OUM

TA.070-9 DISEÑO TÉCNICO DE EXTENSIONES E INTERFACES

ER13\_INV\_Lectura de Kilometros

##### Autor: Pedro Sanchez

##### Fecha de creación: Febrero 14, 2024

##### Última actualización: Febrero 14, 2018

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

##### Version: 1.0

#### Aprobadores:

##### 

|  |
| --- |
| Juan Gonzales |
| Gregorio Flores |

##### 

oracle.gif

# Control de Documento

## Bitácora de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Referencia del cambio** |
| 2024-02-14 | Pedro Sanchez | 1.0 | No Previous Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Revisores

**Nombre**

**Posición**

Contenido

[1 Control de Documento 2](#_Toc158825103)

[1.1 Bitácora de Cambios 2](#_Toc158825104)

[1.2 Revisores 2](#_Toc158825105)

[2 Resumen técnico 1](#_Toc158825106)

[2.1 Diagrama de la integración 2](#_Toc158825107)

[3 Lógica de implementación 8](#_Toc158825108)

[3.1 RecordUnitMetrics 8](#_Toc158825109)

[3.2 RecordUnitMetrics 9](#_Toc158825110)

[3.3 Pruebas 16](#_Toc158825111)

[4 Diseño de Datos 17](#_Toc158825112)

[4.1 Origen de Datos 17](#_Toc158825113)

[4.2 Lógica de Validación 17](#_Toc158825114)

[5 Diseño SQL 18](#_Toc158825115)

[5.1 Sentencias SQL 18](#_Toc158825116)

[6 Reglas de Negocio 19](#_Toc158825117)

[6.1 Diseño del Servicio 19](#_Toc158825118)

[7 Consideraciones de Rendimiento 20](#_Toc158825119)

[7.1 Estrategia de Reinicio 20](#_Toc158825120)

[7.2 Seguridad 20](#_Toc158825121)

[7.3 Personalización 20](#_Toc158825122)

[8 URL de Acceso y seguridad 21](#_Toc158825123)

[9 Temas abiertos y cerrados 22](#_Toc158825124)

[10.1 Temas Abiertos 22](#_Toc158825125)

[10.2 Temas Cerrados 22](#_Toc158825126)

# Resumen técnico

##### Esta especificación documenta el diseño para la lectura de kilometros, cuyo caso de uso se describió en el documento **TA020\_ER13\_INV\_Lectura de Kilometros**.

##### La integración se tiene programada en el servidor SOA para ejecutarse en los horarios establecidos.

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

|  |  |  |
| --- | --- | --- |
| **Componente** | **Propiedad** | **Valor** |
| **Trabajo** | Nombre | ¿ |
| Nombre Mostrado | ¿ |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Definición de trabajo para lectura de kilometros |
| Tipo de trabajo | SyncWebserviceJobType |
| WSDL | http://ersoaprod02lab-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/PROD/RecordUnitMetrics/recordunitmterics\_client\_ep?WSDL |
| Tipo de puerto | RecordUnitMetrics\_pt |
| Operación | process |
| **Planificación** | Nombre |  |
| Nombre Mostrado |  |
| Paquete | /oracle/apps/ess/custom/soa |
| Descripción | Programación diaria |
| Frecuencia | Diariamente |
| Zona horaria | (UTC-06:00) Ciudad de México |

## 2.1 Diagrama de la integración

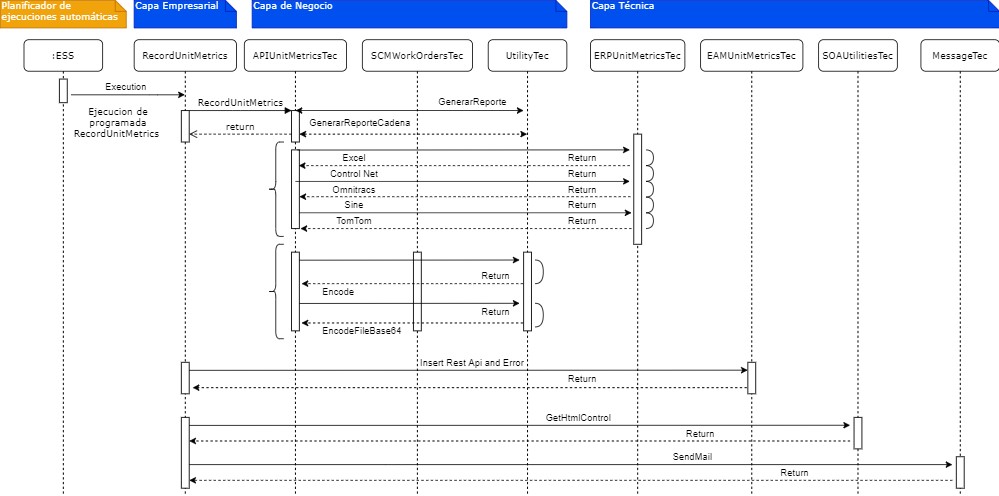
La interfaz RecordUnitMetrics concentrará las funciones de negocio principales para registro de la lectura de kilómetros en

