|  |
| --- |
|  |

OUM

TA.070-9 DISEÑO TÉCNICO DE EXTENSIONES E

INTERFACES

Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

ER04\_AR\_Timbrado\_de\_Transacciones

|  |  |
| --- | --- |
| Autor: | Forte Innovation |
| Fecha de creación: | Enero 15 2018 |
| Última actualización: | 15 de septiembre de 2022 |
| Código de referencia OUM: | DS.140 Design Specification |
| Versión:  **Aprobadores:** | 1.1 |
| Ivan Muñoz |  |

Ángel Flores



**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 4

4.1 Calendarización y Ejecución 5

4.2 InvoicesEnt 6

4.3 InvoicesBiz 8

4.4 DigitalStampGERTec 11

4.5 Pruebas 16

5 Diseño de Datos 17

5.1 Tabla de diseño de datos 17

5.2 Origen de Datos 18

5.3 Lógica de Validación 19

6 Diseño SQL 20

6.1 Sentencias SQL 20

7 Reglas de Negocio 22

7.1 Diseño del Servicio 22

8 Consideraciones de Rendimiento 23

8.1 Estrategia de Reinicio 23

8.2 Seguridad 23

8.3 Personalización 23

8.4 Catálogo de Errores 24

9 Consideraciones de Instalación 25

10 URL de Acceso y seguridad 26

11 Temas abiertos y cerrados 27

11.1 Temas Abiertos 27

11.2 Temas Cerrados 27

# Control de Documento

## Bitácora de Cambios

3

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Referencia del cambio** |
| 02-feb-2018 | Efrain Arellanes | 1.0 | No Previous Document |
| 10-Sep-2021 | Pedro Sanchez | 1.1 | Cambio de PAC AMS |
|  |  |  |  |
|  |  |  |  |

## Revisores

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

# Resumen técnico

Esta especificación documenta el diseño para el timbrado de transacciones, cuyo caso de uso se describió en el documento **TA020-4\_ ER04\_AR\_Timbrado\_de\_Transacciones**.

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

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

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

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

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

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

Existen dos tipos de integraciones:

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

Existen 4 fases importantes para este tipo de integraciones:

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

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

# Diagrama de la integración

Diagrama

Descripción generada automáticamente

# Lógica de implementación

La capa **Ent** administra las capas **Biz** utilizadas para esta integración y es la que se tiene programada en el servidor SOA para ejecutarse en los horarios establecidos.

Como mencionamos anteriormente, se cuenta con tres capas en la integración de facturación electrónica, 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 5 minutos:

|  |  |  |
| --- | --- | --- |
| **Componente** | **Propiedad** | **Valor** |
| **Trabajo** | Nombre | InvoicesEnt |
| Nombre Mostrado | InvoicesEnt |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Definición de trabajo para facturación electrónica |
| Tipo de trabajo | SyncWebserviceJobType |
| WSDL | [http://129.150.110.0:80/soa-](http://129.150.110.0/soa-infra/services/default/InvoicesEnt/InvoicesEnt?WSDL%20)  [infra/services/default/InvoicesEnt/InvoicesEnt?WSDL](http://129.150.110.0/soa-infra/services/default/InvoicesEnt/InvoicesEnt?WSDL%20) |
| Tipo de puerto | InvoicesEntPort |
| Operación | ElecInvoices |
| **Planificación** | Nombre | InvoicesElect |
| Nombre Mostrado | InvoicesElect |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Programación diaria cada 5 minutos |
| Frecuencia | Diariamente |
| Zona horaria | (UTC-06:00) Ciudad de México |

## InvoicesEnt

En la capa InvoicesEnt se realiza una serie de pasos concentrados en un bpel como se muestra en la siguiente imagen:

Diagrama

Descripción generada automáticamente

Como servicio expuesto de lado izquierdo tenemos la operación **ElecInvoices** que es la operación programada en el ESS, posterior utilizamos un mediator para dirigir a **bpelElecInvoice** donde se concentra la lógica para hacer el llamado a las capas Biz correspondientes (InvoicesBiz y ErpIntegrationBiz).

