diagrama

Descripción generada automáticamente

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 |  |
| SFTP |  |

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** |
| SftpReadBancoAmex | FTP Adapter | Adaptador a escucha de un directorio |
| MediatorSftpReadBancoAmex | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| BpelConciliationBankAmex | Bpel | Componente de Orquestación |
| ConciliationTec | OSB    Service | Servicio externo que Reliza operaciones Relacionadas a la conciliación |
| SOAUtilitiesTec | Web    Service | Servicio de Utilidades Custom |
| MessageTec | Web    Service | Servicio de envió de emails |

### Diagrama de integración



### BpelConciliationBankAmex

Como servicio expuesto de lado izquierdo tenemos la operación **SftpReadBancoAmex** que es la operación que se invoca cuando es depositado un archivo en el buzón SFTP del banco, posterior utilizamos un mediator para dirigir a **BpelConciliationBankAmex** donde se concentra la lógica para realizar la sincronización de la información del buzón del banco con la BD de CLOUD.

El bpel realiza las siguientes acciones:

1. Se recibe una entrada del archivo en B64 se decodifica para el manejó de la información .
2. Se utiliza un NXSD que nos ayuda a darle formato a la información decodificada sea manipulable como XML.
3. Realizamos el mapeo para poblar la variable que le enviaremos al servicio ConciliationTec- SendConciliationBankAMEXRq.
4. Invocamos el servicio ConciliationTec- SendConciliationBank
5. Asignamos el error a request que genera el msj de notifiacion.
6. Invocamos el servicio que genera msj de notificación SOAUtilitiesTec- GetHtmlControl.
7. Asignamos el msj de error al requets del servicio de envio de Email.
8. Invocamos servicio de notifiaciones de email MessageTec- SendEmail.

Diagrama

Descripción generada automáticamente

### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del BpelConciliationBankER

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | Java\_decode | Java Enbedding | Ejecuta código java |
| 2 | Trs\_StringFileToXMLPayments | Traslate | Parsea información a XML |
| 3 | Trs\_SendConciliationBankAMEXRq | Transformation | Transformación de datos. |
| 4 | WsConciliationTec | invoke | Invocación de Servicio. |
| 5 | AssignErrorMsg | assign | Asignación de valores. |
| 6 | WsSOAUtilities | invoke | Invocación de Servicio. |
| 7 | assignHtmlResult | assign | Asignación de valores. |
| 8 | WsMessage | invoke | Invocación de Servicio. |

### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo colocando los adjuntos en la ruta del SFTP del banco.

**fsgateway.americanexpress.com/outbox**

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



Response: El servicio no retorna nada.

## ConciliationEnt - Conciliation

En la capa ConciliationEnt – Conciliation se realiza una serie de pasos concentrados en un bpel como se muestra en la siguiente imagen: Diagrama

Descripción generada automáticamente

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** |
| ConciliationEnt | Web Service | Web service de ejecucion |
| MediatorConciliationEnt | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| BpelGetConciliation | Bpel | Componente de Orquestación |
| ConciliationBiz | Web  Service | Servicio externo que tiene la lógica de negocio de la conciliacion |
| ConciliationTec | OSB    Service | Servicio externo que Reliza operaciones Relacionadas a la conciliación |
| SOAUtilitiesTec | Web    Service | Servicio de Utilidades Custom |
| MessageTec | Web    Service | Servicio de envió de emails |

### Diagrama de integración



### BpelGetConciliation

Como servicio expuesto de lado izquierdo tenemos la operación **ConciliationEnt** que es la operación que se invoca por medio del ESS de SOA o manualmente desde la pantalla de usuario de APEX, posterior utilizamos un mediator para dirigir a **BpelGetConciliation** donde se concentra la lógica para realizar La ejecución del Servicio de negocio **ConciliationBiz-ConciliationCashTp** que ejecuta la conciliación.

El bpel realiza las siguientes acciones:

1. Asignamos variables default del Response.
2. Asignamos las variables de entrada de la operación ConciliationTec-GetConciliationsRq si el servicio se ejecuta desde SOA.
3. Transformación que asigna datos de la operación ConciliationTec-GetConciliationsRq si la información proviene de APEX.
4. Invocamos el servicio ConciliationTec-GetConciliations
5. Validamos si el Resultado es incorrecto.
6. Ejecutamos un error para que termine el proceso.
7. Asignamos error a las variables del Response.
8. Asignamos el error a request que genera el msj de notifiacion.
9. Invocamos el servicio que genera msj de notificación SOAUtilitiesTec- GetHtmlControl.
10. Asignamos el msj de error al requets del servicio de envio de Email.
11. Invocamos servicio de notifiaciones de email MessageTec- SendEmail.
12. Replicamos la respuesta en el response para terminar el proceso.

Diagrama

Descripción generada automáticamente

1. Condición si hay resultados para conciliación.
2. Transformación de mapeo de datos para ejecutar servicio ConciliationBiz- ConciliationCashTPRq.
3. Invocamos el Servicio Web ConciliationBiz- ConciliationCashTP.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del BpelConciliationBankER

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | assignRs | Java Enbedding | Ejecuta código java |
| 2 | AssignVars | Traslate | Parsea información a XML |
| 3 | Trs\_conciliacion | Transformation | Transformación de datos. |
| 4 | WsConciliationTecGetConciliations | invoke | Invocación de Servicio. |
| 5 | If\_ErrorQuery | if | Condicional |
| 6 | Throw1 | Throw | Excepción-genera error |
| 7 | If\_conciliations | assign | Asignación de valores. |
| 8 | TrsConciliation | Transformation | Transformación de datos. |
| 9 | ConciliationBiz | invoke | Invocación de Servicio. |
| 10 | assignCatchAll | assign | Asignación de valores. |
| 11 | AssignErrorMsg | assign | Asignación de valores. |
| 12 | WsSOAUtilities | invoke | Invocación de Servicio. |
| 13 | assignHtmlResult | assign | Asignación de valores. |
| 14 | WsMessage | invoke | Invocación de Servicio. |
| 15 | AssignErrorMsg | assign | Asignación de valores. |
| 16 | Reply1 | Reply | Retorna respuesta servicio |

### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección del siguiente WSDL:

[**http://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/PROD/ConciliationEnt/ConciliationEnt?WSDL**](http://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/PROD/ConciliationEnt/ConciliationEnt?WSDL)

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

**Request:** No se necesitan parámetros obligatorios de entrada para la ejecución de esta operación si se puede agregar un filtro personalizado con los campos mostrados en la petición:

**Response:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<ns1:ConciliationEntRq xmlns:ns1="http://soa.estrellaroja.com.mx/ConciliationEnt">

<ns1:BussinessUnit></ns1:BussinessUnit>

<ns1:FromDate></ns1:FromDate>

<ns1:PaymentReference></ns1:PaymentReference>

<ns1:ReceiptMethod></ns1:ReceiptMethod>

</ns1:ConciliationEntRq>

</soap:Body>

</soap:Envelope>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## ConciliationBiz– ConciliationCashTP

En la capa ConciliationBiz – ConciliationCashTP se realiza una serie de pasos concentrados en un bpel como se muestra en la siguiente imagen: Diagrama

Descripción generada automáticamente

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 |  |
| Spring |  |

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** |
| ConciliationBiz | Web Service | Web service de ejecucion |
| MediatorConciliationBiz | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| BpelConciliationCashTp | Bpel | Componente de Orquestación |
| ERPIntegrationBiz | Web  Service | Servicio externo que tiene la lógica de consulta de reportes al ERP |
| ConciliationTec | OSB    Service | Servicio externo que Reliza operaciones Relacionadas a la conciliación |
| SOAUtilitiesTec | Web    Service | Servicio de Utilidades Custom |
| MessageTec | Web    Service | Servicio de envió de emails |

### Diagrama de integración



### BpelConciliationCashTp

Como servicio expuesto de lado izquierdo tenemos la operación **ConciliationBiz** que es la operación que se invoca por medio del web service **ConciliationEnt**, posterior utilizamos un mediator para dirigir a **BpelConciliationCashTp** donde se concentra la lógica para realizar la conciliación.

El bpel realiza las siguientes acciones:

1. Transformación que mapea los datos para la petición del servicio ConciliationSpring- GetPaymentsCashPageRq.
2. Invocamos servicio ConciliationSpring-GetPaymentsCashPage que realiza la paginación de las conciliaciones.
3. Iteramos el resultado de las páginas resultantes del servicio ConciliationSpring-GetPaymentsCashPageRs .
4. Usamos iterador para asignar paginas de pagos a conciliar a variable local.
5. asignación que asigna pago a conciliar a array de pagos.
6. Asignamos el error a request que genera el msj de notificación.
7. Invocamos el servicio que genera msj de notificación SOAUtilitiesTec- GetHtmlControl.
8. Asignamos el msj de error al requets del servicio de envió de Email.
9. Invocamos servicio de notificaciones de email MessageTec- SendEmail.

Diagrama

Descripción generada automáticamente

1. Transformación que mapea los datos para la petición del servicio ConciliationBiz- GetPaymentsCashRq.
2. Invocamos servicio ConciliationBiz- GetPaymentsCashRq que realiza la Extracción de un arreglo de pagos de la información de Cash desde el ERP.
3. Transformación que mapea los datos para la petición del servicio ConciliationTec- ConciliationCashRq donde enviamos la información obtenida de ConciliationBiz- GetPaymentsCashRs.
4. Invocamos servicio ConciliationTec- ConciliationCash que realiza la conciliación de la información obtenida y que se encuentra en tabla de conciliacion.
5. Transformación que mapea los datos para la petición del servicio ConciliationBiz- GetCreditMemoRq.
6. Invocamos servicio ConciliationBiz- GetCreditMemo que realiza obtención de notas de crédito en el ERP que tengan una referencia de los pagos que se están conciliando.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Condición si hay notas de crédito en el ConciliationBiz- GetCreditMemoRs.
2. Transformación que mapea los datos para la petición del servicio ConciliationTec- ConciliationCashRq donde enviamos la información obtenida de ConciliationBiz- GetCreditMemoRs.
3. Invocamos servicio ConciliationTec- ConciliationCash que realiza la conciliación de la información obtenida y que se encuentra en tabla de conciliación.
4. Asignamos el numero de registros procesados en request que genera el msj de notificación.
5. Invocamos el servicio que genera msj de notificación SOAUtilitiesTec- GetHtmlControl.
6. Asignamos el msj de error al requets del servicio de envió de Email.
7. Invocamos servicio de notificaciones de email MessageTec- SendEmail.

Diagrama

Descripción generada automáticamente

### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del BpelConciliationBankER

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | TrsGetPaginasRq | Transformation | Transformación de datos. |
| 2 | GetPaymanetsPage | invoke | Invocación de Servicio. |
| 3 | For\_Page | For | For each itera registros |
| 4 | For\_payments | For | For each itera registros |
| 5 | AssignPayment | assign | Asignación de valores. |
| 6 | TrsGetPaymentsCash | Transformation | Transformación de datos. |
| 7 | BpelGetPaymentsCash | invoke | Invocación de Servicio. |
| 8 | TrsConciliationCashTP | Transformation | Transformación de datos. |
| 9 | WsConciliationCashTP | invoke | Invocación de Servicio. |
| 10 | TrsGetCreditMemo | Transformation | Transformación de datos. |
| 11 | BpelGetGreditMemo | invoke | Invocación de Servicio. |
| 12 | If\_NOTAS\_CREDITO | if | condicional |
| 13 | TrsConciliationCashTransTP | assign | Asignación de valores. |
| 14 | WsConciliationCashTransTP | invoke | Invocación de Servicio. |
| 15 | AssignErrorMsg | assign | Asignación de valores. |
| 16 | WsSOAUtilities | invoke | Invocación de Servicio. |
| 17 | assignHtmlResult | assign | Asignación de valores. |
| 18 | WsMessage | invoke | Invocación de Servicio. |
| 19 | AssignMsg | assign | Asignación de valores. |
| 20 | WsSOAUtilities | invoke | Invocación de Servicio. |
| 21 | assignHtmlResult | assign | Asignación de valores. |
| 22 | WsMessage | invoke | Invocación de Servicio. |

### BpelGetPaymentsCash

Diagrama

Descripción generada automáticamente

Como servicio que consume de lado izquierdo tenemos la operación **BpelConciliationCashTp** que es la operación que se invoca por medio del web service **ConciliationBiz** Expuesto.

