

OUM

TA.070-6 DISEÑO TÉCNICO DE EXTENSIONES E

INTERFACES



# ER20 Importación de Tipo Cambio

|  |  |
| --- | --- |
| Autor: | Forte Innovation |
| Fecha de creación: | Marzo 12 2021 |
| Última actualización: | Marzo 12 2021 |
| Código de referencia OUM: | DS.140 Design Specification |
| Versión:  **Aprobadores:** | 1.0 |
| Ivan Muñoz |  |

Ángel Flores



Contenido

1. Control de Documento ............................................................................................. 3

1.1 Bitácora de Cambios ................................................................................................. 3

1.2 Revisores ................................................................................................................... 3

2 Resumen técnico ....................................................................................................... 4

3 Diagrama de la integración ........................................................................................ 5

4 Lógica de implementación ......................................................................................... 6

4.1 Calendarización y Ejecución........................................................................................ 6

4.2 Implementación de servicio web DailyRatesBiz.......................................................... 7

4.3 Implementación de servicio web BanxicoTec .......................................................... 16

5 Origen de Datos ....................................................................................................... 19

6 Lógica de Validación ................................................................................................. 20

7 Reglas de Negocio .................................................................................................... 21

8 Consideraciones de Rendimiento ............................................................................. 21

8.1 Estrategia de Reinicio................................................................................................ 21

8.2 Seguridad .................................................................................................................. 21

8.3 Personalización ......................................................................................................... 21

9 Catálogo de Errores .................................................................................................. 23

10 Consideraciones de Instalación ................................................................................ 23

11 URL de Acceso y seguridad ....................................................................................... 24

12 Temas abiertos y cerrados ........................................................................................ 24

12.1 Temas Abiertos ......................................................................................................... 24

12.2 Temas Cerrados ........................................................................................................ 24

## 1 Control de Documento

### 1.1 Bitácora de Cambios

3

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Referencia del cambio** |
| 12-marzo-2021 | Pedro Sanchez | 1.0 | Creación de documento |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 1.2 Revisores

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



## 2 Resumen técnico

Este documento presenta el complemento para la integración que conforma la importación de pólizas contables hacia el ERP Cloud. Esta integración es referida por el documento funcional TA020-3\_ER20\_GL\_Importación\_Tipo Cambio.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.

Resumen técnico

## 3 Diagrama de la integración



Diagrama de la integración

## 4 Lógica de implementación

El desarrollo esta calendarizado mediante un ESS de SOA.