Oracle Fusion las cuales se obtienen de los diferentes orígenes de datos.

El principal objetivo de la interfaz es orquestar y llamado y validaciones de cada lectura obtenida para así poder registrarse

en Fusion y determinar cuándo una unidad de transporte requiere de un mantenimiento, el intercambio de mensajes entre

las principales capas de la arquitectura se muestra en el siguiente flujo:



# Lógica de implementación

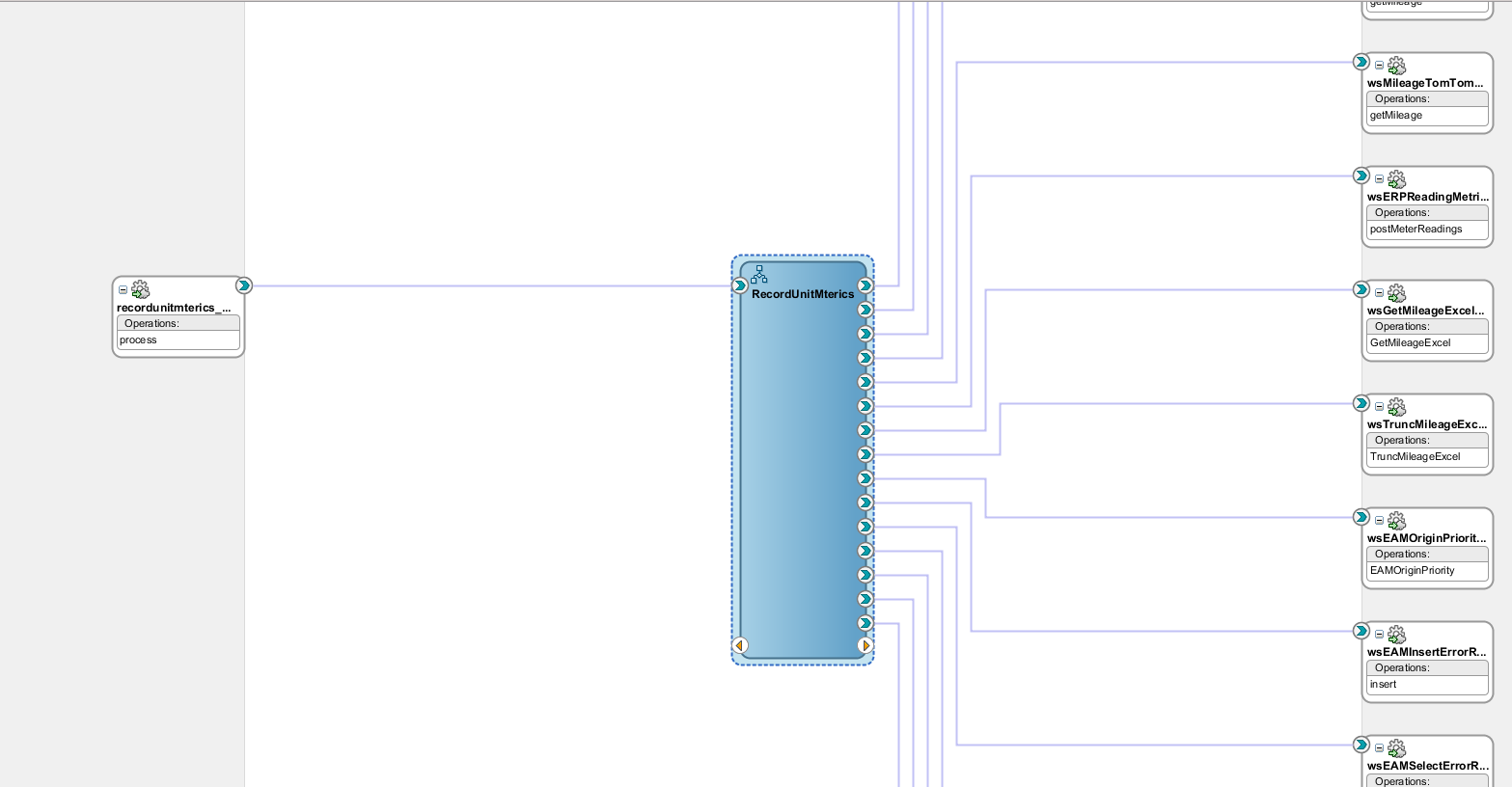
##### El nombre de este servicio es BPEL\_RecordUnitMetrics, por su función de negocio se clasifica como un servicio de