El bpel realiza las siguientes acciones:

1. For que itera los pagos recibidos en la petición.
2. Asignamos lo valores para ejecutar el servicio de consulta de reporte WsErpIntegrationBiz-GenerateReportRq.
3. Invocamos servicio WsErpIntegrationBiz-GenerateReport que consulta reporte en ERP.
4. Asignamos el valor a una variable y paseamos a XML para manipularla.
5. Asignacion de msj de error a response.
6. Reply de envio en caso de error.

Imagen que contiene mapa, interior, tabla, computer

Descripción generada automáticamente

1. Condición Si se encuentran pagos.
2. Asignamos los valores de cabecera del arreglo del pago o arreglo de pagos consultado.
3. For que recorre arreglo de pagos.
4. Asignamos los datos de cada pago a nivel línea.
5. Asigna solo el header de los pagos sin líneas.
6. Reply con respuesta de resultado success.

Interfaz de usuario gráfica, Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del BpelConciliationBankER

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | For\_PaymentsTp | For | For each itera registros |
| 2 | assignParams | assign | Asignación de valores. |
| 3 | WsErpIntegrationBiz | invoke | Invocación de Servicio. |
| 4 | AssignParseStringXML | assign | Asignación de valores. |
| 5 | If\_ExistPaymentAR | if | condicional |
| 6 | AssignHeader | assign | Asignación de valores. |
| 7 | For\_PaymentsCash | For | For each itera registros |
| 8 | AssignPaymentData | assign | Asignación de valores. |
| 9 | AssignPaymentPC | assign | Asignación de valores. |
| 10 | replyOutput | reply | Response servicio |
| 11 | assignCatchAll | assign | Asignación de valores. |
| 12 | replyOutput | reply | Response servicio |

### BpelGetCreditMemo

Diagrama

Descripción generada automáticamente

Como servicio que consume de lado izquierdo tenemos la operación **BpelConciliationCashTp** que es la operación que se invoca por medio del web service **ConciliationBiz** Expuesto.

El bpel realiza las siguientes acciones:

1. For que itera los pagos recibidos en la petición.
2. Asignamos lo valores para ejecutar el servicio de consulta de reporte WsErpIntegrationBiz-GenerateReportRq.
3. Invocamos servicio WsErpIntegrationBiz-GenerateReport que consulta reporte en ERP.
4. Asignamos el valor a una variable y paseamos a XML para manipularla.
5. Asignacion de msj de error a response.
6. Reply de envio en caso de error.

Diagrama

Descripción generada automáticamente con confianza baja

1. Condición Si se encuentran notas de credito.
2. Asignamos los valores de cabecera del arreglo de notas de crédito consultado.
3. For que recorre arreglo de notas de credito.
4. Asignamos los datos de cada pago a nivel línea.
5. Asigna solo el header de los pagos sin líneas.
6. Reply con respuesta de resultado success.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del BpelConciliationBankER

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | For\_PaymentsTpNCL | For | For each itera registros |
| 2 | assignParams | assign | Asignación de valores. |
| 3 | WsErpIntegrationBiz | invoke | Invocación de Servicio. |
| 4 | AssignParseStringXML | assign | Asignación de valores. |
| 5 | If\_ExistPaymentAR | if | condicional |
| 6 | AssignHeader | assign | Asignación de valores. |
| 7 | For\_CreditMemos | For | For each itera registros |
| 8 | AssignCreditMemoData | assign | Asignación de valores. |
| 9 | AssignPaymentPC | assign | Asignación de valores. |
| 10 | replyOutput | reply | Response servicio |
| 11 | assignCatchAll | assign | Asignación de valores. |
| 12 | replyOutput | reply | Response servicio |

### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección del siguiente WSDL:

[**http://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/DEV/ConciliationBiz/ConciliationBiz?WSDL**](http://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/DEV/ConciliationBiz/ConciliationBiz?WSDL)

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

**Request:** No se necesitan parámetros obligatorios de entrada para la ejecución de esta operación si se puede agregar un filtro personalizado con los campos mostrados en la petición:

**Response:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<ns1:ConciliationCashTPRq xmlns:ns1="http://soa.estrellaroja.com.mx/ConciliationBiz">

<ns1:Conciliate>

<ns1:ReceiptMethod>SANTANDER\_3550</ns1:ReceiptMe <ns1:PaymentReference>08052021CAPUT</ns1:PaymentReference>

</ns1:Conciliate>

</ns1:ConciliationCashTPRq>

</soap:Body>

</soap:Envelope>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## ConciliationTec

Este servicio web utiliza una lógica de proceso de varios componentes de OSB que se encargan de enrutar las peticiones hacía los adaptadores de base. A continuación se muestra la lógica de implementación:

1. Petición del cliente dependiendo de la operación que desee ejecutar.
2. Invocación de pipeline de acuerdo con la petición.

