

OUM

TA.070-4 DISEÑO TÉCNICO DE EXTENSIONES E INTERFACES



ER05_AR_Creación y Aplicación de Recibos

Autor:	Oracle Consulting
Fecha de creación:	enero 15, 2018
Última actualización:	enero 15, 2018
Código de referencia OUM:	DS.140 Design Specification
Version:	1.0

Aprobadores:

Ivan Muñoz

Ángel Flores

1 Control de Documento

1.1 Bitácora de Cambios

Fecha	Autor	Versión	Referencia del cambio
15-ene-2018	Gabriel Mares	1.0	No Previous Document

1.2 Revisores

Nombre	Posición

Contenido

1	Control de Documento	ii
1.1	Bitácora de Cambios	ii
1.2	Revisores.....	ii
2	Resumen técnico	1
2.1	Diagrama de la integración	2
3	Lógica de implementación	3
3.1	FinancialsBiz	3
3.2	ReceiptEnt	4
3.3	ReceiptBiz	8
3.4	Pruebas	15
4	Diseño de Datos	16
4.1	Tabla de diseño de datos	16
4.2	Origen de Datos	20
4.3	Lógica de Validación	21
5	Diseño SQL	22
5.1	Sentencias SQL.....	22
6	Reglas de Negocio	24
6.1	Diseño del Servicio.....	24
7	Consideraciones de Rendimiento	25
7.1	Estrategia de Reinicio.....	25
7.2	Seguridad	25
7.3	Personalización	25
7.4	Catálogo de Errores	25
8	Consideraciones de Instalación	26
9	URL de Acceso y seguridad	27
10	Temas abiertos y cerrados	28
10.1	Temas Abiertos	28
10.2	Temas Cerrados.....	28

2 Resumen técnico

Las capas con las que cuenta esta integración son las siguientes:

- ✓ Ent
- ✓ Biz
- ✓ Tec

Todas las especificaciones para la creación y aplicación de Recibos se ubican dentro del documento TA_020-4_ER05_AR_Creacion_y_aplicacion_de_recibos.

- La capa **Ent** se maneja el llamado a diferentes servicios de la capa **Biz** el cual sirve de principal manejador de la información requerida para el resto de las capas.
- La capa **Biz** se encarga de recibir la información del **Ent** y utilizarla para poder crear Recibos y Aplicaciones de Recibos, registrando sus procesos exitosos o erróneos dentro de la base de datos y a la vez notificarlos vía correo electrónico.
- La capa **Tec** se encarga de obtener datos o la manipulación de la información de diferentes fuentes, todo esto para que la capa **Biz**.

La capa **Ent** se configuro para ser ejecutada diariamente a las 8:30 am, toda esta configuración se realizó dentro del ESS. A continuación, se ilustra la información de la configuración:

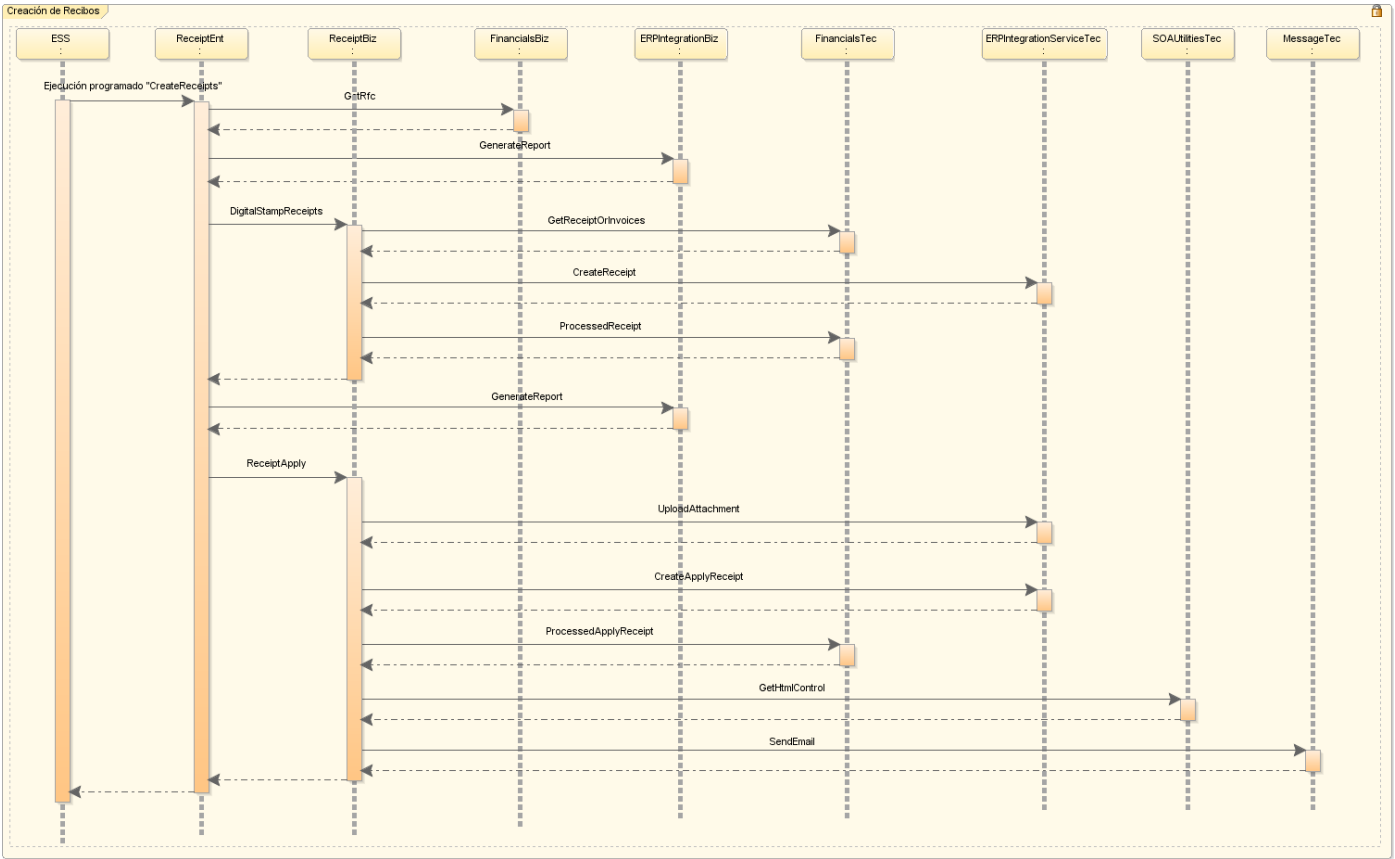
Componente	Propiedad	Valor
Trabajo	Nombre	ReceiptEntJob
	Nombre Mostrado	ReceiptEntJob
	Paquete	/oracle/apps/ess/custom/soa
	Descripción	Job para la Creación y Aplicación de Recibos
	Tipo de trabajo	SyncWebserviceJobType
	WSDL	<a href="http://<host>:<port>/soa-infra/services/default/ReceiptEnt/ReceiptEnt?WSDL">http://<host>:<port>/soa-infra/services/default/ReceiptEnt/ReceiptEnt?WSDL
	Tipo de puerto	ReceiptEntPort
	Operación	CreateReceipts
Planificación	Nombre	ReceiptEntSch
	Nombre Mostrado	ReceiptEnt
	Paquete	/oracle/apps/ess/custom/soa
	Descripción	Ejecución diaria a las 8:30 am
	Frecuencia	Diariamente
	Zona horaria	(UTC-06:00) Ciudad de México

Existen dos tipos de integraciones:

- **Inbound:** En este tipo de integraciones, las aplicaciones legadas llamarán a servicios web SOAP expuestos en la nube, enviando los archivos para ser cargados en ERP Cloud por los mecanismos de integración de datos y estándares que requiera la aplicación.
- **Outbound:** En el ERP Cloud se dispondrán de reportes a través de BI Publisher, que mediante servicios web SOAP serán leídos y la información se procesa según corresponda.

2.1 Diagrama de la integración

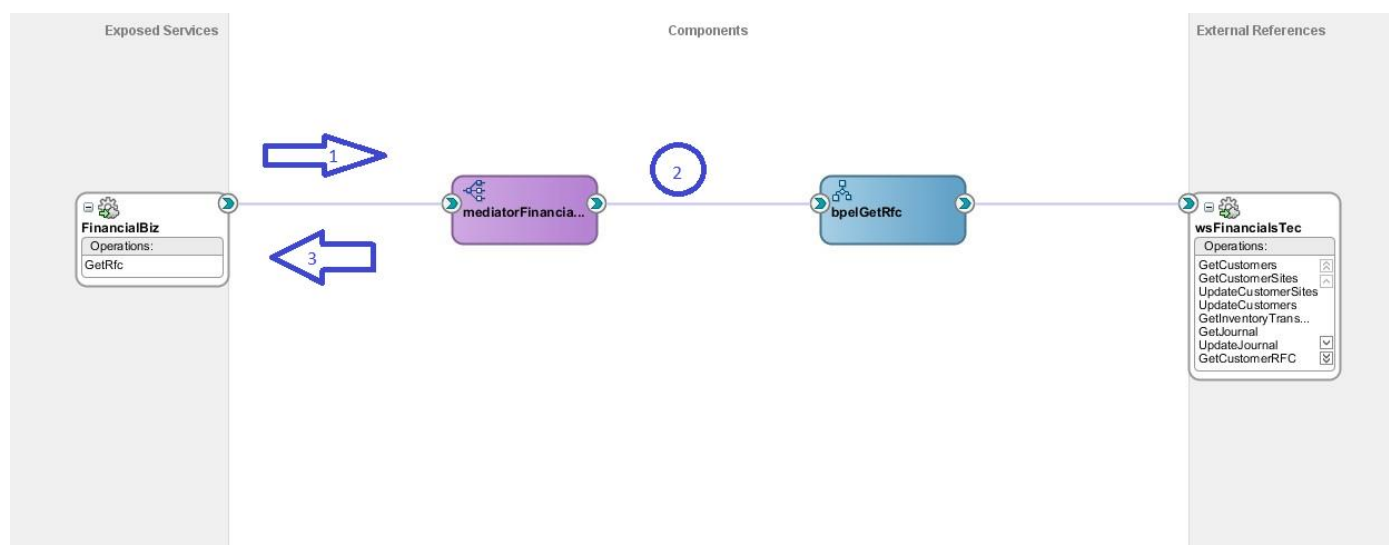
En el siguiente diagrama se muestra el flujo para la Creación y Aplicación de Recibos el cual inicia por la ejecución automática desde el ESS seguido por la capa **Ent** después **Biz**, para terminar en el **Tec**.