El bpel realiza las siguientes acciones:

1. Se crea la variable de Request para la respuesta del servicio InvoicesEnt y se asignan los datos correspondientes para mandar llamar ERPIntegractionBiz.
2. Se invoca el servicio ERPIntegractionBiz.
3. Se evalúa si el ERPIntegractionBiz retorno un nudo de errores.
4. Se asigna mensaje de error a la salida del servicio en caso de que ERPIntegractionBiz haya retornado nodo de errores.
5. Se evalúa que la información que regresa el servicio de ERPIntegractionBiz contenga facturas para ser timbradas.
6. Se asigna la información de la respuesta del servicio ERPIntegractionBiz a la variable de entrada de InvoicesBiz.

Se invoca el servicio de InvoicesBiz.

Interfaz de usuario gráfica, Diagrama, Aplicación

Descripción generada automáticamente

## InvoicesBiz

En la capa **InvoicesBiz** se tiene una mayor cantidad de pasos a realizar debido a que se gestiona toda la lógica del timbrado. Se parte de la operación expuesta DigitalStamp, posterior se tiene un mediador que direcciona al bpelDigitalStamp en donde el servicio hace llamadas a subproceso de bpels para su operación.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

El **bpelDigitalStamp** contiene los siguientes pasos:

1. Se inicializan las variables para el manejo de las notificaciones en 0, se realiza la configuración de los parámetros necesarios para la variable de entrada de ERPIntegrationServiceTec y se le da formato XML a lo información de entrada de InvoicesBiz.
2. Se usa un xsl para colocar correctamente los datos en el namespace de la variable que va iterara las facturas.
3. Se crea un for para iterara las facturas y procesarlas de una a una.
4. Se asigna según el valor del for para iterar las facturas y enviar a la variable de uso temporal, se actualiza la variable de entrada para la actualización de los flex, por último se asigna el valor total de la factura a una variable que servirá como entrada de un servicio para convertir de numero a letra.
5. Se invoca el servicio SOAUtilitiesTec para transformar de número a letra.
6. Se asigna el resultado de transformar de número a letra a la variable temporal para timbrar.
7. Se utiliza un archivo xsl para transferir a la variable de entrada del servicio para timbrar.

Lógica de implementación

1. Se invoca el servicio DigitalStampTec.
2. Se evalúa si existe errores del servicio DigitalStampTec.
3. Se usa un archivo xsl para el mapeo de varios orígenes a la variable de entrada para el registro de errores en la tabla de control.
4. Se incrementa en uno la variable de conteo de errores, se asignan valores a la variable para el registro de errores en la tabla de control y en la salida del servicio.
5. Se actualiza la variable con el dato correspondiente de error para la actualización del flex.
6. Asignación de valores a la variable de entrada de subproceso de bpel para la notificación de errores.
7. Invocación de AuditControlTec para el registro de errores.
8. Se actualizan las variables correspondientes para actualizar los attachments, adicional se incrementa el total de facturas exitosas y se asignan los valores correspondientes para la inserción en las tablas de control con un timbrado exitoso.
9. Se asigna el valor correspondiente para la actualización de los flex con el UUID, folio y estatus.
10. Se invoca el servicio AuditControlTecPs.
11. Se invoca el subproceso del bpel para la actualización de los attachments.
12. Se invoca el servicio ERPIntegrationServiceTec para la actualización de flexs.
13. Condición para saber si existen errores en la variable de salida del servicio InvoicesBiz.
14. Se elimina el nodo de Success en caso de encontrar errores.
15. Se invoca el subproceso para él envió de la notificación de errores.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## DigitalStampGERTec

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.
3. Diagrama

   Descripción generada automáticamenteDespués 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 | Interfaz de usuario gráfica  Descripción generada automáticamente |
| 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:

### Lista de objetos

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** | **GIT** |
| 1 | DigitalStampGERTecPS | Web  Service | Servicio Expuesto |  |
| 2 | DigitalStampGERApi | Adaptador | Adaptador de REST |  |
| 3 | DigitalStampGERTecPL | Tubería | Flujo de tubería |  |