Diagrama

Descripción generada automáticamente

1. Después de haber ejecutado una de las operaciones, el componente pipeline responde la petición.(se utilizan los adaptadores enmarcados)

La implementación OBS mostrada en la imagen anterior, se hace uso de los componentes:

|  |  |
| --- | --- |
| **Componente** | **Icono del componente** |
| Adaptador DataBase |  |
| Tubería |  |
| Adaptador SB |  |

El módulo principal se denomina pipeline , 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. A continuación, se muestra la interface del asistente que se utiliza en tiempo de implementación:

### Lista de objetos

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** | **GIT** |
| 1 | ConciliationTecPs | Web Service | Servicio Expuesto |  |
| 2 | dbConciliation | Adaptador | Adaptador de DataBase |  |
| 3 | dbXXER\_BANK\_AMEX | Adaptador | Adaptador de DataBase |  |
| 4 | dbTMS\_COCILIACION\_BANCOS\_TBL | Adaptador | Adaptador de DataBase |  |
| 5 | ConciliationTecPL | Tubería | Flujo de tubería |  |

### GetConciliations

La operación **GetConciliations** realiza una serie de pasos basados dentro de un pipeline, a continuación, se enlistan los pasos a seguir para consumir dicha operación:

1. Inicia el proceso del pipe line con una operación de entrada.
2. Se confirma que será un servicio asíncrono entrada y salida.
3. Se asigna la entrada del servicio a una variable “varRequest” el mapeo de datos inputPLGetConciliation\_to\_inputdbGetConciliation.
4. Se rutea al servicio saliente correspondiente a esta operación GetConciliation.
5. Reemplazamos la varRequest en el Body de la petición a enviar
6. Se asigna la entrada del servicio a una variable “varResponse” el mapeo de datos outputPLGetConciliation\_to\_outputdbGetConciliation.
7. Error handler cacha los errores ocurridos en la ejecución del routing.
8. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta.
9. Replicamos en error como success.
10. Error handler cacha los errores ocurridos en la ejecución del BranchNode1.
11. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta
12. Replicamos en error como success.

Nota: En caso de existir algún error al momento de invocar al servicio externo, este será devuelto como la salida del servicio.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del ConciliationTecPL- GetConciliations

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | BranchNode1 | BranchNode | Menú de operaciones del servicio |
| 2 | Pipeline Pair Node1 | Pipeline Pair Node | Flujo de petición y response del servicio |
| 3 | Assing | assign | Assingna un valor a una variable |
| 4 | RouteNode1 | RouteNode | Ejecuta un servicio legado |
| 5 | Replace | Replace | Remplaza una variable por otra |
| 6 | Assing | assign | Assingna un valor a una variable |
| 7 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 8 | Replace | Replace | Remplaza una variable por otra |
| 9 | reply | reply | Retorna el resultado en el response |
| 10 | Replace | Replace | Remplaza una variable por otra |
| 11 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 12 | Replace | Replace | Remplaza una variable por otra |
| 13 | reply | reply | Retorna el resultado en el response |

#### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección de Direct Biding:

sb://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/ConciliationTecPs

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

**Request:** Valores de entrada no requeridos

|  |  |  |
| --- | --- | --- |
|  | <soap:Envelope | xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> |

|  |  |
| --- | --- |
|  | <soap:Header> |

|  |  |
| --- | --- |
|  | </soap:Header> |

|  |  |
| --- | --- |
|  | <soap:Body> |

|  |  |  |
| --- | --- | --- |
|  | <con:GetConciliationsRq | xmlns:con="http://soa.estrellaroja.com.mx/ConciliationTec"> |

|  |  |
| --- | --- |
|  | <!--Optional:--> |

|  |  |
| --- | --- |
|  | <con:P\_BUSINESSUNIT/> |

|  |  |
| --- | --- |
|  | <!--Optional:--> |

|  |  |
| --- | --- |
|  | <con:P\_FROMDATE>01/03/2021</con:P\_FROMDATE> |

|  |  |
| --- | --- |
|  | <!--Optional:--> |

|  |  |
| --- | --- |
|  | <con:P\_PAYMENTREFERENCE>AFIL.-009181868-15092021</con:P\_PAYMENTREFERENCE> |

|  |  |
| --- | --- |
|  | <!--Optional:--> |

|  |  |
| --- | --- |
|  | <con:P\_METODO\_RECIBO/> |

|  |  |
| --- | --- |
|  | </con:GetConciliationsRq> |

|  |  |
| --- | --- |
|  | </soap:Body> |

|  |  |
| --- | --- |
|  | </soap:Envelope> |

Response: El servicio retorna los datos que están contenidos en la base de datos de acuerdo a los parámetros seleccionados.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### ConciliationCash

La operación **ConciliationCash** realiza una serie de pasos basados dentro de un pipeline, a continuación, se enlistan los pasos a seguir para consumir dicha operación:

1. Inicia el proceso del pipe line con una operación de entrada.
2. Se confirma que será un servicio asíncrono entrada y salida.
3. Se asigna la entrada del servicio a una variable “varRequest” el mapeo de datos inputPLConciliationCash\_to\_inputdbConciliationCash.
4. Se rutea al servicio saliente correspondiente a esta operación **ConciliationCash**.
5. Reemplazamos la varRequest en el Body de la petición a enviar
6. Se asigna la entrada del servicio a una variable “varResponse” el mapeo de datos outputPLConciliationCash\_to\_outputdbConciliationCash.
7. Error handler cacha los errores ocurridos en la ejecución del routing.
8. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta.
9. Replicamos en error como success.
10. Error handler cacha los errores ocurridos en la ejecución del BranchNode1.
11. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta
12. Replicamos en error como success.

Nota: En caso de existir algún error al momento de invocar al servicio externo, este será devuelto como la salida del servicio.

Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

#### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del ConciliationTecPL- ConciliationCash

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | BranchNode1 | BranchNode | Menú de operaciones del servicio |
| 2 | Pipeline Pair Node1 | Pipeline Pair Node | Flujo de petición y response del servicio |
| 3 | Assing | assign | Assingna un valor a una variable |
| 4 | RouteNode1 | RouteNode | Ejecuta un servicio legado |
| 5 | Replace | Replace | Remplaza una variable por otra |
| 6 | Assing | assign | Assingna un valor a una variable |
| 7 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 8 | Replace | Replace | Remplaza una variable por otra |
| 9 | reply | reply | Retorna el resultado en el response |
| 10 | Replace | Replace | Remplaza una variable por otra |
| 11 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 12 | Replace | Replace | Remplaza una variable por otra |
| 13 | reply | reply | Retorna el resultado en el response |

#### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección de Direct Biding:

sb://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/ConciliationTecPs

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

**Request:** Valores de entrada requeridos .

<con:ConciliationCashRq xmlns:con="http://soa.estrellaroja.com.mx/ConciliationTec">

<!--Zero or more repetitions:-->

<con:ConciliationCash>

<!--Optional:-->

<con:REFERENCE\_PAYMENT>AFIL.-009181868-15092021</con:REFERENCE\_PAYMENT>

<!--Optional:-->

<con:PAYMENT\_NUMBER>AFIL.-009181868-15092021</con:PAYMENT\_NUMBER>

<!--Optional:-->

<con:RECEIPT\_METHOD>SANTANDER\_3550</con:RECEIPT\_METHOD>

<!--Optional:-->

<con:TOTAL>82345.5</con:TOTAL>

</con:ConciliationCash>

</con:ConciliationCashRq>

Response: El servicio retorna los datos que están contenidos en la base de datos de acuerdo a los parámetros seleccionados.

Texto

Descripción generada automáticamente

### ConciliationTransactions

La operación **ConciliationTransactions** realiza una serie de pasos basados dentro de un pipeline, a continuación, se enlistan los pasos a seguir para consumir dicha operación:

1. Inicia el proceso del pipe line con una operación de entrada.
2. Se confirma que será un servicio asíncrono entrada y salida.
3. Se asigna la entrada del servicio a una variable “varRequest” el mapeo de datos inputPLGetConciliation\_to\_inputdbGetConciliation.
4. Se rutea al servicio saliente correspondiente a esta operación **ConciliationTransactions**.
5. Reemplazamos la varRequest en el Body de la petición a enviar
6. Se asigna la entrada del servicio a una variable “varResponse” el mapeo de datos outputPLGetConciliation\_to\_outputdbGetConciliation.
7. Error handler cacha los errores ocurridos en la ejecución del routing.
8. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta.
9. Replicamos en error como success.
10. Error handler cacha los errores ocurridos en la ejecución del BranchNode1.
11. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta
12. Replicamos en error como success.

Nota: En caso de existir algún error al momento de invocar al servicio externo, este será devuelto como la salida del servicio.

Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

#### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del ConciliationTecPL- **ConciliationTransactions**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | BranchNode1 | BranchNode | Menú de operaciones del servicio |
| 2 | Pipeline Pair Node1 | Pipeline Pair Node | Flujo de petición y response del servicio |
| 3 | Assing | assign | Assingna un valor a una variable |
| 4 | RouteNode1 | RouteNode | Ejecuta un servicio legado |
| 5 | Replace | Replace | Remplaza una variable por otra |
| 6 | Assing | assign | Assingna un valor a una variable |
| 7 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 8 | Replace | Replace | Remplaza una variable por otra |
| 9 | reply | reply | Retorna el resultado en el response |
| 10 | Replace | Replace | Remplaza una variable por otra |
| 11 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 12 | Replace | Replace | Remplaza una variable por otra |
| 13 | reply | reply | Retorna el resultado en el response |

#### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección de Direct Biding:

sb://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/ConciliationTecPs

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

**Request:** Valores de entrada no requeridos

<con:ConciliationTransactionsRq xmlns:con="http://soa.estrellaroja.com.mx/ConciliationTec">

<!--Zero or more repetitions:-->

<con:ConciliationTransaction>

<!--Optional:-->

<con:REFERENCE\_PAYMENT>26052021CAPSETEXT</con:REFERENCE\_PAYMENT>

<!--Optional:-->

<con:RECEIPT\_METHOD>SANTANDER\_3550</con:RECEIPT\_METHOD>

<!--Optional:-->

<con:REFERENCE\_TRANSACTIONS\_ID>941366</con:REFERENCE\_TRANSACTIONS\_ID>

<!--Optional:-->

<con:REFERENCE\_TRANSACTIONS>26521113ERP01PERTF</con:REFERENCE\_TRANSACTIONS>

<!--Optional:-->

<con:TOTAL>234</con:TOTAL>

</con:ConciliationTransaction>

<con:ConciliationTransaction>

<!--Optional:-->

<con:REFERENCE\_PAYMENT>26052021CAPSETEXT</con:REFERENCE\_PAYMENT>

<!--Optional:-->

<con:RECEIPT\_METHOD>SANTANDER\_3550</con:RECEIPT\_METHOD>

<!--Optional:-->

<con:REFERENCE\_TRANSACTIONS\_ID>941372</con:REFERENCE\_TRANSACTIONS\_ID>

<!--Optional:-->

<con:REFERENCE\_TRANSACTIONS>26521113ERP01SERTF</con:REFERENCE\_TRANSACTIONS>

<!--Optional:-->

<con:TOTAL>2266</con:TOTAL>

</con:ConciliationTransaction>

</con:ConciliationTransactionsRq>

Response: El servicio retorna los datos que están contenidos en la base de datos de acuerdo a los parámetros seleccionados.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### SendConciliationBank

La operación **SendConciliationBank** realiza una serie de pasos basados dentro de un pipeline, a continuación, se enlistan los pasos a seguir para consumir dicha operación:

1. Inicia el proceso del pipe line con una operación de entrada.
2. Se confirma que será un servicio asíncrono entrada y salida.
3. Se asigna la entrada del servicio a una variable “varRequest” el mapeo de datos inputSendBank\_to\_dbTMS\_BANCOS.
4. Se rutea al servicio saliente correspondiente a esta operación **SendConciliationBank**.
5. Reemplazamos la varRequest en el Body de la petición a enviar
6. Error handler cacha los errores ocurridos en la ejecución del routing.
7. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta.
8. Replicamos en error como success.
9. Error handler cacha los errores ocurridos en la ejecución del BranchNode1.
10. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta
11. Replicamos en error como success.