##### orquestación.

##### 

##### Las operaciones que provee el servicio son las siguientes: input (): esta operación se encarga de recibir con su lógica tecnológica los datos de entrada para filtrar las lecturas de kilómetros.

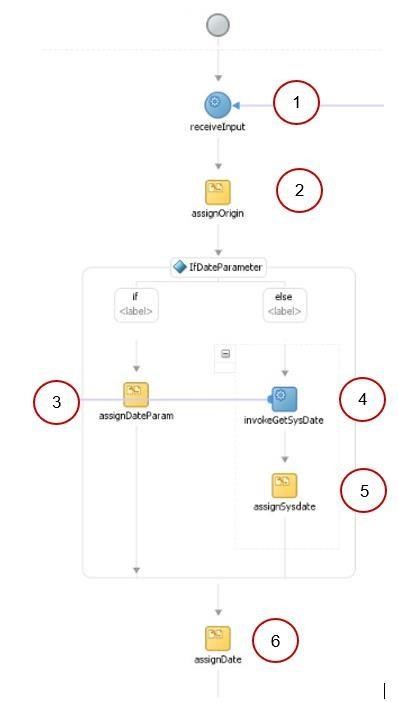
## RecordUnitMetrics

Dentro de la capa SOA se expone el servicio RecordUnitMetrics el cual expone la operación process, ésta a su vez, consume otros servicios comunes para completar el proceso de inicio del proceso de transacciones.

## RecordUnitMetrics

**RecordUnitMetrics**: Esta es la operación que se invoca en el proceso de lectura de Kilómetros, el BPEL que se usa

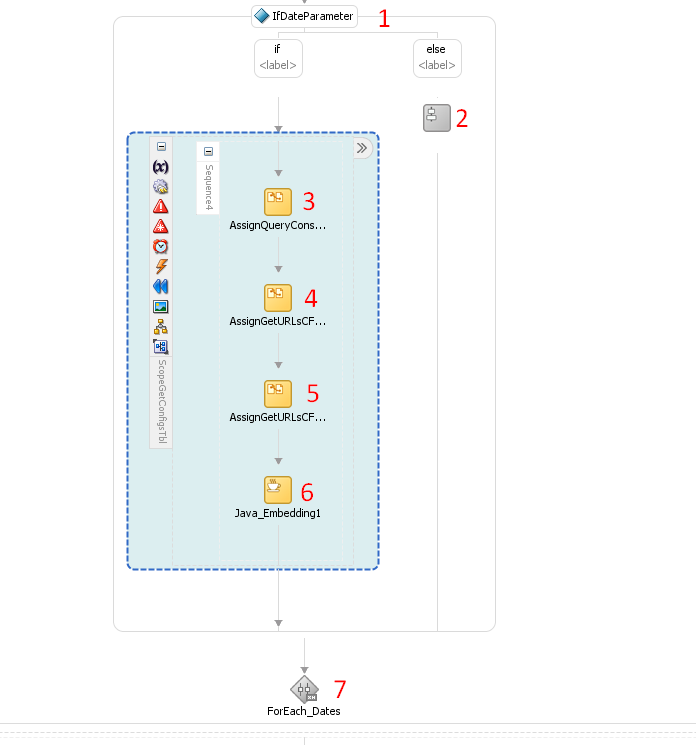
es de tipo oneway a continuación se detallan los componentes que integran esta operación:



1. Se recibe mediante la operación process, los parámetros para la lectura de Kilómetros, Origen, FechaInicio y

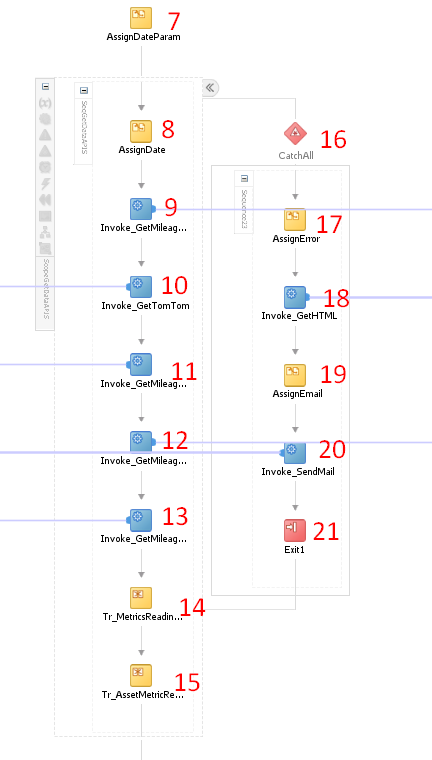
FechaFin.

2. Se asigna el origen el cual se consultará para obtener las lecturas de kilómetros.



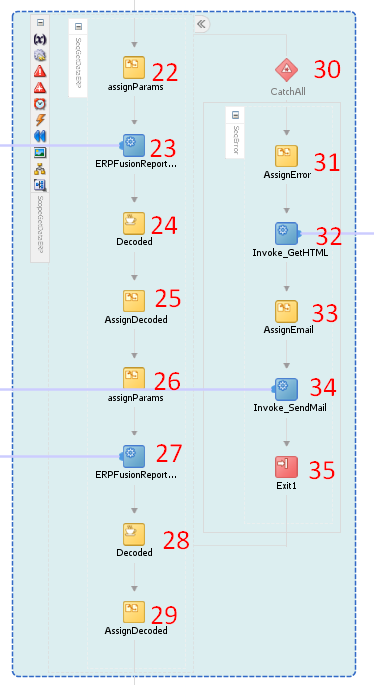
**Descripción**

1. Condición if que evalúa existe el rango de fecha en los valores de entrada de la integracion.
2. Asignación de sysdate por default.
3. Asignación de consulta para obtener listado del rango de fechas recibido.
4. Ejecución de consulta y replace de namespace asignación a variable string.
5. Asignación de variable string a variable node mediante función parse.
6. Muestra de datos mediante java.
7. For para recorrer listado de rango de días.



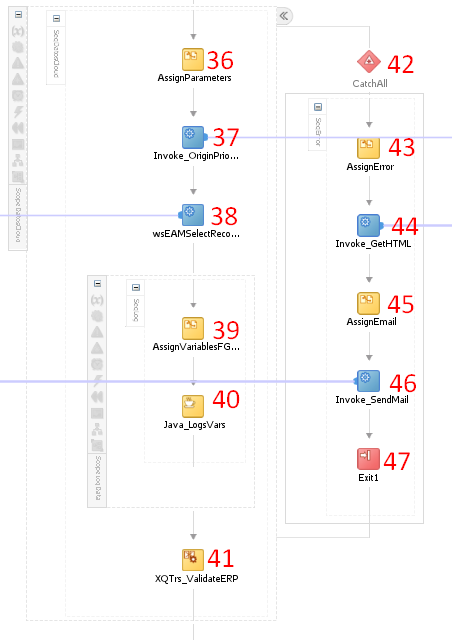
**Descripción**

1. Asignación de datos de entrada de cada API a consultar.
2. Ejecución de API Controlnet.
3. Ejecución de API TomTom.
4. Ejecución de API Omnitracs.
5. Ejecución de API Sine.
6. Ejecución de API Excel.
7. Transformación para Agrupar el resultado de todas las APIS.
8. Transformación para Reagrupar resultado de todas las APIS por unidad y ordenar por fecha más actual.
9. Catch all de errores.
10. Asignamos el error en la variable GetHtmlControlRq.
11. Ejecutamos Invoke\_GetHTML que se encarga de generar el texto de salida en HTML para la notificación.
12. Asignamos el mensaje de notificación en variable In\_SendMail.MessageRq.
13. Ejecutamos Invoke\_SendMail que se encarga enviar la notificación de correo electrónico.
14. Exit del proceso.