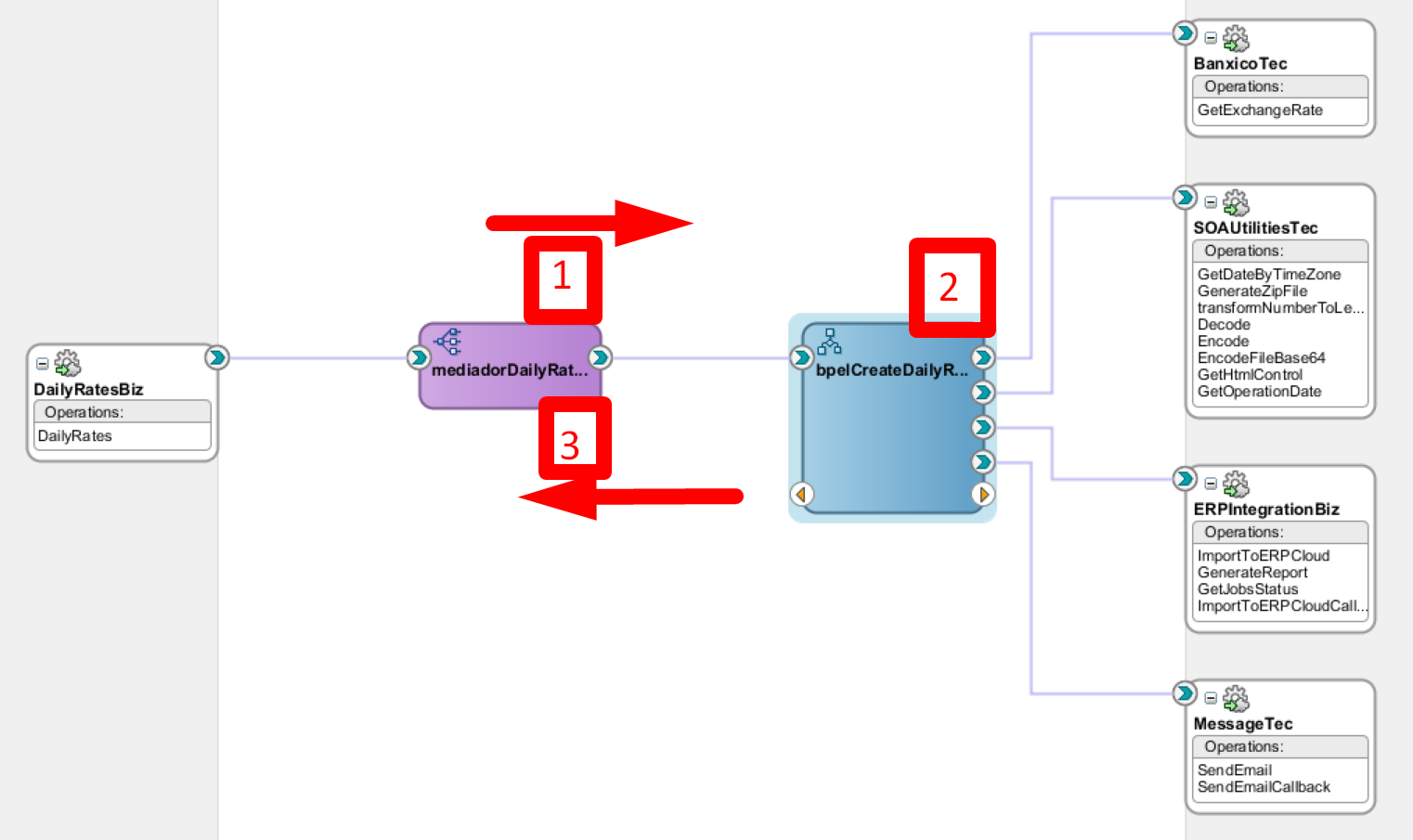
### 4.1 Calendarización y Ejecución

|  |  |  |
| --- | --- | --- |
| Componente | Propiedad | Valor |
| Trabajo | Nombre | DailyRatesBiz |
| Nombre mostrado | DailyRatesBiz |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Creacion de DailyRates en modulo de GL |
| Tipo de trabajo | SyncWebserviceJobType |
| Operación | DailyRates |
| Solicitud | <ns1:DailyRatesRq xmlns:ns1="http://soa.estrellaroja.com.mx/DailyRatesBiz"> <ns1:Series>SF43718</ns1:Series> </ns1:DailyRatesRq> |
| Planificación | Nombre | DailyRatesBiz |
| Nombre mostrado | DailyRatesBiz |
| Paquete | |  |  | | --- | --- | |  | /oracle/apps/ess/custom/soa | |
| Descripción | |  |  | | --- | --- | |  | Creacion de tipo de cambio en GL | |
| Frecuencia | Diario a 07:00:00 a. m. |
|  | Zona horaria | UTC-06:00 Mexico City |

### 4.2 Implementación de servicio web DailyRatesBiz

Este servicio web utiliza una lógica de proceso de varios componentes de SOA que se encargan de enrutar las peticiones hacía los servicios de la capa de negocio. A continuación, se muestra su lógica de implementación:

1. Petición del cliente dependiendo de la operación DailyRates .
2. Invocación de BPEL bpelCreateDailyRates.
3. Después de haber ejecutado una de las operaciones, el servicio responde la petición



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.