Nota: En caso de existir algún error al momento de invocar al servicio externo, este será devuelto como la salida del servicio.

Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

#### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del ConciliationTecPL- **SendConciliationBank**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | BranchNode1 | BranchNode | Menú de operaciones del servicio |
| 2 | Pipeline Pair Node1 | Pipeline Pair Node | Flujo de petición y response del servicio |
| 3 | Assing | assign | Assingna un valor a una variable |
| 4 | RouteNode1 | RouteNode | Ejecuta un servicio legado |
| 5 | Replace | Replace | Remplaza una variable por otra |
| 6 | Assing | assign | Assingna un valor a una variable |
| 7 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 8 | Replace | Replace | Remplaza una variable por otra |
| 9 | reply | reply | Retorna el resultado en el response |
| 10 | Replace | Replace | Remplaza una variable por otra |
| 11 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 12 | Replace | Replace | Remplaza una variable por otra |
| 13 | reply | reply | Retorna el resultado en el response |

#### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección de Direct Biding:

sb://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/ConciliationTecPs

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

**Request:** Valores de entrada no requeridos

<SendConciliationBankRq xmlns="http://soa.estrellaroja.com.mx/ConciliationTec" xmlns:tns="http://soa.estrellaroja.com.mx/ConciliationTec" >

<tns:SendConciliationsBank>

<tns:NumOperacion>426684103</tns:NumOperacion>

<tns:Sucursal>032</tns:Sucursal>

<tns:Referencia>13668251021011013</tns:Referencia>

<tns:Usuario>0670SIUS0</tns:Usuario>

<tns:UsrTrx>13668</tns:UsrTrx>

<tns:TipoPago>Contado</tns:TipoPago>

<tns:Lote />

<tns:NombreArchivo />

<tns:Tarjeta>1519</tns:Tarjeta>

<tns:NombreTH>TDC LIGHT</tns:NombreTH>

<tns:Autorizacion>011525</tns:Autorizacion>

<tns:Afiliacion>8139649</tns:Afiliacion>

<tns:NombreAfiliacion>AUTO PASEO DESTINO 2</tns:NombreAfiliacion>

<tns:Importe>348.00</tns:Importe>

<tns:Moneda>MXN</tns:Moneda>

<tns:FechaOperacion>25/10/2021</tns:FechaOperacion>

<tns:HoraOperacion>01:11:14</tns:HoraOperacion>

<tns:TipoTarjeta>C</tns:TipoTarjeta>

<tns:MarcaTarjeta>MASTERCARD</tns:MarcaTarjeta>

<tns:BancoEmisor>SANTANDER</tns:BancoEmisor>

<tns:TipoOperacion>VENTA</tns:TipoOperacion>

<tns:FechaDeposito>25/10/2021</tns:FechaDeposito>

<tns:Buzon>Estrella Roja</tns:Buzon>

</tns:SendConciliationsBank>

<tns:SendConciliationsBank>

<tns:NumOperacion>426685211</tns:NumOperacion>

<tns:Sucursal>021</tns:Sucursal>

<tns:Referencia>9416251021012109</tns:Referencia>

<tns:Usuario>0670SAUS2</tns:Usuario>

<tns:UsrTrx>9416</tns:UsrTrx>

<tns:TipoPago>Contado</tns:TipoPago>

<tns:Lote />

<tns:NombreArchivo />

<tns:Tarjeta>1113</tns:Tarjeta>

<tns:NombreTH />

<tns:Autorizacion>387639</tns:Autorizacion>

<tns:Afiliacion>9270083</tns:Afiliacion>

<tns:NombreAfiliacion>AUTOB MEX PUE EST ROJA</tns:NombreAfiliacion>

<tns:Importe>696.00</tns:Importe>

<tns:Moneda>MXN</tns:Moneda>

<tns:FechaOperacion>25/10/2021</tns:FechaOperacion>

<tns:HoraOperacion>01:21:38</tns:HoraOperacion>

<tns:TipoTarjeta>D</tns:TipoTarjeta>

<tns:MarcaTarjeta>VISA</tns:MarcaTarjeta>

<tns:BancoEmisor>BBVA</tns:BancoEmisor>

<tns:TipoOperacion>VENTA</tns:TipoOperacion>

<tns:FechaDeposito>25/10/2021</tns:FechaDeposito>

<tns:Buzon>Estrella Roja</tns:Buzon>

</tns:SendConciliationsBank>

</SendConciliationBankRq>

Response: El servicio retorna los datos que están contenidos en la base de datos de acuerdo a los parámetros seleccionados.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

### SendConciliationBankAMEX

La operación **SendConciliationBankAMEX** realiza una serie de pasos basados dentro de un pipeline, a continuación, se enlistan los pasos a seguir para consumir dicha operación:

1. Inicia el proceso del pipe line con una operación de entrada.
2. Se confirma que será un servicio asíncrono entrada y salida.
3. Se asigna la entrada del servicio a una variable “varRequest” el mapeo de datos inputSendBankAmex\_to\_insertBankAmex.
4. Se rutea al servicio saliente correspondiente a esta operación SendConciliationBankAMEX.
5. Reemplazamos la varRequest en el Body de la petición a enviar
6. Error handler cacha los errores ocurridos en la ejecución del routing.
7. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta.
8. Replicamos en error como success.
9. Error handler cacha los errores ocurridos en la ejecución del BranchNode1.
10. Una ves Realizado esto remplazamos el resultado del error en el cuerpo de respuesta
11. Replicamos en error como success.