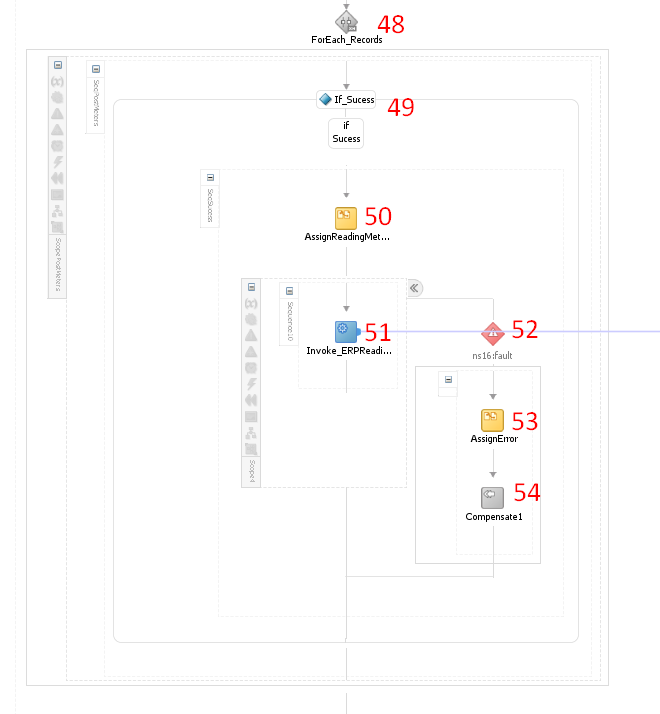
**Descripción**

1. Asignación de datos de variable RunDataModelRq para Consultar reporte ASSETS.
2. Ejecutamos Invoke ERPFusionReportASSETS que se encarga de generar Consultar reporte otbi de ASSETS.
3. Java para decodificar base 64 devuelto por el Invoke ERPFusionReportASSETS.
4. Asignación de string decodificado a variable nodo.
5. Asignación de datos de variable RunDataModelRq para Consultar reporte Readings.
6. Ejecutamos Invoke ERPFusionReportReadings que se encarga de generar Consultar reporte otbi de Readings.
7. Java para decodificar base 64 devuelto por el Invoke ERPFusionReportReadings.
8. Asignación de string decodificado a variable nodo.
9. Catch all de errores.
10. Asignamos el error en la variable GetHtmlControlRq.
11. Ejecutamos Invoke\_GetHTML que se encarga de generar el texto de salida en HTML para la notificación.
12. Asignamos el mensaje de notificación en variable In\_SendMail.MessageRq.
13. Ejecutamos Invoke\_SendMail que se encarga enviar la notificación de correo electrónico.
14. Exit del proceso.



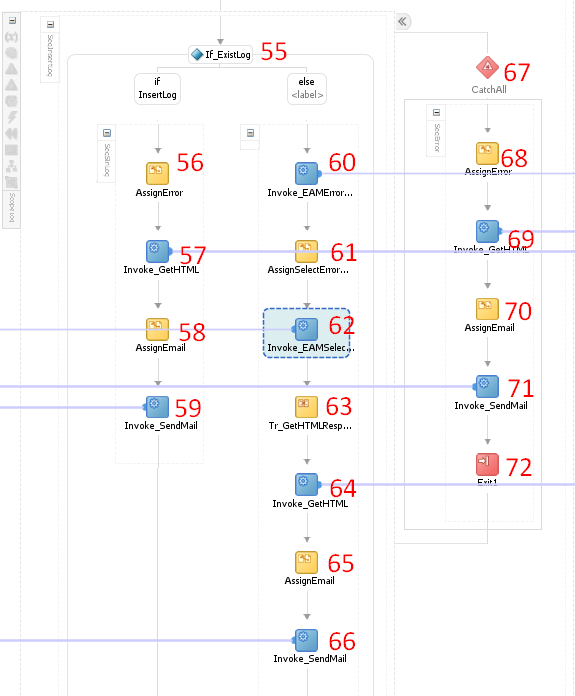
**Descripción**

1. Asignación de datos de variables de entrada para el servicio Invoke\_OriginPriority y wsEAMSelectRecordService.
2. Ejecutamos Invoke Invoke\_OriginPriority nos retorna el listado de prioridad de inserción de origen por unidad.
3. Ejecutamos Invoke wsEAMSelectRecordService nos retorna el ultimo registro success por unidad.
4. Asignación de las variables en string para visualizar en log.
5. Java que codifica string para mostrar en log.
6. Transformación con lógica para validar y calcular kilómetros a partir de todas las consultas previas de información, retorna la salida de errores eh inserciones correctas.
7. Catch all de errores.
8. Asignamos el error en la variable GetHtmlControlRq.
9. Ejecutamos Invoke\_GetHTML que se encarga de generar el texto de salida en HTML para la notificación.
10. Asignamos el mensaje de notificación en variable In\_SendMail.MessageRq.
11. Ejecutamos Invoke\_SendMail que se encarga enviar la notificación de correo electrónico.
12. Exit del proceso.