### createTrxCFDI

La operación **createTrxCFDI** 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 “createTrxCFDIRq”.
2. Se toma el tipo de documento que se va a procesar de document\_type\_description y se coloca en v\_operation.
3. Se procede a mapeo de datos con la transformación DigitalStampRq\_to\_DigitalStampAPIRq.xsl para generar la petición del servicio Rest a consumir y se guarda en la variable varXMLJsonSchema.
4. Ejecutamos un traslate que convierte nuestro Shema XML a Json .
5. Se asigna nuestro json generado al cuerpo de la petición a enviar.
6. Se rutea al servicio saliente correspondiente a esta operación.
7. Dependiendo de la v\_operation ejecutamos la API correspondiente y enviamos la petición generada al Adaptador DigitalStampGERApi
8. Si la respuesta es correcta convertimos la respuesta de Json a XML con un traslate.
9. Después ejecutamos el mapeo de datos de DigitalStampGERApi a DigitalStampGERTecPS con la transformación DigitalStampAPIRs\_to\_DigitalStampRs.xsl.
10. Una ves Realizado esto remplazamos el resultado en el cuerpo de respuesta.

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

### Lista de objetos

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | BranchNode | BranchNode | Menú de operaciones del servicio |
| 2 | Pipeline Pair Node | Pipeline Pair Node | Flujo de petición y response del servicio |
| 3 | Assing | assign | Assingna un valor a una variable |
| 4 | nXSD Translate | nXSD Translate | Transforma datos de un esquema nativo a otros o viceversa |
| 5 | Error Handler | Error Handler | Cacha los errores ocurridos. |
| 6 | If Then | If Then | operación que condición si . |
| 7 | Java Callout | Java Callout | Ejecuta clase java. |
| 8 | Replace | Replace | Remplaza una variable por otra |
| 9 | Condicional Branch | Condicional Branch | Condiciona que servicios ejecutar. |
| 10 | insert | insertt | Inserta valores a la petición o salida. |

### 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/DigitalStampGERTecPS

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

**Request:** Valores de entrada no requeridos

Texto

Descripción generada automáticamente

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, Correo electrónico

Descripción generada automáticamente

## Pruebas

Las pruebas de timbrado de facturas se realizaron mediantela ejecución del compuesto desde el llamado de su endpoint con los parametros requeridos. Donde no require parametros de entrada.

URL del servicio en el ambiente de desarrollo:

<http://ersoacsprod-wls-1:9073/soa-infra/services/default/InvoicesEnt/InvoicesEnt?WSDL>

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

<soap:Body>

<ns1:ElecInvoicesRq xmlns:ns1=http://soa.estrellaroja.com.mx/InvoicesEnt/>

</soap:Body>

</soap:Envelope>

# Diseño de Datos

En este punto abordaremos todo lo referente a la parte de diseño de los datos utilizados durante la integración.

## Tabla de diseño de datos

Los bpels utilizados en esta integración cuentan con los siguientes componentes:

bpelDigitalStamp

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Nombre** | **Tipo** | **Descripción** |
| 1 | Scope2 | scope | Scope. |
| 2 | assignRq | assign | Asignación de valores. |
| 3 | transData\_DS\_To\_Invoice | transformation | Transformación de datos. |
| 4 | ForEachInvoice | forEach | Componente de repetición. |
| 5 | IfErrorsExist | if | Validación de datos. |
| 6 | assignInvoice | assign | Asignación de valores. |
| 7 | invokeWsSOAUtilitiesTecTransformNumberToLetter | invoke | Invocación de Servicio. |
| 8 | assignSOAUtilitiesTec | assign | Asignación de valores. |
| 9 | transDigitalStampTec | transformation | Transformación de datos. |
| 10 | invokeWsDigitalStampTec | invoke | Invocación de Servicio. |
| 11 | IfErrorsDigitalStampTec | if | Validación de datos. |
| 12 | invokeWsERPIntegrationServiceTecUpdateObject | invoke | Invocación de Servicio. |
| 13 | transInsertErrorsRq | transformation | Transformación de datos. |
| 14 | assignErrorsDigitalStamp | assign | Asignación de valores. |
| 15 | assignUpdateError | assign | Asignación de valores. |
| 16 | assignNotifyHtml | assign | Asignación de valores. |
| 17 | invokeWsAuditControlTecInsertArElecInvoiceErrors | invoke | Invocación de Servicio. |
| 18 | assigSuccessDigitalStamp | assign | Asignación de valores. |
| 19 | assignUpdateSuccess | assign | Asignación de valores. |
| 20 | invokeWsAuditControlTecPsInsertArElecInvoice | invoke | Invocación de Servicio. |
| 21 | sbpelUploadAttachments | subprocess | Invocacion de subproceso |
| 22 | invokeWsERPIntegrationServiceTecUpdateObject | invoke | Invocacion de servicio |
| 23 | IfErrorsExist | If | Validación de datos. |
| 24 | assignDeleteSuccess | assign | Asignación de valores. |
| 25 | sbpelNotifyHtmlControl | subprocess | Invocacion de subproceso |