Nota: En caso de existir algún error al momento de invocar al servicio externo, este será devuelto como la salida del servicio.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

#### Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del ConciliationTecPL- SendConciliationBankAMEX

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | BranchNode1 | BranchNode | Menú de operaciones del servicio |
| 2 | Pipeline Pair Node1 | Pipeline Pair Node | Flujo de petición y response del servicio |
| 3 | Assing | assign | Assingna un valor a una variable |
| 4 | RouteNode1 | RouteNode | Ejecuta un servicio legado |
| 5 | Replace | Replace | Remplaza una variable por otra |
| 6 | Assing | assign | Assingna un valor a una variable |
| 7 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 8 | Replace | Replace | Remplaza una variable por otra |
| 9 | reply | reply | Retorna el resultado en el response |
| 10 | Replace | Replace | Remplaza una variable por otra |
| 11 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 12 | Replace | Replace | Remplaza una variable por otra |
| 13 | reply | reply | Retorna el resultado en el response |

#### Pruebas

Las pruebas se ejecutarán en el ambiente de desarrollo tomando la dirección de Direct Biding:

sb://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/ConciliationTecPs

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

**Request:** Valores de entrada requeridos



Response: El servicio retorna los datos que están contenidos en la base de datos de acuerdo a los parámetros seleccionados.

Texto

Descripción generada automáticamente

# Diseño de Datos

Hace referencia a las tablas custom empleadas en esta integración, como es que se consultan y se almacena la información correspondiente a de las integraciones con los buzones de los bancos y los resultados y configuraciones de la conciliación.

## Origen de Datos

Hace referencia a las tablas custom empleadas en esta integración, dichas tablas son para realizar la relación de los datos de la conciliacion.

Tabla: **XXER\_AR\_CONCILIACION\_TBL**

|  |  |  |
| --- | --- | --- |
| **Columna Tipo de dato** | | **Descripción** |
| CONCILIACION\_ID | NUMBER(18,0) | id de la transaccion |
| REFERENCE\_PAYMENT | VARCHAR2(50 BYTE) | Referencia generada por la vista |
| PAYMENT\_NUMBER | VARCHAR2(50 BYTE) | Referencia obtenida en ERP |
| AMOUNT\_TRANSACTIONS | NUMBER | Sumatoria de transaciones relacionadas |
| AMOUNT\_PAYMENT | NUMBER | Monto de pagos consultados en ERP |
| TOLERANCE | NUMBER | Diferencia de tolareancia |
| STATUS | CHAR(3 BYTE) | estatus de conciliacion corto |
| MSJ\_VALIDATION | VARCHAR2(300 BYTE) | descripcion y msj de la conciliacion |
| REFERENCE\_TRANSACTIONS\_ID | CLOB | id de transaciones relacionados |
| REFERENCE\_TRANSACTIONS | CLOB | numeros de documentos y montos de transacion relacionadas al pago. |
| FECHA\_CONCILIACION | TIMESTAMP(6) | Fecha que se realizo la conciliacion. |
| METODO\_RECIBO | VARCHAR2(50 BYTE) | Nombre del metodo de recibo con que se concilio el Pago. |

Se proporciona la info. En base a la cual se crearon las tablas de la información de los bancos:



## Lógica de Validación

Esta lógica hace referencia a las validaciones que se emplean en los origines para poder tener la información congruente y poder procesarla para finalmente poder llegar a una conciliación Exacta.

Sin embargo, podemos mencionar que las principales validaciones de negocio son las siguientes:

* Un pago tiene una o muchas transacciones
* El reference Payment y receipt method son las llaves primarias para la conciliación
* El proceso solo toma transacciones de pagos un mes atrás como máximo
* El desarrollo puede tener mas de un pago en una transacción en TP.
* El proceso concilia transacciones AMEX y AFIL las cuales Extrae la información.
* directamente desde TMS por medio de un JOB que se ejecutara diariamente.

Las cuales se pueden observar en los queries del tema 5.1 Sentencias SQL, también se cuenta con las validaciones explicadas en el punto 3. Lógica de implementación

# Diseño SQL

En esta integración existen principalmente dos consultas, una obtiene los pagos de cash y nos devuelve el monto y otro que consulta si se tiene una nota de crédito aplicada algún pago.

## Sentencias SQL

### Query para Reportes ERP:



# Reglas de Negocio

Las reglas de negocio se describen a continuación.

* Se Concilian solo las Transacciones que existan en TP y tengan dada de alta su configuración en la tabla XXER\_CNF\_INTERFASE\_INGRESOS\_AR y no estén en estatus conciliado en la tabla de resultados XXER\_AR\_CONCILIACION\_TBL.
* Se realiza la sumatoria de los pagos en cash si llegan a ser mas de uno y se coloca el nombre del primero encontrado.
* Las transacciones en tabla puente que tengan la misma referencia se suman y se muestran agrupadas con su monto correspondiente.
* La información que se busca en TMS es en los rangos de un mes atrás.
* Se tiene conciliación semanal(L a D) y diaria.
* Se agrega lógica para AFIL y AMEX para conciliaciones de FIN de SEMANA (V, S y D) la información de estos días se concilia con fecha del día lunes siguiente.

## Diseño del Servicio

Para llevar a cabo el inicio de la integración se tiene que ejecutar el compuesto de ConciliationEnt, el cual recibe 4 parámetros opcionales, los cuales se describen a continuación y no son nesarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Argument Prompt** | | **Value Set** | **Default Value Example** | |
| BussinessUnit | BussinessUnit | String |  | TMS |
| FromDate | FromDate | String |  | 16/10/2021 |
| PaymentReference | PaymentReference | String |  | 08052021CAPUT |
| ReceiptMethod | ReceiptMethod | String |  | SANTANDER\_3550 |

Los campos son opcionales cuando se busca procesar un pago ya fuera del mes o varios de fechas más lejanas.

# Consideraciones de Rendimiento

Este requerimiento ha sido probado con un conjunto de pagos pequeños, tener contemplado que el compuesto puede presentar demora con una cantidad de registros muy grande a procesar.

## Estrategia de Reinicio

* Para llevar a cabo un reinicio de la aplicación no es necesario realizar movimiento en base de datos, la aplicación al ser reiniciada continuará con la ejecución que corresponde.
* Supervisar que al momento del reinicio no existan instancia del proceso en ejecución, de ser así, esperar a que estas terminen para asegurar la congruencia de datos.

## Seguridad

* Se recomienda el monitoreo oportuno de la base de datos para asegurar el correcto espacio para su crecimiento.
* El proceso principal para las Conciliar es ConciliationEnt el cual solo se encuentra expuesto en el ambiente interno de Estrella Roja.
* En caso de requerir exponerlo a un mayor nivel, se recomienda pasar por un servicio OSB para no poner en riesgo la infraestructura del dominio SOA.

## Personalización

* En caso de requerir modificar el proyecto, tomar la versión más reciente del controlador de versiones con el que se cuente.
* La versión del IDE de desarrollo de JDeveloper con la que se implementó dicha solución es JDEVADF\_ 12.2.1.4.0.190828
* La integración de ConciliationEnt cuenta con parámetros que pueden ser personalizables, los cuales se describen a continuación.

En la parte de ConciliationEnt, dentro del proyecto se tiene la carpeta \SOA\Dvms

dvmConciliationEnt donde encontraremos un catálogo actualizable de valores predeterminados, los que podemos modificar son los siguientes:

|  |  |  |
| --- | --- | --- |
| Componente | Código | Valor |
| GetInfoCash | TargetInterface | Conciliation |
| GetInfoCash | ReportName | XXER\_AR\_TRX\_INTERFACE.xdm |
| BancoER | DirName | /prueba |
| BancoER | FileName | prva0670 |
| BancoAmex | DirName | /pruebaAmex |

Tenemos otros catálogos del lado de ConciliationBiz, en la carpeta \SOA\Dvms donde podemos encontrar dvmConciliationBiz que tiene los siguientes valores a personalizables:

|  |  |  |
| --- | --- | --- |
| Componente | Código | Valor |
| GetInfoCash | TargetInterface | Conciliation |
| GetInfoCash | ReportName | XXER\_AR\_CASH.xdm |
| GetInfoCreditMemo | ReportName | XXER\_AR\_NOTA\_CREDITO.xdm |

## Catálogo de Errores

Se cuenta con una lista de errores en el archivo CatalogosErroresEstrellaRoja.xlsx, de manera muy particular particular de esta aplicación, se cuenta con los siguiente:

|  |  |
| --- | --- |
| Código | Descripción |
|  |  |
|  |  |
|  |  |

Sin embargo, también se utilizan códigos generales compartidos con otras integraciones, ver el archivo mencionado para mayor detalle.

# Consideraciones de Instalación

Como Prerrequisitos se requiere un esquema de base de datos (DBCS) previamente configurado, con los data sources correspondientes, consultar el manual de configuración del ambiente.

En este esquema deben existir las siguientes tablas:

* TMS\_COCILIACION\_BANCOS\_TBL
* XXER\_BANCOS\_AMEX\_HEADER\_TBL
* XXER\_BANCOS\_AMEX\_SUMMARY\_TBL
* XXER\_BANCOS\_AMEX\_SUBMISSION\_TBL
* XXER\_BANCOS\_AMEX\_TRANSACTN\_TBL
* XXER\_BANCOS\_AMEX\_CHARGEBACK\_TBL
* XXER\_BANCOS\_AMEX\_ADJUSTMENT\_TBL
* XXER\_BANCOS\_AMEX\_TXNPRICING\_TBL
* XXER\_AR\_CONCILIACION\_TBL
* XXER\_CNF\_INTERFASE\_INGRESOS\_AR

De lo contrario se requiere crearlas, el script de estas tablas se encuentra en el siguiente archivo, para ejecutarlo se debe contar con los permisos necesarios ya sea desde sqlplus o un entorno gráfico.



Instalación de la integración de complemento de pago tanto capa Ent como Biz, instalar como aplicación SOA:



# URL de Acceso y seguridad

A continuación, se definen los servicios utilizados en la integración cuyas URL están definidas con comodines que deben adaptarse dependiendo del ambiente del que se trate. Las dos primeras son propias de las integraciones de Conciliacion, las demás se incorporan para brindar funcionalidades especificas descritas anteriormente en el punto 3. Lógica de implementación

|  |  |
| --- | --- |
| Servicio | URL |
| ConciliationEnt | [http://<hostname>:<puerto>/soa-infra/services/PROD/ConciliationEnt/ConciliationEnt?WSDL](http://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/PROD/ConciliationEnt/ConciliationEnt?WSDL) |
| ConciliationBiz | [http://<hostname>:<puerto>/soa-infra/services/PROD/ConciliationEnt/ConciliationEnt?WSDL](http://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/PROD/ConciliationEnt/ConciliationEnt?WSDL) |
| ERPIntegrationBiz | http://<hostname>:<puerto>/soa-  infra/services/default/ERPIntegrationBiz/ERPIntegrationBiz?WSDL |
| SftpReadBancoAmex | fsgateway.americanexpress.com/outbox |
| SftpReadBancoER | sftp.mitec.com.mx /Outbox |
| SOAUtilitiesTec | http://<hostname>:<puerto>/soa-  infra/services/default/SOAUtilitiesTec/SOAUtilitiesTec?WSDL |
| MessageTec | http://<hostname>:<puerto>/soa-  infra/services/default/MessageTec/MessageTec?WSDL |
| ConciliationTec | sb://<hostname>:<puerto>/ ConciliationTecPS |

# Temas abiertos y cerrados

## Temas Abiertos

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

## Temas Cerrados

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