**Descripción**

1. For que recorre los resultados obtenidos por cada unidad.
2. Condición IF si el estatus es success se inserta la medición.
3. Asignacion de variable de entrada In\_ERPReadingMetrics.
4. Ejecutamos Invoke Invoke\_ERPReadingMetrics servicio que ingresa las mediciones a ERP.
5. Catch all de errores.
6. Asignamos el error en la variable In\_EAMInsertError variable de log de errores.
7. Compásate para seguir iterando.



**Descripción**

1. Condición IF valida si Existen Datos procesados.
2. Asignamos el error de API EMPTY en la variable GetHtmlControlRq.
3. Ejecutamos Invoke\_GetHTML que se encarga de generar el texto de salida en HTML para la notificación.
4. Asignamos el mensaje de notificación en variable In\_SendMail.MessageRq.
5. Ejecutamos Invoke\_SendMail que se encarga enviar la notificación de correo electrónico.
6. Ejecutamos Invoke\_EAMErrorRecord que se encarga de enviar los registros procesados a la tabla de log.
7. Asignamos la variable de entrada In\_EAMSelectErrorRecord.
8. Ejecutamos Invoke\_EAMSelectErrorRecord de consultar el listado de errores por rango de fechas.
9. Transformacion que recibe listado de errores consultado para retornar la variable de entrada GetHtmlControlRq.
10. Ejecutamos Invoke\_GetHTML que se encarga de generar el texto de salida en HTML para la notificación.
11. Asignamos el mensaje de notificación en variable In\_SendMail.MessageRq.
12. Ejecutamos Invoke\_SendMail que se encarga enviar la notificación de correo electrónico.
13. Catch all de errores.
14. Asignamos el error en la variable GetHtmlControlRq.
15. Ejecutamos Invoke\_GetHTML que se encarga de generar el texto de salida en HTML para la notificación.
16. Asignamos el mensaje de notificación en variable In\_SendMail.MessageRq.
17. Ejecutamos Invoke\_SendMail que se encarga enviar la notificación de correo electrónico.
18. Exit del proceso.

## 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://ersoaprod02lab-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/DEV/RecordUnitMetrics/recordunitmterics\_client\_ep?WSDL**](http://ersoaprod02lab-soa-0.snproduccin.vcner.oraclevcn.com:9073/soa-infra/services/DEV/RecordUnitMetrics/recordunitmterics_client_ep?WSDL)

**Request**

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

<soap:Body>

<ns1:RecordUnitMetricsRq xmlns:ns1="http://soa.estrellaroja.com.mx/RecordUnitMterics">