### 4.2.1 Lista de Objetos

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

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Tipo** | **Descripción** |
| DailyRatesBiz | Web  Service | Servicio Expuesto |
| mediadorDailyRatesBiz | Mediator | Componente de SOA utilizado para integrar y enrutar los mensajes entre los servicios web |
| bpelCreateDailyRates | Bpel | Componente de Orquestación |
| ERPIntegrationBiz | Web  Service | Servicio externo que Reliza la importación de los Datos al ERP |
| MessageTec | Web  Service | Servicio externo envio de correos |
| SOAUtilitiesTec | Web  Service | Servicio externo operaciones custom |
| BanxicoTec | Web  Service | Servicio externo consulta Api Rest banxico |

### 4.2.2 DailyRates

La operación *DailyRates* realiza una serie de pasos basados dentro de un bpel.

El proceso BPEL realiza las siguientes acciones:

1. Se procede a asignar los valores de entrada del bpel al servicio DailyRatesBiz - DailyRates.
2. Si el contador de días que se rentan es menor a 10.

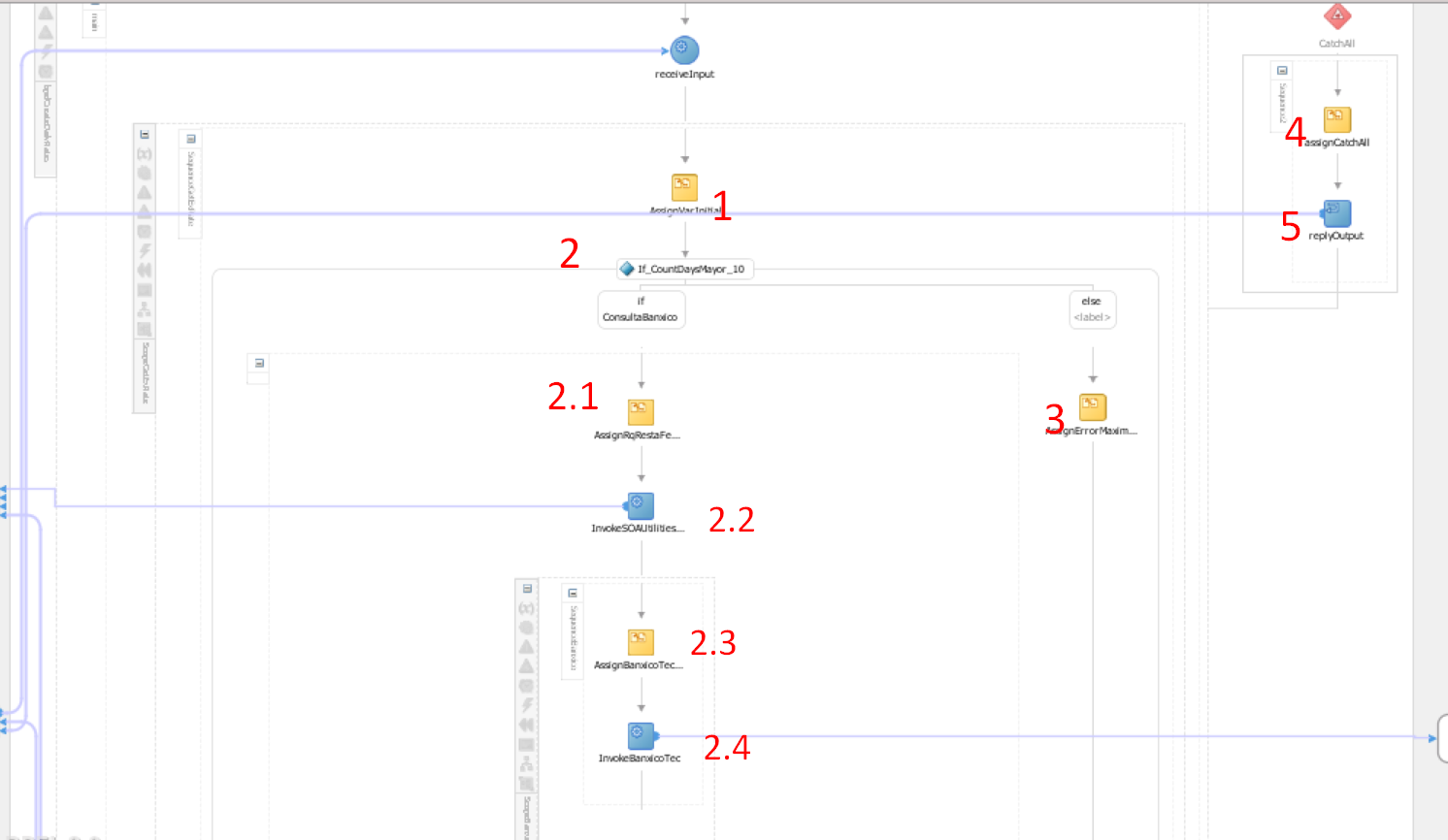
2.1 se Asigna los valores para consumir el servicio que resta días a la fecha con el valor de días a restar.