## Origen de Datos

Hace referencia a las tablas custom empleadas en esta integración, dichas tablas son solo de control.

Tabla: **XXER\_AR\_ELEC\_INVOICE**

La tabla contendrá el registro de las transacciones que hayan sido exitosas al momento de timbrar.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo de dato** | **Descripción** |
| CUSTOMER\_TRX\_ID | NUMBER | Identificador único de la factura. |
| NUMERO\_FACTURA | VARCHAR2 (50 Char) | Identificador número de la factura. |
| TIPOCFD | VARCHAR2 (20 Char) | Tipo de documento. |
| UUID | VARCHAR2 (100 Char) | UUID con que fue generado |
| PAYLOAD | CLOB | Request que se usó para timbrar. |
| ATTRIBUTE1 | VARCHAR2 (150 Char) | Atributo |
| ATTRIBUTE2 | VARCHAR2 (150 Char) | Atributo |
| ATTRIBUTE3 | VARCHAR2 (150 Char) | Atributo |
| ATTRIBUTE4 | VARCHAR2 (150 Char) | Atributo |
| ATTRIBUTE5 | VARCHAR2 (150 Char) | Atributo |
| CREATION\_DATE | DATE | Fecha de creación |
| LAST\_UPDATE\_DATE | DATE | Última fecha de actualización |
| CREATED\_BY | VARCHAR2 (100 Char) | Usuario por que fue creado el registro. |
| LAST\_UPDATED\_BY | VARCHAR2 (100 Char) | Usuario que lo actualizo por última vez. |

Tabla: XXER\_AR\_ELEC\_INVOICE\_ERRORS

Tabla que contendrá el log de los errores que existan al momento de timbrar.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo de dato** | **Descripción** |
| ID\_ERROR | NUMBER | ID de error. |
| CUSTOMER\_TRX\_ID | NUMBER | Identificador único de la factura. |
| NUMERO\_FACTURA | VARCHAR2 (50 Byte) | Identificador número de la factura. |
| TIPOCFD | VARCHAR2 (10 Byte) | Tipo de documento. |
| PAYLOAD | CLOB | Request con que se intentó timbrar. |
| ERROR\_CODE | VARCHAR2 (10 Byte) | Código de error. |
| ERROR\_DESCRIPTION | VARCHAR2 (2500 Byte) | Descripción de error. |

Diseño de Datos

|  |  |  |
| --- | --- | --- |
| ATTRIBUTE1 | VARCHAR2 (250 Byte) | Atributo |
| ATTRIBUTE2 | VARCHAR2 (250 Byte) | Atributo |
| ATTRIBUTE3 | VARCHAR2 (250 Byte) | Atributo |
| ATTRIBUTE4 | VARCHAR2 (250 Byte) | Atributo |
| ATTRIBUTE5 | VARCHAR2 (250 Byte) | Atributo |
| CREATION\_DATE | DATE | Fecha de creación |
| LAST\_UPDATE\_DATE | DATE | Última fecha de actualización |
| CREATED\_BY | VARCHAR2 (100 Char) | Usuario por que fue creado el registro. |
| LAST\_UPDATED\_BY | VARCHAR2 (100 Char) | Usuario que lo actualizo por última vez. |

## Lógica de Validación

Esta lógica hace referencia a las validaciones que se emplean en los origines para que la factura sea candidata para ser timbrada, las principales validaciones de negocio son las siguientes:

* No debe tener UUID.
* No debe tener Folio.
* Debe tener un estatus de Reprocesar, Pago con Anticipo o bien no debe tener un valor.
* El tipo de documento debe ser Factura o Nota de Crédito.

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 el diseño SQL se abarca principalmente las consultas utilizadas para la extracción de la información del ERP Cloud, podremos encontrar más detalle de cómo se realiza la extracción en el documento correspondiente a la capa Ent.

## Sentencias SQL

### invoice – Cabecera Facturas



### item – Líneas de Cabecera



### psv\_vw – Nodo adendas VW



### moneda – Nodo adendas VW de moneda



### proveedor – Nodo adendas VW de proveedor



### referencias – Nodo adendas VW de referencias



### partes – Nodo adendas VW de partes, referencia de parte, y parte



### solicitante – Nodo adendas VW de solicitante

### destino – Nodo adendas VW de destino



### grupo\_ado – Nodo adendas ADO



# Reglas de Negocio

Las reglas de negocio se describen a continuación, para saber más detalle puedes observar el documento **TA020-4\_ ER04\_AR\_Timbrado\_de\_Transacciones** donde se especifica todo el requerimiento.

* Solo se timbran las transacciones de tipo:
  + Factura o Nota de crédito
  + Facturas pagadas con anticipo
* Se enviaran a timbrar las transacciones que tengan en el estatus de “Reprocesar”, “Pago con Anticipo” o bien no tengan valor.
* No se consideran las transacciones que ya tienen Folio y UUID.

## Diseño del Servicio

El servicio no requiere de parámetros para ser ejecutado.

# Consideraciones de Rendimiento

Este requerimiento ha sido probado con un conjunto de transacciones pequeñas, 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.
* En caso de detectar que no existan ejecuciones del proceso calendarizado se deberá comprobar que si este disponible el ESS y que no esté lleno el table space de las ejecuciones.

## Seguridad

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

## 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.
* En caso de requerir personalización a nivel información se deberá configurar el DM correspondiente.  Para agregar o eliminar correos de distribución de errores

## Catálogo de Errores

|  |  |
| --- | --- |
| Código | Descripción |
| SOA-00001 | Error General. |
| SOA-00002 | Error de TimeOut. |
| SOA-00003 | Error de asignación de variable. |
| SOA-00004 | Error de selección. |
| SOA-00005 | Error remoto. |

# 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 los siguientes objetos:

|  |  |
| --- | --- |
| **Nombre** | **Tipo** |
| XXER\_AR\_ELEC\_INVOICE | Tabla |
| XXER\_AR\_ELEC\_INVOICE\_ERRORS | Tabla |
| XXER\_AR\_ELEC\_INVOICE\_ERRORS\_S | Secuencia |

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.



Script DB Facturacion Transacciones.sql

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

* sca\_InvoicesBiz.jar
* sca\_InvoicesEnt.jar

# 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 timbrado de transacciones, las demás se incorporan para brindar funcionalidades específicas descritas anteriormente en el punto 3. Lógica de implementación.

|  |  |
| --- | --- |
| Servicio | URL |
| **InvoicesEnt** | **http://<hostname>:<puerto>/soa-**  **infra/services/default/InvoicesEnt/InvoicesEnt?WSDL** |
| **InvoicesBiz** | **http://<hostname>:<puerto>/soa-**  **infra/services/default/InvoicesBiz/InvoicesBiz?WSDL** |
| ERPIntegrationBiz | http://<hostname>:<puerto>/soa-  infra/services/default/ERPIntegrationBiz/ERPIntegrationBiz?WSDL |
| AuditControlTec | sb://<hostname>:<puerto>/AuditControlTecPs |
| SOAUtilitiesTec | http://<hostname>:<puerto>/soa-  infra/services/default/SOAUtilitiesTec/SOAUtilitiesTec?WSDL |
| MessageTec | http://<hostname>:<puerto>/soa-  infra/services/default/MessageTec/MessageTec?WSDL |
| DigitalStampTec | sb://<hostname>:<puerto>/DigitalStampTecPs |
| ERPIntegrationServiceTec | sb://<hostname>:<puerto>/ ERPIntegrationServiceTec |

# Temas abiertos y cerrados

## Temas Abiertos

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

## Temas Cerrados

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