<ns1:StartDate>2023-12-01</ns1:StartDate>

<ns1:EndDate>2023-12-09</ns1:EndDate>

<ns1:Origin>Excel</ns1:Origin>

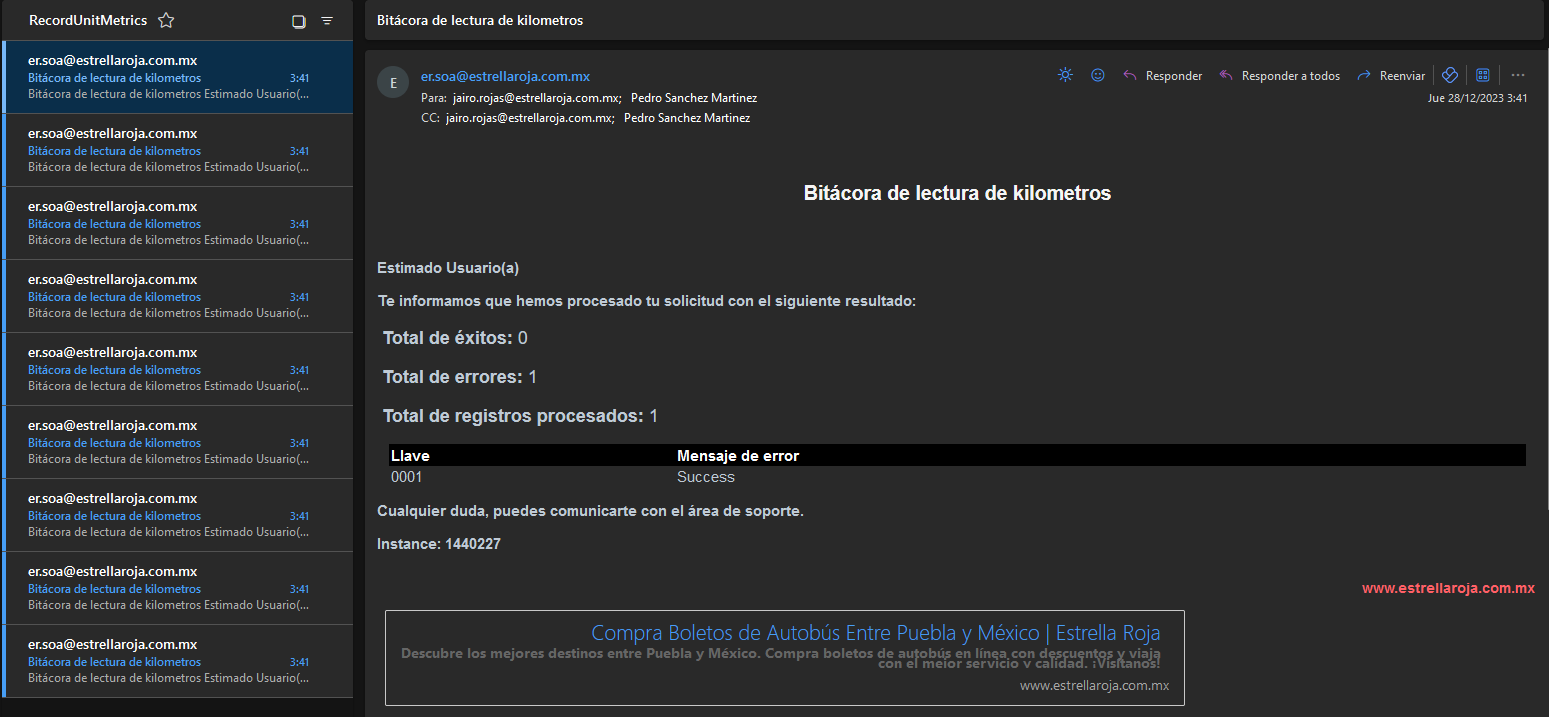
</ns1:RecordUnitMetricsRq>

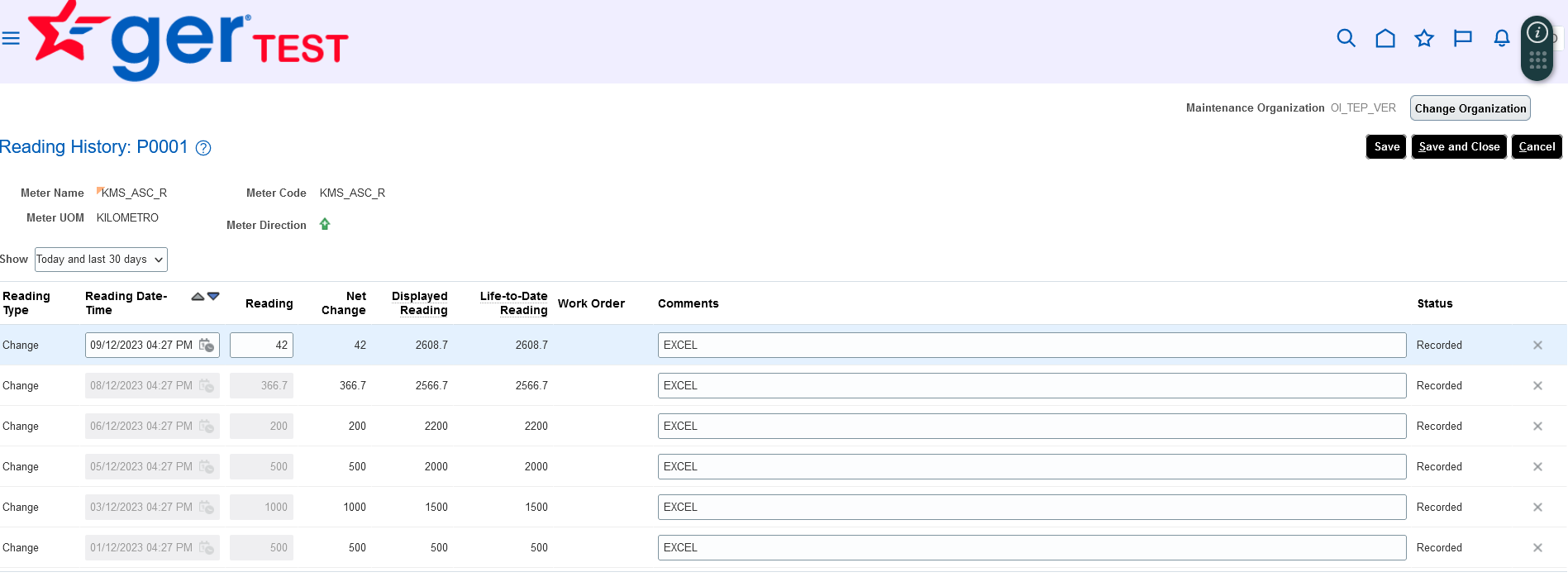
</soap:Body>

</soap:Envelope>

**Response**

Status 200





# Diseño de Datos

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

## Origen de Datos

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

|  |  |  |
| --- | --- | --- |
| **Tabla** | **Columna** | **Comentarios** |
| XXER\_EAM\_DATA\_EXCEL\_TBL | ASSET\_NUMBER | Número de activo |
|  | ORIGIN | Origen de lectura |
|  | READING\_DATE | Fecha de lectura |
|  | READING\_VALUE | Valor de lectura |
|  | STATUS | Estatus de la lectura |
|  | ERROR\_MESSAGE | Mensaje de procesamiento |
| XXER\_EAM\_ERROR\_LOG | ID\_ERROR | Identificador de la tabla |
|  | ASSET\_NUMBER | Número de activo |
|  | READING\_DATE | Fecha de lectura |
|  | READING\_VALUE | Valor de lectura |
|  | ERROR\_MESSAGE | Mensaje de procesamiento |
|  | ORIGIN | Origen de lectura |
| XXER\_EAM\_ORIGIN\_PRIORITY\_TBL | ID | Identificador de la tabla |
|  | ORIGIN | Origen de lectura |
|  | PRIORITY | Asignación de prioridad del origen |
|  | ASSET\_NUMBER | Número de activo |

## Lógica de Validación

##### Esta lógica hace referencia a las validaciones que se emplean en los origines son las siguientes:

# 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.

## 5.1 Sentencias SQL

##### En esta integración existen principalmente dos consultas, una obtiene la información de las unidades y kilometraje.

#### ASSESTS – unidades



#### MEDIDORES – mediciones



# Reglas de Negocio

Las reglas de negocio se describen a continuación, para saber más detalle puedes observar el documento **TA020\_ER13\_INV\_Lectura de Kilometros** donde se especifica todo el requerimiento.