2.2 se invoca el servicio WsSOAUtilitiesTec-GetOperationDateRq para obtener la fecha menos los dias.

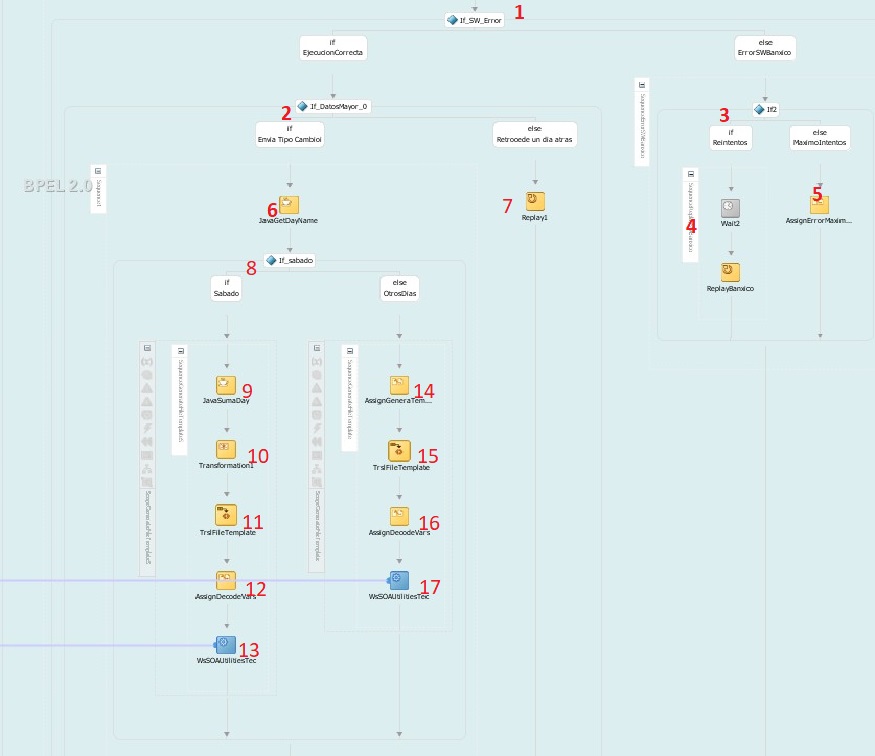
2.3 se Asigna los valores para consumir el servicio de banxico

2.4 se invoca el servicio WsBanxicoTec-GetExchangeRateRq para obtener el tipo de cambio.

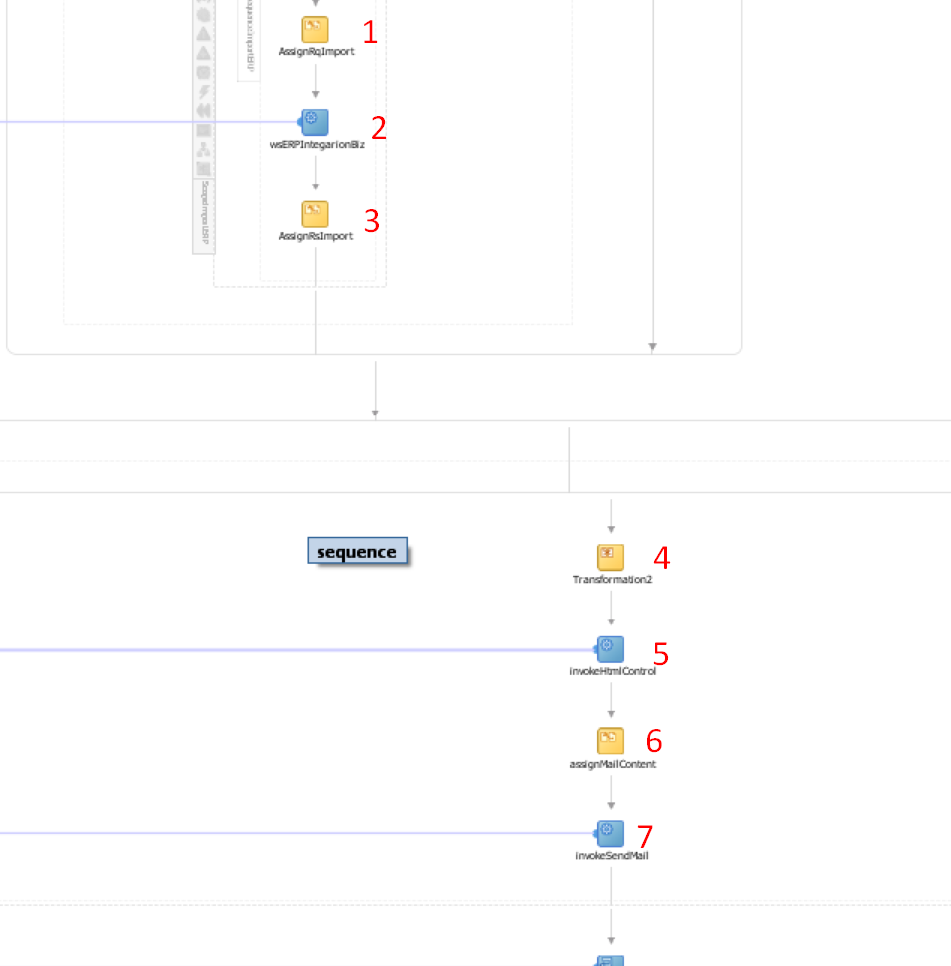
1. SI se excede el numero de días atrás que se puede consultar el servicio de Banxico Se notifica el error.
2. asignación de error General.
3. Respuesta de error.



1. Si el Servicio de Banxico tiene no tiene errores de consulta Errores
2. Validamos Si la consulta trae datos
3. Si el Servicio de Banxico tiene tiene errores de consulta Errores, Validamos los reintentos sin son menor a 10 reintentamos la consulta
4. Esperamos 5 segundos antes de reintentar la consulta
5. Si el numero de reintentos excede a los 10 termina el proceso y notificamos error
6. Java embebido que obtiene el nombre del día.
7. Validamos Si la consulta no trae datos reintentamos un día atrás
8. Validamos si el día que estamos procesando es sábado
9. Obtenemos el día siguiente con un java
10. Ejecutamos transformación que genera template de dos días sábado y domingo
11. Ejecutamos traslate que devuelve el formato en CSV
12. Asignamos las variables de entrada para ejecutar servicio de codificación a B64
13. Ejecutamos Servicio WsSOAUtilitiesTec\_EncodeRq
14. Asignamos la variable que genera template de un dia
15. Ejecutamos traslate que devuelve el formato en CSV
16. Asignamos las variables de entrada para ejecutar servicio de codificación a B64
17. Ejecutamos Servicio WsSOAUtilitiesTec\_EncodeRq



1. Asignamos los datos para enviar el template en base 64
2. Ejecutamos invoke que ejecuta el servicio wsERPIntegarionBiz\_ImportToERPCloudRq que importa el tipo de cambio en GL
3. Asignamos la el id de ejecución del servicio de importación a la respuesta del servicio
4. Transformación que asigna los valores para generar el HTML de respuesta del correo
5. Ejecutamos servicio que genera el HTML del correo
6. Asignamos el HTML del correo a la variable de entrada del servicio que envía la notificación de correo .
7. Ejecutamos Servicio de notificaciones de correo electrónico.



### 4.2.3 Lista de objetos

La siguiente tabla contiene los objetos que fueron usados para la funcionalidad del bpel sendfileJournal.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | assignWsJournalBizGetJournalRq | assign | Asignación de valores. |
| 2 | invokeWsJournalBizGetJournal | invoke | Invocación de Servicio. |
| 3 | IfValidate | if | Validación de datos. |
| 4 | assignVarInterfaceL1 | assign | Asignación de valores. |
| 5 | trWsJournalBizGetJournalRsToBpelRs | transformation | Transformación de datos. |
| 6 | invokeWsERPIntegrationBizImportToERPCloud | invoke | Invocación de Servicio. |
| 7 | assignVarInterfaceL2 | assign | Asignación de valores. |
| 8 | trWsJournalBizGetJournalRsToBpelRs | transformation | Transformación de datos. |
| 9 | invokeWsERPIntegrationBizImportToERPCloud | invoke | Invocación de Servicio. |

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

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

**Request:**Requiere el código que se envia al servicio para la consulta del tipo de cambio ‘SF43718’:



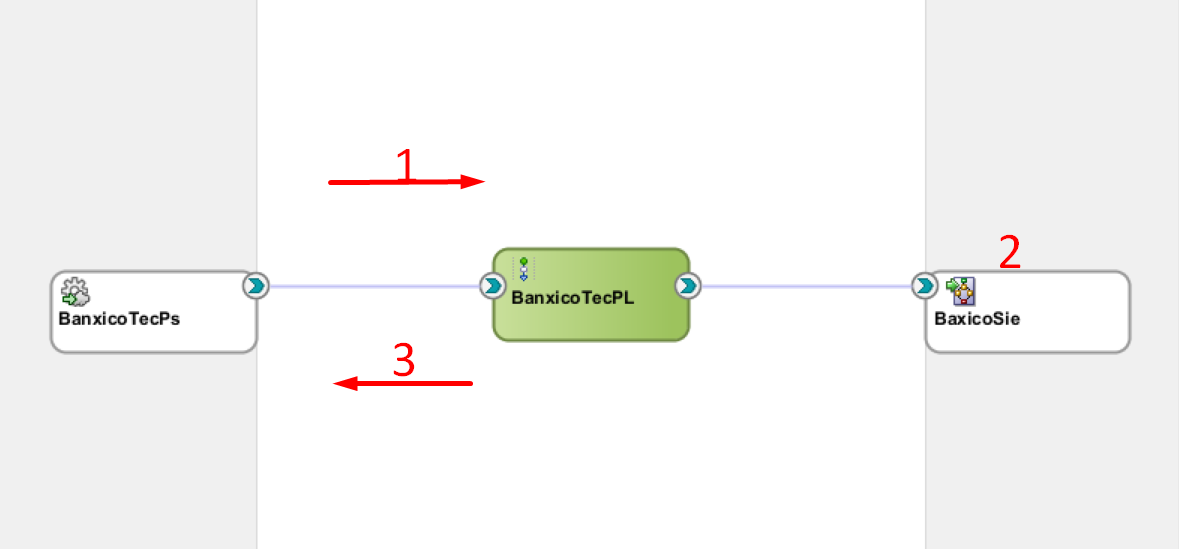
Response: El servicio retorna un id, con el cual se puede consultar la operación JobStatus para checar el estatus de los jobs del flujo.



### 4.3 Implementación de servicio web BanxicoTec

Este servicio web utiliza una lógica de proceso de componentes de OSB que se encargan de enrutar las peticiones hacía el servicio Rest de consulta a Banxico. 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.
3. 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 Rest |  |
| Adaptador Servicio |  |
| Tubería |  |

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:

### 4.4.1 Lista de objetos

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** | **GIT** |
| 1 | BanxicoTecPs | Web  Service | Servicio Expuesto |  |
| 2 | BanxicoSie | Adaptador | Adaptador Rest |  |
| 5 | BanxicoTecPL | Tubería | Flujo de tubería |  |

### 4.4.2 GetExchangeRate

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

1. Se asigna la entrada del servicio a una variable “GetExchangeRateRq”.
2. Se asignan los valores de la variable “GetSeries\_params” al servicio de negocios “BanxicoSie” mediante una transformación.
3. Inserta dentro del header la variable “Bmx-Token”,
4. Se procede a enrutar hacia el servicio de negocios que deseamos consumir “BanxicoSie”.
5. Se recupera el resultado del servicio consumido, transformando y alojando estos dentro de una variable “series”.
6. Se rutea al servicio saliente correspondiente a esta operación.
7. Se asignan los valores de la variable “series” hacía la salida del servicio en la variable “GetExchangeRateRq” mediante una transformacion.

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

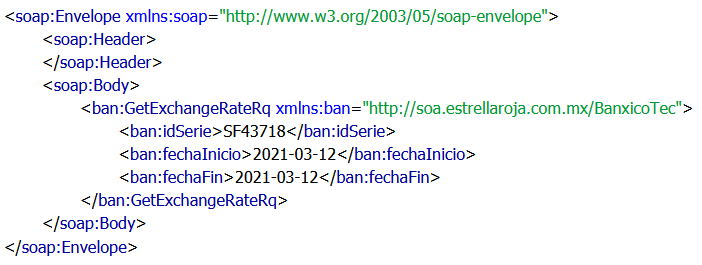
### 4.4.5 Pruebas

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

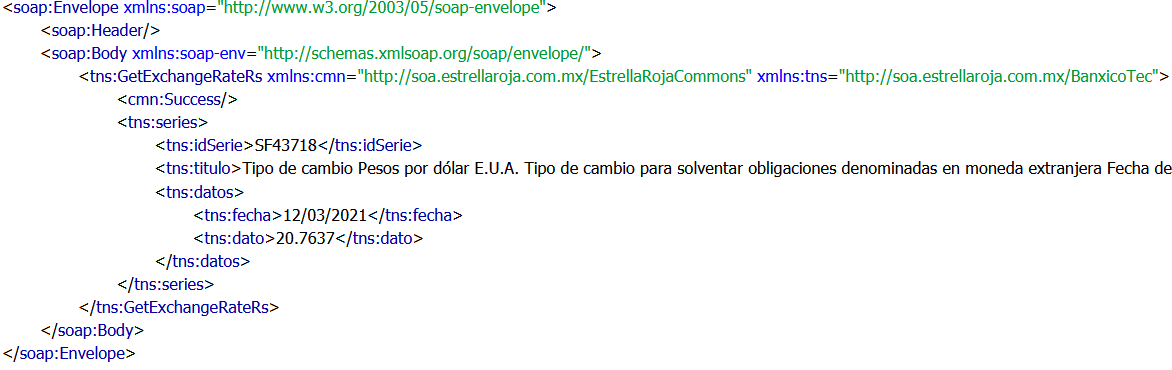
##### https://ersoaprd-soa-0.snproduccin.vcner.oraclevcn.com:9073/BanxicoTecPs

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

**Request:** Se require fecha inicio , fecha din y id de la serie a consultar:



Response: El servicio retorna los datos de el tipo de cambio si es que existen para las fechas proporcionadas en la petición.

:

## 5 Origen de Datos

Los orígenes de datos se encuentran dentro del adaptador de base, por lo cual en este apartado se mencionará aquellos servicios web junto con las operaciones ocupadas para poder realizar dicha integración.

La siguiente tabla muestra los distintos orígenes de datos correspondientes a esta integración dentro de la capa de técnica.

|  |  |  |
| --- | --- | --- |
| **Servicio** | **Operación** | **Descripción** |
| BanxicoTec | GetExchangeRate | Extrae el tipo cambio de Banxico. |

## 6 Lógica de Validación

Esta lógica hace referencia a las validaciones que se emplean en los orígenes para poder tener la información congruente y poder procesarla para finalmente extraer el tipo de cambio , la forma de realizar este proceso lo podemos encontrar en el documento de capas técnicas.

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

 El tipo de cambio debe tener valores en el valor del día consultado.

## 7 Reglas de Negocio

El servicio trabaja sobre sobre la disponibilidad del API Rest de baxico, en caso de no estar disponible el servicio al servicio notifica y no realiza actualizaciones dentro del ERP.

El servicio actualiza con los datos consultados de un día antes para la fecha actual.

Banxico no actualiza tipo de cambios para días no hábiles y sábados y domingos , por lo que la regla de negocio para estos escenarios es tomar el ultimo tipo de cambio proporcionado por Banxico.

El servicio realiza reintentos en caso de haber intermitencia con el servicio de Banxico.

## 8 Consideraciones de Rendimiento

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

### 8.1 Estrategia de Reinicio

* 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.

### 8.2 Seguridad

* El proceso principal para las pólizas es DailyRatesBiz 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.

.

### 8.3 Personalización

En caso de querer 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 Build JDEVADF\_PT.12.2.1.4.0\_GENERIC\_190911.2248.S

## 9 Catálogo de Errores

La lista de errores de SOA que se podrían suscitar, se encuentran dentro del archivo

|  |  |
| --- | --- |
| Código | Descripción |
| SOA-02001 | Fallo al agregar archivos adjuntos |
| SOA-02002 | no se encontro tipo de cambio 10 dias atras |
| SOA-00001 | Servicio Banxico no disponible maximo numero de reintentos |

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

## 10 Consideraciones de Instalación

A continuación, se listan los pasos a seguir para la instalación de esta integración.

1. Los servicios de la capa técnica y de negocio deben de estar desplegados en SOA.
2. Instalar la aplicación compuesta de SOA DailyRatesBiz (Este archivo se encuentra en los entregables, el cual tiene el mismo nombre con extensión Jar).
3. Crear la calendarización de la operación “DailyRatesBiz” del compuesto “DailyRatesBiz”, esta se muestra en el punto 4.1.

## 11 URL de Acceso y seguridad

A continuación se anexa el WSDL correspondiente al servicio JournalEnt, el cual dentro de la infraestructura de la empresa no requiere seguridad.

URL para el acceso al servicio web en el ambiente de desarrollo:

|  |  |
| --- | --- |
| Servicio | URL |
| **DailyRatesBiz** | **http://<hostname>:<puerto>/soa-**  **infra/services/default/DailyRatesBiz /DailyRatesBiz?WSDL** |
| **BanxicoTec** | **sb://<hostname>:<puerto>/TecPs** |

## 12 Temas abiertos y cerrados

### 12.1 Temas Abiertos

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

### 12.2 Temas Cerrados

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