3 Lógica de implementación

Dentro de esta sección se describe cada una de las capas requeridas para la Creación y Aplicación de Recibos. Comenzando por la capa de ReceiptEnt en la cual se obtiene parte de la información requerida que se usará en la siguiente capa ReceiptBiz, tomando en cuenta que dentro de la capa ReceiptEnt se estará haciendo uso de FinancialsBiz para poder encontrar la información que servirá como filtro de los DataModels.

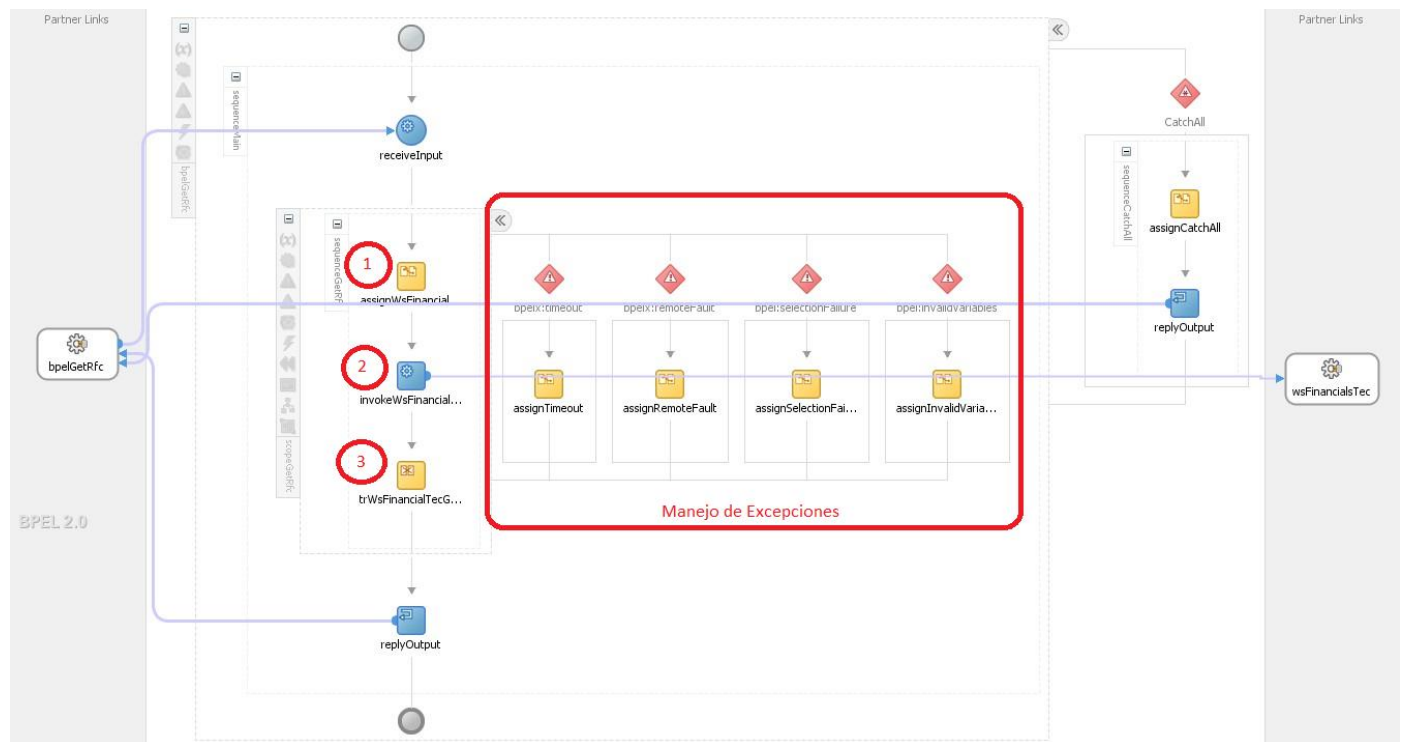
3.1 FinancialsBiz



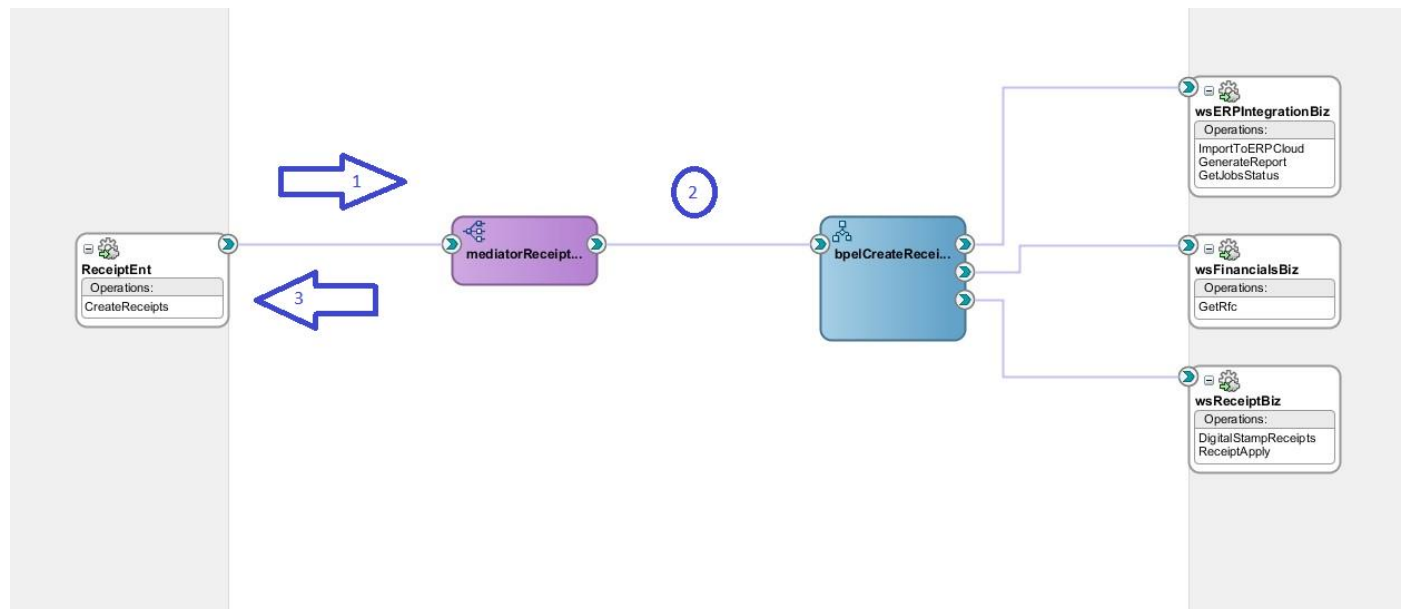
Dentro de la capa FinancialsBiz se expone el servicio con la operación “**GetRfc**” como se muestra el recuadro de la parte izquierda, seguida por el mediator el cual direcciona a “**bpelGetRfc**” en donde se concentra toda la lógica para obtener los RFC disponibles solo para recibos. A continuación, se describirá a detalle la lógica de bpelGetRfc.

1. Se asignan las variables de entrada para el servicio “FinancialsTec”-“GetCustomerRfc”.
2. Se realiza la invocación del servicio “FinancialsTec”-“GetCustomerRfc”.
3. Se realiza la transformación de los datos recibos por el servicio “FinancialsTec”-“GetCustomerRfc”.

También se cuenta con una región llena de componentes en los cuales se engloban el manejo de excepciones que comúnmente se presentan dentro de un Servicio Web.



3.2 ReceiptEnt



Dentro de la capa de “**ReceiptEnt**” se expone el servicio web ReceiptEnt con la operación “**CreateReceipts**” en la cual al consumirlo pasará por el mediator que dirige hacia “**bpelCreateReceipts**”, este se encarga de obtener la información necesaria para la creación y aplicación de Recibos.

1. Asigna variables necesarias para invocar el Servicio de **“FinancialsBiz”-“GetRfc”**, se requiere solo el valor **“RECIBO”** el cual se obtiene de **dvmCreateReceiptEnt.dvm**
2. Se invoca el Servicio de **“FinancialsBiz”-“GetRfc”**
3. Valida la respuesta del Servicio de **“FinancialsBiz”-“GetRfc”** que al menos contenga un elemento.



4. Asigna variables para la invocación del servicio **“ERPIntegrationServicesBiz”-“GenerateReport”**.
5. Transforma los datos para complementar los parámetros de entrada para invocar el servicio **“ERPIntegrationServicesBiz”-“GenerateReport”** con **dvmCreateReceiptEnt.dvm** junto con la respuesta del servicio **“FinancialsBiz”-“GetRfc”**.
6. Se invoca el Servicio **“ERPIntegrationServicesBiz”-“GenerateReport”**.
7. Se transforma el resultado del Servicio para inyectar namespace al data set **“Q_CUSTOMER”**.
8. Se asigna el resultado del Servicio a una variable del mismo tipo.
9. Se transforma cada uno de los datos de la variable anterior por otra variable del mismo tipo para poder tener acceso a ellos.
10. Se valida que contenga al menos un registro de los recibos que se deberán aplicar.

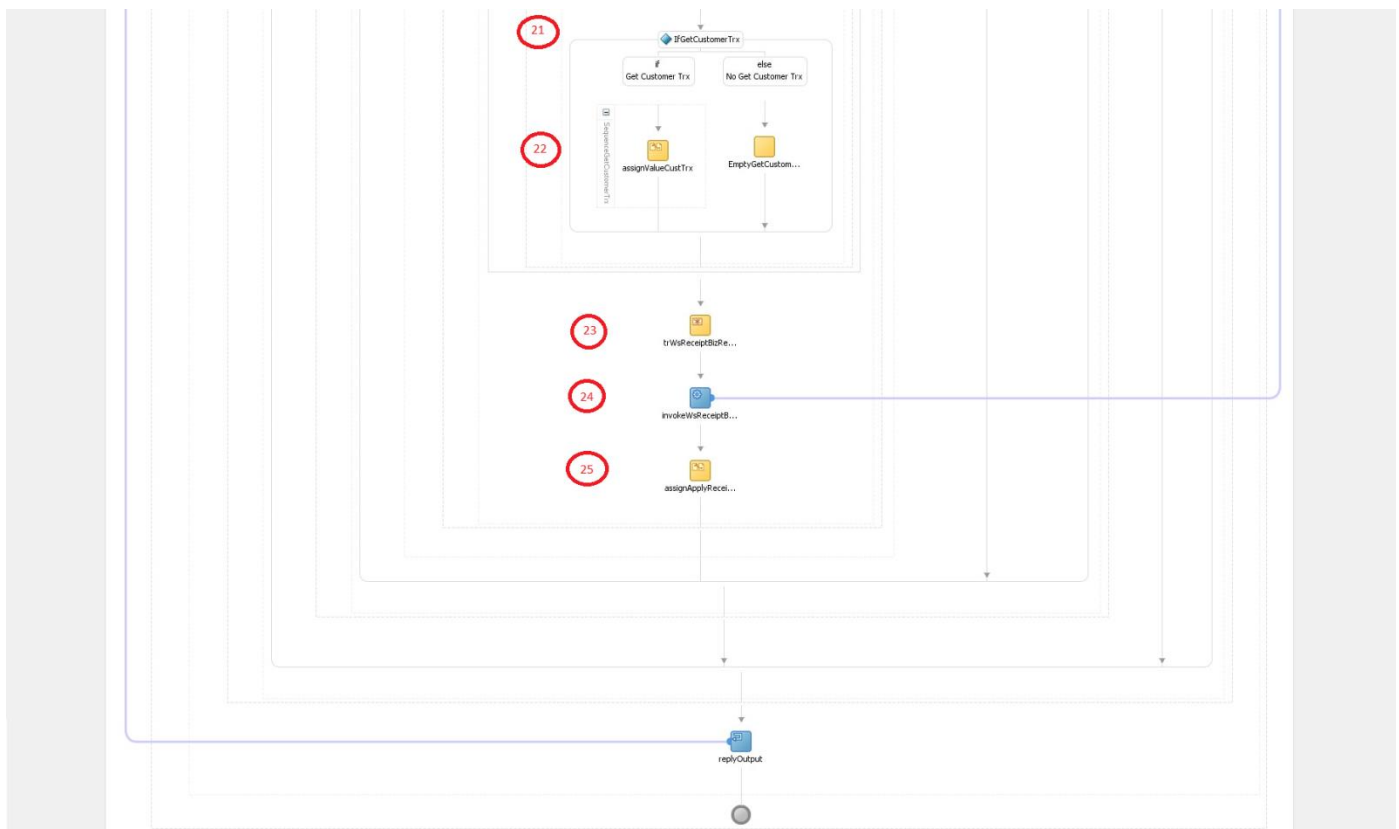


11. Se transforma los datos requeridos para el Servicio **“ReceiptBiz”-“DigitalStampReceipts”**.

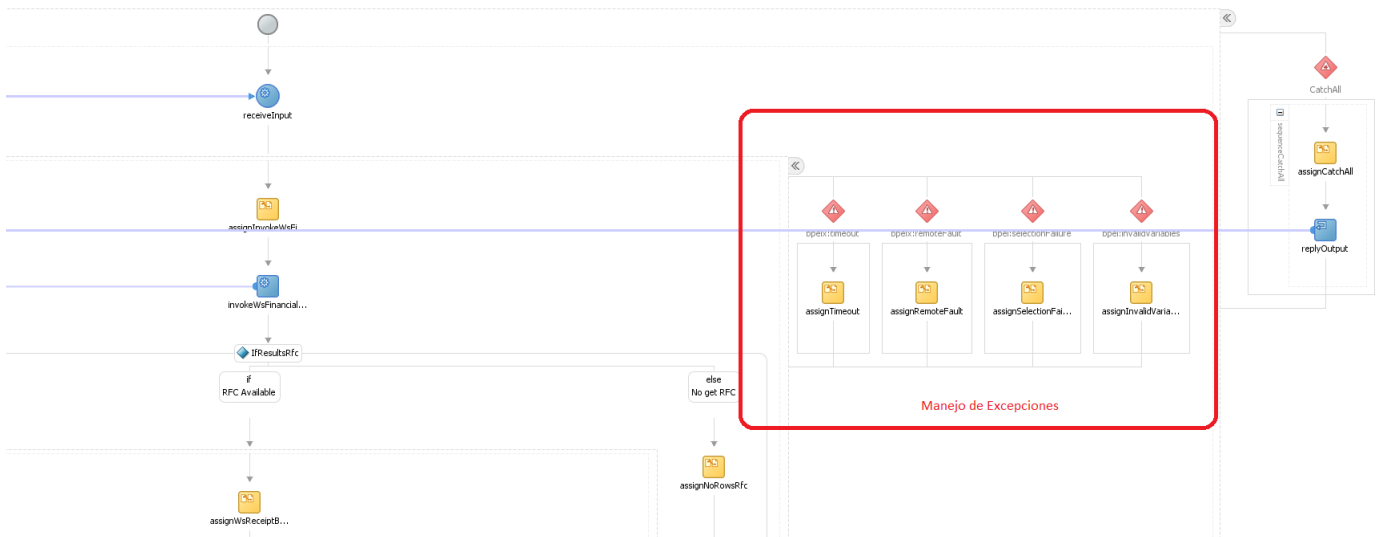
12. Invocación al Servicio **“ReceiptBiz”-“DigitalStampReceipts”** en donde se maneja la creación de Recibos.
13. Inicia **“FOR”** para iterar el resultado del Servicio.
14. Asigna el valor del contador a una variable.
15. Se asignan todos los parámetros necesarios para el Servicio **“ERPIntegrationServicesBiz”-“GenerateReport”**.
16. Se invoca el Servicio **“ERPIntegrationServicesBiz”-“GenerateReport”** para consultar Data Model y complementar la información necesaria para aplicar Recibos.
17. Dentro de la Transformación se inyecta el namespace para el data set **“CUSTOMER_TRX”**.
18. Dentro de la Transformación se inyecta el namespace para el data set **“ORG_UNIT”**.
19. Se asigna el resultado del Servicio a una variable del mismo tipo.
20. Se transforma cada uno de los datos de la variable anterior por otra variable del mismo tipo para poder tener acceso a ellos.



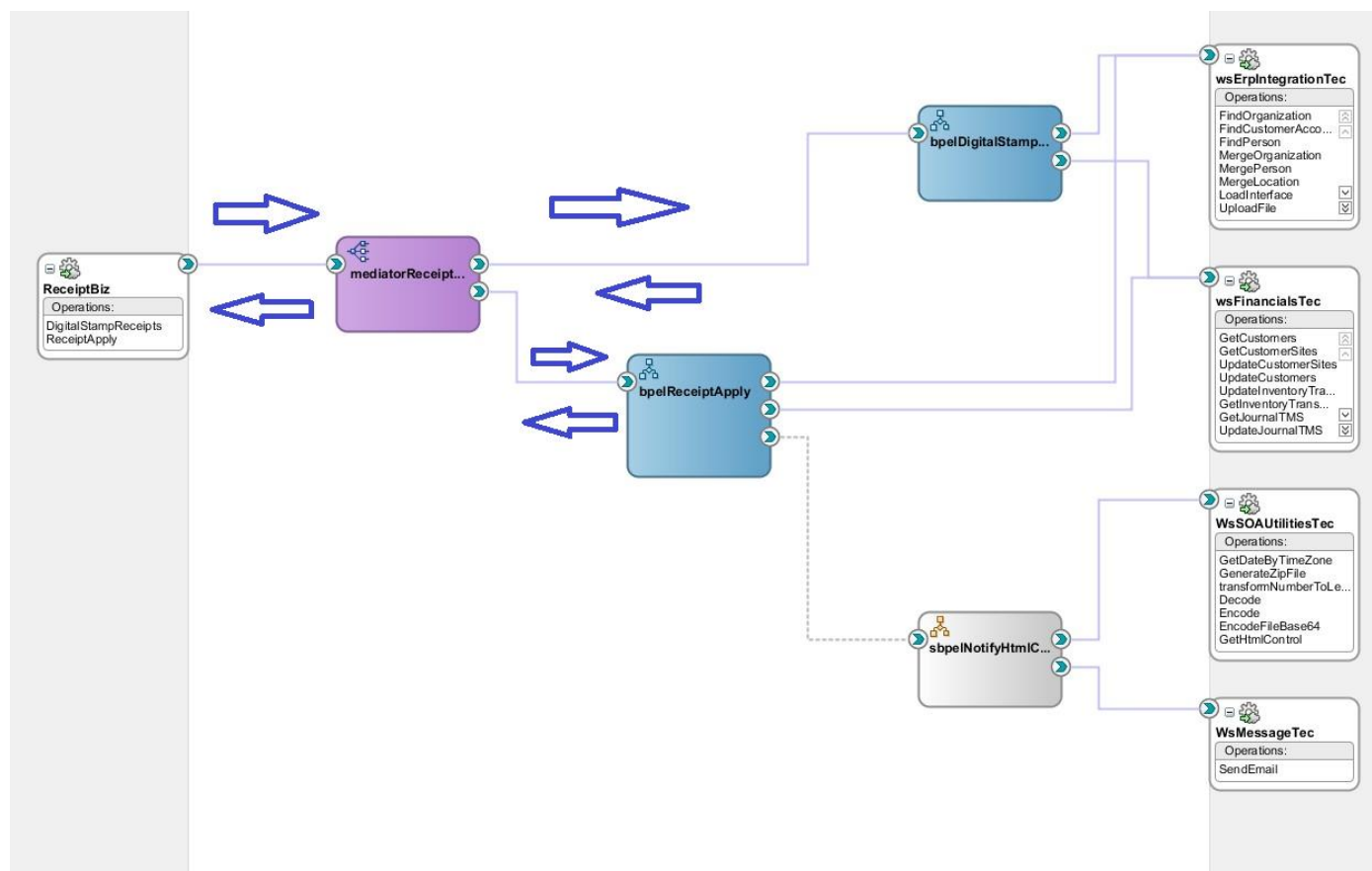
21. Se valida que exista resultado del Servicio.
22. Se asignan las variables que hacen falta para completar la información de respuesta del Servicio **“ReceiptBiz”-“DigitalStampReceipts”**.
23. Dentro de la transformación se establecen los parámetros necesarios para consumir el Servicio **“ReceiptBiz”-“ReceiptApply”** estos datos son los requeridos para aplicar recibos y los errores y cantidad de procesos exitosos.
24. Invoca el Servicio **“ReceiptBiz”-“ReceiptApply”** para los Attachment de Recibos y Aplicación de Recibos.
25. Asigna los valores de respuesta del Servicio **“ReceiptBiz”-“ReceiptApply”** a la salida del Servicio actual.



26. Dentro de este bpel se cuenta con una región en la cual se manejan las excepciones comunes presentadas en los Web Services.



3.3 ReceiptBiz



En esta capa se encuentran dos operaciones: DigitalStampReceipts y ReceiptApply.

- **DigitalStampReceipts**

En esta operación se estarán creando Recibos recibiendo como parámetros la información requerida para crear cada uno de ellos. Como respuesta de este Servicio es la información de cada Recibo creado junto con los errores y mensajes de errores de los Recibos que no se crearon.

- **ReceiptApply**

Dentro de esta operación se Aplicarán los Recibos, recibiendo la información requerida para poder aplicarlos y los errores y mensajes de error devueltos por la operación DigitalStampReceipts. La respuesta del Servicio son los errores globales.

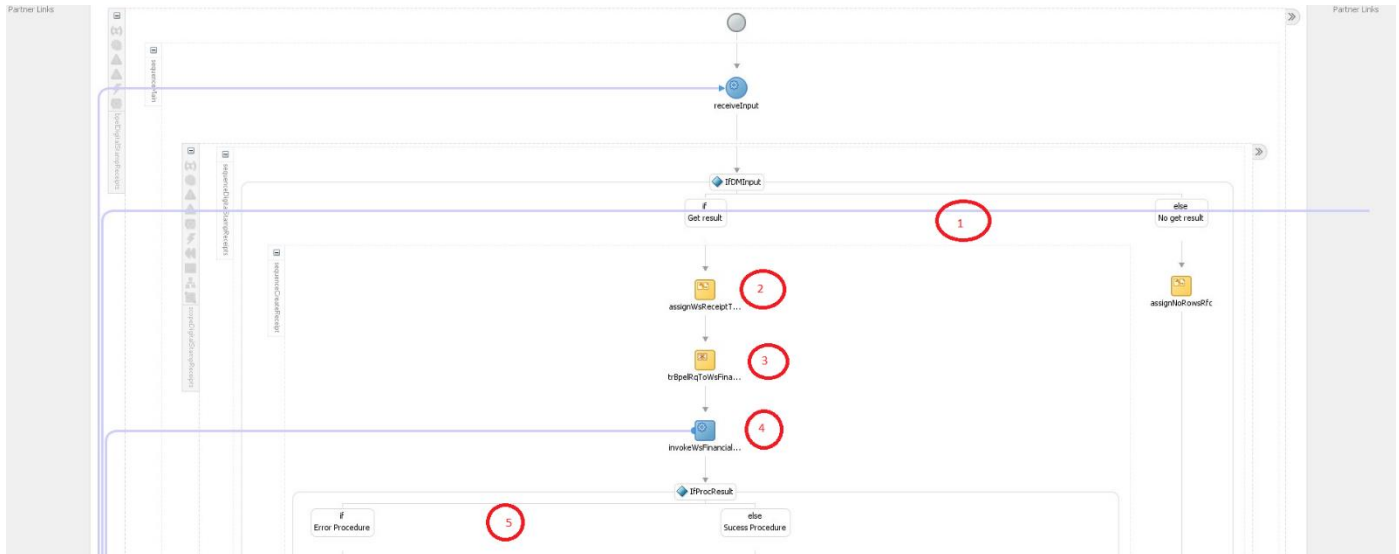
Dentro de este bpel se estarán manejando el envío de las notificaciones con el subproceso sbpelNotifyHtmlControl.

- **sbpelNotifyHtmlControl**

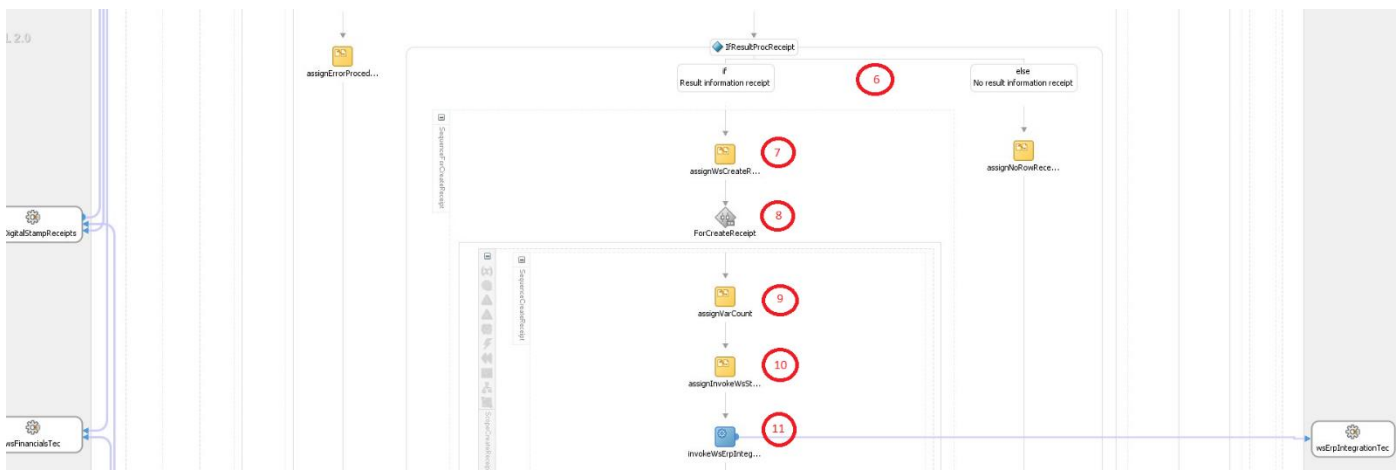
Este Bpel es llamado como subproceso dentro del cual se convierte a HTML la información a notificar y después se envía por correo electrónico.

bpelDigitalStampReceipts.bpel

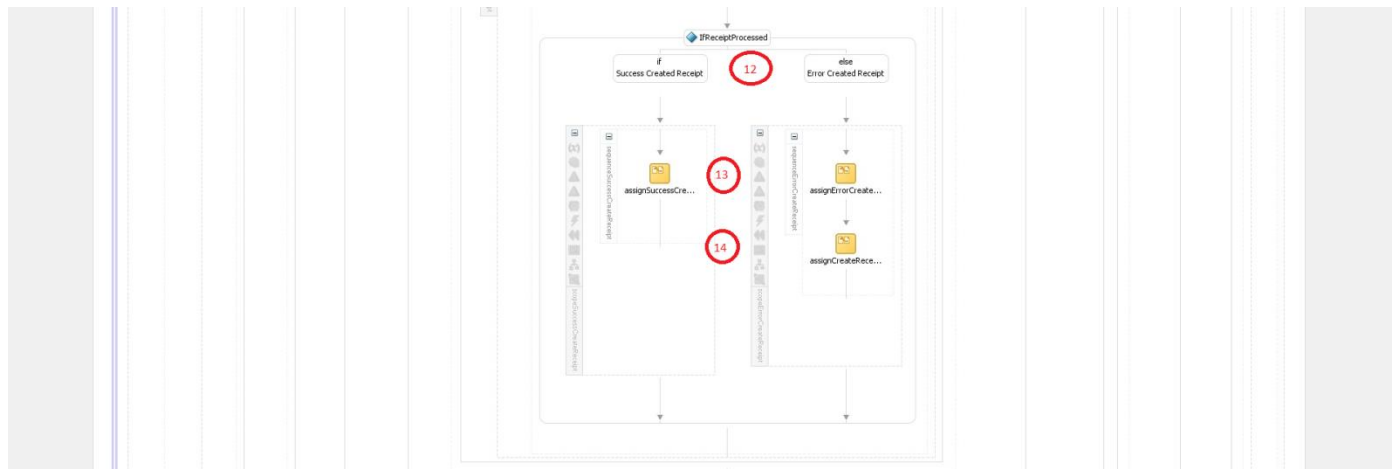
1. Se valida que la entrada del Servicio contenga información requerida para los Recibos.
2. Si no contiene información se termina el flujo del Servicio. En caso contrario establece variable Global.
3. Transforma los datos de entrada hacia la variable de entrada del Servicio “**FinancialsTec**”-“**GetReceiptOrInvoices**”.
4. Se invoca el Servicio “**FinancialsTec**”-“**GetReceiptOrInvoices**” para complementar información en la creación de Recibos.
5. Se valida si existen errores en la respuesta del Servicio “**FinancialsTec**”-“**GetReceiptOrInvoices**”. Si los hay se termina el flujo del servicio. En caso contrario continua con el flujo.



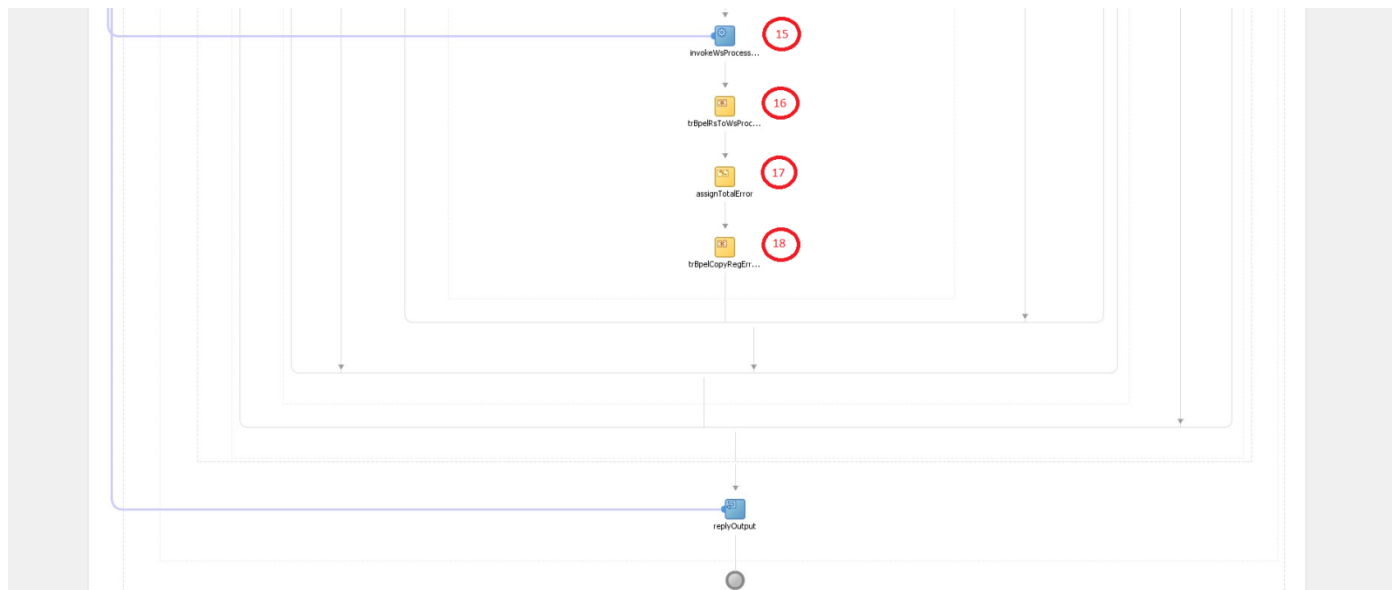
6. Se valida que exista al menos un registro en la respuesta del Servicio anterior. Si no lo hay se termina el flujo. En caso contrario continua con el flujo.
7. Se inicializa variables globales.
8. Ciclo For en el cual se va iterar la respuesta del Servicio anterior.
9. Se asigna el valor del contador del For a variable local.
10. Se asignan valores a los parámetros de entrada del Servicio “**ErpIntegrationServiceTec**”-“**CreateReceipt**”.
11. Se invoca el Servicio “**ErpIntegrationServiceTec**”-“**CreateReceipt**” para la creación de Recibos.



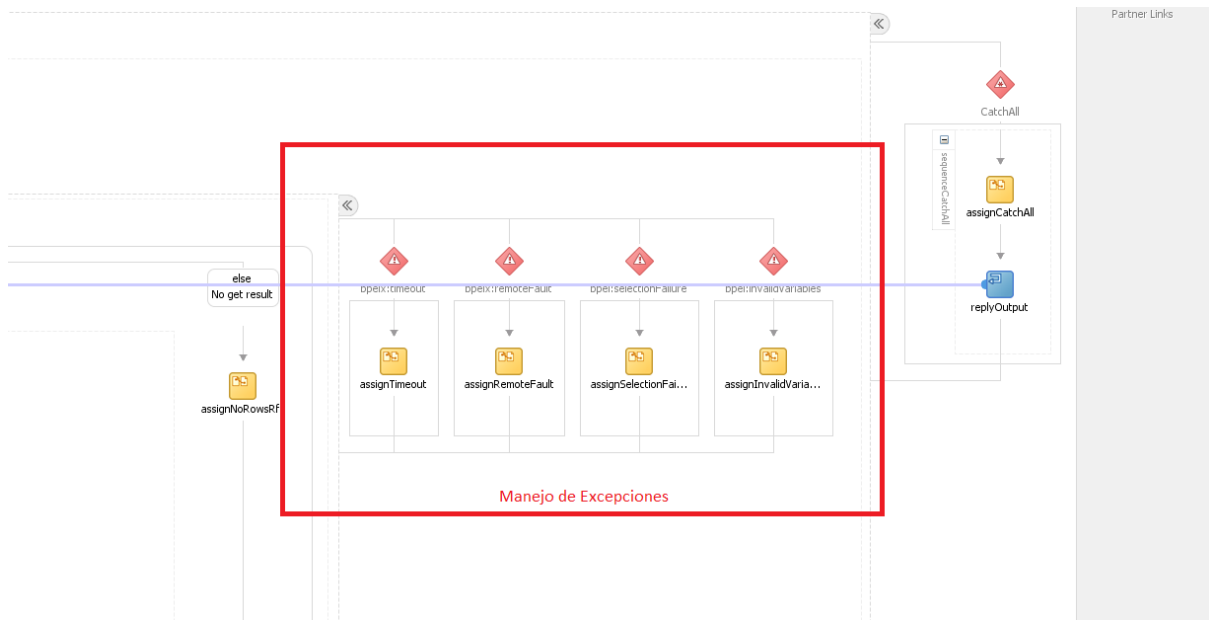
12. Valida la respuesta del Servicio “**ErpIntegrationServiceTec**”-“**CreateReceipt**”.
13. Si la respuesta del Servicio es correcta se incrementa la variable de procesos exitosos, en caso contrario si incrementa la variable de procesos con error.
14. Se establecen a la variable global los errores devueltos por el Servicio anterior. Aparte se llena la variable de entrada del Servicio “**FinancialsTec**”-“**ProcessedReceipt**” usado para registrar en base de datos la respuesta de la creación de los Recibos.



15. Se invoca el Servicio “**FinancialsTec**”-“**ProcessedReceipt**” para registrar la respuesta de la creación de cada Recibo.
16. La respuesta del Servicio anterior incluye la información de los Recibos que se aplicaran, esta información se transforma para ser regresada al proyecto ReceiptEnt.
17. Se asigna el valor de la variable global donde se registra la cantidad de errores hacia la variable de retorno de cantidad de procesos con errores.
18. Se transforma los datos de retorno para el total de procesos exitosos, total de errores, total de procesos y la descripción de los errores junto con sus códigos.

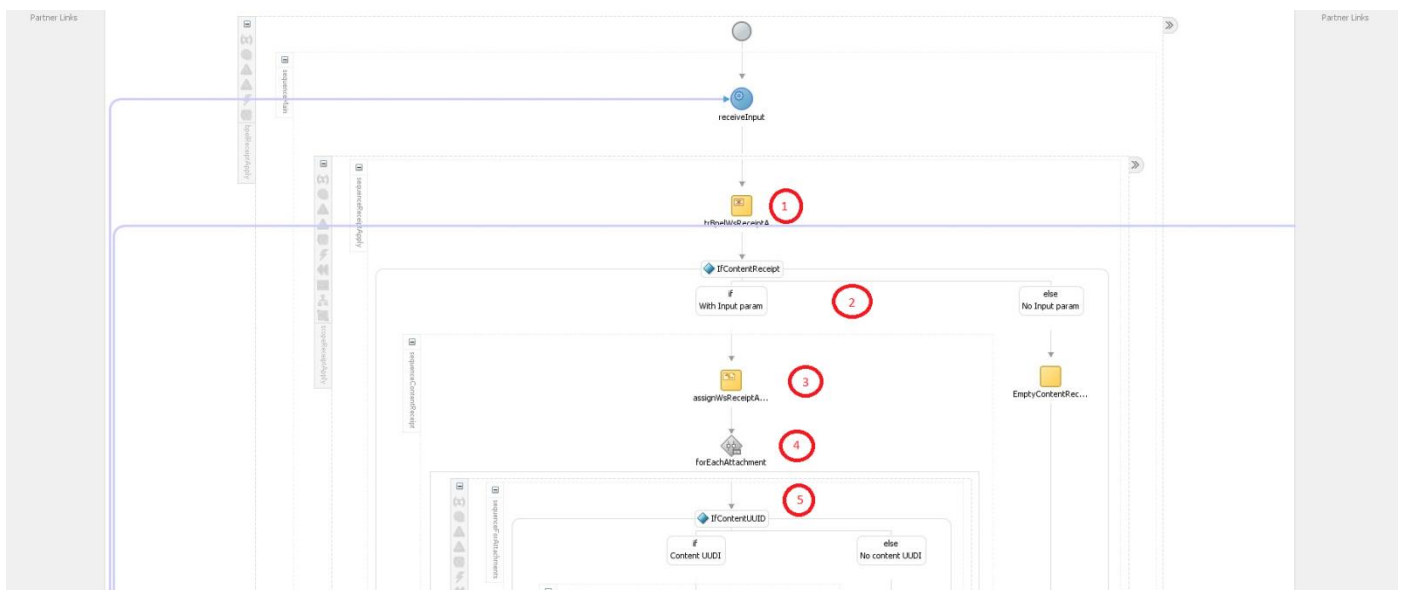


19. Esta región del bpel es utilizada para el manejo de Excepciones comunes en los Servicios Web.



bpelReceiptApply.bpel

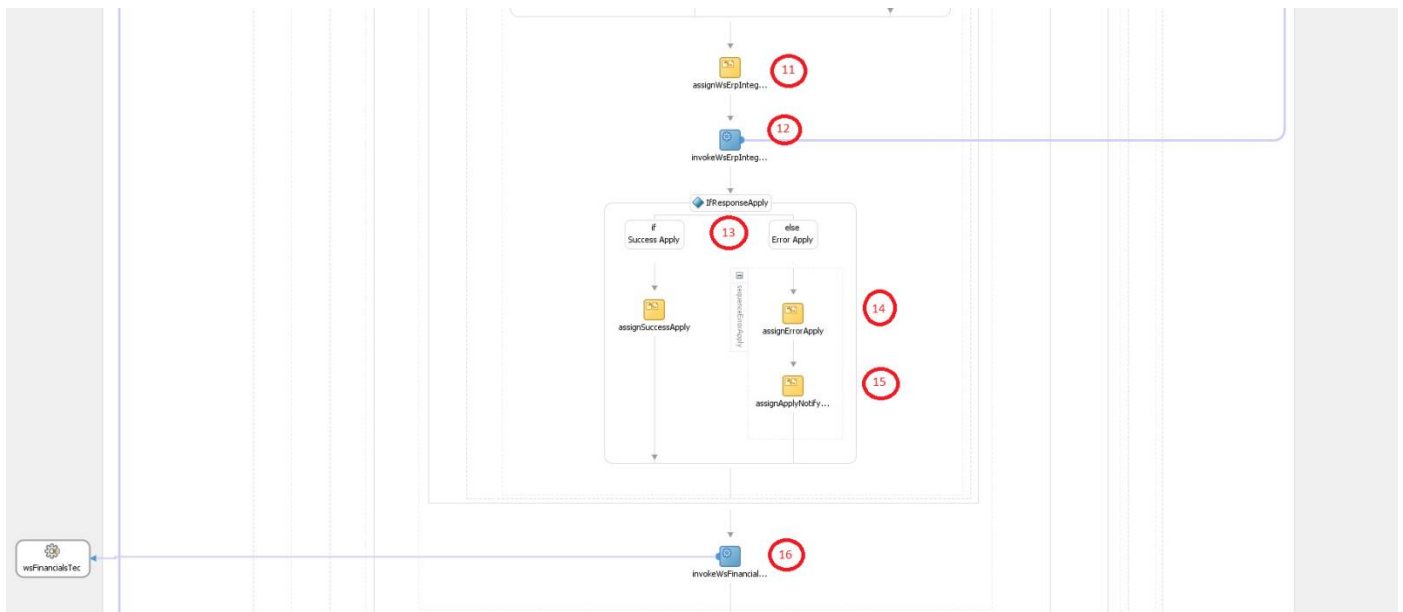
1. Transformación en la que se establecen los valores de entrada, necesarios para enviar la notificación. Estos valores se han enriquecido con los Servicios anteriores.
2. Se valida que existan valores para enviar Recibos a aplicar. Si no los hay el flujo se dirige a enviar la solicitud.
3. Establece variables globales.
4. Ciclo For el cual itera cada Recibo a aplicar.
5. Validación si el recibo a aplicar contiene valor UUID, se deberá agregar como Attachment, en caso contrario pasa al bloque en que se aplica el Recibo.



6. Utilizando los valores de entrada y de **dvmReceiptBiz.dvm** se asigna los valores de entrada para la invocación el Servicio **“ErpIntegrationServiceTec”.”UploadAttachment”**.
7. Se invoca el Servicio **“ErpIntegrationServiceTec”.”UploadAttachment”** en el cual se agregan los UUID por medio de los Attachment.
8. Se valida la respuesta del Servicio **“ErpIntegrationServiceTec”.”UploadAttachment”**.
9. Si la respuesta del Servicio es SUCCEED, incrementa la variable de procesos exitosos. En caso contrario incrementa la variable de procesos con error
10. Se establece el código de error y el mensaje de error para la variable global de las notificaciones.

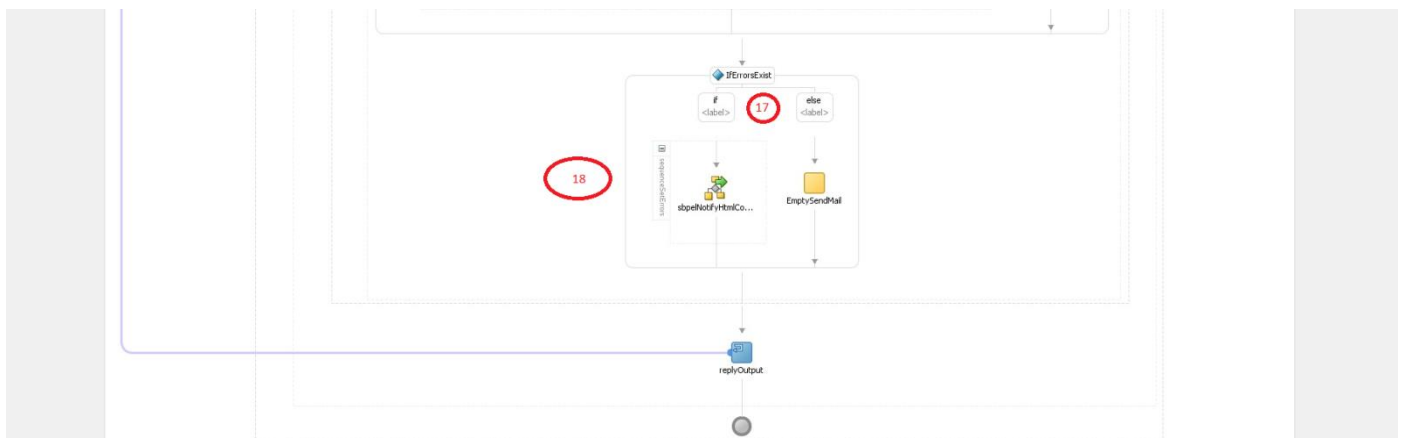


11. Asigna las variables de entrada para la invocación del Servicio **“ErpIntegrationServiceTec”-“CreateApplyReceipt”**.
12. Se invoca el Servicio **“ErpIntegrationServiceTec”-“CreateApplyReceipt”** el cual aplica los Recibos.
13. Se valida la respuesta del Servicio **“ErpIntegrationServiceTec”-“CreateApplyReceipt”**.
14. Si el Servicio **“ErpIntegrationServiceTec”-“CreateApplyReceipt”** crea correctamente la aplicación del Recibo este pasa a la asignación en donde se incrementa la variable de procesos exitosos. En caso contrario se asigna el incremento a la variable de procesos con error.
15. Se asigna el código del error y el mensaje del error a la variable global usada para las notificaciones, aparte se asigna la descripción del error a la variable usada para consumir el servicio **“FinancialsTec”-“ProcessedApplyReceipt”** usado para registrar en la base de datos cuales recibos no se pudieron aplicar.
16. Se invoca el Servicio **“FinancialsTec”-“ProcessedApplyReceipt”** para registrar si los recibos se pudieron o no aplicar.

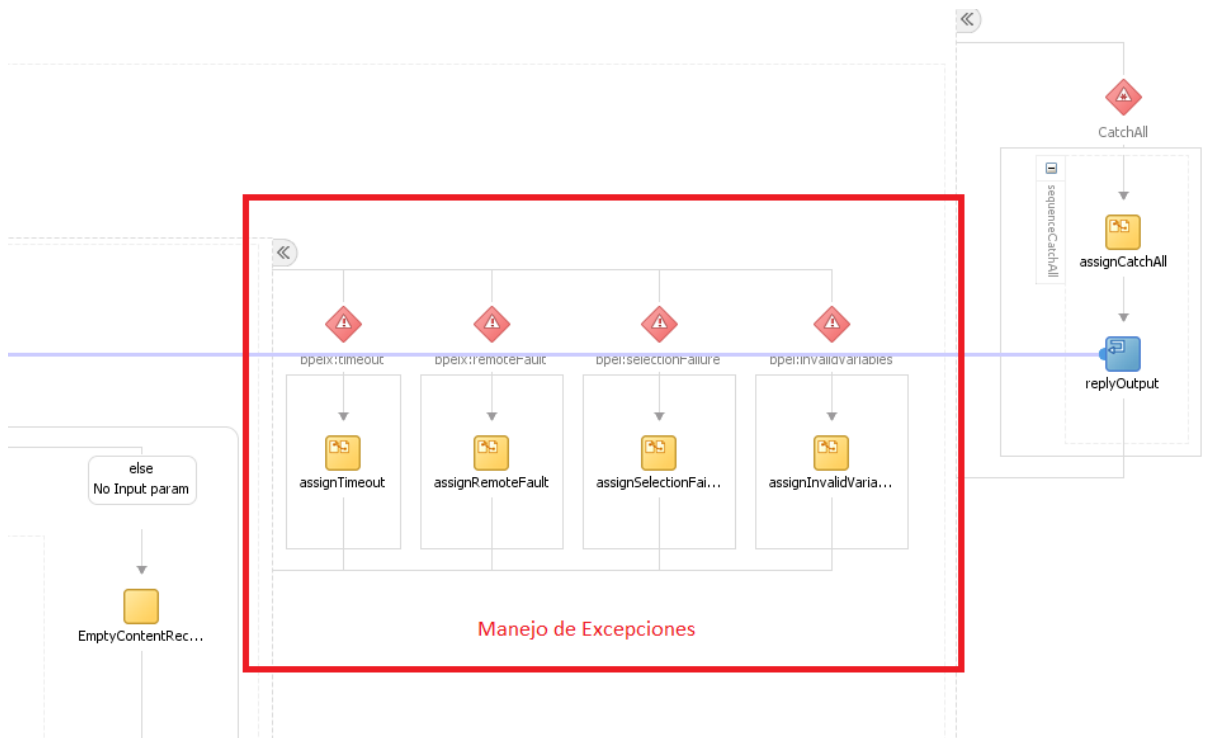


17. Se valida que existan transacciones exitosas y con errores para poder mandar la notificación.

18. Si se cumple la validación anterior se llama al subproceso **sbpelNotifyHtmlControl** dentro del cual se le pasan el total de procesos exitosos y los errores para ser enviados por correo electrónico.

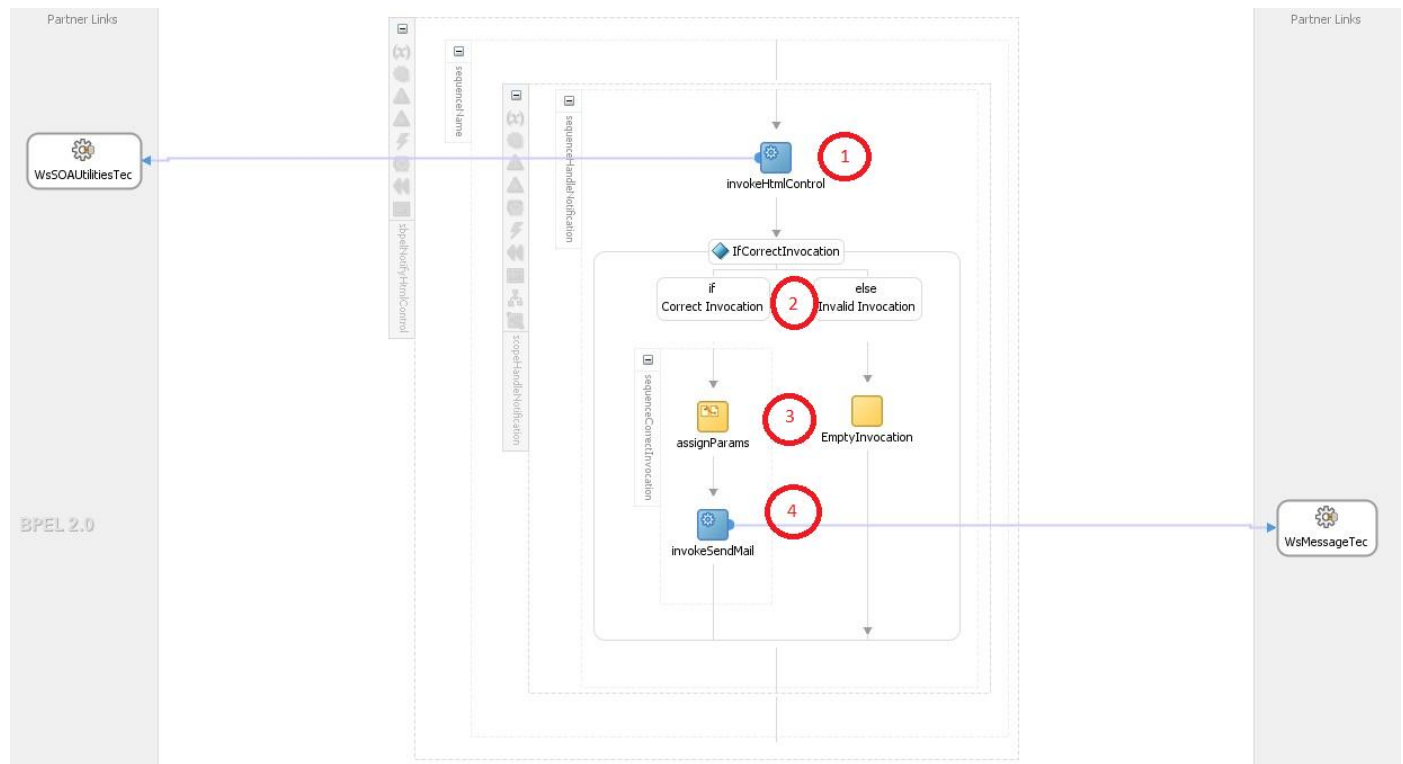


19. Esta región del bpm es utilizada para el manejo de Excepciones comunes en los Servicios Web.



sbpelNotifyHtmlControl.sbpel

1. Invoca al Servicio **“SOAUtilitiesTec”-“GetHtmlControl”** para Obtener un formato en Html con la descripción de los errores.
2. Se valida que no devuelva errores la respuesta de Servicio anterior.
3. Si la respuesta del Servicio es correcta se asigna la respuesta del servicio que es un Html y el valor de la DVM para obtener los destinatarios a los parámetros de entrada del Servicio **“MessageTec”-“SendEmail”**.
4. Se invoca el Servicio **“MessageTec”-“SendEmail”** el cual envía la notificación.



3.4 Pruebas

Para ejecutar una prueba manual es necesario ejecutar el Servicio de ReceiptEnt.

URL del servicio en el ambiente de desarrollo:

<http://<Host>:<Port>/soa-infra/services/default/ReceiptEnt/ReceiptEnt?WSDL>

El servicio no contiene parámetros. A continuación, un ejemplo del endpoint.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns1:CreateReceiptsRq
      xmlns:ns1="http://soa.estrellaroja.com.mx/ReceiptEnt"/>
    </soap:Body>
  </soap:Envelope>
```

4 Diseño de Datos

4.1 Tabla de diseño de datos

Los bpels que se usan, están compuestos por los siguientes componentes.

bpelGetRfc.bpel:

#	Nombre	Tipo	Descripción
1	assignWsFinancialTecReferenceOperationRq	assign	Asignación de valores.
2	invokeWsFinancialTecReferenceOperation	invoke	Invocación de Servicio.
3	trWsFinancialTecGetRfcRsToBpelGetRfcRs	transformation	Transformación de datos.

bpelCreateReceipts.bpel:

#	Nombre	Tipo	Descripción
1	assignInvokeWsFinancialsBiz	assign	Asignación de valores.
2	invokeWsFinancialsBizGetRfc	invoke	Invocación de Servicio.
3	IfResultsRfc	If	Validación de datos.
4	assignWsReceiptBizGenerateReportRq	assign	Asignación de valores.
5	assignNoRowsRfc	assign	Asignación de valores.
6	trBpelRqToWsERPIntegrationBizGenerateReportRq	transformation	Transformación de datos.
7	invokeWsERPIntegrationBizGenerateReport	invoke	Invocación de Servicio.
8	trWsERPIntegrationBizGenerateReportRs_To_WsERPIntegrationBizGenerateReportRs	transformation	Transformación de datos.
9	assignDataReceipt	assign	Asignación de valores.
10	trDataPaymentsReceipt_To_DataPaymentsReceiptResult	transformation	Transformación de datos.
11	IfCashInterfaceDM	If	Validación de datos.
12	trDataPaymentsReceiptResult_To_DataPaymentsReceiptBiz	transformation	Transformación de datos.
13	assignNoRows	assign	Asignación de valores.
14	invokeWsReceiptBizCreateReceipt	invoke	Invocación de Servicio.
15	ForGetCustomerTrx	forEach	Componente de repetición.

16	AssignVarTrxCount	assign	Asignación de valores.
17	assignInvokeWsIntegrationGenerateReportTrxCustomer	assign	Asignación de valores.
18	invokeWsIntegrationGenerateReportTrxCustomer	invoke	Invocación de Servicio.
19	trWsErpIntegrationBizGenerateReportRS_To_GenerateCustTrx	transformatio n	Transformació n de datos.
20	trWsErpIntegrationBizGenerateReportRS_To_GenerateOrgUnit	transformatio n	Transformació n de datos.
21	assignDataReceiptApply	assign	Asignación de valores.
22	trPaymentsReceiptApply_To_PaymentsReceiptApplyResult	transformatio n	Transformació n de datos.
23	IfGetCustomerTrx	If	Validación de datos.
24	assignValueCustTrx	assign	Asignación de valores.
25	EmptyGetCustomerTrx	Empty	Componente vacío.
26	trWsReceiptBizReceiptApplyRq	transformatio n	Transformació n de datos.
27	invokeWsReceiptBizReceiptApply	invoke	Invocación de Servicio.
28	assignApplyReceiptRs	assign	Asignación de valores.

bpelDigitalStampReceipts.bpel:

#	Nombre	Tipo	Descripción
1	IfDMInput	If	Validación de datos.
2	assignWsReceiptTecRq	assign	Asignación de valores.
3	assignNoRowsRfc	assign	Asignación de valores.
4	trBpelRqToWsFinancialsTecGetInformationReceiptRq	transformation	Transformación de datos.
5	invokeWsFinancialsTecProcedure	invoke	Invocación de Servicio.
6	IfProcResult	If	Validación de datos.
7	assignErrorProcedure	assign	Asignación de valores.
8	IfResultProcReceipt	If	Validación de datos.
9	assignWsCreateReceiptRq	assign	Asignación de valores.
10	assignNoRowReceipt	assign	Asignación de valores.
11	ForCreateReceipt	forEach	Componente de repetición.
12	assignVarCount	assign	Asignación de valores.
13	assignInvokeWsStandardReceiptService	assign	Asignación de valores.

14	invokeWsErpIntegrationTecStandardReceipt	invoke	Invocación de Servicio.
15	IfReceiptProcessed	If	Validación de datos.
16	assignSuccessCreateReceipt	assign	Asignación de valores.
17	assignErrorCreateReceipt	assign	Asignación de valores.
18	assignCreateReceiptNotifyHtml	assign	Asignación de valores.
19	invokeWsProcessedReceipt	invoke	Invocación de Servicio.
20	TrBpelRsToWsProcessedReceipt	transformation	Transformación de datos.
21	assignTotalError	assign	Asignación de valores.
22	trBpelCopyRegErrorSuccess	transformation	Transformación de datos.

bpelReceiptApply.bpel:

#	Nombre	Tipo	Descripción
1	trBpelWsReceiptApplyRq	transformation	Transformación de datos.
2	IfContentReceipt	If	Validación de datos.
3	assignWsReceiptApplyRq	assign	Asignación de valores.
4	EmptyContentReceipt	Empty	Componente vacío.
5	forEachAttachment	forEach	Componente de repetición.
6	IfContentUUID	If	Validación de datos.
7	assignUploadAttachment	assign	Asignación de valores.
8	emptyUUID	Empty	Componente vacío.
9	invokeWsErpIntegrationServiceTecUploadAttachment	invoke	Invocación de Servicio.
10	IfResponseAttachment	If	Validación de datos.
11	assignSuccessAttachment	assign	Asignación de valores.
12	assignErrorAttachment	assign	Asignación de valores.
13	assignAttachmentNotifyHtml	assign	Asignación de valores.
14	assignWsErpIntegrationServiceTecCreateApplyReceiptRq	assign	Asignación de valores.
15	invokeWsErpIntegrationServiceTecCreateApplyReceipt	invoke	Invocación de Servicio.
16	IfResponseApply	If	Validación de datos.
17	assignSuccessApply	assign	Asignación de valores.
18	assignErrorApply	assign	Asignación de valores.

19	assignApplyNotifyHtml	assign	Asignación de valores.
20	invokeWsFinancialsTecRegApplyReceipt	invoke	Invocación de Servicio.
21	IfErrorsExist	If	Validación de datos.
22	sbpelNotifyHtmlControl	Subprocess	Componente subproceso bpel
23	EmptySendMail	Empty	Componente vacío.

sbpelNotifyHtmlControl.sbpel:

#	Nombre	Tipo	Descripción
1	invokeHtmlControl	invoke	Invocación de Servicio.
2	IfCorrectInvocation	If	Validación de datos.
3	assignParams	assign	Asignación de valores.
4	EmptyInvocation	Empty	Componente vacío.
5	invokeSendMail	invoke	Invocación de Servicio.

4.2 Origen de Datos

XXER_AR_RECEIPT_TBL

Tabla en donde se almacena los recibos que se crearan y toda la información que necesita el servicio web para crearlo. También se almacena la respuesta del servicio web para crear Recibos. De esta forma se lleva un control de la información.

Table	Column	Source Value
TABLA_PUENTE_ID	NUMBER(18,0)	Identificador de Tabla
RFC	VARCHAR2(25 BYTE)	Indica RFC del cliente
AMOUNT_RECIBO	NUMBER(18,0)	Monto total del Recibo
SITE_USE_ID	NUMBER(18,0)	Indica el site use id
DEPOSIT_DATE	DATE	Indica la fecha de deposito
CUST_ACCOUNT_ID	NUMBER(18,0)	Indica la clave del cliente
RECEIPT_DATE	DATE	Indica la fecha del Recibo
RECEIPT_METHOD_ID	VARCHAR2(20 BYTE)	Indica la clave del Método de Recibo
RECEIPT_NUMBER	VARCHAR2(25 BYTE)	Indica el número de Recibo
ORGANIZATION_ID	NUMBER(18,0)	Indica la Clave de la organización
RECEIPT_ID	NUMBER(18,0)	Indica la clave del Recibo
ESTATUS	CHAR(1 BYTE)	Indica el estatus de la creación del Recibo
ERROR_DESCRIPTION	VARCHAR2(500 BYTE)	Error generado durante la creación del Recibo
CREATED_BY	VARCHAR2(150 BYTE)	Indica la persona que crea el registro
CREATION_DATE	TIMESTAMP(6)	Fecha de creación del registro
LAST_UPDATED_BY	VARCHAR2(150 BYTE)	Indica la persona que actualiza el registro
LAST_UPDATE_DATE	TIMESTAMP(6)	Fecha de actualización del registro
OBJECT_VERSION_NUMBER	NUMBER(9,0)	Indica el número de versión del registro
ATTRIBUTE1	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE2	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE3	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE4	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE5	VARCHAR2(250 BYTE)	Atributo auxiliar
PARTY_NAME	VARCHAR2(360 BYTE)	Indica el Nombre del Cliente
METODO_RECIBO	VARCHAR2(30 BYTE)	Indica el Método del Recibo

XXER_AR_RECEIPT_APPLY_TBL

Dentro de esta tabla se almacena todos los parámetros que requiere el servicio para Aplicar Recibos.

Posteriormente se actualizará cada registro para guardar la respuesta del servicio web de Aplicar Recibos.

Table	Column	Source Value
RECEIPT_APPLY_ID	NUMBER(18,0)	Indica clave de la tabla
TABLA_PUENTE_ID	NUMBER(18,0)	Indica la referencia al recibo
AMOUNT	NUMBER(18,0)	Monto de la factura
PARTY_NAME	VARCHAR2(360 BYTE)	Nombre del Cliente
CUSTOMER_TRX_ID	NUMBER(18,0)	Indica el Customer Trx
TRX_NUMBER	VARCHAR2(20 BYTE)	Indica el número de factura
ESTATUS	CHAR(1 BYTE)	Indica el estatus de la aplicación del recibo
ERROR_DESCRIPTION	VARCHAR2(500 BYTE)	Indica el error que se genera al aplicar el recibo
CREATED_BY	VARCHAR2(150 BYTE)	Indica la persona que crea el registro
CREATION_DATE	TIMESTAMP(6)	Fecha en que se crea el registro
LAST_UPDATED_BY	VARCHAR2(150 BYTE)	Indica la persona que actualiza el registro
LAST_UPDATE_DATE	TIMESTAMP(6)	Fecha en que se actualiza el registro
OBJECT_VERSION_NUMBER	NUMBER(9,0)	Indica el número de la versión
ATTRIBUTE1	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE2	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE3	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE4	VARCHAR2(250 BYTE)	Atributo auxiliar
ATTRIBUTE5	VARCHAR2(250 BYTE)	Atributo auxiliar

4.3 Lógica de Validación

Las validaciones principales que se encuentran dentro de esta integración tenemos las siguientes:

- Se crearán los recibos que estén con estatus **Null** y **E**.
- Se aplicarán todos los recibos que ya fueron antes creados
- El la sumatoria del monto de cada factura no deberá pasar el monto del recibo.

5 Diseño SQL

Se presenta las sentencias utilizadas para obtener la información del ERP Cloud, las cuales complementan la información de los Recibos.

5.1 Sentencias SQL

Los Data Models que se utilizan se ubican en la siguiente ruta:

/Shared Folders/Custom/ER_INTEGRATION/ER05_AR_Creacion_de_recibos/Data Models

1. XXER_AR_CASH_INTERFACE
 - a. Q_CUSTOMER
 - b. Q_ORG
2. XXER_AR_CUST_TRX_INTERFACE
 - a. CUSTOMER_TRX
 - b. ORG_UNIT

Data Set: Q_CUSTOMER

```
SELECT HP.JGZZ_FISCAL_CODE RFC
      , HP.PARTY_NAME
      , HP.PARTY_NUMBER
      , HP.PARTY_ID
      , HCA.ACCOUNT_NUMBER
      , HCA.CUST_ACCOUNT_ID
      , HCAS.CUST_ACCT_SITE_ID
      , HPS.PARTY_SITE_NUMBER
      , HCSU.SITE_USE_ID
      , HCSU.SITE_USE_CODE
      , HCSU.PRIMARY_FLAG
      , HCSU.LOCATION
      , HL.LOCATION_ID
FROM HZ_CUST_ACCOUNTS HCA
      , HZ_PARTIES HP
      , HZ_CUST_ACCT_SITES_ALL HCAS
      , HZ_CUST_SITE_USES_ALL HCSU
      , HZ_PARTY_SITES HPS
```

```

        , HZ_LOCATIONS HL
WHERE 1 = 1
        AND HCA.PARTY_ID = HP.PARTY_ID
        AND HCA.CUST_ACCOUNT_ID = HCAS.CUST_ACCOUNT_ID
        AND HCSU.CUST_ACCT_SITE_ID= HCAS.CUST_ACCT_SITE_ID
        AND HPS.PARTY_SITE_ID = HCAS.PARTY_SITE_ID
        AND HPS.LOCATION_ID = HL.LOCATION_ID
        AND HP.JGZZ_FISCAL_CODE IN (:P_RFC)

```

Data Set: Q_ORG

```

SELECT xr.registration_number RFC_ORIGEN
        , xr.registered_name CLIENTE
        , HO.organization_id
        , HO.NAME
FROM hr_operating_units ho
        , xle_entity_profiles xe
        , xle_registrations xr
WHERE 1 = 1
and xe.legal_entity_id      (+)  = ho.default_legal_context_id
and xe.legal_entity_id      = xr.source_id(+)

```

Data Set: CUSTOMER_TRX

```

SELECT CUSTOMER_TRX_ID
        ,ATTRIBUTE3 AS UUID
FROM RA_CUSTOMER_TRX_ALL
WHERE 1                                = 1
        AND TRX_NUMBER                  = :TrxNumber
        AND ORG_ID                      = :OrgId
        AND BILL_TO_CUSTOMER_ID         = :CustomerId
        AND ROWNUM                      = 1

```

Data Set: ORG_UNIT

```

SELECT NAME AS BUSINESS_UNIT
FROM HR_ALL_ORGANIZATION_UNITS
WHERE 1                                = 1
        AND ORGANIZATION_ID = :OrgId

```

6 Reglas de Negocio

Las reglas de negocio para esta integración es la siguiente:

- Validar la creación de un recibo y su respectiva aplicación y que se aplique a varias facturas siempre y cuando se verifique el estatus del recibo.
- Validar el caso de error en el cual no se genere el recibo y se envíe la notificación del error y en la tabla de superinterfaz se actualice el status E
- Validar un recibo que haya tenido un error y se corrige el registro en la tabla se puede reprocesar
- El recibo se esté creando con las cuentas correctas de acuerdo al método de recibo de origen(Superinterfaz)
- Las siguientes columnas dentro de estas tablas son requeridas para la Creación y Aplicación de los recibos.
 - ✓ XXER_CNF_INTERFASE_INGRESOS_AR:
RECEIPT_METHOD_ID
ORGANIZATION_ID
TIPO_DOCUMENTO = 'RECIBO'
 - ✓ XXER_TBL_PUENTE_HEADER:
TABLA_PUENTE_ID
SISTEMA_ORIGEN
TIPO_DOCUMENTO = 'RECIBO'
TIPO_MOVIMIENTO
UNIDAD_NEGOCIO_ORIGEN
RFC_DESTINO
REFERENCIA_APLICACION
ESTATUS
 - ✓ XXER_TBL_PUENTE_LINES:
TABLA_PUENTE_LINEA_ID
TABLA_PUENTE_ID
TOTAL

6.1 Diseño del Servicio

Para poder iniciar la integración es necesario ejecutar el compuesto **ReceiptEnt**.

Para esta ejecución no requiere el establecer parámetros de salida.

7 Consideraciones de Rendimiento

Este requerimiento ha sido probado en un ambiente que contaba con una pequeña muestra de recibos, tener contemplado que el compuesto puede presentar demora con una cantidad de registros muy grande a procesar.

7.1 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.

7.2 Seguridad

- Se recomienda el monitoreo oportuno de la base de datos para asegurar el correcto espacio para su crecimiento.
- 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.

7.3 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_12.2.1.2.0_GENERIC_161008.1648.S

7.4 Catálogo de Errores

El código de error que se maneja para esta integración se especifican en el documento “**CatalogosErroresEstrellaRoja.xlsx**”.

8 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:

- XXER_AR_RECEIPT_TBL
- XXER_AR_RECEIPT_APPLY_TBL

En este esquema debe existir el siguiente paquete:

- XXER_UTILITIES_PKG

En este esquema debe existir los tipos:

- XXER_AR_APPLY_RECEIPT_TTYPE
- XXER_AR_APPLY_RECEIPT_TYPE
- XXER_AR_PROCESSED_APPLY_RECEIPT_TTYPE
- XXER_AR_PROCESSED_APPLY_RECEIPT_TYPE
- XXER_AR_PROCESSED_RECEIPT_TTYPE
- XXER_AR_PROCESSED_RECEIPT_TYPE
- XXER_AR_RECEIPT_INVOICE_TTYPE
- XXER_AR_RECEIPT_INVOICE_TYPE
- XXER_AR_RECEIPT_TTYPE
- XXER_AR_RECEIPT_TYPE

En este esquema debe existir la vista:

- XXER_AR_CUSTOMER_RFC_V

De lo contrario se requiere crearlas, para ejecutarlo se debe contar con los permisos necesarios ya sea desde sqlplus o un entorno gráfico.



Instalación de la integración de Creación de Recibos para las capas **Ent** y **Biz**, instalar como aplicación SOA:

- sca_FinancialsBiz.jar
- sca_ReceiptBiz.jar
- sca_ReceiptEnt.jar

9 URL de Acceso y seguridad

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

Servicio	URL
ReceiptEnt	http://<host>:<puerto>/soa-infra/services/default/ReceiptEnt/ReceiptEnt?WSDL
ReceiptBiz	http://<host>:<puerto>/soa-infra/services/default/ReceiptBiz/ReceiptBiz?WSDL
FinancialsBiz	http://<host>:<puerto>/soa-infra/services/default/FinancialsBiz/FinancialBiz?WSDL

10 Temas abiertos y cerrados

10.1 Temas Abiertos

ID	Tema	Solución	Responsabilidad	Fecha Objetivo	Fecha impacto

10.2 Temas Cerrados

ID	Tema	Solución	Responsabilidad	Fecha Objetivo	Fecha impacto