* Se determina el origen, fecha de inicio y fin.
* Extrae información de los diferentes orígenes de acuerdo a los parámetros anteriores.
* Se toma la última lectura registrada por día con la lectura mayor.
* Se verifica si ese activo viene de dos o más orígenes, de ser así se verifica la prioridad asignada.
* Se hacen validaciones contra la última lectura registrada en Oracle Fusion.
* De ser una fecha con valor de lectura mayor y ascendente se escribe en ese activo.
* Se envía correo electrónico y escribe en reporte de Oracle Apex con resumen de información procesada.

## 6.1 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.

# 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 |
| **RecordUnitMetrics** | **http://<hostname>:<puerto>/soa-**  **infra/services/default/InvoicesEnt/InvoicesEnt?WSDL** |
| ERPIntegrationBiz | http://<hostname>:<puerto>/soa-  infra/services/default/ERPIntegrationBiz/ERPIntegrationBiz?WSDL |
| APIUnitMetricsTec | sb://<hostname>:<puerto>/ APIUnitMetricsTecPs |
| SOAUtilitiesTec | http://<hostname>:<puerto>/soa-  infra/services/default/SOAUtilitiesTec/SOAUtilitiesTec?WSDL |
| MessageTec | http://<hostname>:<puerto>/soa-  infra/services/default/MessageTec/MessageTec?WSDL |
| ERPUnitMetricsTec | sb://<hostname>:<puerto>/ERPUnitMetricsTecPs |
| 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** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |